Towards Understanding Embeddings of Neural Network

A Theoretical Perspective

by

Junyi Wei

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)

at the
University of Wisconsin-Madison
Year 2024

Date of Final Oral Exam: 11/27/2024
The dissertation is approved by the following members of the Final Oral Committee:
Yingyu Liang, Associate Professor, Computer Science
Qin Li Stechmann, Professor, Mathematics
Sam Stechmann, Professor, Mathematics
Yin Li, Assistant Professor, Biostatistics & Medical Informatics,
Computer Science

Towards Understanding Embeddings of Neural Network

A Theoretical Perspective

Junyi Wei

Abstract

Theoretically understanding the success of modern neural networks remains challenging. In the direction of theoretically understanding fully connected Multilayer Perceptrons (MLPs), current theoretical frameworks, including the Neural Tangent Kernel (NTK), fall short in explaining several crucial capabilities of neural networks, such as the feature learning ability. Furthermore, there's a significant gap between the empirical success and theoretical understanding of novel architectures like transformers, as well as newer training approaches such as fine-tuning of foundation models. This thesis aims to offer new theoretical insights to narrow these gaps.

To explore feature learning in neural networks, we start from analyzing a practical learning problem where labels are determined by a set of class relevant patterns and the inputs are generated from these along with some background patterns. We prove that neural networks trained via gradient descent can efficiently learn effective features from exponentially many candidates, by exploiting the structure of data distribution. In contrast, linear models with data-independent features fail to achieve comparable accuracy.

Building on this, we propose a unified theoretical framework for two-layer networks trained with gradient descent, emphasizing feature learning through gradients. The framework successfully explains multiple phenomena observed in the learning process, such as feature emergence and the lottery ticket hypothesis. This framework can be applied to different learning problems, including Gaussian mixtures and parity functions.

Our theoretical study into foundation models focused on two key aspects: training methodology and architectural design. On the training front, we theoretically studied the effectiveness of fine-tuning. Our theoretical results show that multitask fine-tuning with diverse related tasks reduces target task error compared to the baseline. We introduce diversity and consistency metrics to quantify task relationships and propose a practical task selection algorithm. In terms of network architecture, we investigate transformers and their in-context learning (ICL) abilities in two settings: (1) linear regression with single-head transformers and (2) parity classification with multi-head transformers. Our analysis reveals that smaller models emphasize key features and resist noise, while larger models capture broader features but are more noise-sensitive. These findings illuminate how model scale influences ICL behavior and sensitivity to test-time noise.

Dedication

To mum, dad, and Yifan

Declaration

I declare that this thesis has been composed solely by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified, and that no other sources or learning aids, other than those listed, have been used. Parts of this work have been published in [[238], [242], [302], [247]].

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisors, Professor Yingyu Liang, for his invaluable guidance, support, and encouragement throughout my Ph.D. journey. His profound knowledge and insightful feedback have been instrumental in shaping my research and academic growth.

I am sincerely grateful to the members of my dissertation committee, Professors Qin Li, Sam Stechmann, and Yin Li for serving on my committee, dedicating their time to review this work, and providing valuable advice for its revision.

I would also like to thank my colleagues and collaborators, Zhenmei Shi, Zhuoyan Xu, Xueyan Zou, Yibing Wei, and others, for their inspiring discussions and shared insights that enriched my research experience. The camaraderie and support from my lab/group mates from Lianglab, have made this journey not only productive but also enjoyable.

My heartfelt thanks go to my family, especially my Parents, Bin Wei and Jun Xiao, and my husband Yifan Wei, for their unconditional love, patience, and encouragement during the highs and lows of this process. Their belief in me has been my greatest source of motivation.

Finally, I extend my gratitude to University of Wisconsin-Madison, whose financial and institutional support made this research career possible.

The works presented in the thesis are partially supported by Air Force Grant FA9550-18-1-0166, the National Science Foundation (NSF) Grants 2008559-IIS, 2023239-DMS, and CCF-2046710. Some works also received partial support from grants by McPherson Eye Research Institute and VCGRE at UW Madison, and from the Army Research Lab under contract number W911NF-2020221.

To everyone who contributed to this thesis in ways big and small, thank you.

Junyi Wei

Contents

List of Figures

List of Tables

Chapter 1

Introduction

From Multilayer Perceptron (MLP) [190] to transformer [268], Neural networks have become a cornerstone of modern machine learning and artificial intelligence, thanks to their remarkable empirical performance. Despite this, a comprehensive theoretical understanding of their success remains elusive. Traditional analytical methods are mostly inadequate for this new challenge because practical networks are often highly overparameterized, and their training involves non-convex optimization using gradient descent.

Theoretical Study on MLPs. Fully connected MLPs, given their relative simplicity and longer history, have a more extensive theoretical study. A line of studies(e.g. [134, 161, 57, 75, 15, 329] and many others) have established that, under certain conditions, heavily overparameterized networks can be approximated by linear models. Specifically, they operate as linear functions defined on the Neural Tangent Kernel (NTK), which remains static during training. This perspective relies on the assumption that networks use fixed, data-independent features to learn, making the approach applicable across a broad range of scenarios. However, it also limits the framework to what is known as the kernel regime, where feature learning—the adaptive process of discovering input representations that improve prediction accuracy—is absent. This limitation is critical because feature learning is widely considered a fundamental reason for the outstanding performance of neural networks in many practical applications(e.g., [310, 107, 311, 175]).

Emergence of New Architectures and Training Methods. Meanwhile, deep neural networks with new architectures, such as transformers, have emerged as pivotal tools in modern AI, largely due to their exceptional efficiency when trained on vast datasets. This shift has ushered in a new era of *foundation models* [36]. These models, exemplified by large language models (e.g., BERT [69] and GPT-3 [38]) and vision models (e.g., CLIP [221] and DINOv2 [208]), have demonstrated the ability to generalize across a wide range of downstream tasks, driving some of the most groundbreaking achievements in AI, such as state-of-the-art conversational systems like ChatGPT [207] and GPT4 [206]. While foundation models have shown impressive empirical success [316, 38, 95], they also present significant challenges that must be addressed to unlock their full potential. On this direction of study, this thesis will focus on two critical challenges, which will be introduced in the following sections.

Few-Shot Learning Problem. Adapting a pretrained foundation model to a new task with only a few labeled samples, known as the few-shot learning problem, has long been a core challenge in machine learning [278]. Several strategies have been proposed to tackle this issue. One common approach, *in-context learning*, involves using labeled examples directly within the context prompt during inference [38]. Another method constructs simple classifiers based on the pretrained model's representations [316], while some techniques fine-tune the model using prompts derived from the labeled data [95]. A promising alternative involves fine-tuning a pretrained model on multiple auxiliary tasks that are relevant to the target task. This multitask fine-tuning strategy, closely related to meta-learning [124], has gained attention in both natural language processing (NLP) and computer vision [191, 274, 320, 128, 50, 181]. For instance, recent studies [233, 189] have shown that fine-tuning language models on diverse task sets can lead to strong zero-shot generalization on new, unseen tasks. Despite these advancements, a significant limitation persists: the absence of rigorous theoretical explanations for these methods. This gap raises concerns about their

reliability and capacity to generalize effectively to real-world scenarios [219].

Unexplained Behavior with the In-Context-Learning (ICL) Ability. A key ability that has significantly contributed to the success of foundation models, particularly large language models (LLMs), is *in-context learning* (ICL) [286, 16]. In ICL, the model is presented with a few input–label examples as part of its prompt before being asked to process a new input. Remarkably, this approach enables LLMs to perform few-shot evaluations [38] without requiring parameter updates. Even more surprising is that LLMs can handle tasks they have never encountered during training, achieving strong performance without any additional fine-tuning. This capability highlights the efficiency of ICL in adapting to diverse downstream tasks with minimal sample and computational costs. ICL operates in a fundamentally different way from traditional machine learning paradigms, such as supervised and unsupervised learning. In standard neural networks, learning typically involves parameter updates through gradient descent. In contrast, ICL relies solely on forward inference, with no gradient updates occurring. Recent studies have sought to explain the underlying mechanism of ICL, suggesting that LLMs may implicitly simulate gradient descent or function as meta-optimizers during the forward pass [61, 272, 174] and theoretical investigations [315, 5, 172, 53, 24, 131, 157, 117, 296] have provided insights into this phenomenon. Despite these advances, the mechanism behind ICL remains only partially understood. Further research is needed to deepen our comprehension of how foundation models achieve this remarkable ability.

Recent studies have uncovered intriguing and unexpected observations about in-context learning (ICL) in large language models (LLMs) [182, 213, 288, 237], revealing gaps in current theoretical understanding. For instance, [237] highlights that LLMs exhibit a lack of robustness during ICL, where irrelevant contextual information can easily distract the model, compromising its performance. Similarly, [288] finds that injecting noise into prompts can degrade the ICL abilities of larger language models more significantly than smaller ones. The study speculates that larger models may overfit to noisy prompts, effec-

tively disregarding the prior knowledge acquired during pretraining. In contrast, smaller models appear to retain and rely more on this pretraining knowledge, enabling them to perform more reliably in noisy scenarios. In support of this, [182, 213] demonstrate that adding noise to prompts has minimal impact on the ICL performance of smaller models. These findings suggest that smaller models maintain a stronger bias toward their pretraining knowledge, making them less susceptible to distractions introduced by noisy or irrelevant inputs. These contrasting behaviors between large and small models raise important questions about the mechanisms underlying ICL and call for further investigation.

Our Contribution. In this thesis we will provide theoretical insights in the three directions introduced above, and we will summarize them here: 1) understanding the feature-learning ability of fully-connected MLP, 2)understanding multitask finetuning and few-shot learning of foundation models, and 3)understanding ICL mechanism of transformer based foundation models, particularly LLMs.

1.1 Background

Neural Network Learning Analysis. The theoretical analysis of neural network learning has garnered growing interest recently, with various approaches attempting to explain their remarkable success.

One prominent perspective links sufficiently over-parameterized neural networks to linear models, such as those based on the Neural Tangent Kernel (NTK)(e.g. [134, 328, 161, 176, 330, 210, 154, 201, 304, 75, 15, 57, 212, 20, 40, 137, 42, 102, 186] and more). This framework simplifies the training process by treating it as a convex optimization problem, leveraging the first-order Taylor expansion of the network around its initialization. However, this approach, often referred to as the "lazy training" or kernel regime, excludes the possibility of feature learning [54, 152, 294, 101], limiting its applicability in explaining real-world performance. Empirical and theoretical evidence (e.g. [21, 13, 283, 104, 309, 118, 14, 23, 9, 63, 74, 153, 47, 305, 129, 162, 105, 227, 173, 168, 41, 2] and more) suggests that neural networks outperform NTK-based methods by utilizing their ability to learn features. A second approach, the *mean-field* (MF) theory, models the training dynamics of large-width networks as partial differential equations (PDEs) (e.g. [179, 56, 178, 248, 52, 229, 71] and more). This method assumes smaller initialization than NTK, allowing parameters to evolve significantly during training. However, MF requires unrealistically wide networks and does not provide explicit convergence rates, making it challenging to apply practically. Another framework is the *max-margin* analysis, which examines the implicit bias of gradient-based optimization methods (e.g. [252, 116, 193, 138, 170, 194, 55, 187, 136, 257, 91, 89, 171] and more). While this approach offers insights into convergence behaviors, it typically assumes that training begins with near-perfect accuracy, which conflicts with the fact that feature learning often occurs in the early stages of training. To address these limitations, some studies focus on the intrinsic low-dimensional structure of data [45, 46, 28, 86, 43, 253, 244, 146, 331, 32], while others explore how trained networks recover ground truth or optimal solutions (like teacher network) [76, 18, 192, 217, 211, 326, 7, 6, 188]. These approaches often rely on restrictive assumptions, such as specific

data distributions or idealized network structures. Another intriguing direction examines the multi-phase dynamics of neural network training [108, 279, 81, 1, 269] where feature learning precedes convex-like optimization. This requires conditions such as proxy convexity [88] or PL condition [141] or special data structure.

Feature Learning Based on Gradient Analysis. Recent research emphasizes the role of gradient dynamics in feature emergence. For linearly separable data, studies[10, 90] demonstrate that the initial gradient steps focus on learning features, followed by network fine-tuning to these learned features. Similar patterns have been observed for nonlinear data, such as parity functions [63, 239, 87], where a single dominant feature suffices for accurate prediction. These insights highlight the importance of understanding gradient-based mechanisms in feature learning.

Training Foundation Models. Foundation models [36] are generally trained using self-supervised learning methods applied to extensive and diverse datasets. Two prominent training paradigms dominate this area: *contrastive learning* for vision and multi-modal tasks, and *masked modeling* for natural language processing.

In self-supervised *contrastive learning*, the goal is to bring augmented versions of the same data point closer together in representation space while ensuring that representations of different data points remain distinct. This method has achieved significant success in vision and multi-modal training [205, 48, 123, 258, 110, 221] leading to a surge in research focused on understanding its theoretical foundations. For instance, [19] provided theoretical guarantees on the classification performance of models pretrained with contrastive learning. [119] analyzed the spectral contrastive loss, offering insights into how contrastive learning impacts model performance, particularly when the pretraining and downstream tasks share the same data distribution. Other works [262, 327, 284, 277, 290, 280, 245, 130, 256, 255] have further explored the principles behind contrastive learning, addressing topics such

as alignment, uniformity, and the effectiveness of contrastive methods for downstream adaptation. These studies contribute to a growing theoretical understanding of contrastive learning, although many rely on idealized assumptions about data distribution and task similarity, leaving room for further exploration in more realistic settings.

Masked modeling focuses on predicting masked tokens within an input sequence and serves as the foundation for many large language models [69, 166, 58, 200, 263]. While initially developed for natural language processing, this approach has recently been extended to vision tasks [122]. Theoretical efforts to understand masked modeling include [317], which frames masked language modeling as a standard supervised learning problem, where labels are derived from the input text itself. This work also explores the relationship between pretraining data diversity and model performance on testing data, shedding light on how data characteristics influence learning outcomes.

Adapting Foundation Models. The adaptation of foundation models to downstream tasks has become a critical area of research. In vision, the traditional approach involves using the model's representations for downstream tasks by either freezing the model and training a simple classifier, such as a linear probe, or performing minor fine-tuning across the entire model [271, 100, 48, 123, 122, 243]. In contrast, NLP has increasingly embraced prompt-based fine-tuning [95, 125, 59, 251, 325, 298, 314]. This technique reformulates prediction tasks into masked language modeling problems, making it a flexible and widely adopted solution. Recent advances in large language models have also popularized parameter-efficient tuning methods. For instance, prompt tuning [155, 158, 230] introduces additional prompt tokens for new tasks while minimizing or avoiding updates to the model's core parameters. Another notable adaptation strategy is *in-context learning* [182, 286, 285, 246, 299], where the model is prompted with task-specific examples in its input context and makes predictions without any parameter modifications. This method has gained significant attention for its efficiency, especially in scenarios with limited labeled data.

Multitask Learning. Multitask supervised learning has emerged as a powerful strategy for transferring knowledge to a target task [320, 233, 50, 181, 282]. This approach has been shown to facilitate zero-shot generalization in large language models [233]and supports parameter-efficient adaptation techniques, such as prompt tuning [282]. For instance, [181, 50] emphasize the role of multitask learning in enhancing in-context learning, while [320] explores task conversion, reformulating classification tasks into a question-answering format to improve transferability. From a theoretical perspective, multitask learning has been studied in terms of its impact on error bounds and sample complexity for the target task [77, 266, 245, 301]. For example, [267] developed a multitask learning framework based on the concept of task diversity in training data, focusing on representations derived from multitask supervised pretraining. Their work provides insights into how diverse tasks contribute to effective knowledge transfer and improved performance on new tasks. These findings underscore the potential of multitask learning as a robust method for leveraging shared knowledge across tasks to address diverse challenges in machine learning.

Few-shot Learning and Meta Learning. Few-shot learning focuses on enabling models to generalize effectively to new tasks with only a small number of labeled examples [278, 274, 191, 165, 306, 93]. Training directly on such limited data often leads to overfitting, making this setting particularly challenging. Meta-learning has emerged as a promising approach to address these challenges by equipping models with the ability to adapt efficiently to few-shot scenarios [82, 223]. In meta-learning, the model is trained on a variety of tasks to develop a general strategy for learning, enabling it to adapt quickly to new tasks with minimal labeled data. This methodology has been successfully applied in vision tasks [271, 249, 51, 128], where techniques like matching networks [271] and prototypical networks [249] have demonstrated effective few-shot learning capabilities. These approaches leverage shared knowledge across tasks, reducing the reliance on extensive labeled data for each individual task and addressing the overfitting challenges associated with few-shot settings.

Large language model. Transformer-based neural networks [268] have become the dominant architecture for natural language processing (NLP) tasks. When pretrained on extensive and diverse datasets with billions of parameters, these models are referred to as large language models (LLMs) or foundation models [36]. Notable examples include BERT [69], PaLM [58], Llama[263], ChatGPT [207], GPT4 [206] and so on. LLMs have demonstrated remarkable capabilities in general intelligence [39], achieving strong performance across a wide range of downstream tasks. To adapt LLMs for specific downstream applications, a variety of techniques have been developed. These include:

- Adapter-based methods: Lightweight modules added to pretrained models to enable task-specific fine-tuning while keeping most parameters frozen [126, 314, 94, 245].
- Calibration techniques: Adjustments to improve model reliability and consistency [318, 321].
- Multitask fine-tuning: Finetuning from multiple tasks to enhance generalization [96, 301, 272, 303].
- Prompt-based methods: Including prompt tuning [95, 155], instruction tuning [158, 59, 183], and symbol tuning [289].
- Black-box optimization: Techniques like black-box tuning [254].
- Reasoning augmentations: Methods such as chain-of-thought prompting [285, 143, 307, 319], and scratchpads for intermediate reasoning steps [202]
- Reinforcement learning: Including reinforcement learning from human feedback (RLHF), which fine-tunes models to align with user preferences [209] and many so on.

These adaptation methods highlight the flexibility of LLMs, enabling their application to an increasingly broad array of tasks while optimizing for efficiency, accuracy, and user alignment.

In-context learning. One of the most remarkable emergent abilities of large language models (LLMs) is *in-context learning* (ICL) [286, 38]. This capability allows LLMs to make predictions for unseen test inputs by observing a short sequence of input-output examples (a prompt) related to the task, all without requiring any updates to the model's parameters. ICL has been successfully applied across a wide range of domains, including reasoning [322], negotiation [92], self-correction [220], machine translation [3] and so on. Efforts to enhance the ICL and zero-shot generalization abilities of LLMs have led to numerous advancements [181, 281, 287, 132]. These improvements focus on fine-tuning strategies, data augmentation, and better prompt design to unlock the full potential of ICL. Additionally, a growing body of research has sought to explore the mechanisms underlying transformer learning [103, 297, 98, 135, 17, 156, 164, 11, 169, 261, 260, 323, 31, 300, 112, 113, 111, 114, 115] and in-context learning specifically [61, 172, 225, 24, 5, 272, 213, 157, 160, 159, 8, 313, 315, 131, 53, 291, 296, 117, 226]. These works span both empirical and theoretical investigations, offering insights into how LLMs leverage prompts to infer task structures, simulate learning dynamics, and adapt flexibly to new tasks. Understanding these mechanisms remains an active area of research, crucial for advancing the practical utility and reliability of in-context learning in diverse applications.

1.2 Dissertation Overview

The rest of this dissertation is structured as follows.

- In Chapter 2, we theoretically investigate a fundamental question: *How can effective features emerge from inputs in the training dynamics of gradient descent? Is learning features from inputs necessary for the superior performance?* To address this, we analyze a learning problem where labels are determined by class-relevant patterns within the data, and inputs are composed of these patterns along with additional, irrelevant background patterns. Our study employs a comparative approach: (1) By contrasting the learning approaches with emergence of effective features with approaches using fixed, predefined features, we demonstrate how emerging of effective features provides a significant advantage over relying solely on fixed features. (2) We further analyze the role of input structure by comparing problems with and without structure in the input data. This comparison shows that input structure is critical for enabling feature learning, which in turn leads to improved prediction performance. This chapter is based on a joint work [238] with Zhenmei Shi:

  > Zhenmei Shi, Junyi Wei and Yingyu Liang, "A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs And Advantages Over Fixed Features", *International Conference on Learning Representations (ICLR) 2022*.

  Contributions of the author: Zhenmei Shi and the author of this thesis Junyi Wei has equal and core contribution towards the work.

- In Chapter 3, we pushed the question asked in Chapter 2 further and asked: *Is there a common principle for feature learning in networks via gradient descent? Is there a unified analysis framework that can clarify the principle and also lead to provable error guarantees for prototypical problem settings?* In this chapter, we take a step toward this goal by proposing a gradient feature learning framework for analyzing two-layer network learning by gradient descent. (1) The framework makes essentially

no assumption about the data distribution and can be applied to various problems. (2) It leads to error guarantees competitive with the optimal in a family of networks that use the features induced by gradients on the data distribution.
This chapter is based on a joint work [242] with Zhenmei Shi:

> Zhenmei Shi, Junyi Wei and Yingyu Liang, "Provable Guarantees for Neural Networks via Gradient Feature Learning", *Conference on Neural Information Processing Systems (NeurIPS) 2023*.

Contributions of the author: Zhenmei Shi and the author of this thesis Junyi Wei has equal and core contribution towards the work.

- In Chapter 4, we study the theoretical justification of multitask finetuning. We consider an intermediate step that finetunes a pretrained model with a set of relevant tasks before adapting to a target task. We present a framework for analyzing pretraining followed by multitask finetuning. Our analysis reveal that with limited labeled data from diverse tasks, finetuning can improve the prediction performance on a downstream task. Inspired by our theorem, we design a task selection algorithm for multitask finetuning.
  This chapter is based on a joint work [302] with Zhuoyan Xu and Zhenmei Shi:

  > Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, Yingyu Liang, "Towards Few-Shot Adaption of Foundation Models via Multitask Finetuning", *International Conference on Learning Representations (ICLR) 2024*.

  Contributions of the author: Zhuoyan Xu has main contribution towards this work. The author Junyi Wei contributes in establishing the theoretical proof outline, solving some proof obstacles, help running part of experiments and participating in discussion. Zhuoyan Xu and Zhenmei Shi may submit this work for other degree or professional qualification.

- In Chapter 5, we attempted to give theoretical explanation on the behavioral difference between language models of different size. We study two settings: (1) one-layer single-head linear self-attention network pretrained on linear regression in-context tasks, with rank constraint on the attention weight matrices for studying the effect of the model scale; (2) two-layer multiple-head transformers [157] pretrained on sparse parity classification in-context tasks, comparing small or large head numbers for studying the effect of the model scale. In both settings, we give the closed-form optimal solutions. The analysis gives evidence that smaller models are more robust to label noise and input noise during evaluation, while larger models may easily be distracted by such noises, so larger models may have a worse ICL ability than smaller ones.
  This chapter is based on a joint work [247] with Zhenmei Shi and Zhuoyan Xu:

  > Zhenmei Shi, Junyi Wei, Zhuoyan Xu, Yingyu Liang, "Why Larger Language Models Do In-context Learning Differently?", *International Conference on Machine Learning (ICML) 2024*.

  Contributions of the author: Zhenmei Shi has main contribution towards the work. The author Junyi Wei has core contribution in proving several major lemmas and theorems. Zhuoyan Xu and Zhenmei Shi may submit this work for other degree or professional qualification.

- In Appendix 7 we include complete proofs, more discussions and additional experiments for each chapter.

Chapter 2

A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features

In this chapter, we will focus on the first direction mentioned in the introduction: understanding the feature-learning ability of fully-connected MLP. We would like to theoretically investigate a fundamental question: *How can effective features emerge from inputs in the training dynamics of gradient descent? Is learning features from inputs necessary for the superior performance?*

To provide more insight on this question, we propose to analyze learning problems motivated by practical data, where the labels are determined by a set of class relevant patterns and the inputs are generated from these along with some background patterns. We use comparison for our study: (1) by comparing network learning approaches with fixed feature approaches on these problems, we analyze the emergence of effective features and demonstrate feature learning leads to the advantage over fixed features; (2) by comparing these problems to those with the input structure removed, we demonstrate that the input

structure is crucial for feature learning and prediction performance.

More precisely, we obtain the following results. We first prove that two-layer networks trained by gradient descent can efficiently learn to small errors on these problems, and then prove that no linear models on fixed features of polynomial sizes can learn to as good errors. These two results thus establish the provable advantage of networks and imply that feature learning leads to this advantage. More importantly, our analysis reveals the dynamics of feature learning: the network first learns a rough approximation of the effective features, then improves them to get a set of good features, and finally learns an accurate classifier on these features. Notably, the improvement of the effective features in the second stage is needed for obtaining the provable advantage. The analysis also reveals the emergence and improvement of the effective features are by exploiting the data, and in particular, they rely on the input structure. To formalize this, we further prove the third result: if the specific input structure is removed and replaced by a uniform distribution, then no polynomial algorithm can even weakly learn in the Statistical Query (SQ) learning model, not to mention the advantage over fixed features. Since SQ learning includes essentially all known algorithms (in particular, mini-batch stochastic gradient descent used in practice), this implies that feature learning depends strongly on the input structure. Finally, we perform simulations on synthetic data to verify our results. We also perform experiments on real data and observe similar phenomena, which show that our analysis provides useful insights for the practical network learning.

Our analysis then provides theoretical support for the following principle: *feature learning in neural networks depends strongly on the input structure and leads to the superior performance.* In particular, our results make it explicit that learning features from the input structure is crucial for the superior performance. This suggests that input-distribution-free analysis (e.g., traditional PAC learning) may not be able to explain the practical success, and advocates an emphasis of the input structure in the analysis. While these results are for our proposed problem setting and network learning in practice can be more complicated, the insights obtained match existing empirical observations and are supported by our

experiments. The compelling evidence hopefully can attract more attention to further studies on modeling the input structure and analyzing feature learning.

This chapter is based on a joint work [238] with Zhenmei Shi:

> Zhenmei Shi, Junyi Wei and Yingyu Liang, "A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs And Advantages Over Fixed Features", *International Conference on Learning Representations (ICLR) 2022*.

Contributions of the author: Zhenmei Shi and the author of this thesis Junyi Wei has equal and core contribution towards the work.

2.1 Problem Setup

To motivate our setup, consider images with various kinds of patterns like lines and rectangles. Some patterns are relevant for the labels (e.g., rectangles for distinguishing indoor or outdoor images), while the others are not. If the image contains a sufficient number of the former, then we are confident that the image belongs to a certain class. Dictionary learning or sparse coding is a classic model of such data (e.g., [204, 270, 33]). We thus model the patterns as a dictionary, generate a hidden vector indicating the presence of the patterns, and generate the input and label from this vector.

Let $\mathbf{X} = \mathbb{R}^d$ be the input space, and $\mathcal{Y} = \{\pm 1\}$ be the label space. Suppose $\mathbf{M} \in \mathbb{R}^{d \times D}$ is an unknown dictionary with D columns that can be regarded as patterns. For simplicity, assume $\mathbf{M}$ is orthonormal. Let $\tilde{\phi} \in \{0, 1\}^D$ be a hidden vector that indicates the presence of each pattern. Let $\mathbf{A} \subseteq [D]$ be a subset of size k corresponding to the class relevant patterns. Then the input is generated by $\mathbf{M}\tilde{\phi}$, and the label can be any binary function on the number of class relevant patterns. More precisely, let $P \subseteq [k]$. Given $\mathbf{A}$ and P, we first sample $\tilde{\phi}$ from a distribution $\mathcal{D}_{\tilde{\phi}}$, and then generate the input $\tilde{\mathbf{x}}$ and the class label y from

$\tilde{\phi}$:

$$\tilde{\phi} \sim \mathcal{D}_{\tilde{\phi}}, \quad \tilde{\mathbf{x}} = \mathbf{M}\tilde{\phi}, \quad y = \begin{cases} +1, & \text{if } \sum_{i \in \mathbf{A}} \tilde{\phi}_i \in P, \\ -1, & \text{otherwise.} \end{cases} \tag{2.1}$$

Learning with Input Structure. We allow quite general $\mathcal{D}_{\tilde{\phi}}$ with the following assumptions:

(A0) The class probabilities are balanced: $\Pr[\sum_{i \in \mathbf{A}} \tilde{\phi}_i \in P] = 1/2$.

(A1) The patterns in $\mathbf{A}$ are correlated with the labels with the same correlation: for any $i \in A$, $\gamma = \mathbb{E}[y\tilde{\phi}_i] - \mathbb{E}[y]\mathbb{E}[\tilde{\phi}_i] > 0$.

(A2) Each pattern outside $\mathbf{A}$ is identically distributed and independent of all other patterns. Let $p_{\mathrm{o}} := \Pr[\tilde{\phi}_i = 1]$ and without loss of generality assume $p_{\mathrm{o}} \leq 1/2$.

Let $\mathcal{D}(\mathbf{A}, P, \mathcal{D}_{\tilde{\phi}})$ denote the distribution on $(\tilde{\mathbf{x}}, y)$ for some $\mathbf{A}, P$, and $\mathcal{D}_{\tilde{\phi}}$. Given parameters $\Xi = (d, D, k, \gamma, p_{\mathrm{o}})$, the family $\mathcal{F}_\Xi$ of distributions include all $\mathcal{D}(\mathbf{A}, P, \mathcal{D}_{\tilde{\phi}})$ with $\mathbf{A} \subseteq [D]$, $P \subseteq [k]$, and $\mathcal{D}_{\tilde{\phi}}$ satisfying the above assumptions. The labeling function includes some interesting special cases:

Example 1. Suppose $P = \{i \in [k] : i > k/2\}$ for some threshold, i.e., we will set the label $y = +1$ when more than a half of the relevant patterns are presented in the input.

Example 2. Suppose k is odd, and let $P = \{i \in [k] : i \text{ is odd}\}$, i.e., the labels are given by the parity function on $\tilde{\phi}_j (j \in \mathbf{A})$. This is useful to prove our lower bounds via the properties of parities.

Appendix A.6 presents results for more general settings (e.g., incoherent dictionary, unbalanced classes, etc.). On the other hand, our problem setup does not include some important data models. In particular, one would like to model hierarchical representations often observed in practical data and believed to be important for deep learning. We leave such more general cases for future work.

Learning Without Input Structure. For comparison, we also consider learning problems without input structure. The data are generated as above but with different distribu-

tions $\mathcal{D}_{\tilde{\phi}}$:

(A1') The patterns are uniform over $\{0,1\}^D$: for any $i \in [D], \Pr[\tilde{\phi}_i = 1] = 1/2$ independently.

Given parameters $\Xi_0 = (d, D, k)$, the family $\mathcal{F}_{\Xi_0}$ of distributions without input structure is the set of all the distributions with $\mathbf{A} \subseteq [D]$, $P \subseteq [k]$ and $\mathcal{D}_{\tilde{\phi}}$ satisfying the above assumptions.

2.1.1 Neural Network Learning

Networks. We consider training a two-layer network via gradient descent on the data distribution:

$$g(\mathbf{x}) = \sum_{i=1}^{2m} \mathbf{a}_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + \mathbf{b}_i) \tag{2.2}$$

where $\mathbf{w}_i \in \mathbb{R}^d, \mathbf{b}_i, \mathbf{a}_i \in \mathbb{R}$, and $\sigma(z) = \min(1, \max(z, 0))$ is the truncated rectified linear unit (ReLU) activation function. Let $\theta = \{\mathbf{w}_i, \mathbf{b}_i, \mathbf{a}_i\}_{i=1}^{2m}$ denote all the parameters, and let superscript (t) denote the time step, e.g., $g^{(t)}$ denote the network at time step t with $\theta^{(t)} = \{\mathbf{w}_i^{(t)}, \mathbf{b}_i^{(t)}, \mathbf{a}_i^{(t)}\}$.

Loss Function. Similar to typical practice, we will normalize the data for learning: first compute $\mathbf{x} = (\tilde{\mathbf{x}} - \mathbb{E}[\tilde{\mathbf{x}}])/\tilde{\sigma}$ where $\tilde{\sigma}^2 = \mathbb{E}\sum_{i=1}^d (\tilde{\mathbf{x}}_i - \mathbb{E}[\tilde{\mathbf{x}}_i])^2$ is the variance of the data, and then train on $(\mathbf{x}, y)$. This is equivalent to setting $\phi = (\tilde{\phi} - \mathbb{E}[\tilde{\phi}])/\tilde{\sigma}$ and generating $\mathbf{x} = \mathbf{M}\phi$. For $(\tilde{\mathbf{x}}, y)$ from $\mathcal{D}$ and the normalized $(\mathbf{x}, y)$, we will simply say $(\mathbf{x}, y) \sim \mathcal{D}$.

For the training, we consider the hinge-loss $\ell(y, \hat{y}) = \max\{1 - y\hat{y}, 0\}$. We will inject some noise ξ to the neurons for the convenience of the analysis. (This can be viewed as using a smoothed version of the activation $\tilde{\sigma}(z) = \mathbb{E}_\xi \sigma(z + \xi)$ similar to those in existing studies like [10, 173]. See Section 2.3 for more explanations.) Formally, the loss is:

$$L_{\mathcal{D}}(g; \sigma_\xi) = \mathbb{E}_{(\mathbf{x},y)}[\ell(y, g(\mathbf{x}; \xi))], \text{ where } g(\mathbf{x}; \xi) = \sum_{i=1}^{2m} \mathbf{a}_i \mathbb{E}_\xi[\sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + \mathbf{b}_i + \xi_i)] \tag{2.3}$$

where $\xi \sim \mathcal{N}(0, \sigma_\xi^2 I_{m\times m})$ are independent Gaussian noise. Let $L_{\mathcal{D}}(g)$ denote the typical hinge-loss without noise. We also consider ℓ_2 regularization: $R(g; \lambda_{\mathbf{a}}, \lambda_{\mathbf{w}}) = \sum_{i=1}^{2m} \lambda_{\mathbf{a}} |\mathbf{a}_i|^2 + \lambda_{\mathbf{w}} \|\mathbf{w}_i\|_2^2$ with regularization coefficients $\lambda_{\mathbf{a}}, \lambda_{\mathbf{w}}$.

Training Process. We first perform an unbiased initialization: for every $i \in [m]$, initialize $\mathbf{w}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$ with $\sigma_{\mathbf{w}} = 1/k$, $\mathbf{b}_i^{(0)} \sim \mathcal{N}\left(0, \sigma_{\mathbf{b}}^2\right)$ with $\sigma_{\mathbf{b}} = 1/k^2$, $\mathbf{a}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{a}}^2)$ with $\sigma_{\mathbf{a}} = \tilde{\sigma}^2/(\gamma k^2)$, and then set $\mathbf{w}_{m+i}^{(0)} = \mathbf{w}_i^{(0)}$, $\mathbf{b}_{m+i}^{(0)} = \mathbf{b}_i^{(0)}$, $\mathbf{a}_{m+i}^{(0)} = -\mathbf{a}_i^{(0)}$. We then do gradient updates:

$$\theta^{(t)} = \theta^{(t-1)} - \eta^{(t)} \nabla_\theta \left(L_{\mathcal{D}}(g^{(t-1)}; \sigma_\xi^{(t)}) + R(g^{(t-1)}; \lambda_{\mathbf{a}}^{(t)}, \lambda_{\mathbf{w}}^{(t)}) \right), \text{ for } t = 1, 2, \ldots, T, \quad (2.4)$$

for some choice of the hyperparameters $\eta^{(t)}, \lambda_{\mathbf{a}}^{(t)}, \lambda_{\mathbf{w}}^{(t)}, \sigma_\xi^{(t)}$, and T.

2.2 Main Results

Provable Guarantee for Neural Networks. The network learning has the following guarantee:

Theorem 2.2.1. *For any $\delta, \epsilon \in (0, 1)$, if $k = \Omega\left(\log^2\left(D/(\delta\gamma)\right)\right)$, $p_o = \Omega(k^2/D)$, and $\max\{\Omega(k^{12}/\epsilon^{3/2}), D\} \le m \le$ poly(D), then with properly set hyperparameters, for any $\mathcal{D} \in \mathcal{F}_\Xi$, with probability at least $1 - \delta$, there exists $t \in [T]$ such that $\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \neq y] \le L_{\mathcal{D}}(g^{(t)}) \le \epsilon$.*

The theorem shows that for a wide range of the background pattern probability p_o and the number of class relevant patterns k, the network trained by gradient descent can obtain a small classification error. More importantly, the analysis shows the success comes from feature learning. In the early stages, the network learns and improves the neuron weights such that on the features (i.e., the neurons' outputs) there is an accurate classifier; afterwards it learns such a classifier. The next section will provide a detailed discussion on the feature learning.

Lower Bound for Fixed Features. Empirical observations and Theorem 2.2.1 do not exclude the possibility that some methods without feature learning can achieve similar performance. We thus prove a lower bound for the fixed feature approach, i.e., linear models on data-independent features.

Theorem 2.2.2. *Suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathbf{X} \to [-1, 1]^N$. For $B > 0$, the family of linear models on Ψ with bounded norm B is $\mathcal{H}_B = \{h(\tilde{\mathbf{x}}) : h(\tilde{\mathbf{x}}) = \langle \Psi(\tilde{\mathbf{x}}), w\rangle, \|w\|_2 \leq B\}$. If $3 < k \leq D/16$ and k is odd, then there exists $\mathcal{D} \in \mathcal{F}_\Xi$ such that all $h \in \mathcal{H}_B$ have hinge-loss at least $p_o \left(1 - \frac{\sqrt{2N}B}{2^k}\right)$.*

So using *fixed* features independent of the data cannot get loss nontrivially smaller than p_o unless with exponentially large models. In contrast, viewing the neurons $\sigma(\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i)$ as *learned* features, network learning can achieve any loss $\epsilon \in (0, 1)$ with models of polynomial sizes. We emphasize the lower bound is because the feature map Ψ is independent of the data. Indeed, there exists a small linear model on a small dimensional feature map allowing 0 loss for each data distribution in our problem set $\mathcal{F}_\Xi$ (Lemma 2.3.1). However, this feature map Ψ^* is different for different data distribution in $\mathcal{F}_\Xi$, i.e., depends on the data. On the other hand, the feature map Ψ in the lower bound is data-independent, i.e., fixed before seeing the data. For Ψ to work simultaneously for all distributions in $\mathcal{F}_\Xi$, it needs to have exponential dimensions. Intuitively, it needs a large number of features, so that there are some features to approximate each Ψ_i^*. There are exponentially many data distributions in $\mathcal{F}_\Xi$, and thus exponentially many data-dependent features Ψ_i^*, which requires Ψ to have an exponentially large dimension. Network learning updates the hidden neurons using the data and can learn to move the features to the right positions to approximate the ground-truth data-dependent features Ψ^*, so it does not need an exponentially large dimension feature map.

The theorem directly applies to linear models on fixed finite-dimensional feature maps, e.g., linear models on the input or random feature approaches [224]. It also implies lower bounds to infinite dimensional feature maps (e.g., some kernels) that can be approximated by feature maps of polynomial dimensions. For example, Claim 1 in [224] implies that

a function f using shift-invariant kernels (e.g., RBF) can be approximated by a model $\langle \Psi(\tilde{\mathbf{x}}), w \rangle$ with the dimension N and weight norm B bounded by polynomials of the related parameters of f like its RKHS norm and the input dimension. Then our theorem implies some related parameter of f needs to be exponential in k for f to get nontrivial loss, formalized in Corollary 2.2.3. [140] has more discussions on approximating kernels with finite dimensional maps.

Corollary 2.2.3. *For any function f using a shift-invariant kernel K with RKHS norm bounded by L, or $f(x) = \sum_i \alpha_i K(z_i, x)$ for some data points z_i and $||\alpha||_2 \leq L$. If $3 < k \leq D/16$ and k is odd, then there exists $\mathcal{D} \in \mathcal{F}_\Xi$ such that f has hinge-loss at least $p_o \left(1 - \frac{\text{poly}(d,L)}{2^k}\right) - \frac{1}{\text{poly}(d,L)}$.*

Lower Bound for Without Input Structure. Existing results do not exclude the possibility that some learning methods without exploiting the input structure can achieve strong performance. To show the necessity of the input structure, we consider learning $\mathcal{F}_{\Xi_0}$ with input structure removed. We obtain a lower bound for such learning problems in the classic Statistical Query (SQ) model [142]. In this model, the algorithm can only receive information about the data through statistical queries. A statistical query is specified by some polynomially-computable property predicate Q of labeled instances and a tolerance parameter $\tau \in [0, 1]$. For a query (Q, τ), the algorithm receives a response $\hat{P}_Q \in [P_Q - \tau, P_Q + \tau]$, where $P_Q = \Pr[Q(x, y) \text{ is true}]$. Notice that a query can be simulated using the average of roughly $O(1/\tau^2)$ random data samples with high probability. The SQ model captures almost all common learning algorithms (except Gaussian elimination) including the commonly used mini-batch SGD, and thus is suitable for our purpose.

Theorem 2.2.4. *For any algorithm in the Statistical Query model that can learn over $\mathcal{F}_{\Xi_0}$ to classification error less than $\frac{1}{2} - \frac{1}{\binom{D}{k}^3}$, either the number of queries or $1/\tau$ must be at least $\frac{1}{2}\binom{D}{k}^{1/3}$.*

The theorem shows that without the input structure, polynomial algorithms in the SQ model cannot get a classification error nontrivially smaller than random guessing. The

comparison to the result for with input structure then shows that the input structure is crucial for network learning, in particular, for achieving the advantage over fixed feature models.

2.3 Proof Sketches

Here we provide the sketch of our analysis, focusing on the key intuition and discussing some interesting implications. The complete proofs are included in Appendix A.2-A.4.

2.3.1 Provable Guarantees of Neural Networks

Overall Intuition. We first show that there is a two-layer network that can represent the target labeling function, whose neurons can be viewed as the "ground-truth" features to be learned. We then show that after the first gradient step, the hidden neurons of the trained network become close to the ground-truth: their weights contain large components along the class relevant patterns but small along the background patterns. We further show that in the second gradient step, these features get improved: the "signal-noise" ratio between the components for class relevant patterns and those for the background ones becomes larger, giving a set of good features. Finally, we show that the remaining steps learn an accurate classifier on these features.

Existence of A Good Network. We show that there is a two-layer network that can fit the labels.

Lemma 2.3.1. *For any $\mathcal{D} \in \mathcal{F}_{\Xi}$, there exists a network $g^*(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{a}_i^* \sigma(\langle \mathbf{w}_i^*, \mathbf{x} \rangle + \mathbf{b}_i^*)$ with $y = g^*(x)$ for any $(\mathbf{x}, y) \sim \mathcal{D}$. Furthermore, the number of neurons $n = 3(k+1)$, $|\mathbf{a}_i^*| \leq 32k, 1/(32k) \leq |\mathbf{b}_i^*| \leq 1/2$, $\mathbf{w}_i^* = \tilde{\sigma} \sum_{j \in \mathbf{A}} \mathbf{M}_j/(4k)$, and $|\langle \mathbf{w}_i^*, \mathbf{x} \rangle + \mathbf{b}_i^*| \leq 1$ for any $i \in [n]$ and $(\mathbf{x}, y) \sim \mathcal{D}$.*

In particular, the weights of the neurons are proportional to $\sum_{j \in \mathbf{A}} \mathbf{M}_j$, the sum of the class relevant patterns. We thus focus on analyzing how the network learns such neuron weights.

Feature Emergence in the First Gradient Step. The gradient for $\mathbf{w}_i$ (ignoring the noise) is:

$$\frac{\partial L_{\mathcal{D}}(g)}{\partial \mathbf{w}_i} = -\mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y\mathbb{I}[yg(\mathbf{x}) \le 1]\sigma'[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i]\mathbf{x} \right\} = -\mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y\mathbf{x}\sigma'[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i] \right\}$$

where the last step is due to $g(\mathbf{x}) = 0$ by the unbiased initialization. Let $q_j = \langle \mathbf{M}_j, \mathbf{w}_i \rangle$ denote the component along the direction of the pattern $\mathbf{M}_j$. Then the component of the gradient on $\mathbf{M}_j$ is:

$$\left\langle \mathbf{M}_j, \frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g) \right\rangle = -\mathbf{a}_i \mathbb{E} \left\{ y\phi_j \sigma'[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i] \right\} = -\mathbf{a}_i \mathbb{E} \left\{ y\phi_j \sigma' \left[\sum_{\ell\in[D]} \phi_\ell q_\ell + \mathbf{b}_i \right] \right\}.$$

The key intuition is that with the randomness of ϕ_ℓ (and potentially that of the injected noise ξ), the random variable under σ' is not significantly affected by a small subset of $\phi_\ell q_\ell$. For example, for class relevant patterns $j \in \mathbf{A}$, let $\mathbb{I}_{[D]} := \sigma' \left[\sum_{\ell\in[D]} \phi_\ell q_\ell + \mathbf{b}_i \right]$ and $\mathbb{I}_{-\mathbf{A}} := \sigma' \left[\sum_{\ell\notin\mathbf{A}} \phi_\ell q_\ell + \mathbf{b}_i \right]$. We have $\mathbb{I}_{[D]} \approx \mathbb{I}_{-\mathbf{A}}$ and thus:

$$\left\langle \mathbf{M}_j, \frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g) \right\rangle \propto \mathbb{E} \left\{ y\phi_j \mathbb{I}_{[D]} \right\} \approx \mathbb{E} \left\{ y\phi_j \mathbb{I}_{-\mathbf{A}} \right\} = \mathbb{E} \left\{ y\phi_j \right\} \mathbb{E}[\mathbb{I}_{-\mathbf{A}}] = \frac{\gamma}{\tilde{\sigma}} \mathbb{E}[\mathbb{I}_{-\mathbf{A}}]$$

since y only depends on $\phi_j (j \in \mathbf{A})$. Then the gradient has a nontrivial component along the pattern. Similarly, for background patterns $j \notin \mathbf{A}$, the component of the gradient along $\mathbf{M}_j$ is close to 0.

Lemma 2.3.2 (Informal)**.** *Assume p_o, k as in Theorem 2.2.1 and $\sigma_\xi^{(1)} < 1/k$, then with high probability $\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) = -\mathbf{a}_i^{(0)} \sum_{j=1}^{D} \mathbf{M}_j T_j$ where for a small ϵ_e:*

- *if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \le O(\epsilon_e/\tilde{\sigma})$ with $\beta \in [\Omega(1), 1]$;*
- *if $j \notin \mathbf{A}$, then $|T_j| \le O(\sigma_\phi^2 \epsilon_e \tilde{\sigma})$.*

By setting $\lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)})$, we have $\mathbf{w}_i^{(1)} = \eta^{(1)} \mathbf{a}_i^{(0)} \sum_{j=1}^{D} \mathbf{M}_j T_j \approx \eta^{(1)} \mathbf{a}_i^{(0)} \frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \mathbf{M}_j$. For small p_o, e.g., $p_o = \tilde{O}(k^2/D)$, these neurons can already allow accurate prediction. However, for such small p_o, we cannot show a provable advantage of networks over fixed

features. On the other hand, for larger p_o meaning a significant number of background patterns in the input, the approximation error terms $T_j(j \notin \mathbf{A})$ together can overwhelm the signals $T_j(j \in \mathbf{A})$ and lead to bad prediction, even though each term is small. Fortunately, we will show that the second gradient step can improve the weights by decreasing the ratio between $T_j(j \notin \mathbf{A})$ and $T_j(j \in \mathbf{A})$.

Feature Improvement in the Second Gradient Step. We note that by setting a small $\eta^{(1)}$, after the update we still have $yg(\mathbf{x};\xi) < 1$ for most $(\mathbf{x}, y) \sim \mathcal{D}$ and thus the gradient in the second step is:

$$\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g;\sigma_\xi) \approx -\mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y\mathbf{x}\mathbb{E}_\xi \sigma'[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i + \xi_i] \right\}.$$

We can then follow the intuition for the first step again. For $j \in \mathbf{A}$, the component $\langle \mathbf{M}_j, \frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g)\rangle$ is roughly proportional to $\frac{\gamma}{\tilde{\sigma}}\mathbb{E}[\mathbb{I}_{-A,\xi}]$ where $\mathbb{I}_{-A,\xi} := \sigma'\left[\sum_{\ell\notin\mathbf{A}} \phi_\ell q_\ell + \mathbf{b}_i + \xi_i\right]$. While $\phi_\ell q_\ell$ may not have large enough variance, the injected noise ξ_i makes sure that a nontrivial amount of data activate the neuron.[1] Then $\mathbb{I}_{-A,\xi} \neq 0$, leading to a nontrivial component along $\mathbf{M}_j$, similar to the first step. On the other hand, for $j \notin \mathbf{A}$, the approximation error term T_j depends on how well $\sigma'\left[\sum_{\ell\notin\mathbf{A},\ell\neq j} \phi_\ell q_\ell + \mathbf{b}_i + \xi_i\right]$ approximates $\sigma'\left[\sum_{\ell\in[D]} \phi_\ell q_\ell + \mathbf{b}_i + \xi_i\right]$. Since the q_ℓ's (the weight's component along $\mathbf{M}_\ell$) in the second step are small compared to those in the first step, we can then get a small error term T_j. So the ratio between $T_j(j \notin \mathbf{A})$ over $T_j(j \in \mathbf{A})$ improves after the second step, giving better features allowing accurate prediction.

Classifier Learning Stage. Given the learned features, we are then ready to show the remaining gradient steps can learn accurate classifiers. Intuitively, with small hyperparameter values ($\eta^{(t)} = \frac{k^2}{Tm^{1/3}}, \lambda_{\mathbf{a}}^{(t)} = \lambda_{\mathbf{w}}^{(t)} \leq \frac{k^3}{\tilde{\sigma}m^{1/3}}, \sigma_\xi^{(t)} = 0$ for $2 < t \leq T = m^{4/3}$), the first layer's weights do not change too much and thus the learning is similar to convex learning using

[1] Equivalently, the network uses $\tilde{\sigma}(z) = \mathbb{E}_\xi \sigma(z+\xi)$, a Gaussian smoothed version of σ, and the smoothing allows z slightly outside the activated region of σ to generate gradient for the learning. Empirically it is not needed since typically sufficient data can activate the neurons. One potential reason is that the data have their own noise to achieve a similar effect (a remote analog being noisy gradients can help the optimization). Further analysis on such an effect is left for future work.

the learned features. Formally, our proof uses the online convex optimization technique in [63].

2.3.2 Lower Bounds

The lower bounds are based on the following observation: our problem setup is general enough to include learning sparse parity functions. Consider an odd k, and let $P = \{i \in [k] : i \text{ is odd}\}$. Then y is given by $\Pi_{\mathbf{A}}(z) := \prod_{j\in\mathbf{A}} z_j$ for $z_j = 2\tilde{\phi}_j - 1$, i.e., the parity function on $z_j (j \in \mathbf{A})$. Then known results for learning parity functions can be applied to prove our lower bounds.

Lower Bound for Fixed Features. We show that $\mathcal{F}_\Xi$ contains learning problems that consist of a mixture of two distributions with weights p_{o} and $1 - p_{\text{o}}$ respectively, where in the first distribution $\mathcal{D}^{(1)}_{\mathbf{A}}$, $\tilde{\mathbf{x}}$ is given by the uniform distribution over $\tilde{\phi}$ and the label y is given by the parity function on $\mathbf{A}$. On such $\mathcal{D}^{(1)}_{\mathbf{A}}$, [63] shows that exponentially large models over fixed features is needed to get nontrivial loss. Intuitively, there are exponentially many labeling functions $\Pi_{\mathbf{A}}$ that are uncorrelated (i.e., "orthogonal" to each other): $\mathbb{E}[\Pi_{\mathbf{A}_1}\Pi_{\mathbf{A}_2}] = 0$ for any $\mathbf{A}_1$ and $\mathbf{A}_2$. Note that the best approximation of $\Pi_{\mathbf{A}}$ by a fixed set of features Ψ_i's is its projection on the linear span of the features. Then with polynomial-size models, there always exists some $\Pi_{\mathbf{A}}$ far from the linear span.

Remark. It is instructive to compare to network learning, which finds the effective weights $\sum_{j\in\mathbf{A}} \mathbf{M}_j$ among the exponentially many candidates corresponding to different $\mathbf{A}$'s. This can be done efficiently by exploiting the data since the gradient is roughly proportional to $\mathbb{E}\{y\mathbf{x}\} = \sum_{j\in\mathbf{A}} \mathbf{M}_j$. The network then learns *data-dependent* features on which polynomial size linear models can achieve small loss.

Lower Bound for Learning without Input Structure. Clearly, $\mathcal{F}_{\Xi_0}$ contains the distributions $\mathcal{D}^{(1)}_{\mathbf{A}}$ described above. The lower bound then follows from classic SQ learning results [35].

Remark. The SQ lower bound analysis does not apply to $\mathcal{F}_\Xi$, because in $\mathcal{F}_\Xi$ the input

distribution is related the labeling function. This allows networks to learn with polynomial time/samples. While both the labeling function and the input distribution affect the learning, few existing studies explicitly point out the importance of the input structure. We thus emphasize the input structure is crucial for networks to learn effective features and achieve superior performance.

2.4 Experiments

Our experiments mainly focus on feature learning and the effect of the input structure. We first perform simulations on our learning problems to (1) verify our main theorems on the benefit of feature learning and the effect of input structure; (2) verify our analysis of feature learning in networks. We then check if our insights carry over to real data: (3) whether similar feature learning is presented in real network/data; (4) whether damaging the input structure lowers the performance. The results are consistent with our analysis and provide positive support for the theory. Below we present part of the results and include the complete experimental details and results in Appendix A.5.

Simulation: Verification of the Main Results. We generate data according to our problem setup, with $d = 500, D = 100, k = 5, p_{\mathrm{o}} = 1/2$, a randomly sampled $\mathbf{A}$, and labels given by the parity function. We then train a two-layer network with $m = 300$ following our learning process, and for comparison, we also use two fixed feature methods (the NTK and random feature methods based on the same network). Finally, we also use these three methods on the data distribution with the input structure removed (i.e., $\mathcal{F}_{\Xi_0}$ in Theorem 2.2.4).

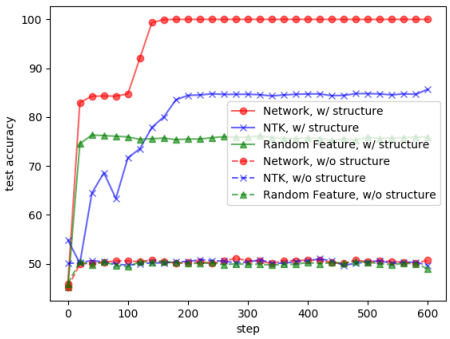


Figure 2.1: Test accuracy on simulated data with or without input structure.

Figure 2.1 shows that the results are consistent with our results. Network learning gets

high test accuracy while the two fixed feature methods get significantly lower accuracy. Furthermore, when the input structure is removed, all three methods get test accuracy similar to random guessing.

Simulation: Feature Learning in Networks. We compute the cosine similarities between the weights $\mathbf{w}_i$'s and visualize them by Multidimensional Scaling. (Recall that our analysis is on the *directions* of the weights without considering their *scaling*, and thus it is important to choose cosine similarity rather than say the typical Euclidean distance.) Figure 2.2 shows that the results are as predicted by our analysis. After the first gradient step, some weights begin to cluster around the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$ (or $-\sum_{j\in\mathbf{A}} \mathbf{M}_j$ due to the $\mathbf{a}_i$ in the gradient update which can be positive or negative). After the second step, the weights get improved and well-aligned with the ground-truth (with cosine similarities > 0.99). Furthermore, if a classifier is trained on the features after the first step, the test accuracy is about 52%; if the same is done after the second step, the test accuracy is about 100%. This demonstrates while some effective features emerge in the first step, they need to be improved in the second step to get accurate prediction.

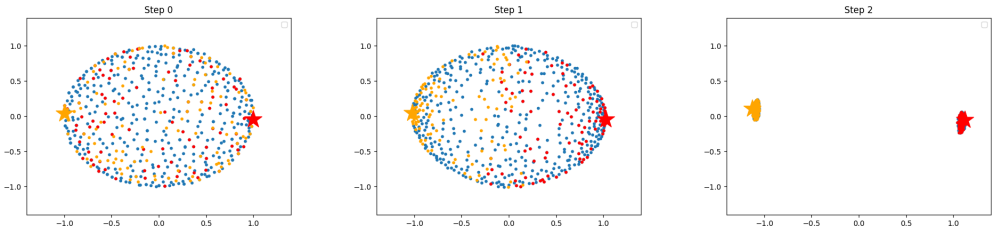


Figure 2.2: Visualization of the weights $\mathbf{w}_i$'s after initialization/one gradient step/two steps in network learning on the synthetic data. The red star denotes the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$; the orange star is $-\sum_{j\in\mathbf{A}} \mathbf{M}_j$. The red/orange dots are the weights closest to the red/orange star, respectively.

Real Data: Feature Learning in Networks. We perform experiments on MNIST [151, 68], CIFAR10 [147], and SVHN [198]. On MNIST, we train a two-layer network with $m = 50$ on the subset with labels 0/1 and visualize the neurons' weights as in the simulation. Figure 2.3 shows a similar feature learning phenomenon: effective features emerge after a

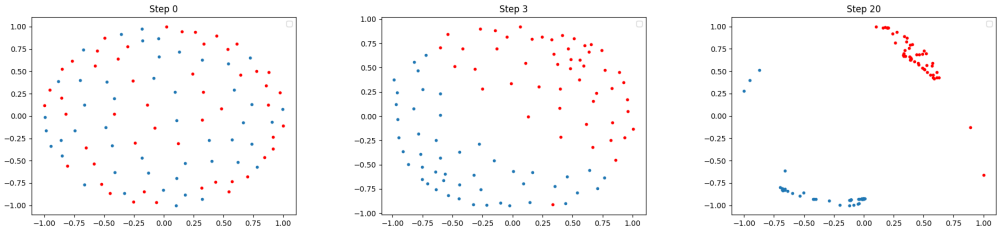
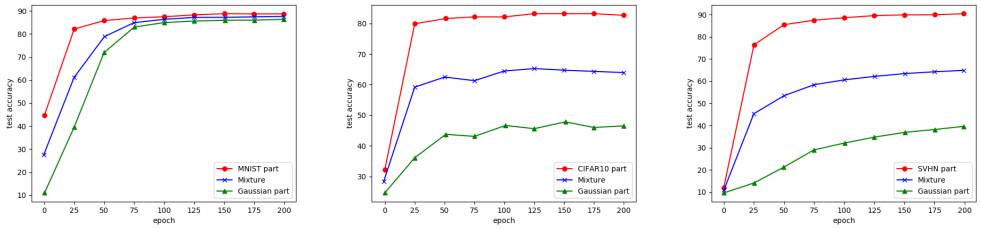


Figure 2.3: Visualization of the neurons' weights in a two-layer network trained on the subset of MNIST data with label 0/1. The weights gradually form two clusters.



(a) (b) (c)

Figure 2.4: Test accuracy at different steps for an equal mixture of Gaussian inputs with data: (a) MNIST, (b) CIFAR10, (c) SVHN.

few steps and then get improved to form two clusters. Similar results are observed on other datasets. These suggest the insights obtained in our analysis are also applicable to the real data.

Real Data: The Effect of Input Structure. Since we cannot directly manipulate the input distribution of real data, we perform controlled experiments by injecting different inputs. For labeled dataset $\mathcal{L}$ and injected input $\mathcal{U}$, we first train a teacher network fitting $\mathcal{L}$, then use the teacher network to give labels on a mixture of inputs from $\mathcal{L}$ and $\mathcal{U}$, and finally train a student network on this new dataset $\mathcal{M}$ consisting of the mixed inputs and the teacher network's labels. Checking the student' performance on different parts of $\mathcal{M}$ and comparing to those by directly training the student on the original data $\mathcal{L}$ can reveal the impact of changing the input structure. We use MNIST, CIFAR10, or SVHN as $\mathcal{L}$, and use Gaussian or images in Tiny ImageNet [150] as $\mathcal{U}$. The networks for MNIST are two-layer with $m = 9$, and those for CIFAR10/SVHN are ResNet-18 convolutional neural networks [120].

Figure 2.4 shows the results on an equal mixture of data and Gaussian. It presents the test accuracy of the student on the original data part, the Gaussian part, and the whole mixture. For example, on CIFAR10, the network learns well over the CIFAR10 part (with accuracy similar to directly training on the original data) but learns slower with worse accuracy on the Gaussian part. Furthermore, the accuracy on the whole mixture is lower than that of training on the original CIFAR10. This shows that the input structure indeed has a significant impact on the learning. While MNIST+Gaussian shows a less significant trend (possibly because the tasks are simpler), the other datasets show similar significant trends as CIFAR10+Gaussian (the results using Tiny ImageNet are in the appendix).

2.5 Acknowledgement

The work in this chapter is partially supported by Air Force Grant FA9550-18-1-0166, the National Science Foundation (NSF) Grants 2008559-IIS and CCF-2046710.

Chapter 3

A Theoretical Framework Towards Provable Guarantees for Neural Networks via Gradient Feature Learning

In chapter **??** we provide theoretical evidence showing that feature learning in neural networks depends strongly on the input structure and leads to the superior performance. In this chapter, we take one step further beyond Chapter 2 by proposing a gradient feature learning framework for analyzing two-layer network learning by gradient descent, from a feature learning point of view. (1) The framework makes essentially no assumption about the data distribution and can be applied to various problems. Furthermore, it is centered around features from gradients, clearly illustrating how gradient descent leads to feature learning in networks and subsequently accurate predictions. (2) It leads to error guarantees competitive with the optimal in a family of networks that use the features induced by gradients on the data distribution. Then for a specific problem with structured data distributions, if the optimal in the induced family is small, the framework gives a small error guarantee.

We then apply the framework to several prototypical problems: mixtures of Gaussians, parity functions, linear data, and multiple-index models. These have been used for studying network learning (in particular, for the feature learning ability), but with different and seemingly unrelated analyses. In contrast, straightforward applications of our framework give small error guarantees, where the main effort is to compute the optimal in the induced family. Furthermore, in some cases, such as parities, we can handle more general data distributions than in the existing work.

Finally, we also demonstrate that the framework sheds light on several interesting network learning phenomena or implications such as feature learning beyond the kernel regime, lottery ticket hypothesis (LTH), simplicity bias, learning over different data distributions, and new perspectives about roadmaps forward. Due to space limitations, we present implications about features beyond the kernel regime and LTH in the main body but defer the other implications in B.2 with a brief here. (1) For simplicity bias, it is generally believed that the optimization has some *implicit regularization* effect that restricts learning dynamics to a low capacity subset of the whole hypothesis class, so can lead to good generalization [199, 106]. Our framework provides an explanation that the learning first learns simpler functions and then more sophisticated ones. (2) For learning over different data distributions, we provide data-dependent non-vacuous guarantees, as our framework can be viewed as using the optimal gradient-induced NN to measure or quantify the "complexity" of the problem. For easier problems, this quantity is smaller, and our framework can give a better error bound to derive guarantees. (3) For new perspectives about roadmaps forward, our framework suggests the strong representation power of NN is actually the key to successful learning, while traditional ones suggest strong representation power leads to vacuous generalization bounds [63, 34]. Thus, we suggest a different analysis road. Traditional analysis typically first reasons about the optimal based on the whole function class then analyzes how NN learns proper features and reaches the optimal. In contrast, our framework defines feature family first, and then reasons about the optimal based on it.

This chapter is based on a joint work [242] with Zhenmei Shi:

> Zhenmei Shi, Junyi Wei and Yingyu Liang, "Provable Guarantees for Neural Networks via Gradient Feature Learning", *Conference on Neural Information Processing Systems (NeurIPS) 2023.*

Contributions of the author: Zhenmei Shi and the author of this thesis Junyi Wei has equal and core contribution towards the work.

3.1 Additional Related Work

[62] considers multiple-index with low-degree polynomials as labeling functions and shows that a one-step gradient update can learn multiple features that lead to accurate prediction. [22, 185] studies one gradient step feature improvements at different learning rates. [222] proposes Recursive Feature Machines to show the mechanism of recursively feature learning but without giving a final loss guarantee. These studies consider specific problems and exploit properties of the data to analyze the gradient delicately, while our work provides a general framework applicable to different problems.

3.2 Gradient Feature Learning Framework

Problem Setup. We denote $[n] := \{1, 2, \dots, n\}$ and $\tilde{O}(\cdot), \tilde{\Theta}(\cdot), \tilde{\Omega}(\cdot)$ to omit the log term inside. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space, $\mathcal{Y} \subseteq \mathbb{R}$ the label space. Let $\mathcal{D}$ be an arbitrary data distribution over $\mathcal{X} \times \mathcal{Y}$. Denote the class of two-layer networks with m neurons as:

$$\mathcal{F}_{d,m} := \left\{ f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \;\middle|\; f_{(\mathbf{a},\mathbf{W},\mathbf{b})}(\mathbf{x}) := \mathbf{a}^\top \Big[\sigma(\mathbf{W}^\top \mathbf{x} - \mathbf{b}) \Big] = \sum_{i \in [m]} \mathbf{a}_i \left[\sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle - \mathbf{b}_i) \right] \right\}, \quad (3.1)$$

where $\sigma(z) = \max(z, 0)$ is the ReLU activation function, $\mathbf{a} \in \mathbb{R}^m$ is the second layer weight, $\mathbf{W} \in \mathbb{R}^{d \times m}$ is the first layer weight, $\mathbf{w}_i$ is the i-th column of $\mathbf{W}$ (i.e., the weight for the i-th neuron), and $\mathbf{b} \in \mathbb{R}^m$ is the bias for the neurons. For technical simplicity, we only

train $\mathbf{a}, \mathbf{W}$ but not $\mathbf{b}$. Let superscript (t) denote the time step, e.g., $f_{(\mathbf{a}^{(t)},\mathbf{W}^{(t)},\mathbf{b})}$ denote the network at time step t. Denote $\Xi := (\mathbf{a}, \mathbf{W}, \mathbf{b})$, $\Xi^{(t)} := (\mathbf{a}^{(t)}, \mathbf{W}^{(t)}, \mathbf{b})$. The goal of neural network learning is to minimize the expected risk, i.e., $\mathcal{L}_{\mathcal{D}}(g) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathcal{L}_{(\mathbf{x},y)}(g)$, where $\mathcal{L}_{(\mathbf{x},y)}(g) = \ell(yg(\mathbf{x}))$ is the loss on an example $(\mathbf{x}, y)$ for some loss function $\ell(\cdot)$, e.g., the hinge loss $\ell(z) = \max\{0, 1 - z\}$, and the logistic loss $\ell(z) = \log[1 + \exp(-z)]$. We also consider ℓ_2 regularization. The regularized loss with regularization coefficient λ is $\mathcal{L}^{\lambda}_{\mathcal{D}}(g) := \mathcal{L}_{\mathcal{D}}(g) + \frac{\lambda}{2}(\|\mathbf{W}\|_F^2 + \|\mathbf{a}\|_2^2)$. Given a training set with n i.i.d. samples $\mathcal{Z} = \{(\mathbf{x}^{(l)}, y^{(l)})\}_{l\in[n]}$ from $\mathcal{D}$, the empirical risk and its regularized version are:

$$\widetilde{\mathcal{L}}_{\mathcal{Z}}(g) := \frac{1}{n}\sum_{l\in[n]}\mathcal{L}_{(\mathbf{x}^{(l)},y^{(l)})}(g), \qquad \widetilde{\mathcal{L}}^{\lambda}_{\mathcal{Z}}(g) := \widetilde{\mathcal{L}}_{\mathcal{Z}}(g) + \frac{\lambda}{2}(\|\mathbf{W}\|_F^2 + \|\mathbf{a}\|_2^2). \tag{3.2}$$

Then the training process is summarized in Algorithm 1.

Algorithm 1 Network Training via Gradient Descent

Initialize $(\mathbf{a}^{(0)}, \mathbf{W}^{(0)}, \mathbf{b})$
for $t = 1$ **to** T **do**
  Sample $\mathcal{Z}^{(t-1)} \sim \mathcal{D}^n$
  $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} - \eta^{(t)}\nabla_{\mathbf{a}}\widetilde{\mathcal{L}}^{\lambda^{(t)}}_{\mathcal{Z}^{(t-1)}}(f_{\Xi^{(t-1)}})$, $\quad \mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \eta^{(t)}\nabla_{\mathbf{W}}\widetilde{\mathcal{L}}^{\lambda^{(t)}}_{\mathcal{Z}^{(t-1)}}(f_{\Xi^{(t-1)}})$
end for

In the whole paper, we need some natural assumptions about the data and the loss.

Assumption 3.2.1. We assume $\mathbb{E}[\|\mathbf{x}\|_2] \le B_{x1}$, $\mathbb{E}[\|\mathbf{x}\|_2^2] \le B_{x2}$, $\|\mathbf{x}\|_2 \le B_x$ and for any label y, we have $|y| \le 1$. We assume the loss function $\ell(\cdot)$ is a 1-Lipschitz convex decreasing function, normalized $\ell(0) = 1, |\ell'(0)| = \Theta(1)$, and $\ell(\infty) = 0$.

Remark 3.2.2. The above are natural assumptions. Most input distributions have the bounded norms required, and the typical binary classification $\mathcal{Y} = \{\pm 1\}$ satisfies the requirement. Also, the most popular loss functions satisfy the assumption, e.g., the hinge loss and logistic loss.

3.2.1 Warm Up: A Simple Setting with Frozen First Layer

To illustrate some high-level intuition, we first consider a simple setting where the first layer is frozen after one gradient update, i.e., no updates to $\mathbf{W}$ for $t \geq 2$ in Algorithm 1.

The first idea of our framework is to provide guarantees compared to the optimal in a family of networks. Here let us consider networks with specific weights for the first layer:

Definition 3.2.3. For some fixed $\mathbf{W} \in \mathbb{R}^{d\times m}, \mathbf{b} \in \mathbb{R}^d$, and a parameter B_{a2}, consider the following family of networks $\mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}}$, and the optimal approximation network loss in this family:

$$\mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}} := \left\{ f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \in \mathcal{F}_{d,m} \;\middle|\; \|\mathbf{a}\|_2 \leq B_{a2} \right\}, \qquad \mathrm{OPT}_{\mathbf{W},\mathbf{b},B_{a2}} := \min_{g\in\mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}}} \mathcal{L}_{\mathcal{D}}(f). \tag{3.3}$$

The second idea is to compare to networks using features from gradient descent. As an illustrative example, we now provide guarantees compared to networks with first layer weights $\mathbf{W}^{(1)}$ (i.e., the weights after the first gradient step):

Theorem 3.2.4 (Simple Setting)**.** *Assume $\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a},\mathbf{W}^{(1)},\mathbf{b})}\right)$ is L-smooth to $\mathbf{a}$. Let $\eta^{(t)} = \frac{1}{L}, \lambda^{(t)} = 0$, for all $t \in \{2,3,\ldots,T\}$. Training by Algorithm 1 with no updates for the first layer after the first gradient step, w.h.p., there exists $t \in [T]$ such that $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) \leq \mathrm{OPT}_{\mathbf{W}^{(1)},\mathbf{b},B_{a2}} + O\Big(\frac{L(\|\mathbf{a}^{(1)}\|_2^2+B_{a2}^2)}{T} + \sqrt{\frac{B_{a2}^2(\|\mathbf{W}^{(1)}\|_F^2 B_x^2+\|\mathbf{b}\|_2^2)}{n}}\Big)$.*

Intuitively, the theorem shows that if the weight $\mathbf{W}^{(1)}$ after a one-step gradient gives a good set of neurons in the sense that there exists a classifier on top of these neurons with low loss, then the network will learn to approximate this good classifier and achieve low loss. The proof is based on standard convex optimization and the Rademacher complexity (details in Appendix B.3.1).

Such an approach, while simple, has been used to obtain interesting results on network learning in existing work, which shows that $\mathbf{W}^{(1)}$ can indeed give good neurons due to the structure of the special problems considered (e.g., parities on uniform inputs [26], or polynomials on a subspace [62]). However, it is unclear whether such intuition can still yield

useful guarantees for other problems. So, for our purpose of building a general framework covering more prototypical problems, the challenge is what features from gradient descent should be considered so that the family of networks for comparison can achieve a low loss on other problems. The other challenge is that we would like to consider the typical case where the first layer weights are not frozen. In the following, we will introduce the core concept of Gradient Features to address the first challenge, and stipulate proper geometric properties of Gradient Features for the second challenge.

3.2.2 Core Concepts in the Gradient Feature Learning Framework

Now, we will introduce the core concept in our framework, Gradient Features, and use it to build the family of networks to derive guarantees. As mentioned, we consider the setting where the first layer is not frozen. After the network learns good features, to ensure the updates in later gradient steps of the first layer are still benign for feature learning, we need some geometric conditions about the gradient features, which are measured by parameters in the definition of Gradient Features. The conditions are general enough, so that, as shown in Section 3.3, many prototypical problems satisfy them and the induced family of networks enjoys low loss, leading to useful guarantees. We begin by considering what features can be learned via gradients. Note that the gradient w.r.t. $\mathbf{w}_i$ is

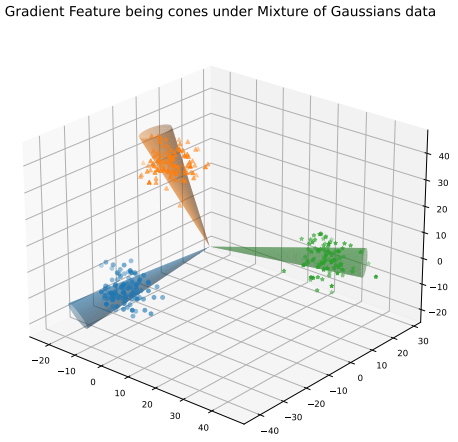


Figure 3.1: An illustration of Gradient Feature, i.e., Definition 3.2.7 with random initialization (Gaussian), under Mixture of three Gaussian clusters in 3-dimension data space with blue/green/orange color. The Gradient Feature stays in three cones, where each center of the cone aligns with the corresponding Gaussian cluster center.

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathcal{D}}(g)}{\partial \mathbf{w}_i} &= \mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)} \left[\ell'(yg(\mathbf{x}))y \left[\sigma'\left(\langle \mathbf{w}_i, \mathbf{x}\rangle - \mathbf{b}_i\right)\right]\mathbf{x}\right] \\ &= \mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)} \left[\ell'(yg(\mathbf{x}))y\mathbf{x}\mathbb{I}[\langle \mathbf{w}_i, \mathbf{x}\rangle > \mathbf{b}_i]\right].\end{aligned}$$

Inspired by this, we define the following notion:

Definition 3.2.5 (Simplified Gradient Vector)**.** For any $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, a Simplified Gradient Vector is

$$G(\mathbf{w}, b) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\mathbf{x}\mathbb{I}[\mathbf{w}^\top \mathbf{x} > b]]. \tag{3.4}$$

Remark 3.2.6. Note that the definition of $G(\mathbf{w}, b)$ ignores the term $\ell'(yg(\mathbf{x}))$ in the gradient, where f is the model function. In the early stage of training (or the first gradient step), $\ell'(\cdot)$ is approximately a constant, i.e., $\ell'(yg(\mathbf{x})) \approx \ell'(0)$ due to the symmetric initialization (see Equation (3.8)).

Definition 3.2.7 (Gradient Feature)**.** For a unit vector $D \in \mathbb{R}^d$ with $\|D\|_2 = 1$, and a $\gamma \in (0, 1)$, a direction neighborhood (cone) $\mathcal{C}_{D,\gamma}$ is defined as:

$$\mathcal{C}_{D,\gamma} := \{\mathbf{w} \ \mid \ |\langle \mathbf{w}, D\rangle|/\|\mathbf{w}\|_2 > (1-\gamma)\}. \tag{3.5}$$

Let $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ be random variables drawn from some distribution $\mathcal{W}, \mathcal{B}$. A Gradient Feature set with parameters p, γ, B_G is defined as:

$$S_{p,\gamma,B_G}(\mathcal{W}, \mathcal{B}) := \big\{(D, s) \ \big| \ \Pr_{\mathbf{w},b}\big[G(\mathbf{w}, b) \in \mathcal{C}_{D,\gamma} \ , \ \|G(\mathbf{w}, b)\|_2 \geq B_G \ , \ s = b/|b|\big] \geq p\big\}. \tag{3.6}$$

Remark 3.2.8. When clear from context, write it as S_{p,γ,B_G}. Gradient features (see Figure 3.1 for illustration) are simply normalized vectors D that are given (approximately) by the simplified gradient vectors. (Similarly, the normalized scalar s is given by the bias b.) To be a useful gradient feature, we require the direction to be "hit" by sufficiently large simplified gradient vectors with sufficient large probability, so as to be distinguished from noise and remain useful throughout the gradient steps. Later we will use the gradient features when $\mathcal{W}, \mathcal{B}$ are the initialization distributions.

To make use of the gradient features, we consider the following family of networks using

these features and with bounded norms, and will provide guarantees compared to the best in this family:

Definition 3.2.9 (Gradient Feature Induced Networks)**.** The Gradient Feature Induced Networks are:

$$\mathcal{F}_{d,m,B_F,S} := \left\{ f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \in \mathcal{F}_{d,m} \;\middle|\; \forall i \in [m],\ |\mathbf{a}_i| \le B_{a1}, \|\mathbf{a}\|_2 \le B_{a2}, (\mathbf{w}_i, \mathbf{b}_i/|\mathbf{b}_i|) \in S,\ |\mathbf{b}_i| \le B_b \right\},$$

where S is some Gradient Feature set and $B_F := (B_{a1}, B_{a2}, B_b)$ are some parameters.

Remark 3.2.10. In above definition, the weight and bias of a neuron are simply the scalings of some item in the feature set S (for simplicity the scaling of $\mathbf{w}_i$ is absorbed into the scaling of $\mathbf{a}_i$ and $\mathbf{b}_i$).

Definition 3.2.11 (Optimal Approximation via Gradient Features)**.** The optimal approximation network and loss using Gradient Feature Induced Networks $\mathcal{F}_{d,r,B_F,S}$ are defined as:

$$g^* := \operatorname*{arg\,min}_{g \in \mathcal{F}_{d,r,B_F,S}} \mathcal{L}_{\mathcal{D}}(f), \qquad \mathrm{OPT}_{d,r,B_F,S} := \min_{g \in \mathcal{F}_{d,r,B_F,S}} \mathcal{L}_{\mathcal{D}}(f). \tag{3.7}$$

3.2.3 Provable Guarantee via Gradient Feature Learning

To obtain the guarantees, we first specify the symmetric initialization. It is convenient for the analysis and is typical in existing analysis (e.g., [63, 62, 10, 239]), though some other initialization can also work. Formally, we train a two-layer network with $4m$ neurons, $f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \in \mathcal{F}_{d,4m}$. We initialize $\mathbf{a}_i^{(0)}, \mathbf{w}_i^{(0)}$ from Gaussians and $\mathbf{b}_i$ from a constant for $i \in \{1, \ldots, m\}$, and initialize the parameters for $i \in \{m+1, \ldots, 4m\}$ accordingly to get a zero output initial network. Specifically:

$$\begin{aligned}
&\text{for } i \in \{1, \ldots, m\}: \quad \mathbf{a}_i^{(0)} \sim \mathcal{N}(0, \sigma_a^2), \mathbf{w}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I), \mathbf{b}_i = \tilde{b}, \\
&\text{for } i \in \{m+1, \ldots, 2m\}: \quad \mathbf{a}_i^{(0)} = -\mathbf{a}_{i-m}^{(0)}, \mathbf{w}_i^{(0)} = -\mathbf{w}_{i-m}^{(0)}, \mathbf{b}_i = -\mathbf{b}_{i-m}, \\
&\text{for } i \in \{2m+1, \ldots, 4m\}: \quad \mathbf{a}_i^{(0)} = -\mathbf{a}_{i-2m}^{(0)}, \mathbf{w}_i^{(0)} = \mathbf{w}_{i-2m}^{(0)}, \mathbf{b}_i = \mathbf{b}_{i-2m},
\end{aligned} \tag{3.8}$$

where $\sigma_a^2, \sigma_{\mathbf{w}}^2, \tilde{b} > 0$ are hyper-parameters. After initialization, $\mathbf{a}, \mathbf{W}$ are updated as in Algorithm 1. We are now ready to present our main result in the framework.

Theorem 3.2.12 (Main Result)**.** *Assume Assumption 3.2.1. For any $\epsilon, \delta \in (0,1)$, if $m \le e^d$ and*

$$
\begin{aligned}
m =& \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right), \\
T =& \Omega\left(\frac{1}{\epsilon}\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}}{\sqrt{B_bB_G}} + \frac{1}{B_{x1}(mp)^{\frac{1}{4}}}\right)\right), \\
\frac{n}{\log n} =& \tilde{\Omega}\left(\frac{m^3pB_x^2B_{a2}^4B_b}{\epsilon^2r^2B_{a1}^2B_G} + \frac{(mp)^{\frac{1}{2}}B_{x2}}{B_bB_G} + \frac{B_x^2}{B_{x2}} + \frac{1}{p} + \left(\frac{1}{B_G^2} + \frac{1}{B_{x1}^2}\right)\frac{B_{x2}}{|\ell'(0)|^2} + \frac{Tm}{\delta}\right),
\end{aligned}
$$

then with initialization (3.8) and proper hyper-parameter values, we have with probability $\ge 1-\delta$ over the initialization and training samples, there exists $t \in [T]$ in Algorithm 1 with:

$$
\begin{aligned}
\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \ne y] &\le \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \\
&\le \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)} + \epsilon.
\end{aligned}
$$

Intuitively, the theorem shows when a data distribution admits a small approximation error by some "ground-truth" network with r neurons using gradient features from S_{p,γ,B_G} (i.e., a small optimal approximate loss $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}$), the gradient descent training can successfully learn good neural networks with sufficiently many m neurons.

Now we discuss the requirements and the error guarantee. Viewing boundedness parameters B_{a1}, B_{x1} etc. as constants, then the number m of neurons learned is roughly $\tilde{\Theta}\left(\frac{r^4}{p\epsilon^4}\right)$, a polynomial overparameterization compared to the "ground-truth" network. The proof shows that such an overparameterization is needed such that some neurons can capture the gradient features given by gradient descent. This is consistent with existing analysis about overparameterization network learning, and also consistent with existing empirical observations.

The error bound consists of three terms. The last term ϵ can be made arbitrarily small,

while the other two depend on the concrete data distribution. Specifically, with larger r and γ, the second term increases. While the first term (the optimal approximation loss) decreases, since a larger r means a larger "ground-truth" network family, and a larger γ means a larger Gradient Feature set S_{p,γ,B_G}. So, there is a trade-off between these two terms. When we later apply the framework to concrete problems (e.g., mixtures of Gaussians, parity functions), we will show that depending on the specific data distribution, we can choose the proper values for r, γ to make the error small. This then leads to error guarantees for the concrete problems and demonstrates the unifying power of the framework. Please refer to Appendix B.3.3 for more discussion about our problem setup and our core concept, e.g., parameter choice, early stopping, the role of s, activation functions, and so on.

Proof Sketch. The intuition in the proof of Theorem 3.2.12 is closely related to the notion of Gradient Features. First, the gradient descent will produce gradients that approximate the features in S_{p,γ,B_G}. Then, the gradient descent update gives a good set of neurons, such that there exists an accurate classifier using these neurons with loss comparable to the optimal approximation loss. Finally, the training will learn to approximate the accurate classifier, resulting in the desired error guarantee. The complete proof is in Appendix B.3 (the population version in Appendix B.3.2 and the empirical version in Appendix B.3.4), including the proper values for hyper-parameters such as $\eta^{(t)}$ in Theorem B.3.17. Below, we briefly sketch the key ideas and omit the technical details.

We first show that a large subset of neurons has gradients at the first step as good features. (The claim can be extended to multiple steps; for simplicity, we follow existing work (e.g., [63, 239]) and present only the first step.) Let ∇_i denote the gradient of the i-th neuron $\nabla_{\mathbf{w}_i}\mathcal{L}_{\mathcal{D}}(f_{\Xi^{(0)}})$. Denote the subset of neurons with nice gradients approximating feature (D, s) as:

$$G_{(D,s),Nice} := \left\{ i \in [2m] : s = \mathbf{b}_i/|\mathbf{b}_i|, \langle \nabla_i, D\rangle > (1-\gamma)\left\|\nabla_i\right\|_2, \left\|\nabla_i\right\|_2 \geq \left|\mathbf{a}_i^{(0)}\right| B_G \right\}. \quad (3.9)$$

Lemma 3.2.13 (Feature Emergence)**.** *For any r size subset $\{(D_1, s_1), \ldots, (D_r, s_r)\} \subseteq$*

S_{p,γ,B_G}, with probability at least $1 - re^{-\Theta(mp)}$, for all $j \in [r]$, we have $|G_{(D_j,s_j),Nice}| \geq \frac{mp}{4}$.

This is because $\nabla_i = \ell'(0)\mathbf{a}_i^{(0)}\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right]\mathbf{x}\right] = \ell'(0)\mathbf{a}_i^{(0)}G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)$. Now consider $s_j = +1$ (the case -1 is similar). Since $\mathbf{w}_i$ is initialized by Gaussians, by ∇_i's connection to Gradient Features, we can see that for all $i \in [m]$, $\Pr\left[i \in G_{(D_j,+1),Nice}\right] \geq \frac{p}{2}$. The lemma follows from concentration via a large enough m, i.e., sufficient overparameterization. The gradients allow obtaining a set of neurons approximating the "ground-truth" network with comparable loss:

Lemma 3.2.14 (Existence of Good Networks)**.** *For any $\delta \in (0,1)$, with proper hyperparameter values, with probability at least $1-\delta$, there is $\tilde{\mathbf{a}}$ such that $\|\tilde{\mathbf{a}}\|_0 = O\left(r\sqrt{mp}\right)$ and $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) = \sum_{i=1}^{4m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)$ satisfies*

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + \sqrt{2}rB_{a1}B_{x1}\left(\sqrt{\gamma} + \sqrt{\frac{2B_b}{\sqrt{mp}B_G}}\right).$$

Given the good set of neurons, we finally show that the remaining gradient steps can learn an accurate classifier. Intuitively, with small step sizes $\eta^{(t)}$, the weights of the first layer $\mathbf{w}_i$ do not change too much (stay in a neighborhood) while the second layer weights grow, and thus the learning is similar to convex learning using the good set of neurons. Technically, we adopt the online convex optimization analysis (Theorem B.3.5) in [63] to get the final loss guarantee in Theorem 3.2.12.

3.3 Applications in Special Cases

In this section we will apply the gradient feature learning framework to some specific problems, corresponding to concrete data distributions $\mathcal{D}$. We primarily focus on prototypical problems for analyzing feature learning in networks. We will present here the results for mixtures of Gaussians and parity functions, and include the complete proofs and some other results in Appendix B.4.

3.3.1 Mixtures of Gaussians

Mixtures of Gaussians are among the most fundamental and widely used statistical models. Recently, it has been used to study neural network learning, in particular, the effect of gradient descent for feature learning of two-layer neural networks and the advantage over fixed feature methods [227, 90].

Data Distributions. We follow notations from [227]. The data are from a mixture of r high-dimensional Gaussians, and each Gaussian is assigned to one of two possible labels in $\mathcal{Y} = \{\pm 1\}$. Let $\mathcal{S}(y) \subseteq [r]$ denote the set of indices of Gaussians associated with the label y. The data distribution is then: $q(\mathbf{x}, y) = q(y)q(\mathbf{x}|y), q(\mathbf{x}|y) = \sum_{j \in \mathcal{S}(y)} p_j \mathcal{N}_j(\mathbf{x})$, where $\mathcal{N}_j(\mathbf{x})$ is a multivariate normal distribution with mean μ_j, covariance Σ_j, and p_j are chosen such that $q(\mathbf{x}, y)$ is correctly normalized. We will make some assumptions about the Gaussians, for which we first introduce some notations.

$$D_j := \frac{\mu_j}{\|\mu_j\|_2}, \quad \tilde{\mu}_j := \mu_j/\sqrt{d}, \quad B_{\mu 1} := \min_{j \in [r]} \|\tilde{\mu}_j\|_2, \quad B_{\mu 2} := \max_{j \in [r]} \|\tilde{\mu}_j\|_2, \quad p_B := \min_{j \in [r]} p_j.$$

Assumption 3.3.1. Let $8 \leq \tau \leq d$ be a parameter that will control our final error guarantee. Assume

- Equiprobable labels: $q(-1) = q(+1) = 1/2$.
- For all $j \in [r]$, $\Sigma_j = \sigma_j I_{d \times d}$. Let $\sigma_B := \max_{j \in [r]} \sigma_j$ and $\sigma_{B+} := \max\{\sigma_B, B_{\mu 2}\}$.
- $r \leq 2d, \quad p_B \geq \frac{1}{2d}, \quad \Omega\left(1/d + \sqrt{\tau {\sigma_{B+}}^2 \log d/d}\right) \leq B_{\mu 1} \leq B_{\mu 2} \leq d.$
- The Gaussians are well-separated: for all $i \neq j \in [r]$, we have $-1 \leq \langle D_i, D_j \rangle \leq \theta$, where $0 \leq \theta \leq \min\left\{\frac{1}{2r}, \frac{\sigma_{B+}}{B_{\mu 2}}\sqrt{\frac{\tau \log d}{d}}\right\}$.

Remark 3.3.2. The first two assumptions are for simplicity; they can be relaxed. We can generalize our analysis to the mixture of Gaussians with unbalanced label probabilities and general covariances. The third assumption is to make sure that each Gaussian has a good amount of probability mass to be learned. The remaining assumptions are to make sure that the Gaussians are well-separated and can be distinguished by the learning algorithm.

We are now ready to apply the framework to these data distributions, for which we only need to compute the Gradient Feature set and the corresponding optimal approximation loss.

Lemma 3.3.3 (Mixtures of Gaussians: Gradient Features)**.** $(D_j, +1) \in S_{p,\gamma,B_G}$ *for all* $j \in [r]$*, where*

$$p = \frac{B_{\mu 1}}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\Theta\left(\tau {\sigma_{B+}}^2 / B_{\mu 1}^2\right)}}, \quad \gamma = \frac{1}{d^{0.9\tau - 1.5}}, \quad B_G = p_B B_{\mu 1}\sqrt{d} - O\left(\frac{\sigma_{B+}}{d^{0.9\tau}}\right).$$

Let $g^*(\mathbf{x}) = \sum_{j=1}^{r} \frac{y_{(j)}}{\sqrt{\tau \log d}\sigma_{B+}} \left[\sigma\left(\langle D_j, \mathbf{x}\rangle - 2\sqrt{\tau \log d}\sigma_{B+}\right)\right]$ *whose hinge loss is at most* $\frac{3}{d^\tau} + \frac{4}{d^{0.9\tau - 1}\sqrt{\tau \log d}}$.

Given the values on gradient feature parameters p, γ, B_G and the optimal approximation loss $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}$, the framework immediately leads to the following guarantee:

Theorem 3.3.4 (Mixtures of Gaussians: Main Result)**.** *Assume Assumption 3.3.1. For any* $\epsilon, \delta \in (0, 1)$*, when Algorithm 1 uses hinge loss with*

$$m = \mathrm{poly}\left(\frac{1}{\delta}, \frac{1}{\epsilon}, d^{\Theta\left(\tau {\sigma_{B+}}^2 / B_{\mu 1}^2\right)}, r, \frac{1}{p_B}\right) \leq e^d, \quad T = \mathrm{poly}\left(m\right), \quad n = \mathrm{poly}\left(m\right)$$

and proper hyper-parameters, then with probability at least $1 - \delta$*, there exists* $t \in [T]$ *such that*

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{\sqrt{2}r}{d^{0.4\tau - 0.8}} + \epsilon.$$

The theorem shows that gradient descent can learn to a small error via learning the gradient features, given proper hyper-parameters. In particular, we need sufficient overparameterization (a sufficiently large number m of neurons). When ${\sigma_{B+}}^2 / B_{\mu 1}^2$ is a constant which is the prototypical interesting case, and we choose a constant τ, then m is polynomial in the key parameters $\frac{1}{\delta}, \frac{1}{\epsilon}, d, r, \frac{1}{p_B}$, and the error bound is inverse polynomial in d. The complete proof is given in Appendix B.4.2.

[90] studies (almost) linear separable cases while our setting includes non-linear separable cases, e.g., XOR. [227] mainly studies neural network classification on 4 Gaussian clusters with XOR structured labels, while our setting is much more general, e.g., our cluster number can extend up to $2d$.

Mixtures of Gaussians: Beyond the Kernel Regime

As discussed in the introduction, it is important for the analysis to go beyond fixed feature methods such as NTK (i.e., the kernel regime), so as to capture the feature learning ability which is believed to be the key factor for the empirical success. We first review the fixed feature methods. Following [63], suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathbf{X} \to [-1, 1]^N$. For $B > 0$, the family of linear models on Ψ with bounded norm B is $\mathcal{H}_B = \{h(\tilde{\mathbf{x}}) : h(\tilde{\mathbf{x}}) = \langle \Psi(\tilde{\mathbf{x}}), w \rangle, \|w\|_2 \leq B\}$. This can capture linear models on fixed finite-dimensional feature maps, e.g., NTK, and also infinite dimensional feature maps, e.g., kernels like RBF, that can be approximated by feature maps of polynomial dimensions [224, 140, 239].

Our framework indeed goes beyond fixed features and shows features from gradients are more powerful than features from random initialization, e.g., NTK. Our framework can show the advantage of network learning over kernel methods under the setting of [227] (4 Gaussian clusters with XOR structured labels). For large enough d, our framework only needs roughly $\Omega(\log d)$ neurons and $\Omega\left((\log d)^2\right)$ samples to achieve arbitrary small constant error (see Theorem B.4.18 when $\sigma_B = 1$), while fixed feature methods need $\Omega(d^2)$ features and $\Omega(d^2)$ samples to achieve nontrivial errors (as proved in [227]). Moreover, [227] uses ODE to simulate the optimization process for the 2-layer networks learning XOR-shaped Gaussian mixture with $\Omega(1)$ neurons and gives convincing evidence that $\Omega(d)$ samples is enough to learn it, yet they do not give a rigorous convergence guarantee for this problem. We successfully derive a convergence guarantee and we require a much smaller sample size $\Omega\left((\log d)^2\right)$. For the proof (detailed in Appendix B.4.3), we only need to calculate the p, γ, B_G of the data distribution carefully and then inject these numbers into

Theorem 3.2.12.

3.3.2 Parity Functions

Parity functions are a canonical family of learning problems in computational learning theory, usually for showing theoretical computational barriers [236]. The typical sparse parties over d-dim binary inputs $\phi \in \{\pm 1\}^d$ are $\prod_{i\in\mathbf{A}} \phi_i$ where $\mathbf{A} \subseteq [d]$ is a subset of dimensions. Recent studies have shown that when the distribution of inputs ϕ has structures rather than uniform, neural networks can perform feature learning and finally learn parity functions with a small error, while methods without feature learning, e.g. NTK, cannot achieve as good results [63, 173, 239]. Thus, this has been a prototypical setting for studying feature learning phenomena in networks. Here we consider a generalization of this problem and show that our framework can show successful learning via gradient descent.

Data Distributions. Suppose $\mathbf{M} \in \mathbb{R}^{d\times D}$ is an unknown dictionary with D columns that can be regarded as patterns. For simplicity, assume $d = D$ and $\mathbf{M}$ is orthonormal. Let $\phi \in \mathbb{R}^d$ be a hidden representation vector. Let $\mathbf{A} \subseteq [D]$ be a subset of size rk corresponding to the class relevant patterns and r is an odd number. Then the input is generated by $\mathbf{M}\phi$, and some function on $\phi_{\mathbf{A}}$ generates the label. WLOG, let $\mathbf{A} = \{1, \ldots, rk\}$, $\mathbf{A}^\perp = \{rk+1, \ldots, d\}$. Also, we split $\mathbf{A}$ such that for all $j \in [r]$, $\mathbf{A}_j = \{(j-1)k+1, \ldots, jk\}$. Then the input $\mathbf{x}$ and the class label y are given by:

$$\mathbf{x} = \mathbf{M}\phi, y = g^*(\phi_{\mathbf{A}}) = \text{sign}\Big(\sum_{j\in[r]} \text{XOR}(\phi_{\mathbf{A}_j})\Big), \tag{3.10}$$

where g^* is the ground-truth labeling function mapping from $\mathbb{R}^{rk}$ to $\mathcal{Y} = \{\pm 1\}$, $\phi_{\mathbf{A}}$ is the sub-vector of ϕ with indices in $\mathbf{A}$, and $\text{XOR}(\phi_{\mathbf{A}_j}) = \prod_{l\in\mathbf{A}_j} \phi_l$ is the parity function. We still need to specify the distribution $\mathbf{X}$ of ϕ, which determines the structure of the input distribution:

$$\mathbf{X} := (1 - 2rp_A)\mathbf{X}_U + \sum_{j\in[r]} p_A(\mathbf{X}_{j,+} + \mathbf{X}_{j,-}). \tag{3.11}$$

For all corresponding $\phi_{\mathbf{A}^\perp}$ in $\mathbf{X}$, we have $\forall l \in \mathbf{A}^\perp$, independently: $\phi_l = \begin{cases} +1, & \text{w.p. } p_o \\ -1, & \text{w.p. } p_o \\ 0, & \text{w.p. } 1-2p_o \end{cases}$,

where p_o controls the signal noise ratio: if p_o is large, then there are many nonzero entries in $\mathbf{A}^\perp$ which are noise interfering with the learning of the ground-truth labeling function on $\mathbf{A}$. For corresponding $\phi_{\mathbf{A}}$, any $j \in [r]$, we have

- In $\mathbf{X}_{j,+}$, $\phi_{\mathbf{A}_j} = [+1, +1, \ldots, +1]^\top$ and $\phi_{\mathbf{A}\setminus\mathbf{A}_j}$ only have zero elements.
- In $\mathbf{X}_{j,-}$, $\phi_{\mathbf{A}_j} = [-1, -1, \ldots, -1]^\top$ and $\phi_{\mathbf{A}\setminus\mathbf{A}_j}$ only have zero elements.
- In $\mathbf{X}_U$, we have $\phi_{\mathbf{A}}$ draw from $\{+1, -1\}^{rk}$ uniformly.

In short, we have r parity functions each corresponding to a block of k dimensions; $\mathcal{X}_{j,+}$ and $\mathcal{X}_{j,-}$ stands for the component providing a strong signal for the j-th parity; $\mathcal{X}_U$ corresponds to uniform distribution unrelated to any parity and providing weak learning signal; $A^\perp$ is the noise part. The label depends on the sum of the r parity functions.

Assumption 3.3.5. Let $8 \leq \tau \leq d$ be a parameter that will control our final error guarantee. Assume k is an odd number and: $k \geq \Omega(\tau \log d)$, $\quad d \geq rk + \Omega(\tau r \log d)$, $\quad p_o = O\left(\frac{rk}{d-rk}\right)$, $\quad p_A \geq \frac{1}{d}$.

Remark 3.3.6. We set up the problem to be more general than the parity function learning in existing work. If $r = 1$, the labeling function reduces to the traditional k-sparse parties of d bits. The assumptions require k, d, and p_A to be sufficiently large so as to provide enough large signals for learning. Note that when $k = \frac{d}{16}, r = 1, p_o = \frac{1}{2}$, our analysis also holds, which shows our framework is beyond the kernel regime (discuss in detail in Section 3.3.2).

To apply our framework, again we only need to compute the Gradient Feature set and the corresponding optimal loss. We first define the Gradient Features: For all $j \in [r]$, let $D_j = \frac{\sum_{l \in \mathbf{A}_j} \mathbf{M}_l}{\| \sum_{l \in \mathbf{A}_j} \mathbf{M}_l \|_2}$.

Lemma 3.3.7 (Parity Functions: Gradient Features)**.** *We have* $(D_j, +1), (D_j, -1) \in S_{p,\gamma,B_G}$

for all $j \in [r]$, where

$$p = \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\Theta(\tau r)}}\right), \qquad \gamma = \frac{1}{d^{\tau-2}}, \qquad B_G = \sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right). \tag{3.12}$$

With gradient features from S_{p,γ,B_G}, let $g^*(\mathbf{x}) = \sum_{j=1}^{r}\sum_{i=0}^{k}(-1)^{i+1}\sqrt{k}\Big[\sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k-1}{\sqrt{k}}\right) - 2\sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k}{\sqrt{k}}\right) + \sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k+1}{\sqrt{k}}\right)\Big]$ whose hinge loss is 0.

Above, we show that D_j is the "indicator function" for the subset A_j so that we can build the optimal neural network based on such directions. Given the values on gradient feature parameters and the optimal approximation loss, the framework immediately leads to the following guarantee:

Theorem 3.3.8 (Parity Functions: Main Result)**.** *Assume Assumption 3.3.5. For any $\epsilon, \delta \in (0,1)$, when Algorithm 1 uses hinge loss with*

$$m = \text{poly}\left(\frac{1}{\delta}, \frac{1}{\epsilon}, d^{\Theta(\tau r)}, k, \frac{1}{p_A}\right) \leq e^d, \quad T = \text{poly}\left(m\right), \quad n = \text{poly}\left(m\right)$$

and proper hyper-parameters, then with probability at least $1-\delta$, there exists $t \in [T]$ such that

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{3r\sqrt{k}}{d^{(\tau-3)/2}} + \epsilon.$$

The theorem shows that gradient descent can learn to a small error in this problem. We also need sufficient overparameterization: When r is a constant (e.g., $r = 1$ in existing work), and we choose a constant τ, m is polynomial in $\frac{1}{\delta}, \frac{1}{\epsilon}, d, k, \frac{1}{p_A}$, and the error bound is inverse polynomial in d. The proof is in Appendix B.4.4. Our setting is more general than that in [63, 173] which corresponds to $\mathbf{M} = I, r = 1, p_A = \frac{1}{4}, p_o = \frac{1}{2}$. [239] study single index learning, where one feature direction is enough for a two-layer network to recover the label, while our setting considers r directions $D_1, \ldots, D_r$, so the network needs to learn multiple directions to get a small error.

Parity Functions: Beyond the Kernel Regime

Again, we show that our framework indeed goes beyond fixed features under parity functions. Our problem setting in Section 3.3.2 is general enough to include the problem setting in [63]. Their lower bound for fixed feature methods directly applies to our case and leads to the following:

Proposition 3.3.9. *There exists a data distribution in the parity learning setting in Section 3.3.2 with* $\mathbf{M} = I, r = 1, p_A = \frac{1}{4}, k = \frac{d}{16}, p_o = \frac{1}{2}$*, such that all* $h \in \mathcal{H}_B$ *have hinge-loss at least* $\frac{1}{2} - \frac{\sqrt{N}B}{2^k\sqrt{2}}$.

This means to get an inverse-polynomially small loss, fixed feature models need to have an exponentially large size, i.e., either the number of features N or the norm B needs to be exponential in k. In contrast, Theorem 3.3.8 shows our framework guarantees a small loss with a polynomially large model, runtime, and sample complexity. Clearly, our framework is beyond the fixed feature methods.

Parities on Uniform Inputs. When $r = 1, p_A = 0$, our problem setting will degenerate to the classic sparse parity function on a uniform input distribution. This has also been used for analyzing network learning [25]. For this case, our framework can get a $k2^{O(k)}\log(k)$ network width bound and a $O(d^k)$ sample complexity bound, matching those in [25]. This then again confirms the advantage of network learning over kernel methods that requires $d^{\Omega(k)}$ dimensions as shown in [25]. See the full statement in Theorem B.4.31, details in Appendix B.4.5, and alternative analysis in Appendix B.4.6.

3.4 Further Implications

Our general framework sheds light on several interesting phenomena in NN learning observed in practice. Feature learning beyond the kernel regime has been discussed in Section 3.3.1 and Section 3.3.2. Here we discuss the LTH and defer more implications such as simplicity bias, learning over different data distributions, and new perspectives about roadmaps forward in Appendix B.2.

Lottery Ticket Hypothesis (LTH). Another interesting phenomenon is the LTH [83]: randomly-initialized networks contain subnetworks that when trained in isolation reach test accuracy comparable to the original network in a similar number of iterations. Later studies (e.g., [84]) show that LTH is more stable when subnetworks are found in the network after a few gradient steps.

Our framework provides an explanation for two-layer networks: the lottery ticket subnetwork contains exactly those neurons whose gradient feature approximates the weights of the "ground-truth" network f^*; they may not exist at initialization but can be found after the first gradient step. More precisely, Lemma 3.2.14 shows that after the first gradient step, there is a *sparse* second-layer weight $\tilde{\mathbf{a}}$ with $\|\tilde{\mathbf{a}}\|_0 = O\left(r\sqrt{mp}\right)$, such that using this weight on the hidden neurons gives a network with a small loss. Let U be the support of $\tilde{\mathbf{a}}$. Equivalently, there is a small-loss subnetwork g_{Ξ}^{U} with only neurons in U and with second-layer weight $\tilde{\mathbf{a}}_U$ on these neurons. Following the same proof of Theorem 3.2.12:

Proposition 3.4.1. *In the same setting of Theorem 3.2.12 but only considering the subnetwork supported on U after the first gradient step, with the same requirements on m and T, with proper hyper-parameter values, we have the same guarantee: with probability $\geq 1-\delta$, there is $t \in [T]$ with* $\Pr[\mathrm{sign}(g_{\Xi^{(t)}}^{U})(\mathbf{x}) \neq y] \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G\sqrt{n}}\right)} + \epsilon$.

This essentially formally proves LTH for two-layer networks, showing (a) the existence of the winning lottery subnetwork and (b) that gradient descent on the subnetwork can learn to similar loss in similar runtime as on the whole network. In particular, (b) is novel and not analyzed in existing work.

We provide this work's limitations (e.g., statement of recovering existing results and some failure cases beyond our framework) in Appendix B.1.

Acknowledgements

The work in this chapter is partially supported by Air Force Grant FA9550-18-1-0166, the National Science Foundation (NSF) Grants 2008559-IIS, 2023239-DMS, and CCF-2046710.

Chapter 4

Towards Few-Shot Adaptation of Foundation Models via Multitask Finetuning

Empirical neural network models are evolving fast, and we don't want to limit our theoretical study in the regime of 2-layer MLPs. From this chapter, we will move on to more recent structures and training method. In this chapter, we focus on the problem of adapting a pretrained foundation model to a new task with a few labeled samples, where the target task can differ significantly from pretraining and the limited labeled data are insufficient for finetuning. As we have discussed in the introduction, latest studies [233, 189] show that multitask finetuning enables strong zero-shot generalization on unseen tasks. Nonetheless, the lack of sound theoretical explanations behind these previous approaches raises doubts about their ability to generalize on real-world tasks [219].

To bridge the gap between empirical explorations and theoretical explanations, we study the theoretical justification of multitask finetuning. We consider an intermediate step that finetunes a pretrained model with a set of relevant tasks before adapting to a target task. Each of these auxiliary tasks might have a small number of labeled samples, and categories of these samples might not overlap with those on the target task. Our key intuition is

that a sufficiently diverse set of relevant tasks can capture similar latent characteristics as the target task, thereby producing meaningful representation and reducing errors in the target task. To this end, we present rigorous theoretical analyses, provide key insight into conditions necessary for successful multitask finetuning, and introduce a novel algorithm for selecting tasks suitable for finetuning.

Our key contributions are three folds. *Theoretically*, we present a framework for analyzing pretraining followed by multitask finetuning. Our analysis (Section 4.2) reveals that with limited labeled data from diverse tasks, finetuning can improve the prediction performance on a downstream task. *Empirically*, we perform extensive experiments on both vision and language tasks (Section 4.3) to verify our theorem. Our results suggest that our theorem successfully predicts the behavior of multitask finetuning across datasets and models. *Practically*, inspired by our theorem, we design *a task selection algorithm* for multitask finetuning. On the Meta-Dataset [265], our algorithm shows significantly improved results in comparison to finetuning using all possible tasks.

This chapter is based on a joint work [302] with Zhuoyan Xu and Zhenmei Shi:

> Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, Yingyu Liang, "Towards Few-Shot Adaption of Foundation Models via Multitask Finetuning", *International Conference on Learning Representations (ICLR) 2024*.

Contributions of the author: Zhuoyan Xu has main contribution towards this work. The author Junyi Wei contributes in establishing the theoretical proof outline, solving some proof obstacles, help running part of experiments and participating in discussion. Zhuoyan Xu and Zhenmei Shi may submit this work for other degree or professional qualification.

4.1 Additional Background: Multitask Finetuning for Few-Shot Learning

This section reviews the pretraining of foundation models and adaptation for few-shot learning, and then formalizes the multitask finetuning approach.

Pretraining Foundation Models. We consider three common pretraining methods: contrastive learning, masked language modeling, and supervised pretraining. *Contrastive learning* is widely considered in vision and multi-modal tasks. This approach pretrains a model ϕ from a hypothesis class Φ of foundation models via loss on contrastive pairs generated from data points x. First sample a point x and then apply some transformation to obtain x^+; independently sample another point x^-. The population contrastive loss is then $\mathcal{L}_{con-pre}(\phi) := \mathbb{E}\left[\ell_u\left(\phi(x)^\top\left(\phi(x^+) - \phi(x^-)\right)\right)\right]$, where the loss function ℓ_u is a non-negative decreasing function. In particular, logistic loss $\ell_u(v) = \log\left(1 + \exp\left(-v\right)\right)$ recovers the typical contrastive loss in most empirical work [167, 205, 48]. *Masked language modeling* is a popular self-supervised learning approach in NLP. It can be regarded as a kind of *supervised pretraining*: the masked word is viewed as the class (see Appendix C.2 for more details). In what follows we provide a unified formulation. On top of the representation function ϕ, there is a linear function $f \in \mathcal{F} \subset \left\{\mathbb{R}^d \to \mathbb{R}^K\right\}$ predicting the labels where K is the number of classes. The supervised loss is: $\mathcal{L}_{sup-pre}(\phi) := \min_{f\in\mathcal{F}} \mathbb{E}\left[\ell\left(f \circ \phi(x), y\right)\right]$, where $\ell(\cdot, y)$ is the cross-entropy loss. To simplify the notation, we unify $\mathcal{L}_{pre}(\phi)$ as the pretraining loss.

Adapting Models for Few-shot Learning. A pretrained foundation model ϕ can be used for downstream target tasks $\mathcal{T}$ by learning linear classifiers on ϕ. We focus on binary classification (the general multiclass setting is in Appendix C.3). A linear classifier on ϕ is given by $\boldsymbol{w}^\top\phi(x)$ where $\boldsymbol{w} \in \mathbb{R}^d$. The supervised loss of ϕ w.r.t the task $\mathcal{T}$ is then:

$$\mathcal{L}_{sup}(\mathcal{T}, \phi) := \min_{\boldsymbol{w}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_\mathcal{T}}\left[\ell\left(\boldsymbol{w}^\top\phi(x), y\right)\right], \tag{4.1}$$

where $\mathcal{D}_\mathcal{T}(x, y)$ is the distribution of data (x, y) in task $\mathcal{T}$. In few-shot learning with novel

classes, there are *limited labeled data points* for learning the linear classifier. Further, the target task $\mathcal{T}_0$ may contain *classes different from those in pretraining*. We are interested in obtaining a model ϕ such that $\mathcal{L}_{sup}(\mathcal{T}_0, \phi)$ is small.

Multitask Finetuning. In the challenging setting of few-shot learning, the data in the target task is limited. On the other hand, we can have prior knowledge of the target task characteristics and its associated data patterns, and thus can collect additional data from relevant and accessible sources when available. Such data may cover the patterns in target task and thus can be used as auxiliary tasks to finetune the pretrained model before adaptation to the target task. Here we formalize this idea in a general form and provide analysis in later sections. Formally, suppose we have M auxiliary tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$, each with m labeled samples $\mathcal{S}_i := \{(x_j^i, y_j^i) : j \in [m]\}$. The finetuning data are $\mathcal{S} := \cup_{i\in[M]}\mathcal{S}_i$. Given a pretrained model $\hat{\phi}$, we further finetune it using the objective:

$$\min_{\phi\in\Phi} \frac{1}{M}\sum_{i=1}^{M} \widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi), \quad \text{where } \widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi) := \min_{\boldsymbol{w}_i\in\mathbb{R}^d} \frac{1}{m}\sum_{j=1}^{m} \ell(\boldsymbol{w}_i^\top \phi(x_j^i), y_j^i). \tag{4.2}$$

This can be done via gradient descent from the initialization $\hat{\phi}$ (see Algorithm 8 in the Appendix). Multitask finetuning is conceptually simple, and broadly applicable to different models and datasets. While its effectiveness has been previously demonstrated [191, 274, 320, 128, 50, 181, 233, 189], the theoretical justification remains to be fully investigated and understood.

4.2 Theoretical Analysis: Benefit of Multitask Finetuning

To understand the potential benefit of multitask finetuning, we will compare the performance of $\hat{\phi}$ (from pretraining) and ϕ' (from pretraining and multitask finetuning) on a target task $\mathcal{T}_0$. That is, we will compare $\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi})$ and $\mathcal{L}_{sup}(\mathcal{T}_0, \phi')$, where $\mathcal{L}_{sup}(\mathcal{T}, \phi)$ is the population supervised loss of ϕ on the task $\mathcal{T}$ defined in Equation (4.1). For the analysis, we first formalize the data distributions and learning models, then introduce the key notions,

and finally present the key theorems.

Data Distributions. Let $\mathcal{X}$ be the input space and $\overline{\mathcal{Z}} \subseteq \mathbb{R}^d$ be the output space of the foundation model. Following [19], suppose there is a set of latent classes $\mathcal{C}$ with $|\mathcal{C}| = K$, and a distribution η over the classes; each class $y \in \mathcal{C}$ has a distribution $\mathcal{D}(y)$ over inputs x. In pretraining using contrastive learning, the distribution $\mathcal{D}_{\text{con}}(\eta)$ of the contrastive data (x, x^+, x^-) is given by: $(y, y^-) \sim \eta^2$ and $x, x^+ \sim \mathcal{D}(y),\ x^- \sim \mathcal{D}(y^-)$. In masked self-supervised or fully supervised pretraining, (x, y) is generated by $y \sim \eta, x \sim \mathcal{D}(y)$. In a task $\mathcal{T}$ with binary classes $\{y_1, y_2\}$, the data distribution $\mathcal{D}_{\mathcal{T}}(x, y)$ is by first uniformly drawing $y \in \{y_1, y_2\}$ and then drawing $x \sim \mathcal{D}(y)$. Finally, let ζ denote the conditional distribution of $(y_1, y_2) \sim \eta^2$ conditioned on $y_1 \neq y_2$, and suppose the tasks in finetuning are from ζ. Note that in few-shot learning with novel classes, the target task's classes may not be the same as those in the pretraining. Let $\mathcal{C}_0$ be the set of possible classes in the target task, which may or may not overlap with $\mathcal{C}$.

Learning Models. Recall that Φ is the hypothesis class of foundation models $\phi : \mathcal{X} \to \overline{\mathcal{Z}}$. To gauge the generalization performance, let $\phi^* \in \Phi$ denote the model with the lowest target task loss $\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*)$ and $\phi^*_\zeta \in \Phi$ denote the model with the lowest average supervised loss over the set of auxiliary tasks $\mathcal{L}_{sup}(\phi^*_\zeta) := \mathbb{E}_{\mathcal{T}\sim\zeta}[\mathcal{L}_{sup}(\mathcal{T}, \phi^*_\zeta)]$. Note that if all $\phi \in \Phi$ have high supervised losses, we cannot expect the method to lead to a good generalization performance, and thus we need to calibrate w.r.t. ϕ^* and ϕ^*_ζ. We also need some typical regularity assumptions.

Assumption 4.2.1 (Regularity Assumptions)**.** $\|\phi\|_2 \leq R$ and linear operator $\|\boldsymbol{w}\|_2 \leq B$. The loss ℓ_u is bounded in $[0, C]$ and L-Lipschitz. The supervised loss $\mathcal{L}_{sup}(\mathcal{T}, \phi)$ is $\tilde{L}$-Lipschitz with respect to ϕ.

Diversity and Consistency. Central to our theoretical analysis lies in the definitions of *diversity* in auxiliary tasks used for finetuning and their *consistency* with the target task.

Definition 1 (Diversity). The averaged representation difference for two model $\phi, \tilde{\phi}$ on a distribution ζ over tasks is $\bar{d}_\zeta(\phi, \tilde{\phi}) := \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T}, \phi) - \mathcal{L}_{sup}(\mathcal{T}, \tilde{\phi})\right] = \mathcal{L}_{sup}(\phi) - \mathcal{L}_{sup}(\tilde{\phi})$.

The worst-case representation difference between representations $\phi, \tilde{\phi}$ on the family of classes $\mathcal{C}_0$ is $d_{\mathcal{C}_0}(\phi, \tilde{\phi}) := \sup_{\mathcal{T}_0 \subseteq \mathcal{C}_0} \left| \mathcal{L}_{sup}(\mathcal{T}_0, \phi) - \mathcal{L}_{sup}(\mathcal{T}_0, \tilde{\phi}) \right|$. We say the model class Φ has ν-diversity (for ζ and $\mathcal{C}_0$) with respect to ϕ^*_ζ, if for any $\phi \in \Phi$, $d_{\mathcal{C}_0}(\phi, \phi^*_\zeta) \leq \bar{d}_\zeta(\phi, \phi^*_\zeta)/\nu$.

Such diversity notion has been proposed and used to derive statistical guarantees (e.g., [267, 317]). Intuitively, diversity measures whether the data from ζ covers the characteristics of the target data in $\mathcal{C}_0$, e.g., whether the span of the linear mapping solutions $\boldsymbol{w}$'s for tasks from ζ can properly cover the solutions for tasks from $\mathcal{C}_0$ [317]. Existing work showed that diverse pretraining data will lead to a large diversity parameter ν and can improve the generalization in the target task. Our analysis will show the diversity in finetuning tasks from ζ can benefit the performance of a target task from $\mathcal{C}_0$.

Definition 2 (Consistency). We say the model class Φ has κ-consistency (for ζ and $\mathcal{C}_0$) with respect to ϕ^* and ϕ^*_ζ, where $\kappa := \sup_{\mathcal{T}_0 \subseteq \mathcal{C}_0} \left[\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \right]$.

This consistency notion measures the similarity between the data in tasks from ζ and the data in the target task from $\mathcal{C}_0$. Intuitively, when the tasks from ζ are similar to the target task $\mathcal{T}_0$, their solutions ϕ^*_ζ and ϕ^* will be similar to each other, resulting in a small κ. Below we will derive guarantees based on the diversity ν and consistency κ to explain the gain from multitask finetuning.

Key Results. We now present the results for a uniform distribution η, and include the full proof and results for general distributions in Appendix C.2 and Appendix C.3. Recall that we will compare the performance of $\hat{\phi}$ (the model from pretraining) and ϕ' (the model from pretraining followed by multitask finetuning) on a target task $\mathcal{T}_0$. For $\hat{\phi}$ without multitask finetuning, we have:

Theorem 4.2.1. (No Multitask Finetuning) Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^* and ϕ^*_ζ. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{pre}(\hat{\phi}) \leq \epsilon_0$. Let $\tau := \Pr_{(y_1, y_2) \sim \eta^2}\{y_1 = y_2\}$. Then for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu} \left[\frac{2\epsilon_0}{1-\tau} - \mathcal{L}_{sup}(\phi^*_\zeta) \right] + \kappa. \tag{4.3}$$

In Theorem 4.2.1, $\widehat{\mathcal{L}}_{pre}(\phi)$ is the empirical loss of $\mathcal{L}_{pre}(\phi)$ with pretraining sample size N. We now consider ϕ' obtained by multitask finetuning. Define the subset of models with pretraining loss smaller than $\tilde{\epsilon}$ as $\Phi(\tilde{\epsilon}) := \left\{\phi \in \Phi : \widehat{\mathcal{L}}_{pre}(\phi) \leq \tilde{\epsilon}\right\}$. Recall the Rademacher complexity of Φ on n points is $\mathcal{R}_n(\Phi) := \underset{\{\sigma_j\}_{j=1}^n, \{x_j\}_{j=1}^n}{\mathbb{E}} \left[\sup_{\phi\in\Phi} \sum_{j=1}^n \sigma_j \phi(x_j)\right]$.

Theorem 4.2.2 below showing that the target prediction performance of the model ϕ' from multitask finetuning can be significantly better than that of $\hat{\phi}$ without multitask finetuning. In particular, achieves an error reduction $\frac{1}{\nu}\left[(1-\alpha)\frac{2\epsilon_0}{1-\tau}\right]$. The reduction is achieved when multitask finetuning is solved to a small loss ϵ_1 for a small α on sufficiently many finetuning data.

Theorem 4.2.2. (With Multitask Finetuning) Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^* and ϕ_ζ^*. Suppose for some constant $\alpha \in (0,1)$, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{2\epsilon_0}{1-\tau}$ and obtain ϕ'. For any $\delta > 0$, if for $\tilde{\epsilon} = \widehat{\mathcal{L}}_{pre}(\phi')$,

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[16LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\alpha\frac{2\epsilon_0}{1-\tau} - \mathcal{L}_{sup}(\phi_\zeta^*)\right] + \kappa. \tag{4.4}$$

The requirement is that the number of tasks M and the total number of labeled samples Mm across tasks are sufficiently large. This implies when M is above the threshold, the total size Mm determines the performance, and increasing either M or m while freezing the other can improve the performance. We shall verify these findings in our experiments (Section 4.3.1).

Theorem 4.2.2 also shows the conditions for successful multitask finetuning, in particular, the impact of the diversity and consistency of the finetuning tasks. Besides small finetuning loss on sufficiently many data, a large diversity parameter ν and a small consistency parameter κ will result in a small target error bound. Ideally, data from the finetuning

tasks should be similar to those from the target task, but also sufficiently diverse to cover a wide range of patterns that may be encountered in the target task. This inspires us to perform finer-grained analysis of diversity and consistency using a simplified data model (Section 4.2.1), which sheds light on the design of an algorithm to select a subset of finetuning tasks with better performance (Section 4.2.2).

4.2.1 Case Study of Diversity and Consistency

Our main results, rooted in notions of diversity and consistency, state the general conclusion of multitask finetuning on downstream tasks. A key remaining question is how relevant tasks should be selected for multitask finetuning in practice. Our intuition is that this task selection should promote both diversity (encompassing the characteristics of the target task) and consistency (focusing on the relevant patterns in achieving the target task's objective). To illustrate such theoretical concepts and connect them to practical algorithms, we specialize the general conclusion to settings that allow easy interpretation of diversity and consistency. In this section, we provide a toy linear case study and we put the proof and also the analysis of a more general setting in Appendix C.4, e.g., more general latent class $\mathcal{C}, \mathcal{C}_0$, more general distribution ζ, input data with noise.

In what follows, we specify the data distributions and function classes under consideration, and present an analysis for this case study. Our goal is to explain the intuition behind diversity and consistency notions: *diversity is about coverage, and consistency is about similarity in the latent feature space.* This can facilitate the design of task selection algorithms.

Linear Data and Tasks. Inspired by classic dictionary learning and recent analysis on representation learning [290, 245], we consider the latent class/representation setting where each latent class $z \in \{0, -1, +1\}^d$ is represented as a feature vector. We focus on individual bi-

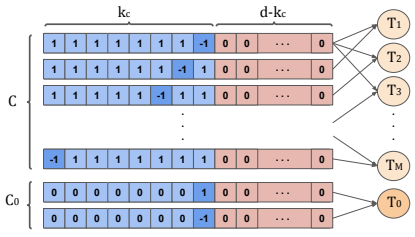


Figure 4.1: Illustration of features in linear data. Blue are the features encoded in $\mathcal{C}$ while red is not.

nary classification tasks, where $\mathcal{Y} = \{-1, +1\}$

is the label space. Thus, each task has two latent classes z, z' (denote the task as $\mathcal{T}_{z,z'}$) and we randomly assign -1 and $+1$ to each latent class. Namely, $\mathcal{T}_{z,z'}$ is defined as: $x = \begin{cases} z, & \text{if } y = -1 \\ z', & \text{if } y = +1 \end{cases}$. We show a diagram in Figure 4.1, we denote each task containing two latent classes, namely (z, z'). Each task in diagram can be represented as (T_1 to $T_{z_1,z_1'}$, T_2 to $T_{z_2,z_2'}$). We further assume a balanced class setting in all tasks, i.e., $p(y = -1) = p(y = +1) = \frac{1}{2}$. Now, we define the latent classes seen in multitask finetuning tasks: $\mathcal{C} = \left\{ (\underbrace{1, 1, \ldots, 1, 1, -1}_{k_\mathcal{C}}, \underbrace{0, \ldots, 0}_{d-k_\mathcal{C}})^\top, (\underbrace{1, 1, \ldots, 1, -1, 1}_{k_\mathcal{C}}, \underbrace{0, \ldots, 0}_{d-k_\mathcal{C}})^\top, \ldots, (\underbrace{-1, 1, \ldots, 1, 1, 1}_{k_\mathcal{C}}, \underbrace{0, \ldots, 0}_{d-k_\mathcal{C}})^\top \right\}$. Note that their feature vectors only encode the first $k_\mathcal{C}$ features, and $|\mathcal{C}| = k_\mathcal{C}$. We let $\mathcal{C}_0 := \{z^{(1)}, z^{(2)}\} \subseteq \{0, -1, +1\}^d$ which is used for the target task, and assume that $z^{(1)}$ and $z^{(2)}$ only differ in 1 dimension, i.e., the target task can be done using this one particular dimension. Let ζ be a distribution uniformly sampling two different latent classes from $\mathcal{C}$. Then, our data generation pipeline for getting a multitask finetuning task is (1) sample two latent classes $(z, z') \sim \zeta$; (2) assign label $-1, +1$ to two latent classes.

Linear Model and Loss Function. We consider a linear model class with regularity Assumption 4.2.1, i.e., $\Phi = \{\phi \in \mathbb{R}^{d\times d} : \|\phi\|_F \leq 1\}$ and linear head $w \in \mathbb{R}^d$ where $\|w\|_2 \leq 1$. Thus, the final output of the model and linear head is $w^\top \phi x$. We use the loss in [245], i.e., $\ell\left(w^\top \phi x, y\right) = -y w^\top \phi x$.

Remark 4.2.2. Although we have linear data, linear model, and linear loss, $\mathcal{L}_{sup}(\phi)$ is a non-linear function on ϕ as the linear heads are different across tasks, i.e., each task has its own linear head.

Now we can link our diversity and consistency to features encoded by training or target tasks.

Theorem 4.2.3 (Diversity and Consistency). If $\mathcal{C}$ encodes the feature in $\mathcal{C}_0$, i.e., the different entry dimension of $z^{(1)}$ and $z^{(2)}$ in $\mathcal{C}_0$ is in the first $k_\mathcal{C}$ dimension, then we have ν is lower bounded by constant $\tilde{c} \geq \frac{2\sqrt{2}-2}{k_\mathcal{C}-1}$ and $\kappa \leq 1 - \sqrt{\frac{1}{k_\mathcal{C}}}$. Otherwise, we have $\nu \to 0$ and $\kappa \geq 1$.

Theorem 4.2.3 establishes $\tilde{c}$-diversity and κ-consistency in Definition 1 and Definition 2.

The analysis shows that diversity can be intuitively understood as the coverage of the finetuning tasks on the target task in the latent feature space: If the key feature dimension of the target task is covered by the features encoded by finetuning tasks, then we have lower-bounded diversity ν; if not covered, then the diversity ν tends to 0 (leading to vacuous error bound in Theorem 4.2.2). Also, consistency can be intuitively understood as similarity in the feature space: when $k_{\mathcal{C}}$ is small, a large fraction of the finetuning tasks are related to the target task, leading to a good consistency (small κ); when $k_{\mathcal{C}}$ is large, we have less relevant tasks, leading to a worse consistency. Such an intuitive understanding of diversity and consistency will be useful for designing practical task selection algorithms.

4.2.2 Task Selection

Our analysis suggests that out of a pool of candidate tasks, a subset S with good consistency (i.e., small κ) and large diversity (i.e., large ν) will yield better generalization to a target task. To realize this insight, we present a greedy selection approach, which sequentially adds tasks with the best consistency, and stops when there is no significant increase in the diversity of the selected subset. In doing so, our approach avoids enumerating all possible subsets and thus is highly practical.

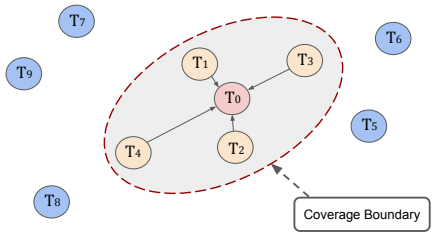


Figure 4.2: Illustration of the similarity and coverage. Target tasks ($\mathcal{T}_0$) with the most similar tasks in yellow and the rest in blue. The ellipsoid spanned by yellow tasks is the coverage for the target task. Adding more tasks in blue to the ellipsoid does not increase the coverage boundary.

A key challenge is to compute the consistency and diversity of the data. While the exact computation deems infeasible, we turn to approximations that capture the key notions of consistency and diversity. We show a simplified diagram for task selection in Figure 4.2. Specifically, given a foundation model ϕ, we assume any task data $\mathcal{T} = \{x_j\}$ follows a Gaussian distribution in the representation space: let $\phi(\mathcal{T}) = \{\phi(x_j)\}$ denote the representation vectors obtained by applying ϕ on the data points in $\mathcal{T}$; compute the

sample mean $\mu_{\mathcal{T}}$ and covariance $C_{\mathcal{T}}$ for $\phi(\mathcal{T})$, and view it as the Gaussian $\mathcal{N}(\mu_{\mathcal{T}}, C_{\mathcal{T}})$. Further, following the intuition shown in the case study, we simplify consistency to similarity: for the target task $\mathcal{T}_0$ and a candidate task $\mathcal{T}_i$, if the cosine similarity $\text{CosSim}(\mathcal{T}_0, \mathcal{T}_i) := \mu_{\mathcal{T}_0}^\top \mu_{\mathcal{T}_i} / (\|\mu_{\mathcal{T}_0}\|_2 \|\mu_{\mathcal{T}_i}\|_2)$ is large, we view $\mathcal{T}_i$ as consistent with $\mathcal{T}_0$. Next, we simplify diversity to coverage: if a dataset D (as a collection of finetuning tasks) largely "covers" the target data $\mathcal{T}_0$, we view D as diverse for $\mathcal{T}_0$. Regarding the task data as Gaussians, we note that the covariance ellipsoid of D covers the target data $\mu_{\mathcal{T}_0}$ iff $(\mu_D - \mu_{\mathcal{T}_0})^T C_D^{-1} (\mu_D - \mu_{\mathcal{T}_0}) \leq 1$. This inspires us to define the following coverage score as a heuristic for diversity: $\text{coverage}(D; \mathcal{T}_0) := 1/(\mu_D - \mu_{\mathcal{T}_0})^\top C_D^{-1} (\mu_D - \mu_{\mathcal{T}_0})$.

Using these heuristics, we arrive at the following selection algorithm: sort the candidate task in descending order of their cosine similarities to the target data; sequentially add tasks in the sorted order to L until $\text{coverage}(L; \mathcal{T}_0)$ has no significant increase. Algorithm 2 illustrates this key idea.

Algorithm 2 Consistency-Diversity Task Selection

Input: Target task $\mathcal{T}_0$, candidate finetuning tasks: $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$, model ϕ, threshold p.

1: Compute $\phi(\mathcal{T}_i)$ and $\mu_{\mathcal{T}_i}$ for $i = 0, 1, \ldots, M$.
2: Sort $\mathcal{T}_i$'s in descending order of similarity $(\mathcal{T}_0, \mathcal{T}_i)$. Denote the sorted list as $\{\mathcal{T}_1', \mathcal{T}_2', \ldots, \mathcal{T}_M'\}$.
3: $L \leftarrow \{\mathcal{T}_1'\}$
4: **for** $i = 2, \ldots, M$ **do**
5: If $\text{coverage}(L \cup \mathcal{T}_i'; \mathcal{T}_0) \geq (1+p) \cdot \text{coverage}(L; \mathcal{T}_0)$, then $L \leftarrow L \cup \mathcal{T}_i'$; otherwise, break.
6: **end for**

Output: selected data L for multitask finetuning.

4.3 Experiments

We now present our main results, organized in three parts. Section 4.3.1 explores how different numbers of finetuning tasks and samples influence the model's performance, offering empirical backing to our theoretical claims. Section 4.3.2 investigates whether our task selection algorithm can select suitable tasks for multitask finetuning. Section 4.3.3 provides a more extensive exploration of the effectiveness of multitask finetuning on various datasets

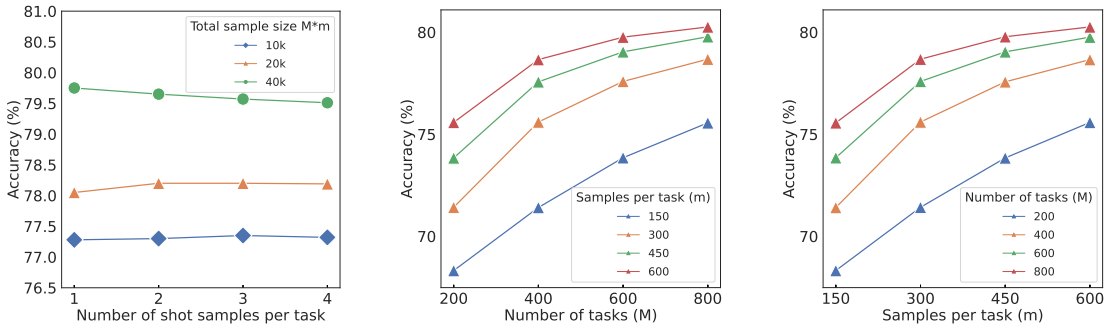
and pretrained models. We defer other results to the appendix. Specifically, Appendix C.5.4 shows that better diversity and consistency of finetuning tasks yield improved performance on target tasks under same sample complexity. Appendix C.5.4 shows that finetuning tasks satisfying diversity yet without consistency lead to no performance gain even with increased sample complexity. Further, Appendix C.6 and Appendix C.7 present additional experiments using NLP and vision-language models, respectively.

Experimental Setup. We use four few-shot learning benchmarks: miniImageNet [271], tieredImageNet [228], DomainNet [218] and Meta-dataset [265]. We use foundation models with different pretraining schemes (MoCo-v3 [49], DINO-v2 [208], and supervised learning with ImageNet [231]) and architectures (ResNet [121] and ViT [73]). We consider few-shot tasks consisting of N classes with K support samples and Q query samples per class (known as N-way K-shot). The goal is to classify the query samples based on the support samples. Tasks used for finetuning are constructed by samples from the training split. Each task is formed by randomly sampling 15 classes, with every class drawing 1 or 5 support samples and 10 query samples. Target tasks are similarly constructed from the test set. We follow [51] for multitask finetuning and target task adaptation. During multitask finetuning, we update all parameters in the model using a nearest centroid classifier, in which all samples are encoded, class centroids are computed, and cosine similarity between a query sample and those centroids are treated as the class logits For adaptation to a target task, we only retain the model encoder and consider a similar nearest centroid classifier. This multitask finetuning protocol applies to all experiments (Sections 4.3.1 to 4.3.3). We provide full experimental set up in Appendix C.5.

4.3.1 Verification of Theoretical Analysis

We conduct experiments on the tieredImageNet dataset to confirm the key insight from our theorem — the impact of the number of finetuning tasks (M) and the number of samples per task (m).

Results. We first investigate the influence of the number of shots. We fix the target



(a) # shots during finetuning. (b) # tasks during finetuning. (c) # samples during finetuning.

Figure 4.3: Results on ViT-B backbone pretrained by MoCo v3. (a) Accuracy v.s. number of shots per finetuning task. Different curves correspond to different total numbers of samples Mm. (b) Accuracy v.s. the number of tasks M. Different curves correspond to different numbers of samples per task m. (c) Accuracy v.s. number of samples per task m. Different curves correspond to different numbers of tasks M.

task as a 1-shot setting but vary the number of shots from 1 to 4 in finetuning, and vary the total sample size $Mm = [10k, 20k, 40k]$. The results in Figure 4.3a show no major change in accuracy with varying the number of shots in finetuning. It is against the common belief that meta-learning like Prototypical Networks [249] has to mimic the exact few-shot setting and that a mismatch will hurt the performance. The results also show that rather than the number of shots, the total sample size Mm determines the performance, which is consistent with our theorem. We next investigate the influence of M and m. We vary the number of tasks ($M = [200, 400, 600, 800]$) and samples per task ($m = [150, 300, 450, 600]$) while keeping all tasks have one shot sample. Figure 4.3b shows increasing M with fixed m improves accuracy, and Figure 4.3c shows increasing m with fixed M has similar behavior. Furthermore, different configurations of M and m for the same total sample size Mm have similar performance (e.g., $M = 400, m = 450$ compared to $M = 600, m = 300$ in Figure 4.3b). These again align with our theorem.

4.3.2 Task Selection

Setup. To evaluate our task selection Algorithm 2, we use the Meta-Dataset [265]. It contains 10 extensive public image datasets from various domains, each partitioned into train/val/test splits. For each dataset except Describable Textures due to small size, we

Pretrained	Selection	INet	Omglot	Acraft	CUB	QDraw	Fungi	Flower	Sign	COCO
CLIP	Random	56.29	65.45	31.31	59.22	36.74	31.03	75.17	33.21	30.16
	No Con.	60.89	72.18	31.50	66.73	40.68	35.17	81.03	37.67	34.28
	No Div.	56.85	73.02	32.53	65.33	40.99	33.10	80.54	34.76	31.24
	Selected	**60.89**	**74.33**	**33.12**	**69.07**	**41.44**	**36.71**	**80.28**	**38.08**	**34.52**
DINOv2	Random	83.05	62.05	36.75	93.75	39.40	52.68	98.57	31.54	47.35
	No Con.	83.21	76.05	36.32	93.96	50.76	53.01	98.58	34.22	47.11
	No Div.	82.82	79.23	36.33	93.96	55.18	52.98	98.59	35.67	44.89
	Selected	**83.21**	**81.74**	**37.01**	**94.10**	**55.39**	**53.37**	**98.65**	**36.46**	**48.08**
MoCo v3	Random	59.66	60.72	18.57	39.80	40.39	32.79	58.42	33.38	32.98
	No Con.	59.80	60.79	18.75	40.41	40.98	32.80	59.55	34.01	33.41
	No Div.	59.57	63.00	18.65	40.36	41.04	32.80	58.67	34.03	33.67
	Selected	**59.80**	**63.17**	**18.80**	**40.74**	**41.49**	**33.02**	**59.64**	**34.31**	**33.86**

Table 4.1: Results evaluating our task selection algorithm on Meta-dataset using ViT-B backbone. No Con.: Ignore consistency. No Div.: Ignore diversity. Random: Ignore both consistency and diversity.

conduct an experiment, where the test-split of that dataset is used as the target task while the train-split from all the other datasets are used as candidate finetuning tasks. Each experiment follows the experiment protocol in Section 4.3. We performed ablation studies on the task selection algorithm, concentrating on either consistency or diversity, while violating the other. See details in Appendix C.5.4.

Results. Table 4.1 compares the results from finetuning with tasks selected by our algorithm to those from finetuning with tasks selected by other methods. Our algorithm consistently attains performance gains. For instance, on Omniglot, our algorithm leads to significant accuracy gains over random selection of 8.9%, 19.7%, and 2.4% with CLIP, DINO v2, and MoCo v3, respectively. Violating consistency or diversity conditions generally result in a reduced performance compared to our approach. These results are well aligned with our expectations and affirm our diversity and consistency conclusions. We provide more ablatioin study on task selection in Table C.7 in Appendix C.5.4. We also apply task selection algorithm on DomainNet in Appendix C.5.5. Furthermore, in Appendix C.6, we employ our algorithm for NLP models on the GLUE dataset.

			miniImageNet		tieredImageNet		DomainNet	
pretrained	**backbone**	**method**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
MoCo v3	ViT-B	Adaptation	75.33 (0.30)	92.78 (0.10)	62.17 (0.36)	83.42 (0.23)	24.84 (0.25)	44.32 (0.29)
		Standard FT	75.38 (0.30)	92.80 (0.10)	62.28 (0.36)	83.49 (0.23)	25.10 (0.25)	44.76 (0.27)
		Ours	**80.62** (0.26)	**93.89** (0.09)	**68.32** (0.35)	**85.49** (0.22)	**32.88** (0.29)	**54.17** (0.30)
	ResNet50	Adaptation	68.80 (0.30)	88.23 (0.13)	55.15 (0.34)	76.00 (0.26)	27.34 (0.27)	47.50 (0.28)
		Standard FT	68.85 (0.30)	88.23 (0.13)	55.23 (0.34)	76.07 (0.26)	27.43 (0.27)	47.65 (0.28)
		Ours	**71.16** (0.29)	**89.31** (0.12)	**58.51** (0.35)	**78.41** (0.25)	**33.53** (0.30)	**55.82** (0.29)
DINO v2	ViT-S	Adaptation	85.90 (0.22)	95.58 (0.08)	74.54 (0.32)	89.20 (0.19)	52.28 (0.39)	72.98 (0.28)
		Standard FT	86.75 (0.22)	95.76 (0.08)	74.84 (0.32)	89.30 (0.19)	54.48 (0.39)	74.50 (0.28)
		Ours	**88.70** (0.22)	**96.08** (0.08)	**77.78** (0.32)	**90.23** (0.18)	**61.57** (0.40)	**77.97** (0.27)
	ViT-B	Adaptation	90.61 (0.19)	97.20 (0.06)	82.33 (0.30)	92.90 (0.16)	61.65 (0.41)	79.34 (0.25)
		Standard FT	91.07 (0.19)	97.32 (0.06)	82.40 (0.30)	93.07 (0.16)	61.84 (0.39)	79.63 (0.25)
		Ours	**92.77** (0.18)	**97.68** (0.06)	**84.74** (0.30)	**93.65** (0.16)	**68.22** (0.40)	**82.62** (0.24)
Supervised pretraining on ImageNet	ViT-B	Adaptation	94.06 (0.15)	97.88 (0.05)	83.82 (0.29)	93.65 (0.13)	28.70 (0.29)	49.70 (0.28)
		Standard FT	95.28 (0.13)	98.33 (0.04)	86.44 (0.27)	94.91 (0.12)	30.93 (0.31)	52.14 (0.29)
		Ours	**96.91** (0.11)	**98.76** (0.04)	**89.97** (0.25)	**95.84** (0.11)	**48.02** (0.38)	**67.25** (0.29)
	ResNet50	Adaptation	81.74 (0.24)	94.08 (0.09)	65.98 (0.34)	84.14 (0.21)	27.32 (0.27)	46.67 (0.28)
		Standard FT	84.10 (0.22)	94.81 (0.09)	74.48 (0.33)	88.35 (0.19)	34.10 (0.31)	55.08 (0.29)
		Ours	**87.61** (0.20)	**95.92** (0.07)	**77.74** (0.32)	**89.77** (0.17)	**39.09** (0.34)	**60.60** (0.29)

Table 4.2: **Results of few-shot image classification.** We report average classification accuracy (%) with 95% confidence intervals on test splits. Adaptation: Direction adaptation without finetuning; Standard FT: Standard finetuning; Ours: Our multitask finetuning; 1-/5-shot: number of labeled images per class in the target task.

4.3.3 Effectiveness of Multitask Finetuning

Setup. We also conduct more extensive experiments on large-scale datasets across various settings to confirm the effectiveness of multitask finetuning. We compare to two baselines: *direct adaptation* where we directly adapt pretrained model encoder on target tasks without any finetuning, and *standard finetuning* where we append encoder with a linear head to map representations to class logits and finetune the whole model. During testing, we removed the linear layer and used the same few-shot testing process with the finetuned encoders. Please refer Table C.12 in Appendix C.5.8 for full results.

Results. Table 4.2 presents the results for various pretraining and finetuning methods, backbones, datasets, and few-shot learning settings. Multitask finetuning consistently outperforms the baselines in different settings. For example, in the most challenging setting of 1-shot on DomainNet, it attains a major gain of 7.1% and 9.3% in accuracy over standard finetuning and direct adaptation, respectively, when considering self-supervised pretraining with DINO v2 and using a Transformer model (ViT-S). Interestingly, multitask finetuning achieves more significant gains for models pretrained with supervised learning than those

pretrained with contrastive learning. For example, on DomainNet, multitask finetuning on supervised pretrained ViT-B achieves a relative gain of 67% and 35% for 1- and 5-shot, respectively. In contrast, multitask finetuning on DINO v2 pretrained ViT-B only shows a relative gain of 10% and 4%. This suggests that models from supervised pretraining might face a larger domain gap than models from DINO v2, and multitask finetuning helps to bridge this gap.

Reproducibility Statement

For theoretical results in the Section 4.2, a complete proof is provided in the Appendix C.2. The theoretical results and proofs for a multiclass setting that is more general than that in the main text are provided in the Appendix C.3. The complete proof for linear case study on diversity and consistency is provided in the Appendix C.4. For experiments in the Section 4.3, complete details and experimental results are provided in the Appendices C.5 to C.7. The source code with explanations and comments is provided in https://github.com/OliverXUZY/Foudation-Model_Multitask.

Acknowledgments

The work in this chapter is partially supported by Air Force Grant FA9550-18-1-0166, the National Science Foundation (NSF) Grants 2008559-IIS, CCF-2046710, and 2023239-DMS. The work also received partial support from grants by McPherson Eye Research Institute and VCGRE at UW Madison, and from the Army Research Lab under contract number W911NF-2020221.

Chapter 5

Why Larger Language Models do In-context Learning Differently

In this chapter, we will continue with our investigation on foundation models. We would like to focus on understanding the ICL mechanism of transformer based foundation models, particularly LLMs.

In the regime of fundation models and LLMs, people have observed various unexpected and unexplanable phenomenon. As we have discussed in the introduction, recently there have been some important and surprising observations [182, 213, 288, 237] that cannot be fully explained by existing studies. In particular, [237] finds that LLM is not robust during ICL and can be easily distracted by an irrelevant context. Furthermore, [288] shows that when we inject noise into the prompts, the larger language models may have a worse ICL ability than the small language models, and conjectures that the larger language models may overfit into the prompts and forget the prior knowledge from pretraining, while small models tend to follow the prior knowledge. On the other hand, [182, 213] demonstrate that injecting noise does not affect the in-context learning that much for smaller models, which have a more strong pretraining knowledge bias. To improve the understanding of the ICL mechanism, to shed light on the properties and inner workings of LLMs, and to inspire efficient and safe use of ICL, we will attempt to answer the following question in

this chapter:

Why do larger language models do in-context learning differently?

To answer this question, we study two settings: (1) one-layer single-head linear self-attention network [234, 272, 8, 5, 315, 172, 296] pretrained on linear regression in-context tasks [98, 225, 272, 8, 24, 172, 315, 5, 160, 131, 296], with rank constraint on the attention weight matrices for studying the effect of the model scale; (2) two-layer multiple-head transformers [157] pretrained on sparse parity classification in-context tasks, comparing small or large head numbers for studying the effect of the model scale. In both settings, we give the closed-form optimal solutions. We show that smaller models emphasize important hidden features while larger models cover more features, e.g., less important features or noisy features. Then, we show that smaller models are more robust to label noise and input noise during evaluation, while larger models may easily be distracted by such noises, so larger models may have a worse ICL ability than smaller ones.

We also conduct in-context learning experiments on five prevalent NLP tasks utilizing various sizes of the Llama model families [263, 264], whose results are consistent with previous work [182, 213, 288] and our analysis.

Our contributions and novelty over existing work:

- We formalize new stylized theoretical settings for studying ICL and the scaling effect of LLM. See Section 5.2 for linear regression and Section 5.3 for parity.
- We characterize the optimal solutions for both settings (Theorem 5.2.1 and Theorem 5.3.1).
- The characterizations of the optimal elucidate different attention paid to different hidden features, which then leads to the different ICL behavior (Theorem 5.2.2, Theorem 5.2.3, Theorem 5.3.2).
- We further provide empirical evidence on large base and chat models corroborating our theoretical analysis (Figure 5.1, Figure 5.2).

Note that previous ICL analysis paper may only focus on (1) the approximation power of transformers [98, 216, 117, 24, 53], e.g., constructing a transformer by hands which can do ICL, or (2) considering one-layer single-head linear self-attention network learning ICL on linear regression [272, 8, 172, 315, 5, 296], and may not focus on the robustness analysis or explain the different behaviors. In this work, (1) we extend the linear model linear data analysis to the non-linear model and non-linear data setting, i.e., two-layer multiple-head transformers leaning ICL on sparse parity classification and (2) we have a rigorous behavior difference analysis under two settings, which explains the empirical observations and provides more insights into the effect of attention mechanism in ICL.

This chapter is based on a joint work [247] with Zhenmei Shi and Zhuoyan Xu:

> Zhenmei Shi, Junyi Wei, Zhuoyan Xu, Yingyu Liang, "Why Larger Language Models Do In-context Learning Differently?", *International Conference on Machine Learning (ICML) 2024*.

Contributions of the author: Zhenmei Shi has main contribution towards the work. The author Junyi Wei has core contribution in proving several major lemmas and theorems. Zhenmei Shi and Zhuoyan Xu may submit this work for other degree or professional qualification.

5.1 Preliminary

Notations. We denote $[n] := \{1, 2, \dots, n\}$. For a positive semidefinite matrix $\mathbf{A}$, we denote $\|\mathbf{x}\|_{\mathbf{A}}^2 := \mathbf{x}^\top \mathbf{A}\mathbf{x}$ as the norm induced by a positive definite matrix $\mathbf{A}$. We denote $\|\cdot\|_F$ as the Frobenius norm. diag() function will map a vector to a diagonal matrix or map a matrix to a vector with its diagonal terms.

In-context learning. We follow the setup and notation of the problem in [315, 172, 5, 131, 296]. In the pretraining stage of ICL, the model is pretrained on prompts. A prompt from a task τ is formed by N examples $(\mathbf{x}_{\tau,1}, y_{\tau,1}), \dots, (\mathbf{x}_{\tau,N}, y_{\tau,N})$ and a query token $\mathbf{x}_{\tau,q}$

for prediction, where for any $i \in [N]$ we have $y_{\tau,i} \in \mathbb{R}$ and $\mathbf{x}_{\tau,i}, \mathbf{x}_{\tau,q} \in \mathbb{R}^d$. The embedding matrix $\mathbf{E}_\tau$, the label vector $\mathbf{y}_\tau$, and the input matrix $\mathbf{X}_\tau$ are defined as:

$$\begin{aligned}
\mathbf{E}_\tau :=& \begin{pmatrix} \mathbf{x}_{\tau,1} & \mathbf{x}_{\tau,2} & \dots & \mathbf{x}_{\tau,N} & \mathbf{x}_{\tau,q} \\ y_{\tau,1} & y_{\tau,2} & \dots & y_{\tau,N} & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}, \\
\mathbf{y}_\tau :=& [y_{\tau,1}, \dots, y_{\tau,N}]^\top \in \mathbb{R}^N, \qquad y_{\tau,q} \in \mathbb{R}, \\
\mathbf{X}_\tau :=& [\mathbf{x}_{\tau,1}, \dots, \mathbf{x}_{\tau,N}]^\top \in \mathbb{R}^{N\times d}, \qquad \mathbf{x}_{\tau,q} \in \mathbb{R}^d.
\end{aligned}$$

Given prompts represented as $\mathbf{E}_\tau$'s and the corresponding true labels $y_{\tau,q}$'s, the pretraining aims to find a model whose output on $\mathbf{E}_\tau$ matches $y_{\tau,q}$. After pretraining, the evaluation stage applies the model to a new test prompt (potentially from a different task) and compares the model output to the true label on the query token.

Note that our pretraining stage is also called learning to learn in-context [181] or in-context training warmup [72] in existing work. Learning to learn in-context is the first step to understanding the mechanism of ICL in LLM following previous works [225, 323, 315, 172, 5, 131, 157, 296].

Linear self-attention networks. The linear self-attention network has been widely studied [234, 272, 8, 5, 315, 172, 296, 4], and will be used as the learning model or a component of the model in our two theoretical settings. It is defined as:

$$f_{\mathrm{LSA},\theta}(\mathbf{E}) = \left[\mathbf{E} + \mathbf{W}^{PV}\mathbf{E} \cdot \frac{\mathbf{E}^\top \mathbf{W}^{KQ}\mathbf{E}}{\rho}\right], \tag{5.1}$$

where $\theta = (\mathbf{W}^{PV}, \mathbf{W}^{KQ})$, $\mathbf{E} \in \mathbb{R}^{(d+1)\times(N+1)}$ is the embedding matrix of the input prompt, and ρ is a normalization factor set to be the length of examples, i.e., $\rho = N$ during pretraining. Similar to existing work, for simplicity, we have merged the projection and value matrices into $\mathbf{W}^{PV}$, and merged the key and query matrices into $\mathbf{W}^{KQ}$, and have a residual connection in our LSA network. The prediction of the network for the query token $\mathbf{x}_{\tau,q}$ will be the bottom right entry of the matrix output, i.e., the entry at location $(d+1), (N+1)$, while other entries are not relevant to our study and thus are ignored. So

only part of the model parameters are relevant. To see this, let us denote

$$\mathbf{W}^{PV} = \begin{pmatrix} \mathbf{W}_{11}^{PV} & \mathbf{w}_{12}^{PV} \\ (\mathbf{w}_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$
$$\mathbf{W}^{KQ} = \begin{pmatrix} \mathbf{W}_{11}^{KQ} & \mathbf{w}_{12}^{KQ} \\ (\mathbf{w}_{21}^{KQ})^\top & w_{22}^{KQ} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $\mathbf{W}_{11}^{PV}, \mathbf{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}$; $\mathbf{w}_{12}^{PV}, \mathbf{w}_{21}^{PV}, \mathbf{w}_{12}^{KQ}, \mathbf{w}_{21}^{KQ} \in \mathbb{R}^d$; and $w_{22}^{PV}, w_{22}^{KQ} \in \mathbb{R}$. Then the prediction is:

$$\begin{aligned} \widehat{y}_{\tau,q} =& f_{\mathrm{LSA},\theta}(\mathbf{E})_{(d+1),(N+1)} \\ =& \begin{pmatrix} (\mathbf{w}_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \left(\frac{\mathbf{E}\mathbf{E}^\top}{\rho}\right) \begin{pmatrix} \mathbf{W}_{11}^{KQ} \\ (w_{21}^{KQ})^\top \end{pmatrix} \mathbf{x}_{\tau,q}. \end{aligned} \tag{5.2}$$

5.2 Linear Regression

In this section, we consider the linear regression task for in-context learning which is widely studied empirically [98, 225, 272, 8, 24] and theoretically [172, 315, 5, 160, 131, 296].

Data and task. For each task τ, we assume for any $i \in [N]$ tokens $\mathbf{x}_{\tau,i}, \mathbf{x}_{\tau,q} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda)$, where Λ is the covariance matrix. We also assume a d-dimension task weight $\mathbf{w}_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{d\times d})$ and the labels are given by $y_{\tau,i} = \langle \mathbf{w}_\tau, \mathbf{x}_{\tau,i} \rangle$ and $y_{\tau,q} = \langle \mathbf{w}_\tau, \mathbf{x}_{\tau,q} \rangle$.

Model and loss. We study a one-layer single-head linear self-attention transformer (LSA) defined in Equation (5.1) and we use $\widehat{y}_{\tau,q} := f_{\mathrm{LSA},\theta}(\mathbf{E})_{(d+1),(N+1)}$ as the prediction. We consider the mean square error (MSE) loss so that the empirical risk over B independent prompts is defined as

$$\widehat{\mathcal{L}}(f_\theta) := \frac{1}{2B} \sum_{\tau=1}^{B} \left(\widehat{y}_{\tau,q} - \langle \mathbf{w}_\tau, \mathbf{x}_{\tau,q} \rangle\right)^2.$$

Measure model scale by rank. We first introduce a lemma from previous work that simplifies the MSE and justifies our measurement of the model scale. For notation simplicity, we denote $\mathbf{U} = \mathbf{W}_{11}^{KQ}, u = w_{22}^{PV}$.

Lemma 5.2.1 (Lemma A.1 in [315]). *Let* $\Gamma := \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_{d\times d} \in \mathbb{R}^{d\times d}$. *Let*

$$\begin{aligned}
\mathcal{L}(f_{\mathrm{LSA},\theta}) &= \lim_{B\to\infty} \widehat{\mathcal{L}}(f_{\mathrm{LSA},\theta}) \\
&= \frac{1}{2}\mathbb{E}_{\mathbf{w}_\tau, \mathbf{x}_{\tau,1},\ldots,\mathbf{x}_{\tau,N},\mathbf{x}_{\tau,q}}\left[\left(\widehat{y}_{\tau,q} - \langle \mathbf{w}_\tau, \mathbf{x}_{\tau,q}\rangle\right)^2\right], \\
\tilde{\ell}(\mathbf{U}, u) &= \operatorname{tr}\left[\frac{1}{2}u^2\Gamma\Lambda\mathbf{U}\Lambda\mathbf{U}^\top - u\Lambda^2\mathbf{U}^\top\right],
\end{aligned}$$

we have $\mathcal{L}(f_{\mathrm{LSA},\theta}) = \tilde{\ell}(\mathbf{U}, u) + C$, *where* C *is a constant independent with* θ.

Lemma 5.2.1 tells us that the loss only depends on $u\mathbf{U}$. If we consider non-zero u, w.l.o.g, letting $u = 1$, then we can see that the loss only depends on $\mathbf{U} \in \mathbb{R}^{d\times d}$,

$$\mathcal{L}(f_{\mathrm{LSA},\theta}) = \operatorname{tr}\left[\frac{1}{2}\Gamma\Lambda\mathbf{U}\Lambda\mathbf{U}^\top - \Lambda^2\mathbf{U}^\top\right].$$

Note that $\mathbf{U} = \mathbf{W}_{11}^{KQ}$, then it is natural to measure the size of the model by rank of $\mathbf{U}$. Recall that we merge the key matrix and the query matrix in attention together, i.e., $\mathbf{W}^{KQ} = (\mathbf{W}^K)^\top\mathbf{W}^Q$. Thus, a low-rank $\mathbf{U}$ is equivalent to the constraint $\mathbf{W}^K, \mathbf{W}^Q \in \mathbb{R}^{r\times d}$ where $r \ll d$. The low-rank key and query matrix are practical and have been widely studied [126, 44, 30, 78, 67, 243]. Therefore, we use $r = \operatorname{rank}(\mathbf{U})$ to measure the scale of the model, i.e., larger r representing larger models. To study the behavior difference under different model scale, we will analyze $\mathbf{U}$ under different rank constraints.

5.2.1 Low Rank Optimal Solution

Since the token covariance matrix Λ is positive semidefinite symmetric, we have eigendecomposition $\Lambda = \mathbf{Q}\mathbf{D}\mathbf{Q}^\top$, where $\mathbf{Q}$ is an orthonormal matrix containing eigenvectors of Λ and $\mathbf{D}$ is a sorted diagonal matrix with non-negative entries containing eigenvalues of Λ, denoting as $\mathbf{D} = \operatorname{diag}([\lambda_1, \ldots, \lambda_d])$, where $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. Then, we have the following theorem.

Theorem 5.2.1 (Optimal rank-r solution for regression). Recall the loss function $\tilde{\ell}$ in

Lemma 5.2.1. Let

$$\mathbf{U}^*, u^* = \underset{\mathbf{U}\in\mathbb{R}^{d\times d},\mathrm{rank}(\mathbf{U})\le r, u\in\mathbb{R}}{\arg\min} \tilde{\ell}(\mathbf{U}, u).$$

Then $\mathbf{U}^* = c\mathbf{Q}\mathbf{V}^*\mathbf{Q}^\top, u = \frac{1}{c}$, where c is any nonzero constant, and $\mathbf{V}^* = \mathrm{diag}([v_1^*, \ldots, v_d^*])$ satisfies for any $i \le r, v_i^* = \frac{N}{(N+1)\lambda_i + \mathrm{tr}(\mathbf{D})}$ and for any $i > r, v_i^* = 0$.

Proof sketch of Theorem 5.2.1. We defer the full proof to Appendix D.2.1. The proof idea is that we can decompose the loss function into different ranks, so we can keep the direction by their sorted "variance", i.e.,

$$\underset{\mathbf{U}\in\mathbb{R}^{d\times d},\mathrm{rank}(\mathbf{U})\le r, u\in\mathbb{R}}{\arg\min} \tilde{\ell}(\mathbf{U}, u) = \sum_{i=1}^{d} T_i \lambda_i^2 \left(v_i^* - \frac{1}{T_i}\right)^2,$$

where $T_i = \left(1 + \frac{1}{N}\right)\lambda_i + \frac{\mathrm{tr}(\mathbf{D})}{N}$. We have that $v_i^* \ge 0$ for any $i \in [d]$ and if $v_i^* > 0$, we have $v_i^* = \frac{1}{T_i}$. Denote $g(x) = x^2 \left(\frac{1}{(1+\frac{1}{N})x + \frac{\mathrm{tr}(\mathbf{D})}{N}}\right)$. We get the conclusion by $g(x)$ is an increasing function on $[0, \infty)$. □

Theorem 5.2.1 gives the closed-form optimal rank-r solution of one-layer single-head linear self-attention transformer learning linear regression ICL tasks. Let $f_{\mathrm{LSA},\theta}$ denote the optimal rank-r solution corresponding to the $\mathbf{U}^*, u^*$ above. In detail, the optimal rank-r solution $f_{\mathrm{LSA},\theta}$ satisfies

$$\mathbf{W}^{*PV} = \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & u \end{pmatrix}, \mathbf{W}^{*KQ} = \begin{pmatrix} \mathbf{U}^* & 0_d \\ 0_d^\top & 0 \end{pmatrix}. \tag{5.3}$$

What hidden features does the model pay attention to? Theorem 5.2.1 shows that the optimal rank-r solution indeed is the truncated version of the optimal full-rank solution, keeping only the most important feature directions (i.e., the first r eigenvectors of the token covariance matrix). In detail, (1) for the optimal full-rank solution, we have for any $i \in [d], v_i^* = \frac{N}{(N+1)\lambda_i + \mathrm{tr}(\mathbf{D})}$; (2) for the optimal rank-r solution, we have for any $i \le r, v_i^* = \frac{N}{(N+1)\lambda_i + \mathrm{tr}(\mathbf{D})}$ and for any $i > r, v_i^* = 0$. That is, the small rank-r model keeps only the first r eigenvectors (viewed as hidden feature directions) and does not cover the

others, while larger ranks cover more hidden features, and the large full rank model covers all features.

Recall that the prediction depends on $\mathbf{U}^*\mathbf{x}_{\tau,q} = c\mathbf{Q}\mathbf{V}^*\mathbf{Q}^\top\mathbf{x}_{\tau,q}$; see Equation (5.2) and (5.3). So the optimal rank-r model only uses the components on the first r eigenvector directions to do the prediction in evaluations. When there is noise distributed in all directions, a smaller model can ignore noise and signals along less important directions but still keep the most important directions. Then it can be less sensitive to the noise, as empirically observed. This insight is formalized in the next subsection.

5.2.2 Behavior Difference

We now formalize our insight into the behavior difference based on our analysis on the optimal solutions. We consider the evaluation prompt to have M examples (may not be equal to the number of examples N during pretraining for a general evaluation setting), and assume noise in labels to facilitate the study of the behavior difference (our results can be applied to the noiseless case by considering noise level $\sigma = 0$). Formally, the evaluation prompt is:

$$\begin{aligned}\widehat{\mathbf{E}} &:= \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_M & \mathbf{x}_q \\ y_1 & y_2 & \dots & y_M & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(M+1)} \\ &= \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_M & \mathbf{x}_q \\ \langle \mathbf{w}, \mathbf{x}_1 \rangle + \epsilon_1 & \dots & \langle \mathbf{w}, \mathbf{x}_M \rangle + \epsilon_M & 0 \end{pmatrix},\end{aligned}$$

where $\mathbf{w}$ is the weight for the evaluation task, and for any $i \in [M]$, the label noise $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$.

Recall $\mathbf{Q}$ are eigenvectors of Λ, i.e., $\Lambda = \mathbf{Q}\mathbf{D}\mathbf{Q}^\top$ and $\mathbf{D} = \mathrm{diag}([\lambda_1, \dots, \lambda_d])$. In practice, we can view the large variance part of $\mathbf{x}$ (top r directions in $\mathbf{Q}$) as a useful signal (like words "positive", "negative"), and the small variance part (bottom $d - r$ directions in $\mathbf{Q}$) as the less important or useless information (like words "even", "just").

Based on such intuition, we can decompose the evaluation task weight $\mathbf{w}$ accordingly:

$\mathbf{w} = \mathbf{Q}(\mathbf{s} + \xi)$, where the r-dim truncated vector $\mathbf{s} \in \mathbb{R}^d$ has $\mathbf{s}_i = 0$ for any $r < i \leq d$, and the residual vector $\xi \in \mathbb{R}^d$ has $\xi_i = 0$ for any $1 \leq i \leq r$. The following theorem (proved in Appendix D.2.2) quantifies the evaluation loss at different model scales r which can explain the scale's effect.

Theorem 5.2.2 (Behavior difference for regression). Let $\mathbf{w} = \mathbf{Q}(\mathbf{s} + \xi) \in \mathbb{R}^d$ where $\mathbf{s}, \xi \in \mathbb{R}^d$ are truncated and residual vectors defined above. The optimal rank-r solution $f_{\mathrm{LSA},\theta}$ in Theorem 5.2.1 satisfies:

$$\begin{aligned}
&\mathcal{L}(f_{\mathrm{LSA},\theta}; \widehat{\mathbf{E}}) \\
:=&\mathbb{E}_{\mathbf{x}_1,\epsilon_1,\ldots,\mathbf{x}_M,\epsilon_M,\mathbf{x}_q} \left(f_{\mathrm{LSA},\theta}(\widehat{\mathbf{E}}) - \langle \mathbf{w}, \mathbf{x}_q \rangle \right)^2 \\
=&\frac{1}{M} \|\mathbf{s}\|^2_{(\mathbf{V}^*)^2\mathbf{D}^3} + \frac{1}{M} \left(\|\mathbf{s} + \xi\|^2_{\mathbf{D}} + \sigma^2 \right) \mathrm{tr}\left((\mathbf{V}^*)^2 \mathbf{D}^2 \right) \\
&+ \|\xi\|^2_{\mathbf{D}} + \sum_{i \in [r]} \mathbf{s}_i^2 \lambda_i \left(\lambda_i v_i^* - 1 \right)^2 .
\end{aligned}$$

Implications. If N is large enough with $N\lambda_r \gg \mathrm{tr}(\mathbf{D})$ (which is practical as we usually pretrain networks on long text), then

$$\mathcal{L}(f_{\mathrm{LSA},\theta}; \widehat{\mathbf{E}}) \approx \|\xi\|^2_{\mathbf{D}} + \frac{1}{M} \left((r+1)\|\mathbf{s}\|^2_{\mathbf{D}} + r\|\xi\|^2_{\mathbf{D}} + r\sigma^2 \right).$$

The first term $\|\xi\|^2_{\mathbf{D}}$ is due to the residual features not covered by the network, so it decreases for larger r and becomes 0 for full-rank $r = d$. The second term $\frac{1}{M}(\cdot)$ is significant since we typically have limited examples in evaluation, e.g., $M = 16 \ll N$. Within it, $(r+1)\|\mathbf{s}\|^2_{\mathbf{D}}$ corresponds to the first r directions, and $r\sigma^2$ corresponds to the label noise. These increase for larger r. So there is a trade-off between the two error terms when scaling up the model: for larger r the first term decreases while the second term increases. This depends on whether more signals are covered or more noise is kept when increasing the rank r.

To further illustrate the insights, we consider the special case when the model already covers all useful signals in the evaluation task: $\mathbf{w} = \mathbf{Q}\mathbf{s}$, i.e., the label only depends on

the top r features (like "positive", "negative" tokens). Our above analysis implies that a larger model will cover more useless features and keep more noise, and thus will have worse performance. This is formalized in the following theorem (proved in Appendix D.2.2).

Theorem 5.2.3 (Behavior difference for regression, special case). Let $0 \leq r \leq r' \leq d$ and $\mathbf{w} = \mathbf{Q}\mathbf{s}$ where $\mathbf{s}$ is r-dim truncated vector. Denote the optimal rank-r solution as f_1 and the optimal rank-r' solution as f_2. Then,

$$\begin{aligned}&\mathcal{L}(f_2; \widehat{\mathbf{E}}) - \mathcal{L}(f_1; \widehat{\mathbf{E}}) \\ =&\frac{1}{M}\left(\|\mathbf{s}\|_{\mathbf{D}}^2 + \sigma^2\right)\left(\sum_{i=r+1}^{r'}\left(\frac{N\lambda_i}{(N+1)\,\lambda_i + \operatorname{tr}(\mathbf{D})}\right)^2\right).\end{aligned}$$

Implications. By Theorem 5.2.3, in this case,

$$\mathcal{L}(f_2; \widehat{\mathbf{E}}) - \mathcal{L}(f_1; \widehat{\mathbf{E}}) \approx \underbrace{\frac{r'-r}{M}\|\mathbf{s}\|_{\mathbf{D}}^2}_{\text{input noise}} + \underbrace{\frac{r'-r}{M}\sigma^2}_{\text{label noise}}.$$

We can decompose the above equation to input noise and label noise, and we know that $\|\mathbf{s}\|_{\mathbf{D}}^2 + \sigma^2$ only depends on the intrinsic property of evaluation data and is independent of the model size. When we have a larger model (larger r'), we will have a larger evaluation loss gap between the large and small models. It means larger language models may be easily affected by the label noise and input noise and may have worse in-context learning ability, while smaller models may be more robust to these noises as they only emphasize important signals. Moreover, if we increase the label noise scale σ^2 on purpose, the larger models will be more sensitive to the injected label noise. This is consistent with the observation in [288, 237] and our experimental results in Section 5.4.

5.3 Sparse Parity Classification

We further consider a more sophisticated setting with nonlinear data which necessitates nonlinear models. Viewing sentences as generated from various kinds of thoughts and knowledge that can be represented as vectors in some hidden feature space, we consider

the classic data model of dictionary learning or sparse coding, which has been widely used for text and images [204, 270, 33]. Furthermore, beyond linear separability, we assume the labels are given by the $(d, 2)$-sparse parity on the hidden feature vector, which is the high-dimensional generalization of the classic XOR problem. Parities are a canonical family of highly non-linear learning problems and recently have been used in many recent studies on neural network learning [64, 27, 240, 242].

Data and task. Let $\mathcal{X} = \mathbb{R}^d$ be the input space, and $\mathcal{Y} = \{\pm 1\}$ be the label space. Suppose $\mathbf{G} \in \mathbb{R}^{d \times d}$ is an unknown dictionary with d columns that can be regarded as features; for simplicity, assume $\mathbf{G}$ is orthonormal. Let $\phi \in \{\pm 1\}^d$ be a hidden vector that indicates the presence of each feature. The data are generated as follows: for each task τ, generate two task indices $\mathbf{t}_\tau = (i_\tau, j_\tau)$ which determines a distribution $\mathcal{T}_\tau$; then for this task, draw examples by $\phi \sim \mathcal{T}_\tau$, and setting $\mathbf{x} = \mathbf{G}\phi$ (i.e., dictionary learning data), $y = \phi_{i_\tau}\phi_{j_\tau}$ (i.e., XOR labels).

We now specify how to generate $\mathbf{t}_\tau$ and ϕ. As some of the hidden features are more important than others, we let $A = [k]$ denote a subset of size k corresponding to the important features. We denote the important task set as $S_1 := A \times A \setminus \{(l, l) : l \in A\}$ and less important task set as $S_2 := [d] \times [d] \setminus (\{(l, l) : l \in [d]\} \cup S_1)$. Then $\mathbf{t}_\tau$ is drawn uniformly from S_1 with probability $1 - p_{\mathcal{T}}$, and uniformly from S_2 with probability $p_{\mathcal{T}}$, where $p_{\mathcal{T}} \in [0, \frac{1}{2})$ is the less-important task rate. For the distribution of ϕ, we assume $\phi_{[d]\setminus\{i_\tau, j_\tau\}}$ is drawn uniformly from $\{\pm 1\}^{d-2}$, and assume $\phi_{\{i_\tau, j_\tau\}}$ has good correlation (measured by a parameter $\gamma \in (0, \frac{1}{4})$) with the label to facilitate learning. Independently, we have

$$\begin{aligned}
\Pr[(\phi_{i_\tau}, \phi_{j_\tau}) = (1, 1)] &= 1/4 + \gamma, \\
\Pr[(\phi_{i_\tau}, \phi_{j_\tau}) = (1, -1)] &= 1/4, \\
\Pr[(\phi_{i_\tau}, \phi_{j_\tau}) = (-1, 1)] &= 1/4, \\
\Pr[(\phi_{i_\tau}, \phi_{j_\tau}) = (-1, -1)] &= 1/4 - \gamma.
\end{aligned}$$

Note that without correlation ($\gamma = 0$), it is well-known sparse parities will be hard to learn, so we consider $\gamma > 0$.

Model. Following [296], we consider the reduced linear self-attention $f_{\mathrm{LSA},\theta}(\mathbf{X}, \mathbf{y}, \mathbf{x}_q) = \frac{\mathbf{y}^\top \mathbf{X}}{N} \mathbf{W}^{KQ} \mathbf{x}_q$ (which is a reduced version of Equation (5.1)), and also denote $\mathbf{W}^{KQ}$ as $\mathbf{W}$ for simplicity. It is used as the neuron in our two-layer multiple-head transformers:

$$g(\mathbf{X}, \mathbf{y}, \mathbf{x}_q) = \sum_{i \in [m]} \mathbf{a}_i \sigma \left[\frac{\mathbf{y}^\top \mathbf{X}}{N} \mathbf{W}^{(i)} \mathbf{x}_q \right],$$

where σ is ReLU activation, $\mathbf{a} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]^\top \in [-1, 1]^m$, $\mathbf{W}^{(i)} \in \mathbb{R}^{d \times d}$ and m is the number of attention heads. Denote its parameters as $\theta = (\mathbf{a}, \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(m)})$.

This model is more complicated as it uses non-linear activation, and also has two layers with multiple heads.

Measure model scale by head number. We use the attention head number m to measure the model scale, as a larger m means the transformer can learn more attention patterns. We consider hinge loss $\ell(z) = \max(0, 1 - z)$, and the population loss with weight-decay regularization:

$$\mathcal{L}^\lambda(g) = \mathbb{E}\left[\ell\left(y_q \cdot g(\mathbf{X}, \mathbf{y}, \mathbf{x}_q)\right)\right] + \lambda \left(\sum_{i \in [m]} \|\mathbf{W}^{(i)}\|_F^2 \right).$$

Suppose $N \to \infty$ and let the optimal solution of $\mathcal{L}^\lambda(g)$ be

$$g^* = \arg\min_g \lim_{\lambda \to 0^+} \mathcal{L}^\lambda(g).$$

5.3.1 Optimal Solution

We first introduce some notations to describe the optimal. Let $\mathrm{bin}(\cdot)$ be the integer to binary function, e.g., $\mathrm{bin}(6) = 110$. Let $\mathrm{digit}(z, i)$ denote the digit at the i-th position (from right to left) of z, e.g., $\mathrm{digit}(01000, 4) = 1$. We are now ready to characterize the optimal solution (proved in Appendix D.3.1).

Theorem 5.3.1 (Optimal solution for parity). Consider $k = 2^{\nu_1}, d = 2^{\nu_2}$, and let g_1^* and g_2^*

denote the optimal solutions for $m = 2(\nu_1 + 1)$ and $m = 2(\nu_2 + 1)$, respectively.

When $0 < p_{\mathcal{T}} < \frac{\frac{1}{4}-\gamma}{\frac{d(d-1)}{2}(\frac{1}{4}+\gamma)+\frac{1}{4}-\gamma}$, g_1^* neurons are a subset of g_2^* neurons. Specifically, for any $i \in [2(\nu_2 + 1)]$, let $\mathbf{V}^{*,(i)}$ be diagonal matrix and

- For any $i \in [\nu_2]$ and $i_\tau \in [d]$, let $\mathbf{a}_i^* = -1$ and $\mathbf{V}^{*,(i)}_{i_\tau,i_\tau} = (2\,\mathrm{digit}(\mathrm{bin}(i_\tau - 1), i) - 1)/(4\gamma)$.
- For $i = \nu_2 + 1$ and any $i_\tau \in [d]$, let $\mathbf{a}_i^* = +1$ and $\mathbf{V}^{*,(i)}_{i_\tau,i_\tau} = -\nu_j/(4\gamma)$ for g_j^*.
- For $i \in [2(\nu_2 + 1)] \setminus [\nu_2 + 1]$, let $\mathbf{a}_i^* = \mathbf{a}^*_{i-\nu_2-1}$ and $\mathbf{V}^{*,(i)} = -\mathbf{V}^{*,(i-\nu_2-1)}$.

Let $\mathbf{W}^{*,(i)} = \mathbf{G}\mathbf{V}^{*,(i)}\mathbf{G}^\top$. Up to permutations, g_2^* has neurons $(\mathbf{a}^*, \mathbf{W}^{*,(1)}, \ldots, \mathbf{W}^{*,(m)})$ and g_1^* has the $\{1, \ldots, \nu_1, \nu_2 + 1, \nu_2 + 2 \ldots, \nu_2 + \nu_1 + 1, 2\nu_2 + 2\}$-th neurons of g_2^*.

Proof sketch of Theorem 5.3.1. The proof is challenging as the non-linear model and non-linear data. We defer the full proof to Appendix D.3.1. The high-level intuition is transferring the optimal solution to patterns covering problems. For small $p_{\mathcal{T}}$, the model will "prefer" to cover all patterns in S_1 first. When the model becomes larger, by checking the sufficient and necessary conditions, it will continually learn to cover non-important features. Thus, the smaller model will mainly focus on important features, while the larger model will focus on all features. □

Example for Theorem 5.3.1. When $\nu_2 = 3$, the optimal has $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{a}_3 = -1$, $\mathbf{a}_4 = +1$ and,

$$\mathbf{V}^{(1)} = 1/4\gamma \cdot \mathrm{diag}([-1, +1, -1, +1, -1, +1, -1, +1])$$

$$\mathbf{V}^{(2)} = 1/4\gamma \cdot \mathrm{diag}([-1, -1, +1, +1, -1, -1, +1, +1])$$

$$\mathbf{V}^{(3)} = 1/4\gamma \cdot \mathrm{diag}([-1, -1, -1, -1, +1, +1, +1, +1])$$

$$\mathbf{V}^{(4)} = 3/4\gamma \cdot \mathrm{diag}([-1, -1, -1, -1, -1, -1, -1, -1])$$

and $\mathbf{V}^{(i+4)} = -\mathbf{V}^{(i)}, \mathbf{a}_{i+4} = \mathbf{a}_i$ for $i \in [4]$.

On the other hand, the optimal g_1^* for $\nu_1 = 1$ has the $\{1, 4, 5, 8\}$-th neurons of g_2^*.

By carefully checking, we can see that the neurons in g_1^* (i.e., the $\{1, 4, 5, 8\}$-th neurons of g_2^*) are used for parity classification task from S_1, i.e, label determined by the first

$k = 2^{\nu_1} = 2$ dimensions. With the other neurons (i.e., the $\{2, 3, 6, 7\}$-th neurons of g_2^*), g_2^* can further do parity classification on the task from S_2, label determined by any two dimensions other than the first two dimensions.

What hidden features does the model pay attention to? Theorem 5.3.1 gives the closed-form optimal solution of two-layer multiple-head transformers learning sparse-parity ICL tasks. It shows the optimal solution of the smaller model indeed is a sub-model of the larger optimal model. In detail, the smaller model will mainly learn all important features, while the larger model will learn more features. This again shows a trade-off when increasing the model scale: larger models can learn more hidden features which can be beneficial if these features are relevant to the label, but also potentially keep more noise which is harmful.

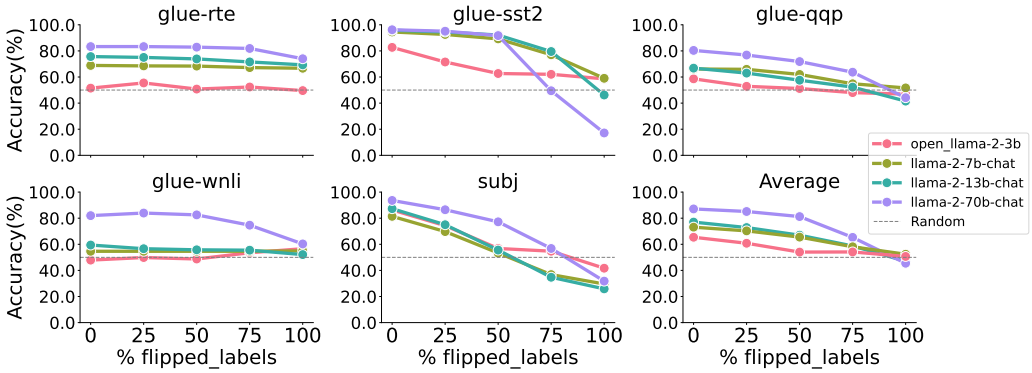


Figure 5.1: Larger models are easier to be affected by noise (flipped labels) and override pretrained biases than smaller models for different datasets and model families (chat/with instruct turning). Accuracy is calculated over 1000 evaluation prompts per dataset and over 5 runs with different random seeds for each evaluation, using $M = 16$ in-context exemplars.

5.3.2 Behavior Difference

Similar to Theorem 5.2.3, to illustrate our insights, we will consider a setting where the smaller model learns useful features for the evaluation task while the larger model covers extra features. That is, for evaluation, we uniformly draw a task $\mathbf{t}_\tau = (i_\tau, j_\tau)$ from S_1, and then draw M samples to form the evaluation prompt in the same way as during pretraining. To present our theorem (proved in Appendix D.3.2 using Theorem 5.3.1), we introduce

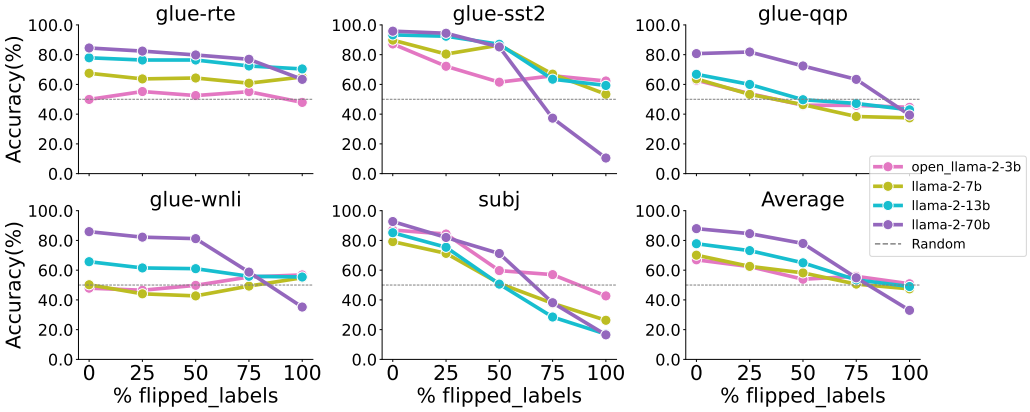


Figure 5.2: Larger models are easier to be affected by noise (flipped labels) and override pretrained biases than smaller models for different datasets and model families (original/without instruct turning). Accuracy is calculated over 1000 evaluation prompts per dataset and over 5 runs with different random seeds for each evaluation, using $M = 16$ in-context exemplars.

some notations. Let

$$\begin{aligned}\mathbf{D}_1 = \big[&\operatorname{diag}(\mathbf{V}^{*,(1)}), \ldots, \operatorname{diag}(\mathbf{V}^{*,(\nu_1)}), \operatorname{diag}(\mathbf{V}^{*,(\nu_2+1)}),\\ &\ldots, \operatorname{diag}(\mathbf{V}^{*,(\nu_2+\nu_1+1)}), \operatorname{diag}(\mathbf{V}^{*,(2\nu_2+2)})\big] \in \mathbb{R}^{d\times 2(\nu_1+1)}\\ \mathbf{D}_2 = \Big[&\operatorname{diag}(\mathbf{V}^{*,(1)}), \ldots, \operatorname{diag}(\mathbf{V}^{*,(2\nu_2+2)})\Big] \in \mathbb{R}^{d\times 2(\nu_2+1)},\end{aligned}$$

where for any $i \in [2(\nu_2+1)]$, $\mathbf{V}^{*,(i)}$ is defined in Theorem 5.3.1. Let $\hat{\phi}_{\tau,q} \in \mathbb{R}^d$ satisfy $\hat{\phi}_{\tau,q,i_\tau} = \phi_{\tau,q,i_\tau}, \hat{\phi}_{\tau,q,j_\tau} = \phi_{\tau,q,j_\tau}$ and all other entries being zero. For a matrix $\mathbf{Z}$ and a vector $\mathbf{v}$, let $P_{\mathbf{Z}}$ denote the projection of $\mathbf{v}$ to the space of $\mathbf{Z}$, i.e., $P_{\mathbf{Z}}(\mathbf{v}) = \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{v}$.

Theorem 5.3.2 (Behavior difference for parity). Assume the same condition as Theorem 5.3.1. For $j \in \{1,2\}$, Let θ_j denote the parameters of g_j^*. For $l \in [M]$, let ξ_l be uniformly drawn from $\{\pm 1\}^d$, and $\Xi = \frac{\sum_{l\in[M]}\xi_l}{M}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the randomness of test data, we have

$$\begin{aligned}g_j^*(\mathbf{X}_\tau, \mathbf{y}_\tau, \mathbf{x}_{\tau,q}) =\ & h(\theta_j, 2\gamma\hat{\phi}_{\tau,q} + P_{\mathbf{D}_j}(\Xi)) + \epsilon_j\\ := \sum_{i\in[m]} \mathbf{a}_i^* \sigma &\left[\operatorname{diag}\left(\mathbf{V}^{*,(i)}\right)^\top \left(2\gamma\hat{\phi}_{\tau,q} + P_{\mathbf{D}_j}(\Xi)\right)\right] + \epsilon_j\end{aligned}$$

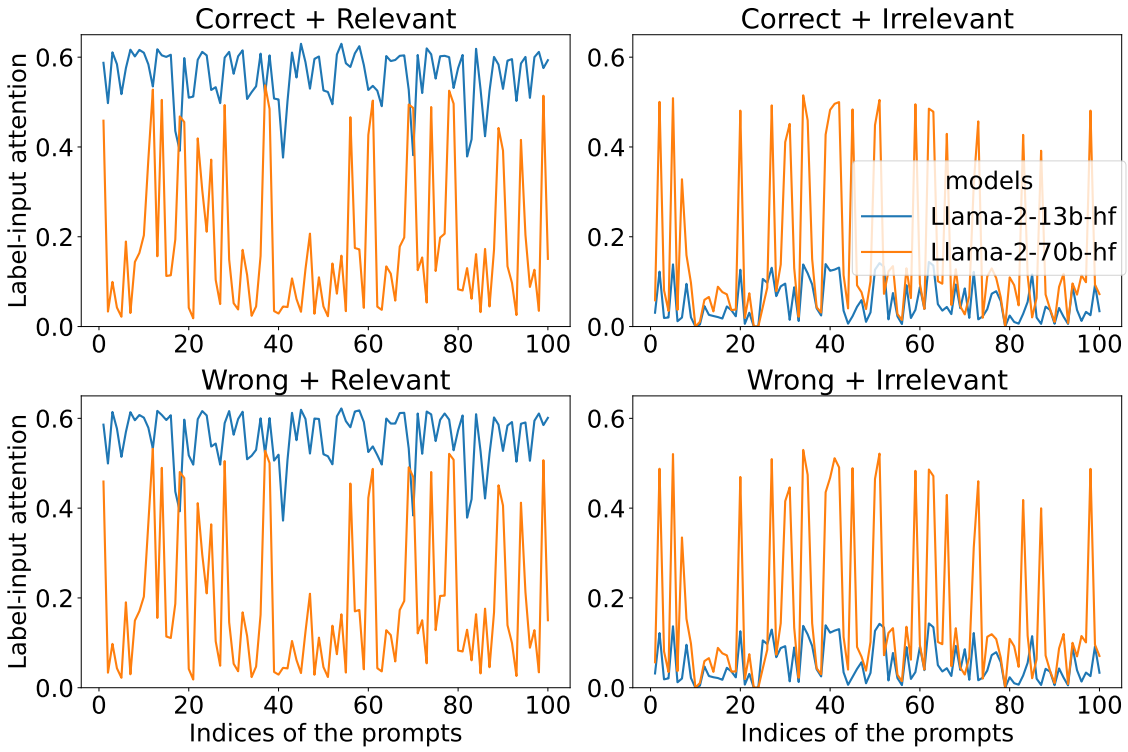


Figure 5.3: The magnitude of attention between the labels and input sentences in Llama 2-13b and 70b on 100 evaluation prompts; see the main text for the details. x-axis: indices of the prompts. y-axis: the norm of the last row of attention maps in the final layer. Correct: original label; wrong: flipped label; relevant: original input sentence; irrelevant: irrelevant sentence from other datasets. The results show that larger models focus on both sentences, while smaller models only focus on relevant sentences.

where $\epsilon_j = O\left(\sqrt{\frac{\nu_j}{M}\log\frac{1}{\delta}}\right)$ and we have

- $2\gamma\hat{\phi}_{\tau,q}$ is the signal useful for prediction: $0 = \ell(y_q \cdot h(\theta_1, 2\gamma\hat{\phi}_{\tau,q})) = \ell(y_q \cdot h(\theta_2, 2\gamma\hat{\phi}_{\tau,q}))$.
- $P_{\mathbf{D}_1}(\Xi))$ and $P_{\mathbf{D}_2}(\Xi))$ is noise not related to labels, and $\frac{\mathbb{E}[\|P_{\mathbf{D}_1}(\Xi))\|_2^2]}{\mathbb{E}[\|P_{\mathbf{D}_2}(\Xi))\|_2^2]} = \frac{\nu_1+1}{\nu_2+1}$.

Implications. Theorem 5.3.2 shows that during evaluation, we can decompose the input into two parts: signal and noise. Both the larger model and smaller model can capture the signal part well. However, the smaller model has a much smaller influence from noise than the larger model, i.e., the ratio is $\frac{\nu_1+1}{\nu_2+1}$. The reason is that smaller models emphasize important hidden features while larger ones cover more hidden features, and thus, smaller models are more robust to noise while larger ones are easily distracted, leading to different

ICL behaviors. This again sheds light on where transformers pay attention to and how that affects ICL.

Remark 5.3.1. Here, we provide a detailed intuition about Theorem 5.3.2. Ξ is the input noise. When we only care about the noise part, we can rewrite the smaller model as $g_1 = h(\theta_1, P_{D_1}(\Xi))$, and the larger model as $g_2 = h(\theta_2, P_{D_2}(\Xi))$, where they share the same h function. Our conclusion says that $E[\|P_{D_1}(\Xi)\|_2^2]/E[\|P_{D_2}(\Xi)\|_2^2] = (\nu_1 + 1)/(\nu_2 + 1)$, which means the smaller model's "effect" input noise is smaller than the larger model's "effect" input noise. Although their original input noise is the same, as the smaller model only focuses on limited features, the smaller model will ignore part of the noise, and the "effect" input noise is small. However, the larger model is the opposite.

5.4 Experiments

Brilliant recent work [288] runs intensive and thorough experiments to show that larger language models do in-context learning differently. Following their idea, we conduct similar experiments on binary classification datasets, which is consistent with our problem setting in the parity case, to support our theory statements.

Experimental setup. Following the experimental protocols in [288, 182], we conduct experiments on five prevalent NLP tasks, leveraging datasets from GLUE [275] tasks and Subj [60]. Our experiments utilize various sizes of the Llama model families [263, 264]: 3B, 7B, 13B, 70B. We follow the prior work on in-context learning [288] and use $M = 16$ in-context exemplars. We aim to assess the models' ability to use inherent semantic biases from pretraining when facing in-context examples. As part of this experiment, we introduce noise by inverting an escalating percentage of in-context example labels. To illustrate, a 100% label inversion for the SST-2 dataset implies that every "positive" exemplar is now labeled "negative". Note that while we manipulate the in-context example labels, the evaluation sample labels remain consistent. We use the same templates as [181], a sample evaluation for SST-2 when $M = 2$:

```
sentence: show us a good time
The answer is positive.
```

```
sentence: as dumb and cheesy
The answer is negative.

sentence: it 's a charming and often
affecting journey
The answer is
```

5.4.1 Behavior Difference

Figure 5.1 shows the result of model performance (chat/with instruct turning) across all datasets with respect to the proportion of labels that are flipped. When 0% label flips, we observe that larger language models have better in-context abilities. On the other hand, the performance decrease facing noise is more significant for larger models. As the percentage of label alterations increases, which can be viewed as increasing label noise σ^2, the performance of small models remains flat and seldom is worse than random guessing while large models are easily affected by the noise, as predicted by our analysis. These results indicate that large models can override their pretraining biases in-context input-label correlations, while small models may not and are more robust to noise. This observation aligns with the findings in [288] and our analysis.

We can see a similar or even stronger phenomenon in Figure 5.2: larger models are more easily affected by noise (flipped labels) and override pretrained biases than smaller models for the original/without instruct turning version (see the "Average" sub-figure). On the one hand, we conclude that both large base models and large chat models suffer from ICL robustness issues. On the other hand, this is also consistent with recent work suggesting that instruction tuning will impair LLM's in-context learning capability.

5.4.2 Ablation Study

To further verify our analysis, we provide an ablation study. We concatenate an irrelevant sentence from GSM-IC [237] to an input-label pair sentence from SST-2 in GLUE dataset.

We use "correct" to denote the original label and "wrong" to denote the flipped label. Then, we measure the magnitude of correlation between label-input, by computing the norm of the last row of attention maps across all heads in the final layer. We do this between "correct"/"wrong" labels and the original/irrelevant inserted sentences. Figure 5.3 shows the results on 100 evaluation prompts; for example, the subfigure Correct+Relevant shows the correlation magnitude between the "correct" label and the original input sentence in each prompt. The results show that the small model Llama 2-13b mainly focuses on the relevant part (original input) and may ignore the irrelevant sentence, while the large model Llama 2-70b focuses on both sentences. This well aligns with our analysis.

5.5 More Discussions about Noise

There are three kinds of noise covered in our analysis:

Pretraining noise. We can see it as toxic or harmful pretraining data on the website (noisy training data). The model will learn these features and patterns. It is covered by ξ in the linear regression case and S_2 in the parity case.

Input noise during inference. We can see it as natural noise as the user's wrong spelling or biased sampling. It is a finite sampling error as x drawn from the Gaussian distribution for the linear regression case and a finite sampling error as x drawn from a uniform distribution for the parity case.

Label noise during inference. We can see it as adversarial examples, or misleading instructions, e.g., deliberately letting a model generate a wrong fact conclusion or harmful solution, e.g., poison making. It is σ in the linear regression case and S_2 in the parity case.

For pretraining noise, it will induce the model to learn noisy or harmful features. During inference, for input noise and label noise, the larger model will pay additional attention to these noisy or harmful features in the input and label pair, i.e., $y \cdot x$, so that the input and label noise may cause a large perturbation in the final results. If there is no pretraining noise, then the larger model will have as good robustness as the smaller model. Also, if there is no input and label noise, the larger model will have as good robustness as the

smaller model. The robustness gap only happens when both pretraining noise and inference noise exist simultaneously.

Acknowledgements

The work in this chapter is partially supported by Air Force Grant FA9550-18-1-0166, the National Science Foundation (NSF) Grants 2008559-IIS, 2023239-DMS, and CCF-2046710.

Chapter 6

Conclusion

In this thesis, we explored three interconnected aspects of modern machine learning, contributing theoretical insights and practical methodologies to the field.

First, we proposed a general framework for analyzing two-layer neural network learning via gradient descent. This framework provides provable guarantees for several prototypical problem settings and goes beyond fixed-feature approaches such as the Neural Tangent Kernel (NTK). It offers insights into phenomena like the lottery ticket hypothesis and simplicity bias, paving the way for deeper understanding of neural network learning. Future work in this direction includes extending the framework to deeper networks and formalizing feature learning dynamics during later stages of training.

Second, we investigated the theoretical justification for multitask finetuning as a means of adapting pretrained foundation models to downstream tasks with limited labeled data. Our analysis demonstrated that, with sufficient sample complexity, finetuning using a diverse set of pertinent tasks can significantly improve target task performance. These findings were validated both theoretically and empirically. Furthermore, we proposed a task selection algorithm for multitask finetuning that yielded superior results compared to using all available tasks. We anticipate that this work will inspire further exploration into the adaptation and optimization of foundation models.

Lastly, we addressed the question of why larger language models exhibit different

behaviors during in-context learning (ICL). Through theoretical and empirical analysis, we revealed that smaller models tend to emphasize critical hidden features, making them more robust to noise, while larger models encompass a broader range of features, which can render them more prone to distraction. These findings deepen our understanding of large language models and their in-context learning mechanisms, offering potential avenues for improving their training and application.

Together, these contributions offer theoretical insights of neural network learning, foundation model adaptation, and large language model behavior. We hope this work will stimulate further research and innovation across these critical areas of machine learning.

Chapter 7

Appendix: Complete Proofs, More Discussions and Additional Experiments

Appendix

Appendix A

Discussions, Complete Proofs and Additional Experiments in Chapter 2: A Theoretical Analysis On Feature Learning In Neural Networks

Section A.1 presents more technical discussion on related work. Section A.2-A.4 provides the complete proofs for our results in the main text. Section A.5 provides the complete details and experimental results for our experiments.

Section A.6 provides the theoretical results and complete proofs for a setting more general than that in the main text, allowing incoherent dictionaries, unbalanced classes, and Gaussian noise in the data.

A.1 More Technical Discussion on Related Work

Advantage of Neural Networks over Linear Models on Fixed Features. A recent line of work has turned to show learning settings where network learning provably has advantage over linear models on fixed features; see the nice summary in [173]. Here we

highlight the results and focuses of the existing related studies and discuss the differences from ours.

[309] shows that the random feature method fails to learn even a single ReLU neuron on Gaussian inputs unless its size is exponentially large in dimension. This points out the limitation of the random feature method (belonging to the fixed feature approach) but does not consider feature learning in networks.

Some studies show that a single ReLU neuron can be learnt by gradient descent [308, 70, 85]. The analysis typically involves feature learning. However, their focus is different: they do not show the advantage over fixed feature methods and do not consider the effect of the input structures.

[326] shows that in a special teacher-student setting, the student network will do exact local convergence in a surprising way that all student neurons will converge to one of the teacher neurons. The work does not consider the effect of the input structure nor the advantage over fixed features.

[74] explains the advantage of network learning by constructing adaptive Reproducing Kernel Hilbert Space (RKHS) indexed by the training process of the neural network, and shows that adaptive RKHS benefits from a smaller function space containing the residue comparing to RKHS. The work shows the statistical advantage of networks over data-independent kernels, but does not consider the optimization for learning the network.

[105] considers data generated from a hidden vector with two subsets of variables, each uniformly distributed in a high-dimensional sphere (with a different radius), while the label is determined by only the first subset of variables. It shows the existence of good neural networks that can overcome the curse of dimensionality by representing the best low-dimensional hidden structure. However, it studies the approximation power of neural networks rather than the learning, i.e., it does not show how to learn the good network.

[80] argues that in the infinite width limit, a two-layer neural network will learn a nearly optimal feature representation in the distribution sense, thanks to the convexity of the limit problem. It is unclear how this result helps to understand the feature learning procedure

for practical networks, which is usually a non-convex process.

[47] considers a fixed, randomly initialized neural network as a representation function fed into another trainable network which is the quadratic Taylor model of a wide two-layer network. It shows that learning over the random representation can achieve improved sample complexities compared to learning over the raw data. However, the representation considered is not learned, which is different from our focus on feature learning.

[9] considers Gaussian inputs with labels given by a multiple-layer network with quadratic activations and skip connections (with the assumption of information gap on the weights), and studies training a deep network with quadratic activation. It shows that the trained network can learn proper representations and obtain small errors while no polynomial fixed feature methods can. On the other hand, it does not focus on the influence of input structure on feature learning: note that its input distribution contains no information about the "ground-truth" features in the target network. It also points out that the learned features get improved during training: higher-level layers will help lower-level layers to improve by backpropagating correction signals. Our analysis also shows feature improvement which however is by signals from the input distribution.

[13] considers PAC learning with labels given by a depth-2 ResNet, and studies training an overparameterized depth-2 ResNet (using uniform inputs over Boolean hypercube as an example). It shows the trained network can obtain small errors while no polynomial kernel methods can obtain as good errors. Similar to [9], it does not focus on the influence of input structure on feature learning or the advantage of networks.

[12] studies how ensemble of deep learning models can improve test accuracy and how the ensemble can be distilled into a single model. It develops a theory which assumes the data has multi-view structure and shows that the ensemble of independently trained networks can provably improve test accuracy and the ensemble can also be provably distilled into a single model. The analysis also relies on showing that the data structure can help the ensemble and the distillation. On the other hand, their focus is on ensembles and is quite different from ours: the analysis is on showing the multi-view input structure allows the

ensembles of networks to improve over single ones and ensembles of fixed feature mappings do not have improvement. While our focus is on supervisedly learning one single network that outperforms the fixed feature approaches.

[63] considers the task of learning sparse parities with two-layer networks, and the analysis suggests that the ability to learn the label-correlated features also seems to be critical towards the success of neural networks, although the authors did not explore much in this direction. [173] also considers similar learning problems but with specifically designed models for the problems. The learning problems considered in [63, 173] have input distributions that leak information about the target labeling function, which is similar to our setting, and their analysis also shows that the first gradient descent can learn a set of good features and later steps can learn an accurate classifier on top. Our work is inspired by their studies, while there are some important differences. First, their focuses are different from ours. [63] focuses on showing neural networks can learn targets (i.e., k-parity functions) that are inherently non-linear. Our analysis generalizes to more general distributions, including practically motivated ones. [173] focuses on strong separations between learning with gradient descent on differentiable models (including typical neural networks) and learning using the corresponding tangent kernels. The analysis is on specific differentiable models, while our work is on two-layer neural networks similar to practical ones. Second, our analysis relies on the feature improvement in the second gradient step. This is not an artifact of the analysis but comes from our problem setup. While in [63] the data distribution allows some neurons to be sufficiently good after the first gradient step and needs no feature improvement, our setup is more general where the data distribution may not have a similar strong benign effect and thus needs feature improvement in the second gradient step.

Most related to our work is [63]. Therefore, we provide a detailed discussion to highlight the connections and differences.

1. Our problem setting is *more general* than that in [63]. To see this, let our dictionary be the identity matrix, the set P to be the odd numbers (i.e., the labeling function is

a sparse parity). Furthermore, let the distribution of the hidden representation be an equal mixture of the following two:

(a) $\mathcal{D}_1$: Uniform distribution over the hypercube.

(b) $\mathcal{D}_2$: Irrelevant patterns $\tilde{\phi}_j (j \notin A)$ have appearance probability $p_0 = 1/2$. And the distribution of relevant patterns $\tilde{\phi}_j (j \in A)$ is: all 0's with probability $1/2$, and all 1's with probability $1/2$.

Then our problem setting reduces to their setting (up to scaling/translation of $\tilde{\phi}_j$'s). On the other hand, in general our setting allows for more choices for the labeling, the dictionary, and the distributions over $\tilde{\phi}$.

2. Upper bound: Because of the more general setting, our upper bound proof requires *technical novelty*. Recall that in their work, the input distribution is essentially a mixture of $\mathcal{D}_1$ and $\mathcal{D}_2$ above. In $\mathcal{D}_2$, the relevant patterns $\tilde{\phi}_j (j \in A)$ have the specific structure of all 0's or all 1's with probability $1/2$. This allows to show that neurons with weight w satisfying $\sum_{j \in A} w_j = 0$ will have good gradients: small components from irrelevant patterns (their Lemma 7) and large components from relevant patterns (their Lemma 8). However, in our setting, the relevant patterns do not have this specific structure, and thus their proof technique is not applicable (or can be applied only when we have an exponentially large number of hidden neurons so that some hit the good positions at random initialization). What we showed is that the gradient has some correlation with the good feature direction. So after the first gradient step, the neuron weights are not good yet but are in a better position for further improvement (in particular, their setting corresponds to $p_0 = 1/2$ which means large noise in the weights after the first step; see discussion after our Lemma 2.3.2 in Section 2.3). Then the latter gradient steps are able to improve the weights to better "signal-to-noise-ratio". In summary, our proof does not rely on their specific input structure or an exponentially large number of hidden neurons for hitting some good positions. The key is that the good feature will emerge with the help of the input

structure, and once in a better position, the neurons' weights can be improved to the desired quality.

3. Lower bound: On the other hand, our lower bound is proved by a reduction to the lower bound results in [63]. They have shown that $\mathcal{D}_1$ above can lead to large errors for fixed feature models of polynomial size. Our proof is essentially constructing a mixture of $\mathcal{D}_1$ and $\mathcal{D}_2$ with mixture weights p_0 and $(1 - p_0)$, and applying their lower bound for $\mathcal{D}_1$. See our proof in Appendix A.3.

4. Conceptually, our work belongs to the same line of research as [63], to analyze how feature learning leads to the superior performance of networks. While their analysis also relies on feature learning from good gradients induced by input structure, their focus is more on separating network learning and fixed feature models and has not explicitly explored the impact of input structures (while we agree that such an explicit study will not be difficult in their setting). More importantly, their input distribution is specific and atypical in practice, which allows a specific type of feature learning (as explained in the above discussion on upper bounds). Our work thus considers a more general setting that is motivated by practical problems. Our results then bring theoretical insights closer for explaining the feature learning in practice and provide some positive evidence for the importance of analysis under proper models of the input distributions.

Sparse Coding and Subspace Data Models. To analyze neural networks' performance, various data models have been considered. A practical way to model the underlying structure of data is by assuming that a set of hidden variables exists and the input data is a high dimensional projection of the hidden vector (possibly with noise). Along this line, the classic sparse coding model has been used in existing works for analyzing networks. [144] considers such a data distribution where the label is given by a linear function on the hidden sparse vector, but studies the approximation power of networks and classic polynomial methods rather than the learning. [10] considers similar data distributions, but studies

the performance of networks under adversarial perturbations. Another type of related data models assumes that the label is determined by a subset of hidden variables. [105] considers a hidden vector with two subsets of variables, each uniformly distributed in a high-dimensional sphere (with a different radius), while the label is determined by only the first subset of variables. However, [105] studies the approximation power of neural networks rather than the learning. Compared to these studies, our work assumes the input is given by a dictionary multiplied with a hidden vector (not necessarily sparse) while the label is determined by a subset of the hidden vector, as motivated by pattern recognition applications in practice. Furthermore, we focus on the learning ability of networks instead of approximation.

A.2 Complete Proofs for Provable Guarantees of Neural Networks

We first make a few remarks about the proof.

Remark. The analysis can be carried out for more gradient steps following similar intuition, while we analyze two steps for simplicity.

Remark. Readers may notice that the network can be overparameterized. With sufficient overparameterization and proper initialization and step sizes, network learning becomes approximately NTK. However, here our learning scheme allows going beyond this kernel regime: we use aggressive gradient updates $\lambda_{\mathbf{w}}^{(t)} = 1/(2\eta^{(t)})$ in the first two steps, completely forgetting the old weights to learn effective features. Using proper initialization and aggressive updates early to escape the kernel regime has been studied in existing work (e.g., [294, 163]). Our result thus adds another concrete example.

Notations. For a vector v and an index set I, let v_I denote the vector containing the entries of v indexed by I, and v_{-I} denote the vector containing the entries of v with indices outside I.

By initialization, $\mathbf{w}_i^{(0)}$ for $i \in [m]$ are i.i.d. copies of the same random variable $\mathbf{w}^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$; similar for $\mathbf{a}^{(0)}$ and $\mathbf{b}^{(0)}$. Let $q_\ell := \langle \mathbf{w}^{(0)}, \mathbf{M}_\ell \rangle$, then $\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle = \langle \phi, q \rangle$. Similarly, define $q_{i,\ell}^{(t)} := \langle \mathbf{w}_i^{(t)}, \mathbf{M}_\ell \rangle$. Let $\sigma_\phi^2 := p_{\mathrm{o}}(1-p_{\mathrm{o}})/\tilde{\sigma}^2$ denote the variance of ϕ_ℓ for $\ell \notin \mathbf{A}$.

We also define the following sets to denote typical initialization. For a fixed $\delta \in (0,1)$, define

$$\mathcal{G}_{\mathbf{w}}(\delta) := \Bigg\{ \mathbf{w} \in \mathbb{R}^d : q_\ell = \langle \mathbf{w}, \mathbf{M}_\ell \rangle, \frac{\sigma_{\mathbf{w}}^2 (D-k)}{2} \le \sum_{\ell \notin \mathbf{A}} q_\ell^2 \le \frac{3\sigma_{\mathbf{w}}^2 (D-k)}{2}, \max_\ell |q_\ell| \le \sigma_{\mathbf{w}} \sqrt{2\log(Dm/\delta)} \Bigg\}, \tag{A.1}$$

$$\mathcal{G}_{\mathbf{a}}(\delta) := \{\mathbf{a} \in \mathbb{R} : |\mathbf{a}| \leq \sigma_{\mathbf{a}}\sqrt{2\log(m/\delta)}\}. \tag{A.2}$$

$$\mathcal{G}_{\mathbf{b}}(\delta) := \{\mathbf{b} \in \mathbb{R} : |\mathbf{b}| \leq \sigma_{\mathbf{b}}\sqrt{2\log(m/\delta)}\}. \tag{A.3}$$

A.2.1 Existence of A Good Network

we first show that there exists a network that can fit the data distribution.

Lemma A.2.1. *For some $s, a, b \in \mathbb{R}$ with $a, b \geq 0$, define a function $\delta_{s,a,b} : \mathbb{R} \to \mathbb{R}$ as*

$$\delta_{s,a,b}(z) = a\sigma_{\mathrm{r}}(z - s + b) - 2a\sigma_{\mathrm{r}}(z - s) + a\sigma_{\mathrm{r}}(z - s - b). \tag{A.4}$$

where $\sigma_{\mathrm{r}}(z) = \max\{z, 0\}$ is the ReLU activation function. Then

$$\delta_{s,a,b}(z) = \begin{cases} 0 & \text{when } z \leq s - b, \\ a(z - s) + ab & \text{when } s - b \leq z \leq s, \\ -a(z - s) + ab & \text{when } s \leq z \leq s + b, \\ 0 & \text{when } s + b \leq z. \end{cases} \tag{A.5}$$

That is, $\delta_{s,a,b}(z)$ linearly interpolates between $(s-b, 0), (s, ab), (s+b, 0)$ when $z \in [s-b, s+b]$, and is 0 elsewhere.

Proof of Lemma A.2.1. This can be simply verified for the four cases of the value of z. □

Lemma A.2.2 (Restatement of Lemma 2.3.1)**.** *For any $\mathcal{D} \in \mathcal{F}_{\Xi}$, there exists a network $g^*(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{a}_i^* \sigma(\langle \mathbf{w}_i^*, \mathbf{x}\rangle + \mathbf{b}_i^*)$ with $y = g^*(x)$ for any $(\mathbf{x}, y) \sim \mathcal{D}$. Furthermore, the number of neurons $n = 3(k + 1)$, $|\mathbf{a}_i^*| \leq 32k, 1/(32k) \leq |\mathbf{b}_i^*| \leq 1/2$, $\mathbf{w}_i^* = \tilde{\sigma} \sum_{j \in \mathbf{A}} \mathbf{M}_j/(4k)$, and $|\langle \mathbf{w}_i^*, \mathbf{x}\rangle + \mathbf{b}_i^*| \leq 1$ for any $i \in [n]$ and $(\mathbf{x}, y) \sim \mathcal{D}$.*

Proof of Lemma 2.3.1. Let $\mathbf{w} = \tilde{\sigma} \sum_{j \in \mathbf{A}} \mathbf{M}_j$ and let $\mu = \sum_{j \in \mathbf{A}} \mathbb{E}[\tilde{\phi}_j]$. We have

$$\langle \mathbf{w}, \mathbf{x} \rangle = \tilde{\sigma} \sum_{j \in \mathbf{A}} \langle \mathbf{M}_j, \mathbf{M}\phi \rangle = \tilde{\sigma} \sum_{j \in \mathbf{A}} \phi_j = \sum_{j \in \mathbf{A}} \tilde{\phi}_j - \mu. \tag{A.6}$$

Then by Lemma A.2.1,

$$g_1^*(x) := \sum_{p \in P} \delta_{p-\mu,2,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) - \sum_{p \notin P, 0 \le p \le k} \delta_{p-\mu,2,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) \tag{A.7}$$

$$= \sum_{p \in P} \delta_{p,2,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + \mu) - \sum_{p \notin P, 0 \le p \le k} \delta_{p,2,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + \mu) \tag{A.8}$$

$$= \sum_{p \in P} \delta_{p,2,1/2} \left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j \right) - \sum_{p \notin P, 0 \le p \le k} \delta_{p,2,1/2} \left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j \right) \tag{A.9}$$

$$= y \tag{A.10}$$

for any $(x, y) \sim \mathcal{D}$. Similarly,

$$g_2^*(x) := \sum_{p \in P} \delta_{p-\mu+1/4,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) - \sum_{p \notin P, 0 \le p \le k} \delta_{p-\mu+1/4,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) \tag{A.11}$$

$$= \sum_{p \in P} \delta_{p+1/4,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + \mu) - \sum_{p \notin P, 0 \le p \le k} \delta_{p+1/4,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + \mu) \tag{A.12}$$

$$= \sum_{p \in P} \delta_{p+1/4,4,1/2} \left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j \right) - \sum_{p \notin P, 0 \le p \le k} \delta_{p+1/4,4,1/2} \left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j \right) \tag{A.13}$$

$$= y \tag{A.14}$$

for any $(x, y) \sim \mathcal{D}$. Note that the bias terms in g_1^* and g_2^* have distance at least $1/4$, then at least one of them satisfies that all its bias terms have absolute value $\ge 1/8$. Pick that one and denote it as $g(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{a}_i \sigma_{\mathrm{r}}(\langle \mathbf{w}_i, \mathbf{x} \rangle + \mathbf{b}_i)$. By the positive homogeneity of σ_{r}, we have

$$g(\mathbf{x}) = \sum_{i=1}^{n} 4k \mathbf{a}_i \sigma_{\mathrm{r}}(\langle \mathbf{w}_i, \mathbf{x} \rangle / (4k) + \mathbf{b}_i / (4k)). \tag{A.15}$$

Since for any $(\mathbf{x}, y) \sim \mathcal{D}$, $|\langle \mathbf{w}_i, \mathbf{x}\rangle/(4k) + \mathbf{b}_i/(4k)| \leq 1$, then

$$g(\mathbf{x}) = \sum_{i=1}^{n} 4k\mathbf{a}_i \sigma(\langle \mathbf{w}_i, \mathbf{x}\rangle/(4k) + \mathbf{b}_i/(4k)) \tag{A.16}$$

where σ is the truncated ReLU. Now we can set $\mathbf{a}_i^* = 4k\mathbf{a}_i, \mathbf{w}_i^* = \mathbf{w}_i/(4k), \mathbf{b}_i^* = \mathbf{b}_i/(4k)$, to get our final g^*. □

A.2.2 Initialization

We first show that with high probability, the initial weights are in typical positions.

Lemma A.2.3. *For any $\delta \in (0,1)$, with probability at least $1 - \delta - 2\exp(-\Theta(D-k))$ over $\mathbf{w}^{(0)}$,*

$$\sigma_{\mathbf{w}}^2(D-k)/2 \leq \sum_{\ell \notin \mathbf{A}} q_\ell^2 \leq 3\sigma_{\mathbf{w}}^2(D-k)/2,$$

$$\max_\ell |q_\ell| \leq \sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}.$$

With probability at least $1-\delta$ over $\mathbf{b}^{(0)}$,

$$|\mathbf{b}^{(0)}| \leq \sigma_{\mathbf{b}}\sqrt{2\log(1/\delta)}.$$

With probability at least $1-\delta$ over $\mathbf{a}^{(0)}$,

$$|\mathbf{a}^{(0)}| \leq \sigma_{\mathbf{a}}\sqrt{2\log(1/\delta)}.$$

Proof of Lemma A.2.3. From $q \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$, we have:

- With probability $\geq 1 - \delta/2$, $\max_\ell |q_\ell| \leq \sqrt{2\sigma_{\mathbf{w}}^2 \log\frac{D}{\delta}}$, and
- For any subset $S \subseteq [D]$, with probability $\geq 1 - 2\exp(-\Theta(|S|))$, $\|q_S\|_2^2 \in \left(\frac{|S|\sigma_{\mathbf{w}}^2}{2}, \frac{3|S|\sigma_{\mathbf{w}}^2}{2}\right)$.

Similar for $\mathbf{b}^{(0)}$ and $\mathbf{a}^{(0)}$. The lemma then follows. □

Lemma A.2.4. *We have:*

- *With probability* $\geq 1-\delta-2m\exp(-\Theta(D-k))$ *over* $\mathbf{w}_i^{(0)}$*'s, for all* $i\in[2m]$, $\mathbf{w}_i^{(0)}\in\mathcal{G}_{\mathbf{w}}(\delta)$.
- *With probability* $\geq 1-\delta$ *over* $\mathbf{b}_i^{(0)}$*'s, for all* $i\in[2m]$, $\mathbf{b}_i^{(0)}\in\mathcal{G}_{\mathbf{b}}(\delta)$.
- *With probability* $\geq 1-\delta$ *over* $\mathbf{a}_i^{(0)}$*'s, for all* $i\in[2m]$, $\mathbf{a}_i^{(0)}\in\mathcal{G}_{\mathbf{a}}(\delta)$.

Proof of Lemma A.2.4. This follows from Lemma A.2.3 by union bound. □

The following lemma about the typical $\mathbf{w}_i^{(0)}$'s will be useful for later analysis.

Lemma A.2.5. *Fix* $\delta\in(0,1)$. *For any* $\mathbf{w}_i^{(0)}\in\mathcal{G}_{\mathbf{w}}(\delta)$, *we have*

$$\Pr_{\phi}\left[\sum_{\ell\notin\mathbf{A}}\phi_\ell q_{i,\ell}^{(0)}\geq\Theta\left(\sqrt{(D-k)\sigma_\phi^2\sigma_{\mathbf{w}}^2}\right)\right]=\Theta(1)-\frac{O(\log^{3/2}(Dm/\delta))}{\sqrt{(D-k)\sigma_\phi^2\tilde{\sigma}^2}}. \tag{A.17}$$

Consequently, when $p_o=\Omega(k^2/D)$ *and* $k=\Omega(\log^2(Dm/\delta))$,

$$\Pr_{\phi}\left[\sum_{\ell\notin\mathbf{A}}\phi_\ell q_{i,\ell}^{(0)}\geq\Theta\left(\sigma_{\mathbf{w}}\right)\right]=\Theta(1)-\frac{O(1)}{k^{1/4}}. \tag{A.18}$$

Proof of Lemma A.2.5. Note that for $\ell\notin\mathbf{A}$, $\mathbb{E}[\phi_\ell]=0$, $\mathbb{E}[\phi_\ell^2]=\sigma_\phi^2$, and $\mathbb{E}[|\phi_\ell|^3]=\Theta(\sigma_\phi^2/\tilde{\sigma})$. Then the statement follows from Berry-Esseen Theorem. □

A.2.3 Some Auxiliary Lemmas

The expression of the gradients will be used frequently.

Lemma A.2.6.

$$\frac{\partial}{\partial\mathbf{w}_i}L_{\mathcal{D}}(g;\sigma_\xi)=-\mathbf{a}_i\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg(\mathbf{x};\xi)\leq 1]\mathbb{E}_{\xi_i}\mathbb{I}[\langle\mathbf{w}_i,\mathbf{x}\rangle+\mathbf{b}_i+\xi_i\in(0,1)]\mathbf{x}\right\}, \tag{A.19}$$

$$\frac{\partial}{\partial\mathbf{b}_i}L_{\mathcal{D}}(g;\sigma_\xi)=-\mathbf{a}_i\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg(\mathbf{x};\xi)\leq 1]\mathbb{E}_{\xi_i}\mathbb{I}[\langle\mathbf{w}_i,\mathbf{x}\rangle+\mathbf{b}_i\in(0,1)]\right\}, \tag{A.20}$$

$$\frac{\partial}{\partial\mathbf{a}_i}L_{\mathcal{D}}(g;\sigma_\xi)=-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg(\mathbf{x};\xi)\leq 1]\mathbb{E}_{\xi_i}\sigma(\langle\mathbf{w}_i,\mathbf{x}\rangle+\mathbf{b}_i+\xi_i)\right\}. \tag{A.21}$$

Proof of Lemma A.2.6. It follows from straightforward calculation. □

We now show that a small subset of the entries in ϕ, q does not affect the probability distribution of $\langle \phi, q \rangle$ much.

Lemma A.2.7. *Suppose $\nu \sim \mathcal{N}(0, \sigma^2)$. For any $B \supseteq \mathbf{A}$ and any b:*

$$\left| \Pr_{\phi_{-B}, \nu} \{\langle \phi, q \rangle + \nu \geq b\} - \Pr_{\phi_{-B}, \nu} \{\langle \phi_{-B}, q_{-B} \rangle + \nu \geq b\} \right| \tag{A.22}$$

$$\leq O\left(\frac{|\langle \phi_B, q_B \rangle|}{(\sigma_\phi^2 \|q_{-B}\|_2^2 + \sigma^2)^{1/2}} + \frac{\sigma^3 + \sigma_\phi^2 \|q_{-B}\|_3^3 / \tilde{\sigma}}{(\sigma^2 + \sigma_\phi^2 \|q_{-B}\|_2^2)^{3/2}} \right). \tag{A.23}$$

Similarly,

$$\left| \Pr_{\phi_{-B}} \{\langle \phi, q \rangle \geq b\} - \Pr_{\phi_{-B}} \{\langle \phi_{-B}, q_{-B} \rangle \geq b\} \right| \tag{A.24}$$

$$\leq O\left(\frac{|\langle \phi_B, q_B \rangle|}{\sigma_\phi \|q_{-B}\|_2} + \frac{\|q_{-B}\|_3^3}{\tilde{\sigma} \sigma_\phi \|q_{-B}\|_2^3} \right). \tag{A.25}$$

Proof of Lemma A.2.7. Note that for $\ell \notin \mathbf{A}$, $\mathbb{E}[\phi_\ell] = 0$, $\mathbb{E}[\phi_\ell^2] = \sigma_\phi^2$, and $\mathbb{E}[|\phi_\ell|^3] = \Theta(\sigma_\phi^2 / \tilde{\sigma})$. Let $t = |\langle \phi_B, q_B \rangle|$. Then by the Berry-Esseen Theorem,

$$\left| \Pr_{\phi_{-B}} \{\langle \phi, q \rangle + \nu \geq b\} - \Pr_{\phi_{-B}} \{\langle \phi_{-B}, q_{-B} \rangle + \nu \geq b\} \right| \tag{A.26}$$

$$\leq \Pr_{\phi_{-B}} \{\langle \phi_{-B}, q_{-B} \rangle + \nu \in [-t + b, t + b]\} \tag{A.27}$$

$$\leq \frac{2t}{(\sigma_\phi^2 \|q_{-B}\|_2^2 + \sigma^2)^{1/2}} + \frac{O(\sigma^3 + \sigma_\phi^2 \|q_{-B}\|_3^3 / \tilde{\sigma})}{(\sigma^2 + \sigma_\phi^2 \|q_{-B}\|_2^2)^{3/2}}. \tag{A.28}$$

The second statement follows from a similar argument. □

We also have the following auxiliary lemma for later calculations.

Lemma A.2.8.

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{y\} = 0, \tag{A.29}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\} = 1, \tag{A.30}$$

$$\mathbb{E}_{\phi_j}\{|\phi_j|\} = 2\sigma_\phi^2\tilde{\sigma},\ \textit{for } j \notin \mathbf{A}, \tag{A.31}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{y\phi_j\} = \frac{\gamma}{\tilde{\sigma}}, \tag{A.32}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|y\phi_j|\} \leq \frac{1}{\tilde{\sigma}},\ \textit{for all } j \in [D]. \tag{A.33}$$

Proof of Lemma A.2.8.

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{y\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{y|y=v\}\Pr[y=v] \tag{A.34}$$

$$= \frac{1}{2}\sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{y|y=v\} \tag{A.35}$$

$$= 0. \tag{A.36}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\,|y=v\}\Pr[y=v] \tag{A.37}$$

$$= \frac{1}{2}\sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\,|y=v\} \tag{A.38}$$

$$= 1. \tag{A.39}$$

$$\mathbb{E}_{\phi_j}\{|\phi_j|\} = \frac{|-p_{\mathrm{o}}|(1-p_{\mathrm{o}}) + |1-p_{\mathrm{o}}|p_{\mathrm{o}}}{\tilde{\sigma}} = 2\sigma_\phi^2\tilde{\sigma}. \tag{A.40}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{y\phi_j\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\frac{\tilde{\phi}_j - \mathbb{E}[\tilde{\phi}_j]}{\tilde{\sigma}}\right\} \tag{A.41}$$

$$= \frac{1}{\tilde{\sigma}}\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\tilde{\phi}_j - y\mathbb{E}[\tilde{\phi}_j]\right\} \tag{A.42}$$

$$= \frac{\gamma}{\tilde{\sigma}}. \tag{A.43}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|y\phi_j|\} = \mathbb{E}_{\phi_{\mathbf{A}}}\{|\phi_j|\} \tag{A.44}$$

$$\leq \frac{1}{\tilde{\sigma}}. \tag{A.45}$$

□

A.2.4 Feature Emergence: First Gradient Step

We will show that w.h.p. over the initialization, after the first gradient step, there are neurons that represent good features.

We begin with analyzing the gradients.

Lemma A.2.9 (Full version of Lemma 2.3.2)**.** *Fix $\delta \in (0,1)$ and suppose $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta)$ for all $i \in [2m]$. Let*

$$\epsilon_e := \frac{k \log^{1/2}(Dm/\delta) + \log^{3/2}(Dm/\delta)}{\sqrt{\sigma_\phi^2 \tilde{\sigma}^2 (D-k)}}.$$

If $p_o = \Omega(k^2/D)$, $k = \Omega(\log^2(Dm/\delta))$, and $\sigma_\xi^{(1)} < 1/k$, then

$$\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) = -\mathbf{a}_i^{(0)} \sum_{j=1}^{D} \mathbf{M}_j T_j \tag{A.46}$$

where T_j satisfies:

- *if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \le O(\epsilon_e/\tilde{\sigma})$, where $\beta \in [\Omega(1), 1]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;*
- *if $j \notin \mathbf{A}$, then $|T_j| \le O(\sigma_\phi^2 \epsilon_e \tilde{\sigma})$.*

Proof of Lemma A.2.9. Consider one neuron index i and omit the subscript i in the parameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x}; \xi^{(1)}) = 0$, we have

$$\frac{\partial}{\partial \mathbf{w}} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) \tag{A.47}$$

$$= -\mathbf{a}^{(0)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y \mathbb{I}[y g^{(0)}(\mathbf{x}; \xi^{(1)}) \le 1] \mathbb{E}_{\xi^{(1)}} \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \mathbf{x} \right\} \tag{A.48}$$

$$= -\mathbf{a}^{(0)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}, \xi^{(1)}} \left\{ y \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \mathbf{x} \right\} \tag{A.49}$$

$$= -\mathbf{a}^{(0)} \sum_{j=1}^{D} \mathbf{M}_j \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}, \xi^{(1)}} \left\{ y \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \phi_j \right\}}_{:= T_j}. \tag{A.50}$$

First, consider $j \in \mathbf{A}$.

$$T_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{y\mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\phi_j\right\} \tag{A.51}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{y\phi_j \Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\}. \tag{A.52}$$

Let

$$I_a := \Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.53}$$

$$I'_a := \Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.54}$$

We have

$$|\mathbb{E}_{\xi^{(1)}}(I_a - I'_a)| \tag{A.55}$$

$$\leq \mathbb{E}_{\xi^{(1)}}\left|\Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0\right] - \Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0\right]\right| \tag{A.56}$$

$$+ \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 1\right] + \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 1\right]. \tag{A.57}$$

Then by Lemma A.2.7,

$$\left|\Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0\right] - \Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0\right]\right| = O(\epsilon_{\mathrm{e}}). \tag{A.58}$$

Note that $\sum_{\ell\notin\mathbf{A}} \mathrm{Var}(\phi_\ell q_\ell) = \Theta(\sigma_\phi^2\sigma_{\mathbf{w}}^2(D-k)) = \Theta(\sigma_{\mathbf{w}}^2)$, and $|\phi_\ell| \leq \frac{1}{\tilde{\sigma}}$, $\max_\ell |q_\ell| \leq \sigma_{\mathbf{w}}\sqrt{2\log(Dm/\delta)}$. Applying Bernstein's inequality for bounded distributions, we have:

$$\Pr_{\phi_{-\mathbf{A}}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle \geq 1/4\right] = \exp(-\Omega(k)) = O(\epsilon_{\mathrm{e}}). \tag{A.59}$$

We also have:

$$\Pr_{\xi^{(1)}}\left[\mathbf{b}^{(0)} + \xi^{(1)} \geq 1/4\right] = \exp(-\Omega(k)) = O(\epsilon_{\mathrm{e}}). \tag{A.60}$$

Therefore,

$$\Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle\phi,q\rangle+\mathbf{b}^{(0)}+\xi^{(1)}\geq 1\right]=\exp(-\Omega(k))=O(\epsilon_{\mathrm{e}}) \tag{A.61}$$

where the last step follows from the assumption on $\sigma_{\mathbf{w}}$ and k. A similar argument gives:

$$\Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle\phi_{-\mathbf{A}},q_{-\mathbf{A}}\rangle+\mathbf{b}^{(0)}+\xi^{(1)}\geq 1\right]=\exp(-\Omega(k))=O(\epsilon_{\mathrm{e}}). \tag{A.62}$$

Then we have

$$\begin{aligned}
&\left|T_j-\mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{y\phi_j I'_a\right\}\right| && \text{(A.63)}\\
&\leq \mathbb{E}_{\phi_{\mathbf{A}}}\left\{|y\phi_j|\left|\mathbb{E}_{\xi^{(1)}}(I_a-I'_a)\right|\right\} && \text{(A.64)}\\
&\leq O(\epsilon_{\mathrm{e}})\mathbb{E}_{\phi_{\mathbf{A}}}\left\{|y\phi_j|\right\} && \text{(A.65)}\\
&\leq O(\epsilon_{\mathrm{e}}/\tilde{\sigma}) && \text{(A.66)}
\end{aligned}$$

where the last step is from Lemma A.2.8. Furthermore,

$$\begin{aligned}
&\mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{y\phi_j I'_a\right\} && \text{(A.67)}\\
&=\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j\right\}\mathbb{E}_{\xi^{(1)}}[I'_a] && \text{(A.68)}\\
&=\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j\right\}\Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle\phi_{-\mathbf{A}},q_{-\mathbf{A}}\rangle+\mathbf{b}^{(0)}+\xi^{(1)}\in(0,1)\right] && \text{(A.69)}
\end{aligned}$$

By Lemma A.2.5, the assumption on p_{o}, and (A.59), we have

$$\begin{aligned}
&\Pr_{\phi_{-\mathbf{A}}}\left[\langle\phi_{-\mathbf{A}},q_{-\mathbf{A}}\rangle+\mathbf{b}^{(0)}\in(0,1/2)\right]\geq\Omega(1)-O(1/k^{1/4}), && \text{(A.70)}\\
&\Pr_{\xi^{(1)}}\left[\xi^{(1)}\in(0,1/2)\right]=1/2-\exp(-\Omega(k)), && \text{(A.71)}
\end{aligned}$$

This leads to

$$\beta:=\mathbb{E}_{\xi^{(1)}}[I'_a]=\Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle\phi_{-\mathbf{A}},q_{-\mathbf{A}}\rangle+\mathbf{b}^{(0)}+\xi^{(1)}\in(0,1)\right]\geq\Omega(1). \tag{A.72}$$

By Lemma A.2.8, $\mathbb{E}_{\phi_{\mathbf{A}}}\{y\phi_j\} = \gamma/\tilde{\sigma}$. Therefore,

$$|T_j - \beta\gamma/\tilde{\sigma}| \leq O(\epsilon_{\mathrm{e}}/\tilde{\sigma}). \tag{A.73}$$

Now, consider $j \notin \mathbf{A}$. Let B denote $\mathbf{A} \cup \{j\}$.

$$T_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{y\phi_j \mathbb{I}\left[\langle\phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\} \tag{A.74}$$

$$= \mathbb{E}_{\phi_B}\mathbb{E}_{\phi_{-B},\xi^{(1)}}\left\{y\phi_j \mathbb{I}\left[\langle\phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\} \tag{A.75}$$

$$= \mathbb{E}_{\phi_B,\xi^{(1)}}\left\{y\phi_j \Pr_{\phi_{-B}}\left[\langle\phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\}. \tag{A.76}$$

Let

$$I_b := \Pr_{\phi_{-B}}\left[\langle\phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.77}$$

$$I'_b := \Pr_{\phi_{-B}}\left[\langle\phi_{-B}, q_{-B}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.78}$$

Similar as above, we have $|\mathbb{E}_{\xi^{(1)}}(I_b - I'_b)| \leq O(\epsilon_{\mathrm{e}})$ by Lemma A.2.7. Then by Lemma A.2.8,

$$\left|T_j - \mathbb{E}_{\phi_B,\xi^{(1)}}\left\{y\phi_j I'_b\right\}\right| \tag{A.79}$$

$$\leq \mathbb{E}_{\phi_B}\left\{|y\phi_j||\mathbb{E}_{\xi^{(1)}}(I_b - I'_b)|\right\} \tag{A.80}$$

$$\leq O(\epsilon_{\mathrm{e}})\mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\}\,\mathbb{E}_{\phi_j}\{|\phi_j|\} \tag{A.81}$$

$$\leq O(\epsilon_{\mathrm{e}}) \times O(\sigma_\phi^2\tilde{\sigma}) \tag{A.82}$$

$$= O(\sigma_\phi^2\epsilon_{\mathrm{e}}\tilde{\sigma}). \tag{A.83}$$

Furthermore,

$$\mathbb{E}_{\phi_B,\xi^{(1)}}\left\{y\phi_j I'_b\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\{y\}\,\mathbb{E}_{\phi_j}\{\phi_j\}\,\mathbb{E}_{\xi^{(1)}}[I'_b] = 0. \tag{A.84}$$

Therefore,

$$|T_j| \leq O(\sigma_\phi^2 \epsilon_{\mathrm{e}} \tilde{\sigma}). \tag{A.85}$$

□

Lemma A.2.10. *Under the same assumptions as in Lemma A.2.9,*

$$\frac{\partial}{\partial \mathbf{b}_i} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) = -\mathbf{a}_i^{(0)} T_b \tag{A.86}$$

where $|T_b| \leq O(\epsilon_e)$.

Proof of Lemma A.2.10. Consider one neuron index i and omit the subscript i in the parameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x}; \xi^{(1)}) = 0$, we have

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) && \text{(A.87)}\\
&= -\mathbf{a}^{(0)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y \mathbb{I}[y g^{(0)}(\mathbf{x}; \xi) \leq 1] \mathbb{E}_{\xi^{(1)}} \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\} && \text{(A.88)}\\
&= -\mathbf{a}^{(0)} \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}, \xi^{(1)}} \left\{ y \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\}}_{:= T_b}. && \text{(A.89)}
\end{aligned}$$

Similar to the proof in Lemma 2.3.2,

$$\left| \Pr_{\phi_{-\mathbf{A}}} [\langle \phi, q \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] - \Pr_{\phi_{-\mathbf{A}}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right| = O(\epsilon_{\mathrm{e}}). \tag{A.90}$$

Then

$$\left|T_b - \mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{y \Pr_{\phi_{-\mathbf{A}}}[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\}\right| \tag{A.91}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{|y|\left|\Pr_{\phi_{-\mathbf{A}}}[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] - \Pr_{\phi_{-\mathbf{A}}}[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right|\right\} \tag{A.92}$$

$$\leq O(\epsilon_{\mathrm{e}})\mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\} \tag{A.93}$$

$$\leq O(\epsilon_{\mathrm{e}}). \tag{A.94}$$

Also,

$$\mathbb{E}_{\phi_{\mathbf{A}},\xi^{(1)}}\left\{y \Pr_{\phi_{-\mathbf{A}}}[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\} \tag{A.95}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\{y\} \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \tag{A.96}$$

$$= 0. \tag{A.97}$$

Therefore, $|T_b| \leq O(\epsilon_{\mathrm{e}})$. □

Lemma A.2.11. *We have*

$$\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) = -T_a \tag{A.98}$$

where $|T_a| \leq O(\max_\ell q_{i,\ell}^{(0)})$. *So if* $w_i^{(0)} \in \mathcal{G}(\delta)$, $|T_a| \leq O(\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$.

Proof of Lemma A.2.11. Consider one neuron index i and omit the subscript i in the parameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x}; \xi^{(1)}) = 0$, we have

$$\frac{\partial}{\partial \mathbf{a}} L_{\mathcal{D}}(g^{(0)}; \sigma_\xi^{(1)}) \tag{A.99}$$

$$= -\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg^{(0)}(\mathbf{x}; \xi^{(1)}) \leq 1]\mathbb{E}_{\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\} \tag{A.100}$$

$$= -\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{y\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\}}_{:=T_a}. \tag{A.101}$$

Let $\phi'_{\mathbf{A}}$ be an independent copy of $\phi_{\mathbf{A}}$, ϕ' be the vector obtained by replacing in ϕ the entries $\phi_{\mathbf{A}}$ with $\phi'_{\mathbf{A}}$, and let $x' = \mathbf{M}\phi'$ and its label is y'. Then

$$|T_a| = \left|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\}\right| \tag{A.102}$$

$$\leq \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=1\right\} \tag{A.103}$$

$$- \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=-1\right\}\Bigg| \tag{A.104}$$

$$\leq \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=1\right\} \tag{A.105}$$

$$- \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)}, \mathbf{x}'\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y'=-1\right\}\Bigg|. \tag{A.106}$$

Since σ is 1-Lipschitz,

$$|T_a| \leq \frac{1}{2}\mathbb{E}_{\phi_{\mathbf{A}},\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}}}\left|\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle - \langle \mathbf{w}^{(0)}, \mathbf{x}'\rangle\right| |y=1, y'=-1\right\} \tag{A.107}$$

$$\leq \frac{1}{2}\mathbb{E}_{\phi_{-\mathbf{A}}}\left(\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle\right| |y=1\right\} + \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(0)}, \mathbf{x}'\rangle\right| |y'=-1\right\}\right) \tag{A.108}$$

$$= \mathbb{E}_{\phi_{-\mathbf{A}},\phi_{\mathbf{A}}}\left|\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle\right| \tag{A.109}$$

$$= \mathbb{E}_{\mathbf{x}}\left|\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle\right| \tag{A.110}$$

$$\leq \sqrt{\mathbb{E}_{\mathbf{x}}\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle^2} \tag{A.111}$$

$$\leq \max_{\ell} q_{i,\ell}^{(0)} \sqrt{\mathbb{E}_{\mathbf{x}}\left(\sum_{\ell\in[D]}\phi_\ell^2 + \sum_{j\neq\ell: j,\ell\in\mathbf{A}}|\phi_j\phi_\ell|\right)} \tag{A.112}$$

$$\leq \max_{\ell} q_{i,\ell}^{(0)} \sqrt{\mathbb{E}_{\mathbf{x}}\left(1 + O(1)\right)} \tag{A.113}$$

$$= \Theta(\max_{\ell} q_{i,\ell}^{(0)}). \tag{A.114}$$

□

With the bounds on the gradient, we now summarize the results for the weights after the first gradient step.

Lemma A.2.12. *Set*

$$\lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \lambda_{\mathbf{a}}^{(1)} = \lambda_{\mathbf{b}}^{(1)} = 0, \sigma_{\xi}^{(1)} = 1/k^{3/2}.$$

Fix $\delta \in (0,1)$ *and suppose* $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta)$ *for all* $i \in [2m]$. *If* $p_o = \Omega(k^2/D)$, $k = \Omega(\log^2(Dm/\delta))$, *then for all* $i \in [m]$, $\mathbf{w}_i^{(1)} = \sum_{\ell=1}^{D} q_{i,\ell}^{(1)} \mathbf{M}_\ell$ *satisfying*

- *if* $\ell \in A$, *then* $|q_{i,\ell}^{(1)} - \eta^{(1)}\mathbf{a}_i^{(0)}\beta\gamma/\tilde{\sigma}| \leq O\left(\frac{|\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_e}{\tilde{\sigma}}\right)$, *where* $\beta \in [\Omega(1), 1]$ *and depends only on* $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;
- *if* $\ell \notin \mathbf{A}$, *then* $|q_{i,\ell}^{(1)}| \leq O\left(\sigma_\phi^2 |\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_e\tilde{\sigma}\right)$;

and

- $\mathbf{b}_i^{(1)} = \mathbf{b}_i^{(0)} + \eta^{(1)}\mathbf{a}_i^{(0)} T_b$ *where* $|T_b| = O(\epsilon_e)$;
- $\mathbf{a}_i^{(1)} = \mathbf{a}_i^{(0)} + \eta^{(1)} T_a$ *where* $|T_a| = O(\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$.

Proof of Lemma A.2.12. This follows from Lemma A.2.4 and Lemma A.2.9-A.2.11. □

A.2.5 Feature Improvement: Second Gradient Step

We first show that with properly set $\eta^{(1)}$, for most $\mathbf{x}$, $|g^{(1)}(\mathbf{x}; \sigma_\xi^{(2)})| < 1$ and thus $yg^{(1)}(\mathbf{x}; \sigma_\xi^{(2)}) < 1$.

Lemma A.2.13. *Fix* $\delta \in (0,1)$ *and suppose* $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta), \mathbf{a}_i^{(0)} \in \mathcal{G}_{\mathbf{a}}(\delta)$ *for all* $i \in [2m]$. *If* $p_o = \Omega(k^2/D)$, $k = \Omega(\log^2(Dm/\delta))$, $\sigma_{\mathbf{a}} \leq \tilde{\sigma}^2/(\gamma k^2)$, $\eta^{(1)} = O\left(\frac{\gamma}{km\sigma_{\mathbf{a}}}\right)$, *and* $\sigma_\xi^{(2)} \leq 1/k$, *then with probability* $\geq 1 - \exp(-\Theta(k))$ *over* $(\mathbf{x}, y)$, *we have* $yg^{(1)}(\mathbf{x}; \sigma_\xi^{(2)}) < 1$. *Furthermore, for any* $i \in [2m]$, $\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| = \left|\langle q_i^{(1)}, \phi\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, $\left|\langle (q_i^{(1)})_{-\mathbf{A}}, \phi_{-\mathbf{A}}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, *and* $\left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right| = O(|\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_e)$.

Proof of Lemma A.2.13. Note that $\mathbf{w}_i^{(0)} = \mathbf{w}_{m+i}^{(0)}$, $\mathbf{b}_i^{(0)} = \mathbf{b}_{m+i}^{(0)}$, and $\mathbf{a}_i^{(0)} = -\mathbf{a}_{m+i}^{(0)}$. Then the gradient for $\mathbf{w}_i$ is the negation of that for $\mathbf{w}_{m+i}$, the gradient for $\mathbf{b}_i$ is the negation of that for $\mathbf{b}_{m+i}$, and the gradient for $\mathbf{a}_i$ is the same as that for $\mathbf{a}_{m+i}$. With probability $\geq 1 - \exp(-\Theta(\max\{2p_o(D-k), k\}))$, among all $j \notin \mathbf{A}$, we have that at most $2p_o(D-k)+k$

of ϕ_j are $(1-p_{\mathrm{o}})/\tilde{\sigma}$, while the others are $-p_{\mathrm{o}}/\tilde{\sigma}$. For data points with ϕ satisfying this, we have:

$$\left|g^{(1)}(\mathbf{x};\sigma_{\xi}^{(2)})\right| \tag{A.115}$$

$$= \left|\sum_{i=1}^{2m} \mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right| \tag{A.116}$$

$$= \left|\sum_{i=1}^{m} \left(\mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle + \mathbf{b}_{m+i}^{(1)} + \xi_{m+i}^{(2)})\right)\right| \tag{A.117}$$

$$\leq \left|\sum_{i=1}^{m} \left(\mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right)\right| \tag{A.118}$$

$$+ \left|\sum_{i=1}^{m} \left(-\mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle + \mathbf{b}_{m+i}^{(1)} + \xi_i^{(2)})\right)\right|. \tag{A.119}$$

Then we have

$$\left|g^{(1)}(\mathbf{x};\sigma_{\xi}^{(2)})\right| \leq \sum_{i=1}^{m} \left|2\eta^{(1)} T_a \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right| \tag{A.120}$$

$$+ \sum_{i=1}^{m} \left|\mathbf{a}_{m+i}^{(1)}\right| \left(\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| + \left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right|\right) \tag{A.121}$$

$$\leq \sum_{i=1}^{m} \left|2\eta^{(1)} T_a\right| \left(\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)}\right| + \mathbb{E}_{\xi^{(2)}} \left|\xi_i^{(2)}\right|\right) \tag{A.122}$$

$$+ \sum_{i=1}^{m} \left|\mathbf{a}_{m+i}^{(1)}\right| \left(\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| + \left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right|\right). \tag{A.123}$$

We have $|T_a| = O(\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$, and

$$\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| \le O(|\eta^{(1)}\mathbf{a}_i^{(0)}|)\,(\beta\gamma/\tilde{\sigma} + \epsilon_{\mathrm{e}}/\tilde{\sigma})\,\frac{k}{\tilde{\sigma}} \tag{A.124}$$

$$+ O(|\eta^{(1)}\mathbf{a}_i^{(0)}|\sigma_\phi^2\epsilon_{\mathrm{e}}\tilde{\sigma})\,((2p_{\mathrm{o}}(D-k)+k)(1-p_{\mathrm{o}})/\tilde{\sigma} + p_{\mathrm{o}}D/\tilde{\sigma}) \tag{A.125}$$

$$\le O(|\eta^{(1)}\mathbf{a}_i^{(0)}|)\left(k\gamma/\tilde{\sigma}^2 + \epsilon_{\mathrm{e}}k/\tilde{\sigma}^2 + (k+p_{\mathrm{o}}D)\sigma_\phi^2\epsilon_{\mathrm{e}}\right) \tag{A.126}$$

$$\le O(\eta^{(1)}(1+p_{\mathrm{o}}\tilde{\sigma})/\gamma). \tag{A.127}$$

$$\left|\mathbf{b}_i^{(1)}\right| \le \left|\mathbf{b}_i^{(0)}\right| + \left|\eta^{(1)}\mathbf{a}_i^{(0)}T_b\right| \tag{A.128}$$

$$\le \frac{\sqrt{\log(m/\delta)}}{k^2} + \left|\eta^{(1)}\mathbf{a}_i^{(0)}\frac{\epsilon_{\mathrm{e}}}{\tilde{\sigma}}\right|. \tag{A.129}$$

$$\mathbb{E}_{\xi^{(2)}}\left|\xi_i^{(2)}\right| \le O(\sigma_\xi^{(2)}). \tag{A.130}$$

$$|\mathbf{a}_{m+i}^{(1)}| \le |\mathbf{a}_i^{(0)}| + |\eta^{(1)}T_a| \le |\mathbf{a}_i^{(0)}| + O(\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}). \tag{A.131}$$

$$\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| = 2\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| = O(\eta^{(1)}(1+p_{\mathrm{o}}\tilde{\sigma})/\gamma). \tag{A.132}$$

$$\left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right| = 2|\eta^{(1)}\mathbf{a}_i^{(0)}T_b| = O(|\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}). \tag{A.133}$$

Then we have

$$\left|g^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right| \le O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\right)\left(\frac{\eta^{(1)}}{\gamma} + \frac{\sqrt{\log(m/\delta)}}{k^2} + \left|\eta^{(1)}\mathbf{a}_i^{(0)}\frac{\epsilon_{\mathrm{e}}}{\tilde{\sigma}}\right| + \sigma_\xi^{(2)}\right) \tag{A.134}$$

$$+ O\left(m(|\mathbf{a}_i^{(0)}| + \eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})\right)\left(\frac{\eta^{(1)}}{\gamma} + \left|\eta^{(1)}\mathbf{a}_i^{(0)}\frac{\epsilon_{\mathrm{e}}}{\tilde{\sigma}}\right|\right) \tag{A.135}$$

$$= O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\frac{\log(Dm/\delta)}{k} + m|\mathbf{a}_i^{(0)}|\left(\frac{\eta^{(1)}}{\gamma} + \left|\eta^{(1)}\mathbf{a}_i^{(0)}\frac{\epsilon_{\mathrm{e}}}{\tilde{\sigma}}\right|\right)\right) \tag{A.136}$$

$$= O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\frac{\log(Dm/\delta)}{k} + m|\mathbf{a}_i^{(0)}|\frac{\eta^{(1)}}{\gamma} + m\sigma_{\mathbf{a}}\eta^{(1)}\frac{k}{\gamma}\right) \tag{A.137}$$

$$< 1. \tag{A.138}$$

Then $\left|yg^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right| < 1$. Finally, the statement on $\left|\langle (q_i^{(1)})_{-\mathbf{A}}, \phi_{-\mathbf{A}}\rangle\right|$ follows from a similar

calculation on $\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| = \left|\langle q_i^{(1)}, \phi\rangle\right|$. □

We are now ready to analyze the gradients in the second gradient step.

Lemma A.2.14. *Fix $\delta \in (0,1)$ and suppose $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta), \mathbf{a}_i^{(0)} \in \mathcal{G}_{\mathbf{a}}(\delta)$ for all $i \in [2m]$. Let $\epsilon_{e2} := O\left(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_e)}{\tilde{\sigma}^2\sigma_\xi^{(2)}}\right) + \exp(-\Theta(k))$. If $k = \Omega(\log^2(Dm/\delta))$ and $k = O(D)$, $\sigma_{\mathbf{a}} \le \tilde{\sigma}^2/(\gamma k^2)$, $\eta^{(1)} = O\left(\frac{\gamma}{km\sigma_{\mathbf{a}}}\right)$, and $\sigma_\xi^{(2)} = 1/k^{3/2}$, then*

$$\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) = -\mathbf{a}_i^{(1)} \sum_{j=1}^{D} \mathbf{M}_j T_j \tag{A.139}$$

where T_j satisfies:

- *if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \le O(\epsilon_{e2}/\tilde{\sigma} + \eta^{(1)}/\sigma_\xi^{(2)} + \eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_e/(\tilde{\sigma}\sigma_\xi^{(2)}))$, where $\beta \in [\Omega(1), 1]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;*
- *if $j \notin \mathbf{A}$, then $|T_j| \le \frac{1}{\tilde{\sigma}}\exp(-\Theta(k)) + O(\sigma_\phi^2\epsilon_{e2}\tilde{\sigma})$.*

Proof of Lemma A.2.14. Consider one neuron index i and omit the subscript i in the parameters. By Lemma A.2.13, $\Pr[yg^{(1)}(\mathbf{x}; \xi^{(2)}) > 1] \le \exp(-\Theta(k))$. Let $\mathbb{I}_{\mathbf{x}} = \mathbb{I}[yg^{(1)}(\mathbf{x}; \xi^{(2)}) \le 1]$.

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{w}} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) && \text{(A.140)}\\
&= -\mathbf{a}^{(1)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y\mathbb{I}_{\mathbf{x}} \mathbb{E}_{\xi^{(2)}} \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\mathbf{x} \right\} && \text{(A.141)}\\
&= -\mathbf{a}^{(1)} \sum_{j=1}^{D} \mathbf{M}_j \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y\mathbb{I}_{\mathbf{x}} \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\phi_j \right\}}_{:=T_j}. && \text{(A.142)}
\end{aligned}$$

Let $T_{j1} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\phi_j \right\}$. We have

$$\begin{aligned}
&|T_j - T_{j1}| && \text{(A.143)}\\
&= \left| \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y(1-\mathbb{I}_{\mathbf{x}})\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\phi_j \right\} \right| && \text{(A.144)}\\
&\le \frac{1}{\tilde{\sigma}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} |1 - \mathbb{I}_{\mathbf{x}}| && \text{(A.145)}\\
&\le \frac{1}{\tilde{\sigma}} \exp(-\Theta(k)). && \text{(A.146)}
\end{aligned}$$

So it is sufficient to bound T_{j1}. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

First, consider $j \in \mathbf{A}$.

$$T_{j1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{y\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\phi_j\right\} \tag{A.147}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j \Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\}. \tag{A.148}$$

Let

$$I_a := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right], \tag{A.149}$$

$$I'_a := \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.150}$$

By the property of the Gaussian $\xi^{(2)}$, that $|\langle \phi_{\mathbf{A}}, q_{\mathbf{A}}\rangle| = O(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_{\mathrm{e}})}{\tilde{\sigma}^2})$, and that $|\langle \phi, q\rangle| = |\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle| = O(\eta^{(1)}/\gamma) < O(1/k)$ and $|\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle| = O(\eta^{(1)}/\gamma) < O(1/k)$, we have

$$|I_a - I'_a| \le \left|\Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \ge 0\right] - \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \ge 0\right]\right| \tag{A.151}$$

$$+ \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \ge 1\right] + \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \ge 1\right] \tag{A.152}$$

$$= O\left(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_{\mathrm{e}})}{\tilde{\sigma}^2\sigma_\xi^{(2)}}\right) + \exp(-\Theta(k)) = O(\epsilon_{\mathrm{e}2}). \tag{A.153}$$

This leads to

$$\left|T_{j1} - \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{y\phi_j I'_a\right\}\right| \tag{A.154}$$

$$\le \mathbb{E}_{\phi_{\mathbf{A}}}\left\{|y\phi_j|\left|\mathbb{E}_{\phi_{-\mathbf{A}}}(I_a - I'_a)\right|\right\} \tag{A.155}$$

$$\le O(\epsilon_{\mathrm{e}2})\mathbb{E}_{\phi_{\mathbf{A}}}\left\{|y\phi_j|\right\} \tag{A.156}$$

$$\le O(\epsilon_{\mathrm{e}2}/\tilde{\sigma}) \tag{A.157}$$

where the last step is from Lemma A.2.8. Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{y\phi_j I'_a\right\} \tag{A.158}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j\right\}\mathbb{E}_{\phi_{-\mathbf{A}}}[I'_a] \tag{A.159}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j\right\}\Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.160}$$

By Lemma A.2.13, we have $|\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle| \leq O(\eta^{(1)}\tilde{\sigma}/\gamma)$. Also, $|b^{(1)} - b^{(0)}| \leq O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}})$. By the property of $\xi^{(2)}$,

$$\left|\Pr_{\xi^{(2)}}\left[\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right] - \Pr_{\xi^{(2)}}\left[\mathbf{b}^{(0)} + \xi^{(2)} \in (0,1)\right]\right| \tag{A.161}$$

$$\leq O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)})) + O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/\sigma_\xi^{(2)}). \tag{A.162}$$

On the other hand,

$$\beta := \Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\mathbf{b}^{(0)} + \xi^{(2)} \in (0,1)\right] = \Pr_{\xi^{(2)}}\left[\xi^{(2)} \in (-\mathbf{b}^{(0)}, 1 - \mathbf{b}^{(0)})\right] \tag{A.163}$$

$$= \Omega(1) \tag{A.164}$$

and β only depends on $\mathbf{b}^{(0)}$. By Lemma A.2.8, $\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\phi_j\right\} = \gamma/\tilde{\sigma}$. Therefore,

$$|T_{j1} - \beta\gamma/\tilde{\sigma}| \leq O(\epsilon_{\mathrm{e}2}/\tilde{\sigma}) + O(\eta^{(1)}/\sigma_\xi^{(2)}) + O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/(\tilde{\sigma}\sigma_\xi^{(2)})). \tag{A.165}$$

Now, consider $j \notin \mathbf{A}$. Let B denote $\mathbf{A} \cup \{j\}$.

$$T_{j1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{y\phi_j\mathbb{I}\left[\langle\phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\} \tag{A.166}$$

$$= \mathbb{E}_{\phi_B}\mathbb{E}_{\phi_{-B},\xi^{(2)}}\left\{y\phi_j\mathbb{I}\left[\langle\phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\} \tag{A.167}$$

$$= \mathbb{E}_{\phi_B}\left\{y\phi_j \Pr_{\phi_{-B},\xi^{(2)}}\left[\langle\phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\}. \tag{A.168}$$

Let

$$I_b := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right], \tag{A.169}$$

$$I'_b := \Pr_{\xi^{(2)}}\left[\langle \phi_{-B}, q_{-B}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.170}$$

Similar as above, we have $|I_b - I'_b| \le \epsilon_{\mathrm{e}2}$. Then by Lemma A.2.8,

$$\left|T_{j1} - \mathbb{E}_{\phi_B,\phi_{-B}}\left\{y\phi_j I'_b\right\}\right| \tag{A.171}$$

$$\le \mathbb{E}_{\phi_B}\left\{|y\phi_j||\mathbb{E}_{\phi_{-B}}(I_b - I'_b)|\right\} \tag{A.172}$$

$$\le O(\epsilon_{\mathrm{e}2})\mathbb{E}_{\phi_j}\left\{|\phi_j|\right\} \tag{A.173}$$

$$\le O(\epsilon_{\mathrm{e}}) \times O(\sigma_\phi^2\tilde{\sigma}) \tag{A.174}$$

$$= O(\sigma_\phi^2\epsilon_{\mathrm{e}2}\tilde{\sigma}). \tag{A.175}$$

Furthermore,

$$\mathbb{E}_{\phi_B,\phi_{-B}}\left\{y\phi_j I'_b\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\right\}\mathbb{E}_{\phi_j}\left\{\phi_j\right\}\mathbb{E}_{\phi_{-B}}[I'_b] = 0. \tag{A.176}$$

Therefore,

$$|T_{j1}| \le O(\sigma_\phi^2\epsilon_{\mathrm{e}2}\tilde{\sigma}). \tag{A.177}$$

□

Lemma A.2.15. *Under the same assumptions as in Lemma A.2.14,*

$$\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) = -\mathbf{a}_i^{(1)} T_b \tag{A.178}$$

where $|T_b| \le \exp(-\Omega(k)) + O(\epsilon_{e2})$.

Proof of Lemma A.2.15. Consider one neuron index i and omit the subscript i in the param-

eters. By Lemma A.2.13, $\Pr[yg^{(1)}(\mathbf{x};\xi^{(2)}) > 1] \le \exp(-\Omega(k))$. Let $\mathbb{I}_{\mathbf{x}} = \mathbb{I}[yg^{(1)}(\mathbf{x};\xi^{(2)}) \le 1]$.

$$\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) \tag{A.179}$$

$$= -\mathbf{a}^{(1)} \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ y\mathbb{I}_{\mathbf{x}} \mathbb{E}_{\xi^{(2)}} \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\}}_{:=T_b}. \tag{A.180}$$

Let $T_{b1} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\}$. We have

$$|T_b - T_{b1}| \tag{A.181}$$

$$= \left| \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y(1-\mathbb{I}_{\mathbf{x}})\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\} \right| \tag{A.182}$$

$$\le \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} |1 - \mathbb{I}_{\mathbf{x}}| \tag{A.183}$$

$$\le \exp(-\Omega(k)). \tag{A.184}$$

So it is sufficient to bound T_{b1}. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

$$T_{b1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ y\mathbb{I}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\} \tag{A.185}$$

$$= \mathbb{E}_{\phi_A} \mathbb{E}_{\phi_{-A},\xi^{(2)}} \left\{ y\mathbb{I}\left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\} \tag{A.186}$$

$$= \mathbb{E}_{\phi_A} \left\{ y \Pr_{\phi_{-A},\xi^{(2)}} \left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\}. \tag{A.187}$$

Let

$$I_b := \Pr_{\xi^{(2)}} \left[\langle \phi, q\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right], \tag{A.188}$$

$$I_b' := \Pr_{\xi^{(2)}} \left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right]. \tag{A.189}$$

Similar as in Lemma A.2.14, we have $|I_b - I_b'| \le \epsilon_{\text{e2}}$. Then by Lemma A.2.8,

$$\left|T_{b1} - \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{yI_b'\right\}\right| \tag{A.190}$$

$$\le \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\left|\mathbb{E}_{\phi_{-\mathbf{A}}}(I_b - I_b')\right|\right\} \tag{A.191}$$

$$\le O(\epsilon_{\text{e2}}). \tag{A.192}$$

Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{yI_b'\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\right\}\mathbb{E}_{\phi_{-\mathbf{A}}}[I_b'] = 0. \tag{A.193}$$

Therefore, $|T_{b1}| \le O(\epsilon_{\text{e2}})$ and the statement follows. □

Lemma A.2.16. *Under the same assumptions as in Lemma A.2.14,*

$$\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) = -T_a \tag{A.194}$$

where $|T_a| = O(\eta^{(1)}\tilde{\sigma}/\gamma) + \exp(-\Omega(k))\text{poly}(Dm)$.

Proof of Lemma A.2.16. Consider one neuron index i and omit the subscript i in the parameters. By Lemma A.2.13, $\Pr[yg^{(1)}(\mathbf{x}; \xi^{(2)}) > 1] \le \exp(-\Omega(k))$. Let $\mathbb{I}_{\mathbf{x}} = \mathbb{I}[yg^{(1)}(\mathbf{x}; \xi^{(2)}) \le 1]$.

$$\frac{\partial}{\partial \mathbf{a}} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) \tag{A.195}$$

$$= -\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}_{\mathbf{x}}\mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}}_{:=T_a}. \tag{A.196}$$

Let $T_{a1} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}$. We have

$$|T_a - T_{a1}| \tag{A.197}$$

$$= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y(1-\mathbb{I}_{\mathbf{x}})\mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}\right| \tag{A.198}$$

$$\le \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left|1 - \mathbb{I}_{\mathbf{x}}\right| \tag{A.199}$$

$$\le \exp(-\Omega(k)). \tag{A.200}$$

So it is sufficient to bound T_{a1}. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

Let $\phi'_{\mathbf{A}}$ be an independent copy of $\phi_{\mathbf{A}}$, ϕ' be the vector obtained by replacing in ϕ the entries $\phi_{\mathbf{A}}$ with $\phi'_{\mathbf{A}}$, and let $x' = \mathbf{M}\phi'$ and its label is y'. Then

$$
\begin{aligned}
|T_{a1}| &:= \left|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{y\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}\right| && \text{(A.201)}\\
&\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(1)})|y=1\right\} && \text{(A.202)}\\
&\quad - \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y=-1\right\}\Bigg| && \text{(A.203)}\\
&\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y=1\right\} && \text{(A.204)}\\
&\quad - \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}'\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y'=-1\right\}\Bigg| && \text{(A.205)}\\
&\le \frac{1}{2}\mathbb{E}_{\phi_{\mathbf{A}},\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}}}\left|\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle - \langle \mathbf{w}^{(1)}, \mathbf{x}'\rangle\right| |y=1, y'=-1\right\} && \text{(A.206)}\\
&\le \frac{1}{2}\mathbb{E}_{\phi_{-\mathbf{A}}}\left(\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle\right| |y=1\right\} + \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(1)}, \mathbf{x}'\rangle\right| |y'=-1\right\}\right) && \text{(A.207)}\\
&\le \mathbb{E}_{\phi_{-\mathbf{A}},\phi_{\mathbf{A}}}\left|\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle\right| && \text{(A.208)}\\
&= \mathbb{E}_{\mathbf{x}}\left|\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle\right| && \text{(A.209)}\\
&= O(\eta^{(1)}\tilde{\sigma}/\gamma) + \exp(-\Omega(k)) \times D \times \|q^{(1)}\|_\infty \|\phi\|_\infty && \text{(A.210)}\\
&= O(\eta^{(1)}\tilde{\sigma}/\gamma) + \exp(-\Omega(k))\frac{D|\eta^{(1)}\mathbf{a}^{(0)}|(\gamma+\epsilon_{\mathrm{e}})}{\tilde{\sigma}^2} && \text{(A.211)}\\
&= O(\eta^{(1)}\tilde{\sigma}/\gamma) + \exp(-\Omega(k))\mathrm{poly}(Dm) && \text{(A.212)}
\end{aligned}
$$

where the fourth step follows from that σ is 1-Lipschitz, the third to the last step from Lemma A.2.13, and the second to the last step from Lemma A.2.12. □

With the above lemmas about the gradients, we are now ready to show that at the end of the second step, we get a good set of features for accurate prediction.

Lemma A.2.17. *Set*

$$\eta^{(1)} = \frac{\gamma^2\tilde{\sigma}}{km^3}, \lambda_{\mathbf{a}}^{(1)} = 0, \lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \sigma_\xi^{(1)} = 1/k^{3/2}, \tag{A.213}$$

$$\eta^{(2)} = 1, \lambda_{\mathbf{a}}^{(2)} = \lambda_{\mathbf{w}}^{(2)} = 1/(2\eta^{(2)}), \sigma_\xi^{(2)} = 1/k^{3/2}. \tag{A.214}$$

Fix $\delta \in (0, O(1/k^3))$. *If* $p_o = \Omega(k^2/D)$, $k = \Omega\left(\log^2\left(\frac{Dm}{\delta\gamma}\right)\right)$, *and* $m \geq \max\{\Omega(k^4), D\}$, *then with probability at least* $1-\delta$ *over the initialization, there exist* $\tilde{\mathbf{a}}_i$*'s such that* $\tilde{g}(\mathbf{x}) := \sum_{i=1}^{2m} \tilde{\mathbf{a}}_i \sigma(\langle \mathbf{w}_i^{(2)}, \mathbf{x}\rangle + \mathbf{b}_i^{(2)})$ *satisfies* $L_{\mathcal{D}}(\tilde{g}) = 0$. *Furthermore,* $\|\tilde{\mathbf{a}}\|_0 = O(m/k)$, $\|\tilde{\mathbf{a}}\|_\infty = O(k^5/m)$, *and* $\|\tilde{\mathbf{a}}\|_2^2 = O(k^9/m)$. *Finally,* $\|\mathbf{a}^{(2)}\|_\infty = O\left(\frac{1}{km^2}\right)$, $\|\mathbf{w}_i^{(2)}\|_2 = O(\tilde{\sigma}/k)$, *and* $|\mathbf{b}_i^{(2)}| = O(1/k^2)$ *for all* $i \in [2m]$.

Proof of Lemma A.2.17. By Lemma 2.3.1, there exists a network $g^*(\mathbf{x}) = \sum_{\ell=1}^{3(k+1)} \mathbf{a}_\ell^* \sigma(\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*)$ satisfying $g^*(\mathbf{x})$ for all $(\mathbf{x}, y) \sim \mathcal{D}$. Furthermore, $|\mathbf{a}_i^*| \leq 32k, 1/(32k) \leq |\mathbf{b}_i^*| \leq 1/2$, $\mathbf{w}_i^* = \tilde{\sigma}\sum_{j\in\mathbf{A}} \mathbf{M}_j/(4k)$, and $|\langle \mathbf{w}_i^*, \mathbf{x}\rangle + \mathbf{b}_i^*| \leq 1$ for any $i \in [n]$ and $(\mathbf{x}, y) \sim \mathcal{D}$. Now we fix an ℓ, and show that with high probability there is a neuron in $g^{(2)}$ that can approximate the ℓ-th neuron in g^*.

By Lemma A.2.4, with probability $1-2\delta$ over $\mathbf{w}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{w}}(\delta)$; with probability $1-\delta$ over $\mathbf{a}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{a}}(\delta)$; with probability $1-\delta$ over $\mathbf{b}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{b}}(\delta)$. Under these events, by Lemma A.2.12, Lemma A.2.14 and A.2.15, for any neuron $i \in [2m]$, we have

$$\mathbf{w}_i^{(2)} = \mathbf{a}_i^{(1)} \sum_{j=1}^{D} \mathbf{M}_j T_j, \tag{A.215}$$

$$\mathbf{b}_i^{(2)} = \mathbf{b}_i^{(1)} + \mathbf{a}_i^{(1)} T_b. \tag{A.216}$$

where

- if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \leq \epsilon_{\mathbf{w}1} := O(\epsilon_{\mathrm{e}2}/\tilde{\sigma} + \eta^{(1)}/\sigma_\xi^{(2)} + \eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/(\tilde{\sigma}\sigma_\xi^{(2)}))$, where $\beta \in [\Omega(1), 1]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;
- if $j \notin \mathbf{A}$, then $|T_j| \leq \epsilon_{\mathbf{w}2} := \frac{1}{\tilde{\sigma}}\exp(-\Theta(k)) + O(\sigma_\phi^2 \epsilon_{\mathrm{e}2}\tilde{\sigma})$.

- $|T_b| \le \epsilon_b := \frac{1}{\tilde{\sigma}} \exp(-\Theta(k)) + O(\epsilon_{\text{e2}})$.

Given the initialization, with probability $\Omega(1)$ over $\mathbf{b}_i^{(0)}$, we have

$$|\mathbf{b}_i^{(0)}| \in \left[\frac{1}{2k^2}, \frac{2}{k^2}\right], \text{sign}(\mathbf{b}_i^{(0)}) = \text{sign}(\mathbf{b}_\ell^*). \tag{A.217}$$

Finally, since $\frac{4k|\mathbf{b}_\ell^*|\beta\gamma}{|\mathbf{b}_i^{(0)}|\tilde{\sigma}^2} \in [\Omega(k^2\gamma/\tilde{\sigma}^2), O(k^3\gamma/\tilde{\sigma}^2)]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$, we have that for $\epsilon_{\mathbf{a}} = \Theta(1/k^2)$, with probability $\Omega(\epsilon_{\mathbf{a}}) > \delta$ over $\mathbf{a}_i^{(0)}$,

$$\left|\frac{4k|\mathbf{b}_\ell^*|\beta\gamma}{|\mathbf{b}_i^{(0)}|\tilde{\sigma}^2}\mathbf{a}_i^{(0)} - 1\right| \le \epsilon_{\mathbf{a}}, \quad |\mathbf{a}_i^{(0)}| = O\left(\frac{\tilde{\sigma}^2}{k^2\gamma}\right). \tag{A.218}$$

Let $n_a = \epsilon_{\mathbf{a}} m/4$. For the given value of m, by (A.215)-(A.218) we have with probability $\ge 1 - 5\delta$ over the initialization, for each ℓ there is a different set of neurons $I_\ell \subseteq [m]$ with $|I_\ell| = n_a$ and such that for each $i_\ell \in I_\ell$,

$$|\mathbf{b}_{i_\ell}^{(0)}| \in \left[\frac{1}{2k^2}, \frac{2}{k^2}\right], \quad \text{sign}(\mathbf{b}_{i_\ell}^{(0)}) = \text{sign}(\mathbf{b}_\ell^*), \tag{A.219}$$

$$\left|\frac{4k|\mathbf{b}_\ell^*|\beta\gamma}{|b_{i_\ell}^{(0)}|\tilde{\sigma}^2}\mathbf{a}_{i_\ell}^{(0)} - 1\right| \le \epsilon_{\mathbf{a}}, \quad |\mathbf{a}_{i_\ell}^{(0)}| = O\left(\frac{\tilde{\sigma}^2}{k^2\gamma}\right). \tag{A.220}$$

We also have

$$\left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.221}$$

$$\le \left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| + \left|\frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.222}$$

$$= \left|\mathbf{a}_{i_\ell}^{(1)} \sum_{j=1}^{D} T_j\phi_j - \frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \phi_j\right| + \left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| \left|\frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \phi_j\right| \tag{A.223}$$

$$= \left|\mathbf{a}_{i_\ell}^{(1)}\right| \left|\sum_{j=1}^{D} T_j\phi_j - \frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \phi_j\right| + \left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| \left|\frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \phi_j\right|. \tag{A.224}$$

We have $\left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| = O(\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$, and

$$\left|\sum_{j=1}^{D} T_j\phi_j - \frac{\beta\gamma}{\tilde{\sigma}}\sum_{j\in\mathbf{A}}\phi_j\right| \le \left|\sum_{j\in\mathbf{A}}(T_j - \frac{\beta\gamma}{\tilde{\sigma}})\phi_j\right| + \left|\sum_{j\notin\mathbf{A}} T_j\phi_j\right| \tag{A.225}$$

$$\le O(k\epsilon_{\mathbf{w}1}/\tilde{\sigma}) + O(D\epsilon_{\mathbf{w}2}/\tilde{\sigma}) =: \epsilon_\phi. \tag{A.226}$$

For the given values of parameters, we have

$$\epsilon_{\mathrm{e}2} = O\left(\frac{\gamma}{m^2}\right), \tag{A.227}$$

$$\epsilon_{\mathbf{w}1} = O\left(\frac{k\gamma}{m^2\tilde{\sigma}} + \frac{\gamma\epsilon_{\mathrm{e}}}{km^2}\right), \tag{A.228}$$

$$\epsilon_{\mathbf{w}2} = O\left(\frac{\gamma}{m^2\tilde{\sigma}}\right), \tag{A.229}$$

$$\epsilon_b = O\left(\frac{\gamma}{m^2}\right), \tag{A.230}$$

$$\epsilon_\phi = O\left(\frac{k^2\gamma}{m^2\tilde{\sigma}^2} + \frac{\gamma\epsilon_{\mathrm{e}}}{m^2\tilde{\sigma}} + \frac{\gamma}{m\tilde{\sigma}^2}\right). \tag{A.231}$$

Therefore,

$$\left|\langle\mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}}\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.232}$$

$$\le \left|\mathbf{a}_{i_\ell}^{(1)}\right|\epsilon_\phi + \left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right|\frac{k\gamma}{\tilde{\sigma}^2} \tag{A.233}$$

$$\le O\left(\frac{\tilde{\sigma}^2}{k^2\gamma} + \eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\right)\left(\frac{k^2\gamma}{m^2\tilde{\sigma}^2} + \frac{\gamma\epsilon_{\mathrm{e}}}{m^2\tilde{\sigma}} + \frac{\gamma}{m\tilde{\sigma}^2}\right) \tag{A.234}$$

$$+ O\left(\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\right)\frac{k\gamma}{\tilde{\sigma}^2} \tag{A.235}$$

$$\le O\left(\frac{1}{m}\right). \tag{A.236}$$

We also have by Lemma A.2.12 and A.2.15:

$$|\mathbf{b}_{i_\ell}^{(2)} - \mathbf{b}_{i_\ell}^{(0)}| \le O\left(\eta^{(1)}|\mathbf{a}_{i_\ell}^{(0)}|\epsilon_{\mathrm{e}} + |\mathbf{a}_{i_\ell}^{(1)}|\left(\frac{1}{\tilde{\sigma}}\exp(-\Theta(k)) + \epsilon_{\mathrm{e}2}\right)\right) \le O\left(\frac{1}{m}\right). \tag{A.237}$$

Now, construct $\tilde{\mathbf{a}}$ such that $\tilde{\mathbf{a}}_{i_\ell} = \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a}$ for each ℓ and each $i_\ell \in I_\ell$, and $\tilde{\mathbf{a}}_i = 0$ elsewhere. Then

$$|\tilde{g}(\mathbf{x}) - 2g^*(\mathbf{x})| \tag{A.238}$$

$$= \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \tilde{\mathbf{a}}_{i_\ell} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \sum_{\ell=1}^{3(k+1)} 2\mathbf{a}_\ell^* \sigma\left(\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*\right) \right| \tag{A.239}$$

$$= \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a} \sigma\left(\frac{|b_{i_\ell}^{(0)}|}{|\mathbf{b}_\ell^*|}\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + b_{i_\ell}^{(0)}\right) \right| \tag{A.240}$$

$$\leq \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{1}{n_a} \left(\frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(0)} \right) \right) \right| \tag{A.241}$$

$$+ \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{1}{n_a} \left(\frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(0)} \right) - \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{|b_{i_\ell}^{(0)}|}{|\mathbf{b}_\ell^*|}\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + b_{i_\ell}^{(0)} \right) \right) \right| \tag{A.242}$$

$$\leq 3(k+1) \max_\ell \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|\mathbf{b}_{i_\ell}^{(0)}|} O\left(\frac{1}{m}\right) + \tag{A.243}$$

$$3(k+1) \max_\ell \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|\mathbf{b}_{i_\ell}^{(0)}|} \frac{\tilde{\sigma}|\mathbf{b}_{i_\ell}^{(0)}|}{4k|\mathbf{b}_\ell^*|} \left| \frac{4k\mathbf{a}_{i_\ell}^{(0)}\beta\gamma|\mathbf{b}_\ell^*|}{\tilde{\sigma}^2|\mathbf{b}_{i_\ell}^{(0)}|} - 1 \right| \frac{k}{\tilde{\sigma}} \tag{A.244}$$

$$= O\left(\frac{k^4}{m} + k^2\epsilon_{\mathbf{a}}\right) \tag{A.245}$$

$$\leq 1. \tag{A.246}$$

Here the second equation follows from that σ is positive-homogeneous in $[0,1]$, $|\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*| \leq 1$, $|\mathbf{b}_{i_\ell}^{(0)}|/|\mathbf{b}_\ell^*| \leq 1$. This guarantees $y\tilde{g}(\mathbf{x}) \geq 1$. Changing the scaling of δ leads to the statement.

Finally, the bounds on $\tilde{\mathbf{a}}$ follow from the above calculation. The bound on $\|\mathbf{a}^{(2)}\|_2$ follows from Lemma A.2.16, and those on $\|\mathbf{w}_i^{(2)}\|_2$ and $\|\mathbf{b}_i^{(2)}\|_2$ follow from (A.215)(A.216) and the bounds on $\mathbf{a}_i^{(1)}$ and $\mathbf{b}_i^{(1)}$ in Lemma A.2.12. □

A.2.6 Classifier Learning Stage

Once we have a set of good features, we are now ready to prove that the later steps will learn an accurate classifier. The intuition is that the first layer's weights do not change

much and the second layer's weights get updated till achieving good accuracy. In particular, we will employ the online optimization technique from [63].

We begin by showing that the first layer's weights do not change too much.

Lemma A.2.18. *Assume the same conditions as in Lemma A.2.17. Suppose for $t > 2$, $\lambda_{\mathbf{a}}^{(t)} = \lambda_{\mathbf{w}}^{(t)} = \lambda$, $\eta^{(t)} = \eta$ for some $\lambda, \eta \in (0,1)$, and $\sigma_{\xi}^{(t)} = 0$. Then for any $t > 2$ and $i \in [2m]$,*

$$|\mathbf{a}_i^{(t)}| \leq \eta t + O\left(\frac{1}{km^2}\right), \tag{A.247}$$

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(2)}\|_2 \leq O\left(\frac{t\eta\lambda\tilde{\sigma}}{k}\right) + \eta^2 t^2 + O\left(\frac{t}{km^2}\right), \tag{A.248}$$

$$|\mathbf{b}_i^{(t)} - \mathbf{b}_i^{(2)}| \leq O\left(\frac{t\eta\lambda}{k^2}\right) + \eta^2 t^2 + O\left(\frac{t}{km^2}\right). \tag{A.249}$$

Proof of Lemma A.2.18. First, we bound the size of $|\mathbf{a}_i^{(t)}|$:

$$|\mathbf{a}_i^{(t)}| = \left|(1-2\eta\lambda)\mathbf{a}_i^{(t-1)} - \eta\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}(g^{(t-1)})\right| \tag{A.250}$$

$$\leq \left|(1-2\eta\lambda)\mathbf{a}_i^{(t-1)} - \eta\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg^{(t-1)}(\mathbf{x}) \leq 1]\sigma(\langle \mathbf{w}_i^{(t-1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t-1)})\right\}\right| \tag{A.251}$$

$$\leq |\mathbf{a}_i^{(t-1)}| + \eta \tag{A.252}$$

which leads to

$$|\mathbf{a}_i^{(t)}| \leq \eta t + |\mathbf{a}_i^{(2)}| \tag{A.253}$$

where $|\mathbf{a}_i^{(2)}| = O\left(\frac{1}{km^2}\right)$. We are now to bound the change of $\mathbf{w}_i^{(t)}$ and $\mathbf{b}_i^{(t)}$.

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(2)}\|_2 \tag{A.254}$$

$$= \left\|(1-2\eta\lambda)\mathbf{w}_i^{(t-1)} - \eta\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}(g^{(t-1)}) - \mathbf{w}_i^{(2)}\right\|_2 \tag{A.255}$$

$$\leq \Bigg\|(1-2\eta\lambda)\mathbf{w}_i^{(t-1)} \tag{A.256}$$

$$+ \eta\mathbf{a}_i^{(t-1)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg^{(t-1)}(\mathbf{x}) \leq 1]\mathbb{I}[\langle \mathbf{w}_i^{(t-1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t-1)} \in (0,1)]\mathbf{x}\right\} - \mathbf{w}_i^{(2)}\Bigg\|_2 \tag{A.257}$$

$$\leq \left\|(1-2\eta\lambda)\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(2)}\right\|_2 \tag{A.258}$$

$$+ \eta\left\|\mathbf{a}_i^{(t-1)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg^{(t-1)}(\mathbf{x}) \leq 1]\mathbb{I}[\langle \mathbf{w}_i^{(t-1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t-1)} \in (0,1)]\mathbf{x}\right\}\right\|_2 \tag{A.259}$$

$$\leq (1-2\eta\lambda)\left\|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(2)}\right\|_2 + 2\eta\lambda\left\|\mathbf{w}_i^{(2)}\right\|_2 + \eta\left|\mathbf{a}_i^{(t-1)}\right| \tag{A.260}$$

leading to

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(2)}\|_2 \leq 2t\eta\lambda\left\|\mathbf{w}_i^{(2)}\right\|_2 + \eta^2t^2 + t|\mathbf{a}_i^{(2)}|. \tag{A.261}$$

Note that $\|\mathbf{w}_i^{(2)}\|_2 = O(\tilde{\sigma}/k)$.

$$|\mathbf{b}_i^{(t)} - \mathbf{b}_i^{(2)}| = \left|\mathbf{b}_i^{(t-1)} - \eta\frac{\partial}{\partial \mathbf{b}_i} L_{\mathcal{D}}(g^{(t-1)}) - \mathbf{b}_i^{(2)}\right| \tag{A.262}$$

$$\leq \left|\mathbf{b}_i^{(t-1)} - \mathbf{b}_i^{(2)}\right| \tag{A.263}$$

$$+ \eta\left|\mathbf{a}_i^{(t-1)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{y\mathbb{I}[yg^{(t-1)}(\mathbf{x}) \leq 1]\mathbb{I}[\langle \mathbf{w}_i^{(t-1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t-1)} \in (0,1)]\right\}\right| \tag{A.264}$$

$$\leq \left|\mathbf{b}_i^{(t-1)} - \mathbf{b}_i^{(2)}\right| + \eta\left|\mathbf{a}_i^{(t-1)}\right| \tag{A.265}$$

leading to

$$|\mathbf{b}_i^{(t)} - \mathbf{b}_i^{(2)}| \leq \eta^2t^2 + t|\mathbf{a}_i^{(2)}|. \tag{A.266}$$

Note that $\left|\mathbf{b}_i^{(2)}\right| = O(1/k^2)$. □

Lemma A.2.19. *Assume the same conditions as in Lemma A.2.18. Let* $g_{\tilde{\mathbf{a}}}^{(t)}(\mathbf{x}) = \sum_{i=1}^m \tilde{\mathbf{a}}_i \sigma(\langle \mathbf{w}_i^{(t)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t)})$. *Then*

$$|\ell(g_{\tilde{\mathbf{a}}}^{(t)}(\mathbf{x}), y) - \ell(g_{\tilde{\mathbf{a}}}^{(2)}(\mathbf{x}), y)| \leq \|\tilde{\mathbf{a}}\|_2 \sqrt{\|\tilde{\mathbf{a}}\|_0} \left(O\left(\frac{t\eta\lambda\tilde{\sigma}}{k}\right) + \eta^2 t^2 + O\left(\frac{t}{km^2}\right)\right). \quad \text{(A.267)}$$

Proof of Lemma A.2.19. It follows from that

$$\begin{aligned}
&|\ell(g_{\tilde{\mathbf{a}}}^{(t)}(\mathbf{x}), y) - \ell(g_{\tilde{\mathbf{a}}}^{(2)}(\mathbf{x}))| && \text{(A.268)}\\
&\leq |g_{\tilde{\mathbf{a}}}^{(t)}(\mathbf{x}) - g_{\tilde{\mathbf{a}}}^{(2)}(\mathbf{x})| && \text{(A.269)}\\
&\leq \|\tilde{\mathbf{a}}\|_2 \sqrt{\|\tilde{\mathbf{a}}\|_0} \max_{i\in[2m]} \left|\sigma(\langle \mathbf{w}_i^{(t)}, \mathbf{x}\rangle + \mathbf{b}_i^{(t)}) - \sigma(\langle \mathbf{w}_i^{(2)}, \mathbf{x}\rangle + \mathbf{b}_i^{(2)})\right| && \text{(A.270)}\\
&\leq \|\tilde{\mathbf{a}}\|_2 \sqrt{\|\tilde{\mathbf{a}}\|_0} \max_{i\in[2m]} \left(\left|\langle \mathbf{w}_i^{(t)} - \mathbf{w}_i^{(2)}, \mathbf{x}\rangle\right| + \left|\mathbf{b}_i^{(t)} - \mathbf{b}_i^{(2)}\right|\right). && \text{(A.271)}
\end{aligned}$$

and Lemma A.2.18. □

A.2.7 Proof of Theorem 2.2.1

Based on the above lemmas, following the same argument as in the proof of Theorem 2 in [63], we get our main theorem.

Theorem A.2.20 (Full version of Theorem 2.2.1)**.** *Set*

$$\begin{aligned}
&\eta^{(1)} = \frac{\gamma^2\tilde{\sigma}^2}{km^3}, \lambda_{\mathbf{a}}^{(1)} = 0, \lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \sigma_\xi^{(1)} = 1/k^2, && \text{(A.272)}\\
&\eta^{(2)} = 1, \lambda_{\mathbf{a}}^{(2)} = \lambda_{\mathbf{w}}^{(2)} = 1/(2\eta^{(2)}), \sigma_\xi^{(2)} = 1/k^2, && \text{(A.273)}\\
&\eta^{(t)} = \eta = \frac{k^2}{Tm^{1/3}}, \lambda_{\mathbf{a}}^{(t)} = \lambda_{\mathbf{w}}^{(t)} = \lambda \leq \frac{k^3}{\tilde{\sigma}m^{1/3}}, \sigma_\xi^{(t)} = 0, \text{ for } 2 < t \leq T. && \text{(A.274)}
\end{aligned}$$

For any $\delta \in (0,1)$, *if* $p_o = \Omega(k^2/D)$, $k = \Omega\left(\log^2\left(\frac{D}{\delta\gamma}\right)\right)$, $\max\{\Omega(k^4), D\} \leq m \leq poly(D)$,

then we have for any $\mathcal{D} \in \mathcal{F}_\Xi$, with probability at least $1-\delta$, there exists $t \in [T]$ such that

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \neq y] \leq L_{\mathcal{D}}(g^{(t)}) = O\left(\frac{k^8}{m^{2/3}} + \frac{k^3 T}{m^2} + \frac{k^2 m^{2/3}}{T}\right). \tag{A.275}$$

Consequently, for any $\epsilon \in (0,1)$, if $T = m^{4/3}$, and $\max\{\Omega(k^{12}/\epsilon^{3/2}), D\} \leq m \leq \text{poly}(D)$, then

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \neq y] \leq L_{\mathcal{D}}(g^{(t)}) \leq \epsilon. \tag{A.276}$$

Proof of Theorem 2.2.1. Consider $\tilde{L}_{\mathcal{D}}(g^{(t)}) = \mathbb{E}[\ell(g^{(t)}, y)] + \lambda_{\mathbf{a}}^{(t)} \|\mathbf{a}^{(t)}\|_2^2$. Note that the gradient update using $\tilde{L}_{\mathcal{D}}(g^{(t)})$ is the same as the update in our learning algorithm. Then by Theorem A.2.21, Lemma A.2.17, and Lemma A.2.19,

$$\frac{1}{T}\sum_{t=3}^{T} \tilde{L}_{\mathcal{D}}(g^{(t)}) \leq \frac{\|\tilde{\mathbf{a}}\|_2^2}{2} + \|\tilde{\mathbf{a}}\|_2 \sqrt{\|\tilde{\mathbf{a}}\|_0} \left(O\left(\frac{T\eta\lambda\tilde{\sigma}}{k}\right) + \eta^2 T^2 + O\left(\frac{T}{km^2}\right)\right) \tag{A.277}$$

$$+ \frac{\|\tilde{\mathbf{a}}\|_2^2}{2\eta T} + \|\mathbf{a}^{(2)}\|_2 \sqrt{m} + \eta m \tag{A.278}$$

$$\leq O\left(\frac{k^9}{m} + k^4\eta^2 T^2 + \frac{k^3 T}{m^2} + \frac{k^9}{\eta T m} + \eta m\right). \tag{A.279}$$

$$\leq O\left(\frac{k^8}{m^{2/3}} + \frac{k^3 T}{m^2} + \frac{k^2 m^{2/3}}{T}\right). \tag{A.280}$$

The statement follows from that 0-1 classification error is bounded by the hinge-loss. □

Theorem A.2.21 (Theorem 13 in [63])**.** *Fix some η, and let $f_1, \ldots, f_T$ be some sequence of convex functions. Fix some θ_1, and assume we update $\theta_{t+1} = \theta_t - \eta \nabla f_t(\theta_t)$. Then for every θ^* the following holds:*

$$\frac{1}{T}\sum_{t=1}^{T} f_t(\theta_t) \leq \frac{1}{T}\sum_{t=1}^{T} f_t(\theta^*) + \frac{1}{2\eta T}\|\theta^*\|_2^2 + \|\theta_1\|_2 \frac{1}{T}\sum_{t=1}^{T} \|\nabla f_t(\theta_t)\|_2 + \eta \frac{1}{T}\sum_{t=1}^{T} \|\nabla f_t(\theta_t)\|_2^2. \tag{A.281}$$

A.3 Lower Bound for Linear Models on Fixed Feature Mappings

Theorem A.3.1 (Restatement of Theorem 2.2.2)**.** *Suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathbf{X} \to [-1,1]^N$. Define for $B > 0$:*

$$\mathcal{H}_B = \{h(\tilde{\mathbf{x}}) : h(\tilde{\mathbf{x}}) = \langle \Psi(\tilde{\mathbf{x}}), w \rangle, \|w\|_2 \leq B\}. \tag{A.282}$$

Then, if $3 < k \leq D/16$ and k is odd, then there exists $\mathcal{D} \in \mathcal{F}_\Xi$ such that all $h \in \mathcal{H}_B$ have hinge-loss at least $p_o \left(1 - \frac{\sqrt{2N}B}{2^k}\right)$.

Proof of Theorem 2.2.2. We first show that $\mathcal{F}_\Xi$ contains some distributions that are essentially sparse parity learning problems, and then we invoke the lower bound result from existing work for such problems.

Consider $\mathcal{D}$ defined as follows.

- Let $P = \{i \in [k] : i \text{ is odd}\}$. That is, if there are odd numbers of 1's in $\tilde{\phi}_{\mathbf{A}}$, then $y = +1$.
- Let $\mathcal{D}^{(0)}_{\tilde{\phi}}$ be a distribution where all entries $\tilde{\phi}_j$ are i.i.d. with $\Pr[\tilde{\phi}_j = 0] = \Pr[\tilde{\phi}_j = 1] = 1/2$. Let $\mathcal{D}^{(0)}$ be the distribution over $(\tilde{\mathbf{x}}, y)$ induced by $\mathcal{D}^{(0)}_{\tilde{\phi}}$ and the above P.
- Let $\mathcal{D}^{(1)}_{\tilde{\phi}}$ be a distribution where all entries $\tilde{\phi}_j$ for $j \notin \mathbf{A}$ are i.i.d. with $\Pr[\tilde{\phi}_j = 1] = p_{\mathrm{o}}/(2 - 2p_{\mathrm{o}})$, while $\Pr[\tilde{\phi}_{\mathbf{A}} = (0, 0, \ldots, 0)] = \Pr[\tilde{\phi}_{\mathbf{A}} = (1, 1, \ldots, 1)] = 1/2$. Let $\mathcal{D}^{(1)}$ be the distribution over $(\tilde{\mathbf{x}}, y)$ induced by $\mathcal{D}^{(1)}_{\tilde{\phi}}$ and the above P.
- Let $\mathcal{D}^{\mathrm{mix}}_{\mathbf{A}} = p_{\mathrm{o}}\mathcal{D}^{(0)} + (1 - p_{\mathrm{o}})\mathcal{D}^{(1)}$.

It can be verified that such distributions are included in $\mathcal{F}_\Xi$ for $\gamma = \Theta(1)$.

Assume for contradiction that for all $\mathcal{D} \in \mathcal{F}_\Xi$, there exists $h^* \in \mathcal{H}_B$ such that $h = \langle \Psi, w^* \rangle$ loss smaller than $p_{\mathrm{o}} \left(1 - \frac{\sqrt{2N}B}{2^k}\right)$. Then for all the distributions $\mathcal{D}^{\mathrm{mix}}_{\mathbf{A}}$ defined above, we

have

$$\mathbb{E}_{\mathcal{D}^{(0)}}[\ell(h^*(\tilde{\mathbf{x}}), y)] < 1 - \frac{\sqrt{2N}B}{2^k}. \tag{A.283}$$

Now let $\mathcal{D}_z$ be a distribution over $z \in \{-1, +1\}^D$ with i.i.d. entries z_j and $\Pr[z_j = -1] = \Pr[z_j = +1] = 1/2$. Let $f_{\mathbf{A}}(z) = \prod_{j \in \mathbf{A}} z_j$ be the k-sparse parity functions. Let $\Psi'(z) = \Psi(\mathbf{M}(z+1)/2)$. Then we have $h'(z) = \langle \Psi'(z), w^* \rangle$ such that for all $\mathbf{A}$,

$$\mathbb{E}_{\mathcal{D}_z}[\ell(h'(z), f_{\mathbf{A}}(z))] < 1 - \frac{\sqrt{2N}B}{2^k}. \tag{A.284}$$

This is contradictory to Theorem A.3.2. □

The following theorem is implicit in the proof in Theorem 1 in [63].

Theorem A.3.2. *For a subset $\mathbf{A} \subseteq [D]$ of size k, let the distribution $\mathcal{D}_{\mathbf{A}}$ over (z, y) defined as follows: z is uniform over $\{\pm 1\}^D$ and $y = \prod_{i \in \mathbf{A}} z_i$. Fix some $\Psi : \{\pm 1\}^D \to [-1, +1]^N$, and define:*

$$\mathcal{H}_\Psi^B = \{z \to \langle \Psi(z), w \rangle : \|w\|_2 \leq B\}.$$

If k is odd and $k \leq D/16$, then there exists some $\mathbf{A}$ such that

$$\min_{h \in \mathcal{H}_\Psi^B} \mathbb{E}_{\mathcal{D}_{\mathbf{A}}}[\ell(h(z), y)] \geq 1 - \frac{\sqrt{2N}B}{2^k}.$$

We now prove the corollary.

Corollary A.3.3 (Restatement of Corollary 2.2.3)**.** *For any function f using a shift-invariant kernel K with RKHS norm bounded by L, or $f(x) = \sum_i \alpha_i K(z_i, x)$ for some data points z_i and $||\alpha||_2 \leq L$. If $3 < k \leq D/16$ and k is odd, then there exists $\mathcal{D} \in \mathcal{F}_\Xi$ such that f have hinge-loss at least $p_o(1 - \frac{\text{poly}(d,L)}{2^k}) - \frac{1}{\text{poly}(d,L)}$.*

Proof. By Claim 1 in [224], for any $\nu > 0$, there exists $N = \text{poly}(d, 1/\nu)$ Fourier features Ψ_j that can approximate the shift-invariant kernel up to error ν. For any $\epsilon > 0$, consider $\sum_i \alpha_i \langle \Psi(z_i), \Phi(x) \rangle = \langle \sum_i \alpha_i \Psi(z_i), \Psi(x) \rangle$. Let $w = \sum_i \alpha_i \Psi(z_i)$ and let $\nu = O(\frac{\epsilon}{L})$, then

$\langle \Psi(x), w \rangle$ approximates $f(x)$ upto error ϵ and $N = \text{poly}(d, L, 1/\epsilon)$ and the norm of w bounded by $B = \text{poly}(d, L, 1/\epsilon)$. The reasoning is the same for f in the RKHS form, replacing sum with integral. By Theorem 2.2.2, $\langle \Psi(x), w \rangle$ has hinge-loss at least $p_0(1 - \frac{\sqrt{2NB}}{2^k})$. Thus, the function f has loss at least $p_0(1 - \frac{\text{poly}(d,L,1/\epsilon)}{2^k}) - \epsilon$. Choose $\epsilon = \frac{1}{\text{poly}(d,L)}$, we get the bound. □

A.4 Lower Bound for Learning without Input Structure

First recall the Statistical Query model [142]. In this model, the learning algorithm can only receive information about the data through statistical queries. A statistical query is specified by some property predicate Q of labeled instances, and a tolerance parameter $\tau \in [0, 1]$. When the algorithm asks a statistical query (Q, τ), it receives a response $\hat{P}_Q \in [P_Q - \tau, P_Q + \tau]$, where $P_Q = \Pr[Q(x, y) \text{ is true}]$. Q is also required to be polynomially computable, i.e., for any (x, y) $Q(x, y)$ can be computed in polynomial time. Notice that a statistical query can be simulated by empirical average of a large random sample of data of size roughly $O(1/\tau^2)$ to assure the tolerance τ with high probability.

[35] introduces the notion of Statistical Query dimension, which is convenient for our purpose.

Definition A.4.1 (Definition 2 in [35])**.** For concept class C and distribution $\mathcal{D}$, the statistical query dimension SQ-DIM$(C, \mathcal{D})$ is the largest number d such that C contains d concepts $c_1, \ldots, c_d$ that are nearly pairwise uncorrelated: specifically, for all $i \neq j$,

$$|\Pr_{x \sim \mathcal{D}}[c_i(x) = c_j(x)] - \Pr_{x \sim \mathcal{D}}[c_i(x) \neq c_j(x)]| \leq 1/d^3. \tag{A.285}$$

Theorem A.4.2 (Theorem 12 in [35])**.** *In order to learn C to error less than $1/2 - 1/d^3$ in the Statistical Query model, where $d =$ SQ-DIM$(C, \mathcal{D})$, either the number of queries or $1/\tau$ must be at least $\frac{1}{2}d^{1/3}$.*

We now use the above tools to prove our lower bound.

Theorem A.4.3 (Restatement of Theorem 2.2.4)**.** *For any algorithm in the Statistical Query model that can learn over $\mathcal{F}_{\Xi_0}$ to error less than $\frac{1}{2} - \frac{1}{\binom{D}{k}^3}$, either the number of queries or $1/\tau$ must be at least $\frac{1}{2}\binom{D}{k}^{1/3}$.*

Proof of Theorem 2.2.4. Consider the following concept class and marginal distribution:

- Let $\mathcal{D}$ be the distribution over $\tilde{\mathbf{x}}$, given by $\tilde{\mathbf{x}} = \mathbf{M}\tilde{\phi}$ and $\tilde{\phi}_j$ are i.i.d. with $\Pr[\tilde{\phi}_j = 0] = \Pr[\tilde{\phi}_j = 1] = 1/2$.
- Let C be the class of functions $y = g_{\mathbf{A}}(\tilde{\phi}) = \mathbb{I}[\sum_j (1 - \tilde{\phi}_j) \text{ is odd}]$ for different $\mathbf{A} \subseteq [D]$.

The distributions over $(\tilde{\mathbf{x}}, y)$ induced by $(C, \mathcal{D})$ are a subset of $\mathcal{F}_{\Xi_0}$. It is then sufficient to show that SQ-DIM$(C, \mathcal{D}) \geq \binom{D}{k}$.

It is easy to see that C are essentially the sparse parity functions: if $z_j = 2\tilde{\phi}_j - 1$, then $g_{\mathbf{A}}(\tilde{\phi}) = \prod_{j \in \mathbf{A}} z_j$. This then implies that the $g_{\mathbf{A}}$'s are uncorrelated, so SQ-DIM$(C, \mathcal{D}) \geq \binom{D}{k}$. □

A.5 Complete Experimental Results

Our experiments mainly focus on feature learning and the effect of the input structure. We first perform simulations on our learning problems to (1) verify our main theorems on the benefit of feature learning and the effect of input structure (2) verify our analysis of feature learning in networks. We then check if our insights carry over to real data: (3) whether similar feature learning is presented in real network/data; (4) whether damaging the input structure lowers the performance. The results are consistent with our analysis and provide positive support for the theory.

The experiments were ran 5 times with different random seeds, and the average results (accuracy) are reported. The standard deviations of the results are smaller than 0.5% and thus we do not present them for clarity. The hardware specifications are 4 Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 16 GB RAM, and one NVIDIA GPU GTX1080.

A.5.1 Simulation

We train a two-layer network following our learning process. We use two fixed feature methods: the NTK [79] and random feature (RF) methods based on the same network and random initialization as the network learning. More precisely, in the NTK method, we randomly initialize the network and take its NTK and learn a classifier on it. In the RF method, we freeze the first layer of the network, and train the second layer (on the random features given by the frozen neurons). The training step number is the same as that in network learning. We also test these three methods on the data distribution with input structure removed (i.e., $\mathcal{F}_{\Xi_0}$ in Theorem 2.2.4). For comparison, we take the representation of our two-layer network at step one/step two, named One Step/Two Step (fix the weight of the 1st layer after the first step/second step to train the weight of the second layer), and train the best classifiers on top of them.

Recall that our analysis is on the *directions* of the weights without considering their *scaling*, and thus it is important to choose cosine similarity rather than the typical ℓ_2 distance. Thus, we use metric Cos Similarity $\max_{\{i\in[2m]\}} \cos(\mathbf{w}_i, \sum_{j\in\mathbf{A}} \mathbf{M}_j)$ in our tables,

and use Multidimensional Scaling to plot the weights distribution. The simulation dataset size is 50000. During training, the batch size is 1000, while for the first two steps we use the approximate full gradient (batch size is 50000). Each step is corresponding to one weights update.

Parity Labeling

Setting. We generate data according to the parity function data distributions used in our proof of the lower bound for fixed features (Theorem 2.2.2), with $d = 500, D = 100, k = 5, p_{\mathrm{o}} = 1/2$, with a randomly sampled $\mathbf{A}$. More precisely, we consider $\mathcal{D}$ defined as follows.

- Let $P = \{i \in [k] : i \text{ is odd}\}$. That is, if there are odd numbers of 1's in $\tilde{\phi}_{\mathbf{A}}$, then $y = +1$.
- Let $\mathcal{D}_{\tilde{\phi}}^{(0)}$ be a distribution where all entries $\tilde{\phi}_j$ are i.i.d. with $\Pr[\tilde{\phi}_j = 0] = \Pr[\tilde{\phi}_j = 1] = 1/2$. Let $\mathcal{D}^{(0)}$ be the distribution over $(\tilde{\mathbf{x}}, y)$ induced by $\mathcal{D}_{\tilde{\phi}}^{(0)}$ and the above P.
- Let $\mathcal{D}_{\tilde{\phi}}^{(1)}$ be a distribution where all entries $\tilde{\phi}_j$ for $j \notin \mathbf{A}$ are i.i.d. with $\Pr[\tilde{\phi}_j = 1] = p_{\mathrm{o}}/(2 - 2p_{\mathrm{o}})$, while $\Pr[\tilde{\phi}_{\mathbf{A}} = (0, 0, \ldots, 0)] = \Pr[\tilde{\phi}_{\mathbf{A}} = (1, 1, \ldots, 1)] = 1/2$. Let $\mathcal{D}^{(1)}$ be the distribution over $(\tilde{\mathbf{x}}, y)$ induced by $\mathcal{D}_{\tilde{\phi}}^{(1)}$ and the above P.
- Let $\mathcal{D}_{\mathbf{A}}^{\mathrm{mix}} = p_{\mathrm{o}}\mathcal{D}^{(0)} + (1 - p_{\mathrm{o}})\mathcal{D}^{(1)}$.

The network and the training follow Section 2.1, where the network size is $m = 300$ and the training time $T = 600$ steps.

Model	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc (%)	100.0	84.0	74.7	51.3	100.0	100.0
Test Acc (%)	100.0	86.4	76.0	52.2	100.0	52.0
Cos Similarity	0.997	NA	0.114	0.848	0.997	0.253

Table A.1: Parity labeling results in six methods. The cosine similarity is computed between the ground-truth $\sum_{j \in \mathbf{A}} \mathbf{M}_j$ and the closest neuron weight.

Verification of the Main Results. Figure A.1 shows that the results are consistent with our analysis. Network learning gets high test accuracy while the two fixed feature

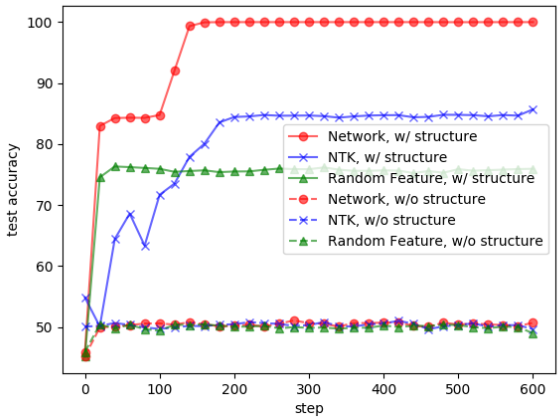


Figure A.1: Test accuracy on simulated data under parity labeling with or without input structure.

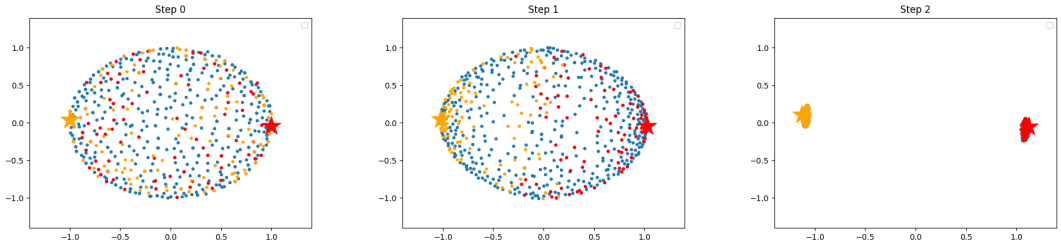


Figure A.2: Visualization of the weights $\mathbf{w}_i$'s after initialization/one gradient step/two gradient steps in network learning under parity labeling. The red star denotes the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$; the orange star is $-\sum_{j\in\mathbf{A}} \mathbf{M}_j$. The red dots are the weights closest to the red star after two steps; the orange ones are for the orange star.

methods get significantly lower accuracy. Furthermore, when the input structure is removed, all three methods get test accuracy similar to random guessing.

Feature Learning in Networks. Figure A.2 shows that the results are as predicted by our analysis. After the first gradient step, some weights begin to cluster around the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$ (or $-\sum_{j\in\mathbf{A}} \mathbf{M}_j$ due to we have $\mathbf{a}_i$ in the gradient update which can be positive or negative). After the second step the weights get improved and well-aligned with the ground-truth (with cosine similarity > 0.99).

Table A.1 shows the results for different methods. Recall that the Cos Similarity metric is $\max_{\{i\in[2m]\}} \cos(\mathbf{w}_i, \sum_{j\in\mathbf{A}} \mathbf{M}_j)$, which reports the cosine value of the closest one. One

Step refers to the method where we take the neurons after one gradient step, freeze their weights, and train a classifier on top; similar for Two Step. One Step gets test accuracy about 52%, while Two Step gets accuracy about 100%. This demonstrates that while some effective feature emerge in the first step, they need to be improved in the second step for accurate prediction. NTK, random feature, One Step all failed, while Network and Two Step can achieve 100% test accuracy. Network w/o structure refers to training the network on data without the input structure. It overfits the training dataset with 52% test accuracy.

Interval Labeling

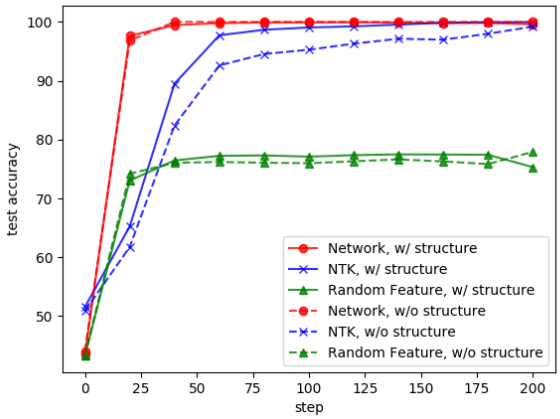


Figure A.3: Test accuracy on simulated data under interval labeling with or without input structure.

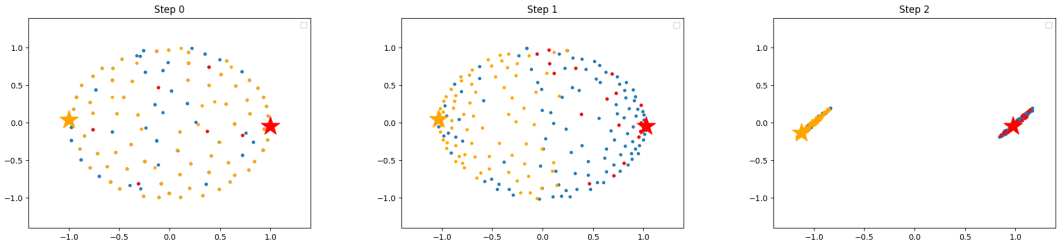


Figure A.4: Visualization of the weights $\mathbf{w}_i$'s after initialization/one gradient step/two gradient steps in network learning under interval labeling. The red star denotes the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$; the orange star is $-\sum_{j\in\mathbf{A}} \mathbf{M}_j$. The red dots are the weights closest to the red star after two steps; the orange ones are for the orange star.

Model	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc (%)	100.0	100.0	76.4	44.1	100.0	100.0
Test Acc (%)	100.0	100.0	73.2	41.0	100.0	100.0
Cos Similarity	1.00	NA	0.153	0.901	0.994	0.965

Table A.2: Interval labeling results in six methods.

Setting. We also tried interval function, where $y = 1$ if $\sum_{i \in \mathbf{A}} \tilde{\phi}_i$ is in the range $[t_1, t_2]$ with $t_1 = 20$ and $t_2 = 30$, otherwise $y = -1$. We use $d = 500, D = 100, k = 30$. The $\tilde{\phi}_i$'s are independent, and $\Pr[\tilde{\phi}_i = 1] = 2/3$ for any $i \in A$, and $\Pr[\tilde{\phi}_i = 1] = 1/2$ otherwise. When the input structure is removed, we set $\Pr[\tilde{\phi}_i = 1] = 1/2$ for all i's.

The network and training again follows Section 2.1 with a network size $m = 100$ and the training time $T = 200$ steps.

Verification of the Main Results. Figure A.3 shows that network learning learns the fastest, NTK learns slower but reaches similar test accuracy, while random feature can only reach a decent but lower accuracy. This is because for such simpler labeling functions, fixed feature methods can still achieve good performance (note that the lower bound does not hold for such a case), while the performance depends on what fixed features to use.

Furthermore, when the input structure is removed, the methods still get similar (or only slightly worse) performance as with input structure. This shows that when the labeling function is simple, the help of the input structure for learning may not be needed. In the experiments on real data, we will show that when the input structure is changed, it indeed leads to lower performance which suggests that the labeling function in practice is typically more complicated than this interval labeling setting, and the help of the input structure is significant for learning.

Feature Learning in Networks. Figure A.4 shows the phenomenon of feature learning similar to that in the parity labeling setting. Table A.2 shows the test accuracy of six different methods. Random feature and One Step failed, while Network, NTK and Two Step succeed showing that interval labeling setting is a simpler case than parity labeling

setting.

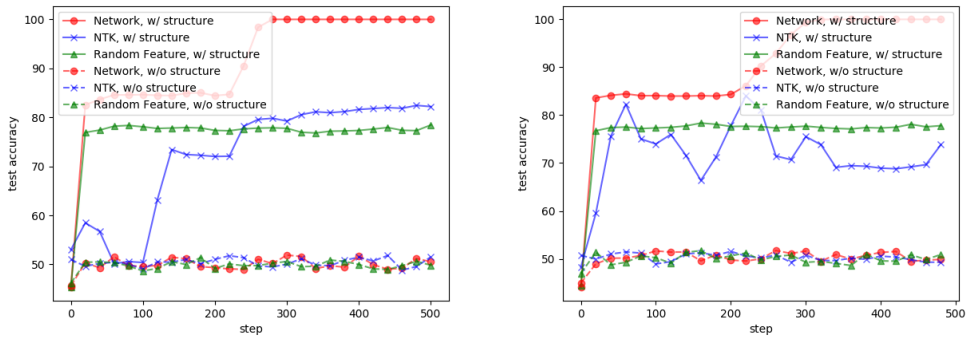
A.5.2 More Simulation Result in Various Settings

We show the robustness of our simulation results by studying the learning behaviors in a variety of settings including different sample size, input data dimension and class imbalance. We reuse the same setting as the simulation in the main text (details in A.5.1), vary different parameters, and report the accuracy, the cosine similarities between the learned weights, and the visualization of the neuron weights.

Varying Input Data Dimension

In the simulation experiments in the main text, the input data dimension d is 500. Here we change the input data dimension to 100 and 2000. All other configurations follow A.5.1.

Verification of the Main Results. Figure A.5 shows that our claim is robust under different input data dimensions. The performance of network learning is superior over NTK and random feature approaches on inputs with structure, and on inputs without structure, all three methods fail.



(a) $d = 100$ (b) $d = 2000$

Figure A.5: Test accuracy on simulated data under different input data dimensions.

Feature Learning in Networks. Figure A.6 visualizes the neuron weights. It shows similar results to that in A.5.1: the weights gets updated to to the effective feature in the

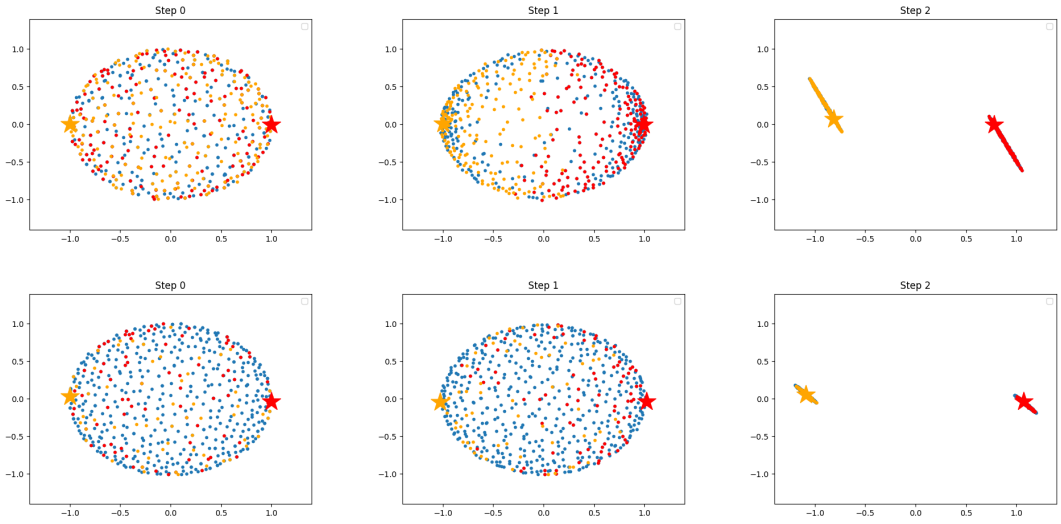


Figure A.6: Visualization of the weights $\mathbf{w}_i$'s in early steps under different input data dimensions. Upper row: input data dimension $d = 100$; lower row: $d = 2000$.

$d = 100$	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	83.1	78.9	53.0	100.0	100.0
Test Acc	100.0	81.5	78.3	51.1	100.0	51.0
Cos Similarity	1.000	NA	0.354	0.967	1.000	0.331
$d = 2000$	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	75.6	80.0	50.22	100.0	100.0
Test Acc	100.0	75.4	77.0	50.01	100.0	52.5
Cos Similarity	0.998	NA	0.056	0.560	0.998	0.309

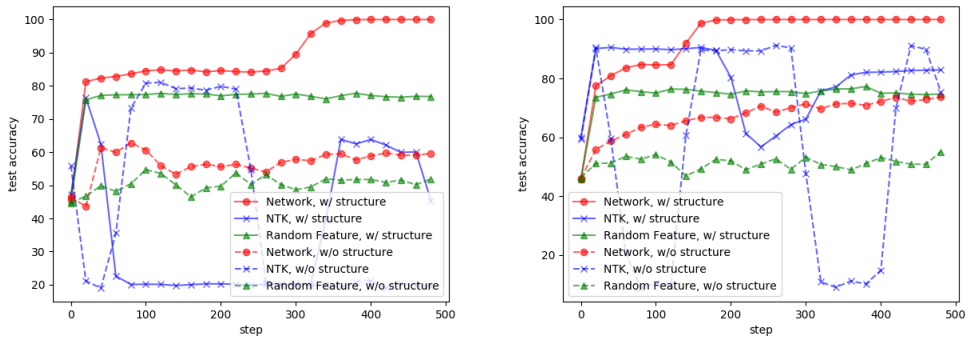
Table A.3: Results of six methods for different input data dimensions. The cosine similarity is computed between the ground-truth $\sum_{j\in\mathbf{A}} \mathbf{M}_j$ and the closest neuron weight.

first two steps, forming clusters. Table A.3 shows some quantitative results. In particular, the average cosine similarities between neuron weights and the effective features after two steps are close to 1, showing that they match the effective features.

Varying Class Imbalance Ratio

The experiments in the main text has 25000 training samples for each class. Here we keep the total sample size 50000 but use different class imbalance ratios, which is the class -1 sample size divide by the total sample size.

Verification of the Main Results. Figure A.7 shows that our claim is robust under different class imbalance ratios. The results are similar to those for balanced classes, except that NTK becomes less stable.



(a) Negative class ratio = 0.8 (b) Negative class ratio = 0.9

Figure A.7: Test accuracy on simulated data under different negative class ratios.

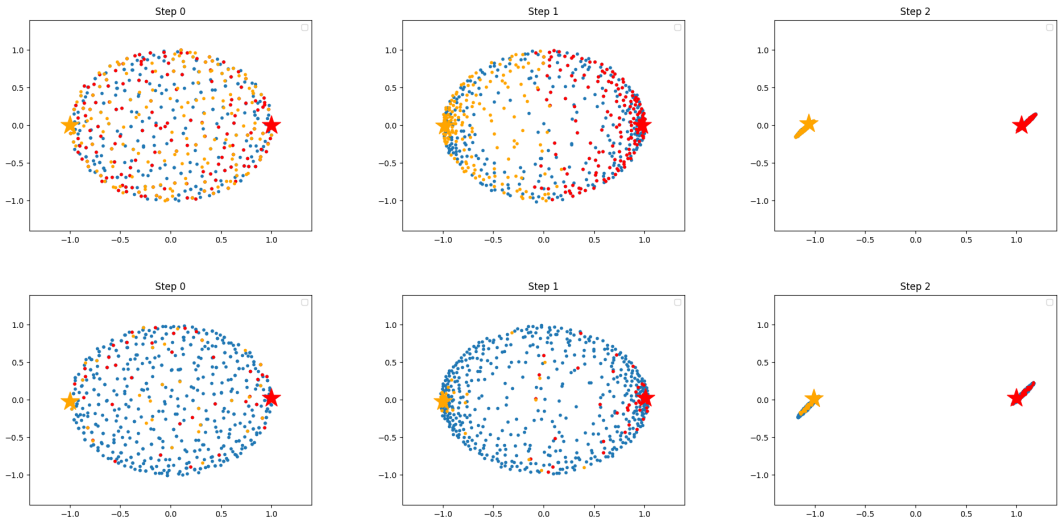


Figure A.8: Visualization of the weights $\mathbf{w}_i$'s in early steps under different class imbalance ratios. Upper row: negative class ratio 0.8; lower row: 0.9.

Feature Learning in Networks. Figure A.8 visualizes the neurons' weights. Again, the observation is similar to that for balanced classes. Table A.4 shows some quantitative results which are also similar to those for balanced classes.

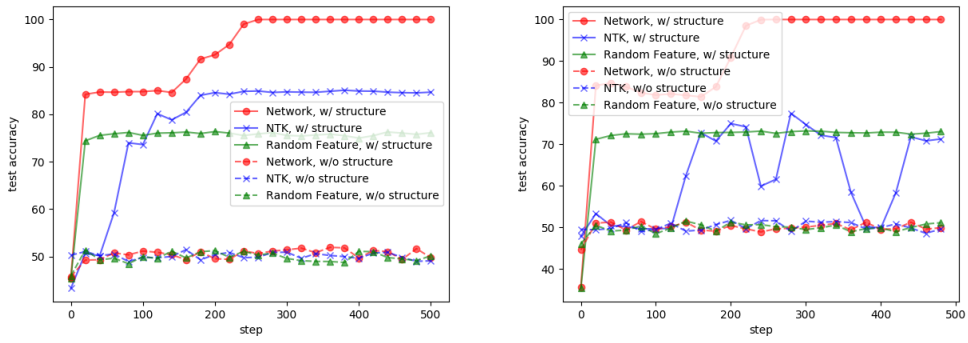
ratio = 0.8	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	62.9	72.7	78.3	100.0	100.0
Test Acc	100.0	82.7	70.4	75.7	100.0	61.7
Cos Similarity	0.999	NA	0.293	0.950	0.999	0.218
ratio = 0.9	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	84.0	73.6	92.3	100.0	100.0
Test Acc	100.0	81.7	72.4	89.2	100.0	71.8
Cos Similarity	0.997	NA	0.296	0.956	0.997	0.286

Table A.4: Results of six methods under different negative class ratios.

Varying Sample Size

Here we change the sample size 50000 in Section A.5.1 to be 25000 and 10000. For sample size 25000, we observe similar results. For sample size 10000, we observe over-fitting (test accuracy much lower than train accuracy). Therefore, for sample size 10000 we reduces the size of the network (i.e., number of hidden neurons) from $m = 300$ to $m = 50$.

Verification of the Main Results. Figure A.9 shows that our claim is robust under different sample sizes. In particular, the network learning still outperforms the NTK and random feature approaches on structured inputs.



(a) $n = 25000$ (b) $n = 10000$

Figure A.9: Test accuracy on simulated data under different sample sizes n.

Feature Learning in Networks. Figure A.10 and Table A.5 show that the phenomenon of feature learning for different samples is similar to that in A.5.1.

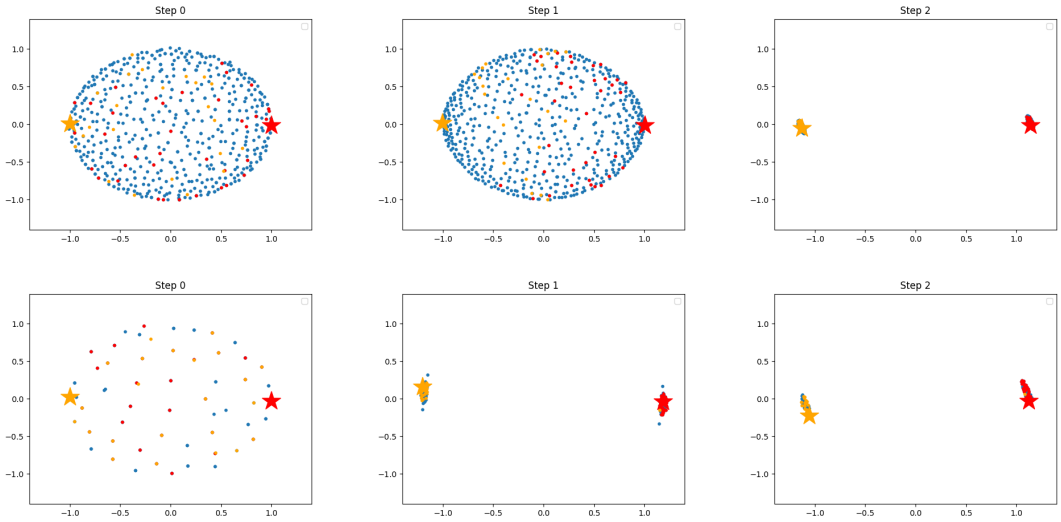


Figure A.10: Visualization of the weights $\mathbf{w}_i$'s in early steps under different sample sizes. Upper row: sample size 25000; lower row: 10000.

$n = 25000$	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	84.0	78.6	50.6	100.0	100
Test Acc	100.0	84.1	74.7	50.0	100.0	50.2
Cos Similarity	0.997	NA	0.105	0.851	0.997	0.230
$n = 10000$	Network	NTK	RF	One Step	Two Step	Network w/o structure
Train Acc	100.0	73.9	71.6	50.7	100.0	100.0
Test Acc	100.0	75.0	74.3	50.3	100.0	52.2
Cos Similarity	0.995	NA	0.096	0.974	0.994	0.176

Table A.5: Results of six methods for different sample size.

A.5.3 Experiments on More Data Generation Models

In this section we consider some additional data distributions and run the simulation experiments, in particular, focusing on the feature learning phenomenon. Note that our analysis is for the setting where the input distributions have structure revealing some information about the labeling function. (More precisely, the labeling function is specified by $\mathbf{A}$ and P, while the input distribution also depends on them.) We therefore consider two other data generation mechanisms where the labeling function also has connections to the input distributions.

Hidden Representation Labeling

Here we consider the following data model: first uniformly at random select $\tilde{\phi}_{\mathbf{A}}$ from a set of binary vectors, and assign label 1 to some and -1 to others; sample irrelevant patterns $\tilde{\phi}_{-\mathbf{A}}$ uniformly at random; generate the input $\mathbf{x} = \mathbf{M}\tilde{\phi}$. We randomly select 50 binary vectors for each label, with $d = 500, D = 250, k = 50, p_{\mathrm{o}} = 1/2$.

This is a generalization of the distribution $\mathcal{D}^{(1)}$, a component in the distribution of our simulation experiments (see the proof of Theorem 2.2.2 for details). Recall the definition of $\mathcal{D}^{(1)}$: $\tilde{\phi}_A$ is uniform on only two values $[+1, \ldots, +1]$ and $[0, \ldots, 0]$, and uniform over irrelevant patterns; the value $[+1, \ldots, +1]$ corresponds to one class and $[0, \ldots, 0]$ correspond to another class. Our data model here generalizes $\mathcal{D}^{(1)}$ to more than 2 values.

The visualization is shown in Figure A.11. We can observe similar feature learning phenomena, and the neuron weights are updated to form clusters.

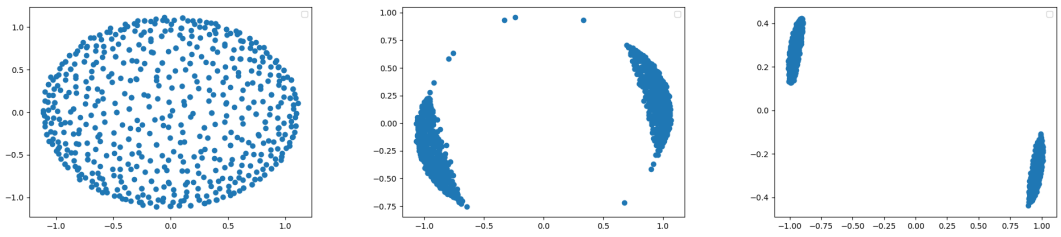


Figure A.11: Visualization of the weights $\mathbf{w}_i$'s after initialization/one gradient step/two gradient steps in network learning under hidden representation labeling.

Two-layer Networks on Mixture of Gaussians

To further support our intuition of feature learning, we run experiments on mixture of Gaussians.

Data. Let $\mathbf{X} = \mathbb{R}^d$ be the input space, and $\mathcal{Y} = \{\pm 1\}$ be the label space. Suppose $\mathbf{M} \in \mathbb{R}^{d \times k}$ is an dictionary with k orthonormal columns. Let $\varepsilon_i, i = 1, \ldots, k$ be i.i.d symmetric Bernoulli random variables, and $g \sim \mathcal{N}(0, \sigma_r^2 \frac{k}{d} \mathbb{I}_d)$. Then we generate the input

x and class label y by:

$$\mathbf{x} = \sum_{i=1}^{k} \varepsilon_i \mathbf{M}_{:i} + g, \quad y = \prod_{i=1}^{k} \varepsilon_i \tag{A.286}$$

In this case, 2^k Gaussian clusters will be created. The centers of the Gaussian clusters $\sum_{i=1}^{k} \pm \mathbf{M}_{:i}$ lie on the vertices of a hyper cube, and the label of each Gaussian cluster is determined by the parity function on the vertices of the hyper cube.

Note that the labeling function is roughly equivalent to a network: $y = \sum_{i=1}^{n} a_i \mathrm{ReLU}(\langle c_i, \mathbf{x} \rangle)$ where c_i's are the Gaussian centers, and $a_i \propto 1$ for Gaussian components with label 1 and $a_i \propto -1$ for those with label -1.

Setting. We then train a two-layer network with $m = 800$ hidden neurons on data sets generated as above with different chosen k's and d's. The training follows typical practice (not the hyperparameters in our analysis). In this setting, we expect the neural network to learn the effective features: the directions of Gaussian cluster centers.

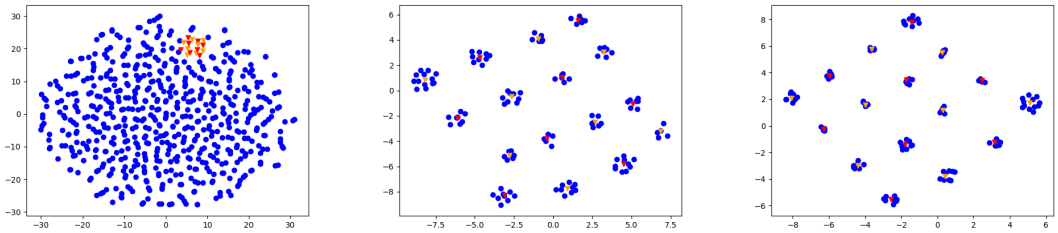
Result. We run experiments with different settings. The parameters are shown in Table A.6. From Figure A.12 we can see that some neurons learn the directions of Gaussian centers, and each Gaussian center is covered by some neurons, which matches our expectation.

Parameters	d	k	Number of Clusters	σ_r
Experiment 1	100	4	16	1
Experiment 2	25	4	16	0.7
Experiment 3	100	5	32	1

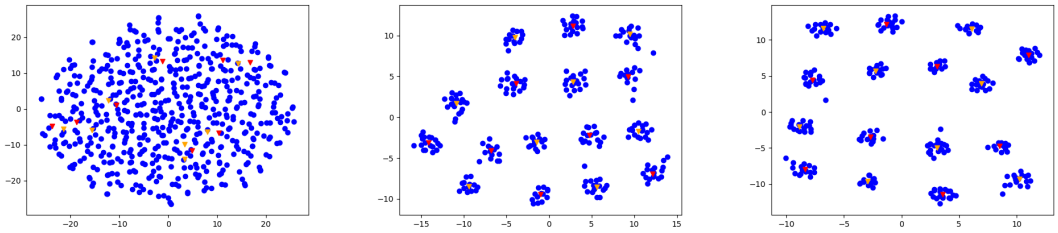
Table A.6: Gaussian mixture setting.

A.5.4 Real Data: Feature Learning in Networks

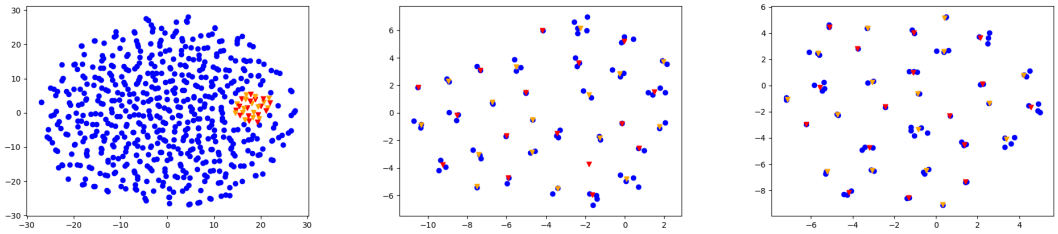
We take the subset of MNIST [68] with labels 0/1, CIFAR10 [147] with labels airplane/automobile and SVHN [198] with labels 0/1, and train a two-layer network with $m = 50$. We use traditional weight initialization method (random Gaussian) and training



(a) Experiment 1 with epoch 0/50/80



(b) Experiment 2 with epoch 0/30/50



(c) Experiment 3 with epoch 0/50/80

Figure A.12: Visualization of the weights w_i's (blue dots) and Gaussian centers (red for positive labeled clusters and orange for negative labeled clusters).

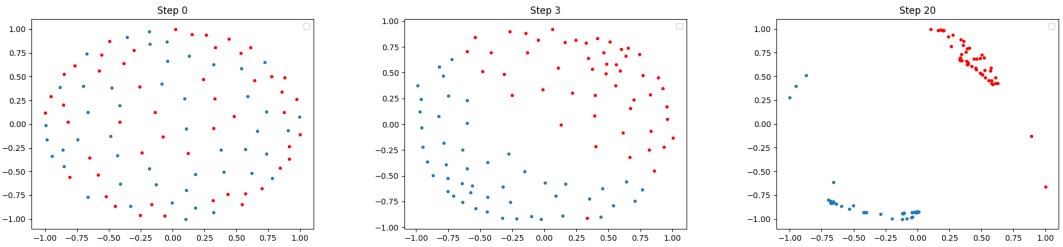


Figure A.13: Visualization of the neurons' weights in a two-layer network trained on the subset of MNIST data with label 0/1. The weights gradually form two clusters.

method (SGD with momentum = 0.95 without regularization) in this section, for our purpose of investigating the training dynamics in practice.

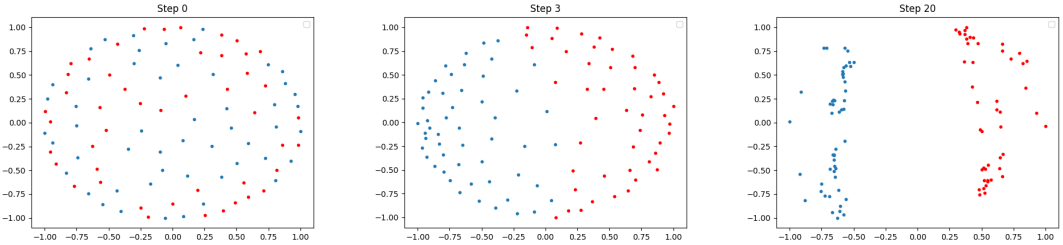


Figure A.14: Visualization of the neurons' weights in a two-layer network trained on the subset of CIFAR10 data with label airplane/automobile. The weights gradually form two clusters.

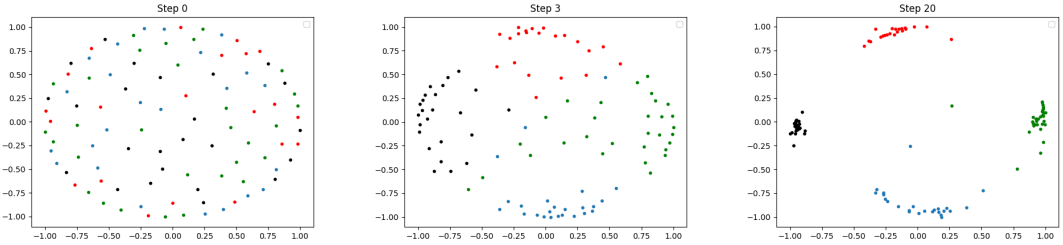


Figure A.15: Visualization of the neurons' weights in a two-layer network trained on the subset of SVHN data with label 0/1. The weights gradually form four clusters.

Then we visualize the neurons' weights following the same method in the simulation. Figure A.13, Figure A.14 and Figure A.15 show a similar feature learning phenomenon: effective features emerge after a few steps, and then get improved to form clusters. This shows the insights obtained on our learning problems are also applicable to the real data.

CNNs on Binary Cifar10: Feature Learning in Networks

Setting. We use ResNet(m), which is a ResNet-18 convolutional neural network [120] with m filters in the first residual block. It is obtained by scaling the number of filters in each block proportionally from the standard ResNet-18 network which is ResNet(64). We use ResNet(128) and ResNet(256) in this experiment. We train our model on Binary CIFAR10 [147] with labels airplane/automobile for 20 epochs. The final test accuracy of ResNet(128) is 95.75% and that of ResNet(256) is 93.8%.

	$\cos(v_1, \bar{v})$	$\cos(v_2, \bar{v})$	$\cos(v_3, \bar{v})$	$\cos(v_1, v_2)$	$\cos(v_1, v_3)$	$\cos(v_2, v_3)$
ResNet(128)	0.9727	0.8655	0.6549	0.7454	0.5083	0.6533
ResNet(256)	0.8646	0.9665	0.9121	0.7087	0.6919	0.9135

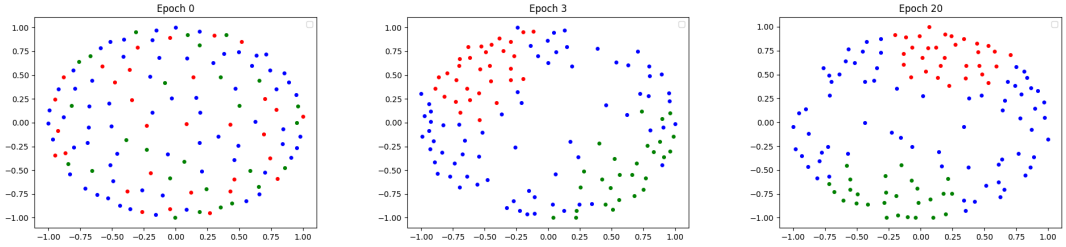
Table A.7: Cosine similarities between the gradients in the early steps. We choose the neuron weight closest to the average weight of the green cluster at the end of the training (in Figure A.16 for ResNet(128) and Figure A.17 for ResNet(256)). We record the gradients of the first 30 steps and divide them to three trunks of 10 steps evenly and sequentially. For the three trunks, we get the average gradients v_1, v_2, v_3. We calculate their cosine similarities to their average $\bar{v} = (v_1 + v_2 + v_3)/3$ and those between them.

Results. Figure A.16 visualizes the filters' weights of different residual blocks in ResNet(128) at Epoch 0, 3, and 20, and Figure A.17 shows those in ResNet(256). They show that feature learning happens in the early stage, and show that there are some clusters of weights (e.g., the red and green points). These colored points are selected at Epoch 20. We first visualize the weights at Epoch 20, and then hand pick the points that roughly form two clusters (i.e., the points in the same cluster are close to each other while those in different clusters are far away). We assign red and green colors to the two clusters at Epoch 20, and then assign these weights with the same color in Epoch 0 and 3. Finally, we compute the cosine similarities and show that the hand picked points are indeed roughly clusters in the high-dimension.

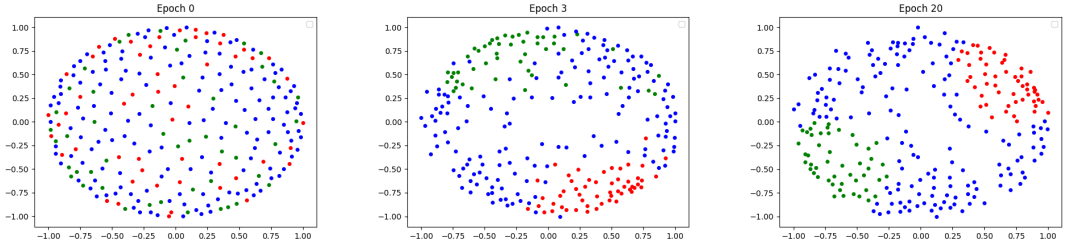
In particular, we have the following three observations.

First, we can see that the filter weights change significantly during the early stage of the training, indicating feature learning happens in the early stage: the change between Epoch 0 and Epoch 3 is much more significant than that between Epoch 3 and Epoch 20.

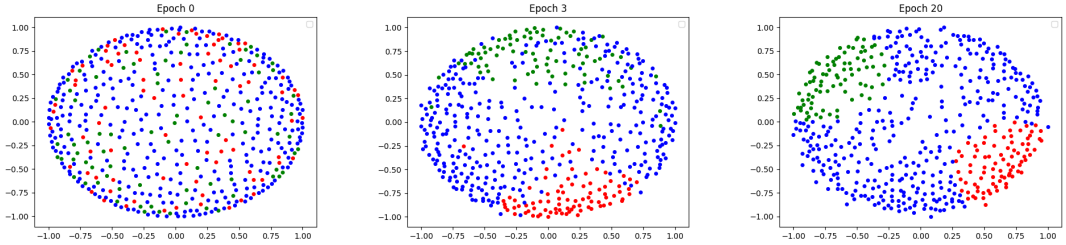
Second, we can also verify that the feature learning is guided by the gradients: the gradients of a filter in the early gradient steps point to similar directions (and thus the updated filter will learn this direction). More precisely, for a selected filter, we average the gradients every 10 gradient steps (so to reduce the variance due to mini-batch), and get v_1, v_2 and v_3 for the first 30 steps and compute their cosine similarities and those to their average. Table A.7 shows the results. In general the similarities are high indicating they point to similar directions. (Note that a similarity of 0.6 is regarded as very significant as



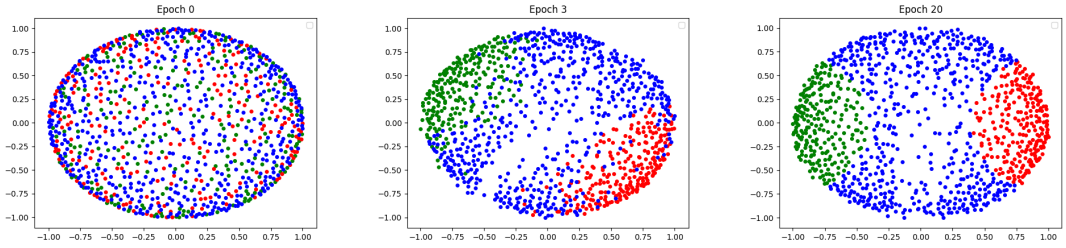
(a) Residual block 1: (Green: 0.6152, Red: 0.6973, Two Centers: -0.7245)



(b) Residual block 2: (Green: 0.5528, Red: 0.6000, Two Centers: -0.7509)

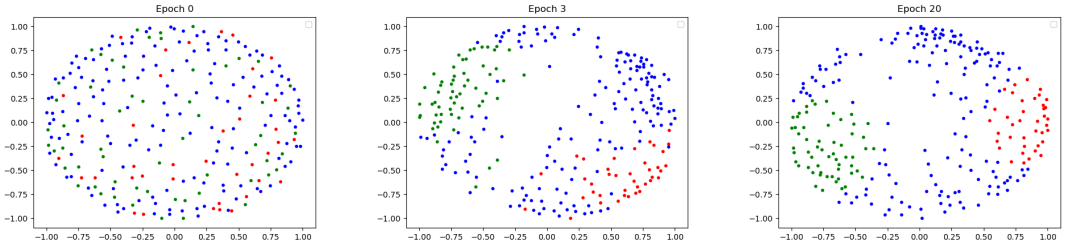


(c) Residual block 3: (Green: 0.4260, Red: 0.5006, Two Centers: -0.5099)

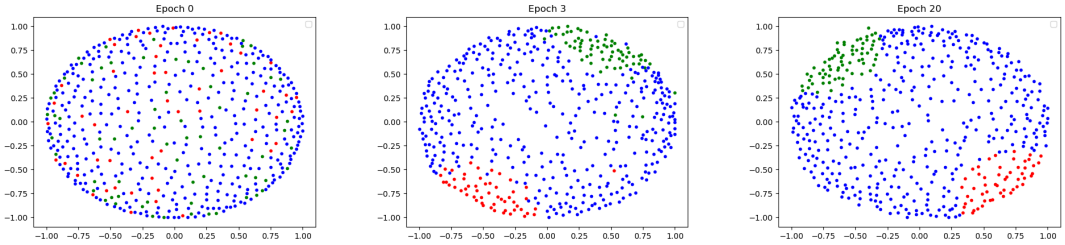


(d) Residual block 4: (Green: 0.5584, Red: 0.5697, Two Centers: -0.9074)

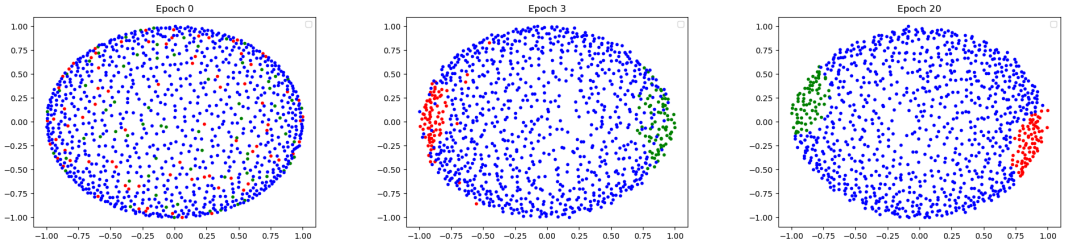
Figure A.16: Visualization of the normalized convolution weights in all Residual block of ResNet(128) trained on the subset of CIFAR10 data with labels airplane/automobile. We show the weights after 0/3/20 epochs in network learning. The weights gradually form two clusters in all Residual blocks. We also report average cosine similarity between the green/red points in the clusters to their centers and cosine similarity between two cluster centers as (Green, Red, Two Centers).



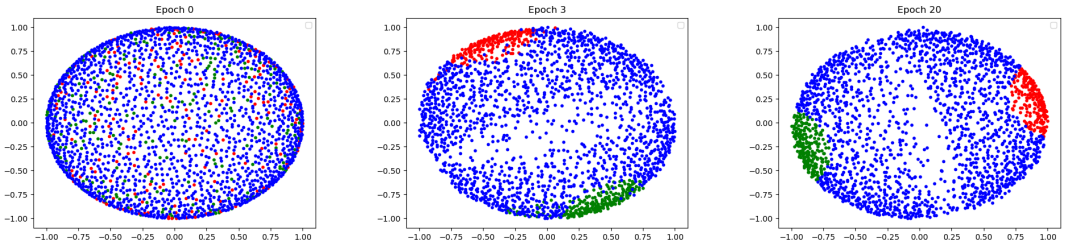
(a) Residual block 1: (Green: 0.7065, Red: 0.6551, Two Centers: -0.7875)



(b) Residual block 2: (Green: 0.6599, Red: 0.5299, Two Centers: -0.9004)



(c) Residual block 3: (Green: 0.5193, Red: 0.6267, Two Centers: -0.9258)



(d) Residual block 4: (Green: 0.7386, Red: 0.6839, Two Centers: -0.9740)

Figure A.17: Visualization of the normalized convolution weights in all Residual block of ResNet(256) trained on the subset of CIFAR10 data with labels airplane/automobile. We show the weights after 0/3/20 epochs in network learning. The weights gradually form two clusters in all Residual blocks. We also report average cosine similarity between the green/red points in the clusters to their centers and cosine similarity between two cluster centers as (Green, Red, Two Centers).

the filters are in a high dimension of $3 \times 3 \times 1024 = 9216$).

Third, we also observe some clustering effect of the filter weights, though not as significant as in our simulations. For example, in the red and green clusters in Figure A.16(a) for the first residual block, the average cosine similarity for filter weights in the red cluster is about 0.62 and that for the green is about 0.7, while the cosine similarity between the two clusters' centers is about -0.72. This shows significant similarities within the cluster while difference between clusters.

Note that the clustering is less significant than our simulation experiments. This is because practical data have more patterns (i.e., effective feature directions) to be learned than our synthetic data, and also the practical network is not as overparameterized as in our simulation. Then filters are likely to learn different patterns (or their mixtures) without forming significant clusters. The results of ResNet(256) show more significant clustering than ResNet(128), which supports our explanation. On the other hand, we emphasize that the key insight of our analysis is that the gradient guides the learning of effective features in the early stage of training (rather than the clustering), which is verified as discussed above.

A.5.5 Real Data: The Effect of Input Structure

To study the influence of the input structure, we propose to keep the labeling function unchanged, vary the input distributions, and exam the change of the loss surface and the training dynamics. We first describe the detailed experimental methodology, which allows us to generate data with similar labeling function but different input distributions. Then we perform experiments on the generated datasets to investigate the change of the learning due to the change in the input distributions, and present the experimental results. Finally, we also perform experiments to verify the intuition behind our experimental method.

Experimental Methodology

We consider the following experimental method. Given an original dataset $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^n$ (e.g., CIFAR10) and an unlabeled dataset $\mathcal{U} = \{\tilde{x}_i\}_{i=1}^m$ from a proposed distribution $P_{\mathcal{U}}$

(e.g., Gaussians), first extend the labeling function of $\mathcal{L}$ to $\mathcal{U}$, giving synthetic labels $\tilde{y}_i$ to $\tilde{x}_i$. Then train a neural network on the union of $\mathcal{L}$ and the synthetic data $\mathcal{L}_{\mathcal{U}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m}$. By investigating the new training dynamics, in particular the difference on the original part $\mathcal{L}$ and the synthetic part $\mathcal{L}_{\mathcal{U}}$, we can see the effect of the input structure. The original dataset should be from real-world data, since one of our goals is to compare them with synthetic data, and identify the properties of real-world data important for the success of learning.

A natural idea is to first learn a powerful network $f(x)$ (called the teacher) on $\mathcal{L}$ to approximate the true labeling function, then apply f on $\mathcal{U}$ to generate synthetic labels, and finally train another network (called the student) on the synthetic data and original data. However, we found that naïvely implementation of this idea fails miserably: the support of $\mathcal{L}$ and $\mathcal{U}$ can be typically different, and the powerful network learned over $\mathcal{L}$ can have entirely different behavior on $\mathcal{U}$. Therefore, we need to control the size of the teacher f so that the labeling on $\mathcal{U}$ has similar complexity as that on $\mathcal{L}$. For our purpose, we can define the complexity of the labeling on $\mathcal{L}$ as the minimum size of the teacher achieving an approximation error ϵ for a chosen ϵ, if the ground-truth data distribution of $\mathcal{L}$ is known. However, given only limited data, we cannot faithfully estimate the needed size of the teacher, and need to take into account the variance introduced by the finite data.

Our key idea is to use the U-shaped curve of the bias-variance trade-off and select the size of the teacher at the minimum of the U-shaped curve. Since recent works [29, 196] show that neural networks can have a double descent curve for the error v.s. model complexity, we thus plot the double descent curve, and find the minimum in the classical regime (corresponding to the traditional U-shape curve).

Our method is designed based on the following two reasons. First, on the U-shaped curve, the complexity of the network is still roughly controlled by that of the number of parameters. The local minimum of the U-shaped curve is a good measurement of the complexity of the data. If the ground-truth is much more complicated than the teacher, then increasing the teacher's size leads to a significant decrease in the approximation error

(bias) compared to a small increase in the variance, that is, we will be on the left-hand side of the U-shaped. In contrast, on the right-hand side of the U-shaped, increasing the teacher's size leads to a small decrease in the bias compared to a significant increase in the variance. That is, the complexity of the ground-truth is comparable to or lower than the teacher. So the local minimum approximates the complexity of the ground-truth labeling function.

Second, the local minimum point is chosen to get the best approximation of the true labels. This helps to maintain the labeling from the real-world data and thus helps our investigation on the input, since too drastic change in the labeling can affect the training.

We note that the method is not perfect. First, the teacher at the local minimum of U-shape may not have very high accuracy, especially on more complicated data. To alleviate this, we also use the teacher to give synthetic labels y_i' to x_i in $\mathcal{L}$, and train the student network on $\mathcal{L}' = \{(x_i, y_i')\}_{i=1}^n$. Though this introduces some differences from the original labels, it is acceptable for our purpose of studying the inputs. Furthermore, ensuring the consistency of the labels on the original input in $\mathcal{L}$ and $\mathcal{U}$ is important in our experiments. Second, the measurement is an approximation due to variance. Since only limited labeled data is available, it's important and necessary to calibrate the measurement w.r.t. the level of variance on the given dataset.

Method Description. Algorithm 3 presents the details. For a fixed network architecture for the teacher f, it first varies the network size and plots the double descent curve. Then it selects the local minimum in the classic regime of U-shape and trains the teacher with the corresponding size. In practice, we observed that the teacher might have unbalanced probabilities for different classes on $\mathcal{U}$ if its training does not take into account $\mathcal{U}$. Therefore, we propose the following heuristic regularization using $x \in \mathcal{U}$, where λ is a regularization

weight, and $f(x)$ is the probabilities over classes given by the teacher:

$$R(x) = R_1(x) + \lambda R_2(x) \tag{A.287}$$

$$R_1(x) = \sum_j \left(\frac{\sum_i f(x)_j}{m} \ln \frac{\sum_i f(x)_j}{m} \right) \tag{A.288}$$

$$R_2(x) = -\frac{1}{m} \sum_i \sum_j (f(x)_j \ln(f(x)_j)). \tag{A.289}$$

Here, $R_1(x)$ guarantees that each kind of label has the same average probability to be generated, and $R_2(x)$ pushes the probability away from uniform to avoid the case that the class probabilities for each data point converge to uniform.

Algorithm 3 Learning the teacher network to generate synthetic labels for studying the effect of the input structure

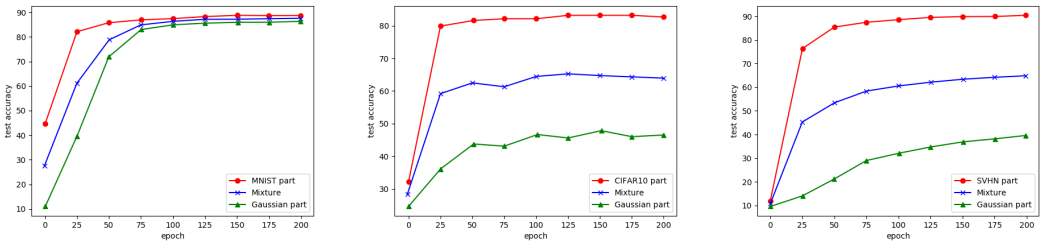
Input: teacher architecture f, labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^n$, unlabeled dataset $\mathcal{U} = \{\tilde{x}_i\}_{i=1}^m$.
Let i to be the size of f, f_i to be the teacher of size i.
for $i = 1$ to n **do**
  Train f_i on $\mathcal{L}$ and let l_i denote the test loss
end for
Plot l_i v.s. i, identify the classical regime, and the size i_t corresponding to the local minimum in classical regime.
Train f_{i_t} on $\mathcal{L}$ with a regularizer $R(x)$ on $\mathcal{U}$ defined in (A.287).
Output: f_{i_t}

Experimental Results

Network models. Here we use one-hidden-layer fully-connected networks with m hidden units and quadratic activation functions. The network is denoted as FC(m). We use ResNet(m), which is a ResNet-18 convolutional neural network [120] with m filters in the first residual block. It is obtained by scaling the number of filters in each block proportionally from the standard ResNet-18 network which is ResNet(64).

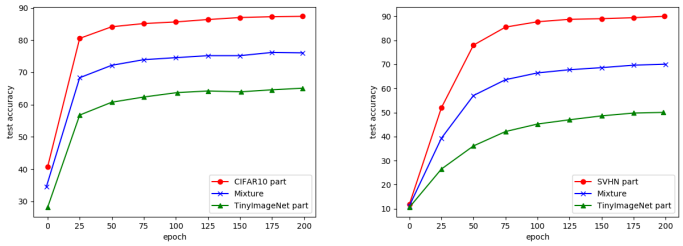
Datasets. We use MNIST [68], CIFAR10 [147] and SVHN [198] as $\mathcal{L}$, and use Gaussian and images in Tiny ImageNet [150] as $\mathcal{U}$. We generate the mixture data, where the fraction of the unlabeled data is denoted as α.

Setup. We first use Algorithm 3 on the labeled data $\mathcal{L}$ and the unlabeled data $\mathcal{U}$ to get a synthetic labeling function (the teacher network) and then use it to give synthetic labels on a mixture of inputs from $\mathcal{L}$ and $\mathcal{U}$. For MNIST, the teacher network learned is FC(9), where the number of the hidden units is determined by Algorithm 3. See empirical verification in Figure A.22. For CIFAR10 and SVHN, the teacher networks are ResNet(5) and ResNet(2), respectively, as determined by our method. The student network for MNIST is FC(9), and those for CIFAR10 and SVHN are ResNet(9) and ResNet(8), respectively. Finally, we train the student networks on these new datasets with perturbed input distributions.



(a) (b) (c)

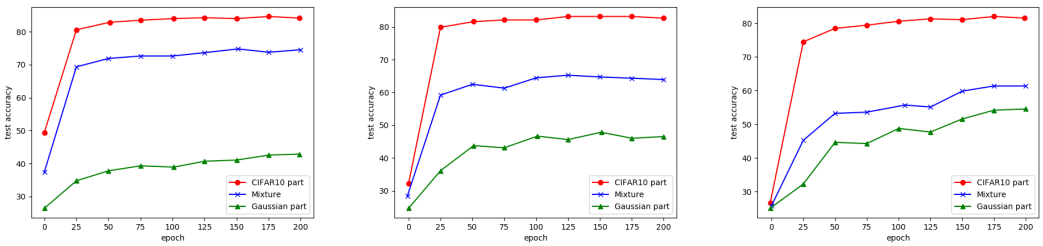
Figure A.18: Test accuracy at different steps for an equal mixture $\alpha = 0.5$ of Gaussian inputs with data: (a) MNIST, (b) CIFAR10, (c) SVHN.



(a) (b)

Figure A.19: Test accuracy at different steps for an equal mixture $\alpha = 0.5$ of Tiny ImageNet inputs with data: (a) CIFAR10, (b) SVHN.

Figure A.18 shows the results on an equal mixture of data and Gaussian. It presents the test accuracy of the student on the original data part, the Gaussian part, and the whole mixture. For example, for CIFAR10, the test accuracy on the whole mixture is lower than that of training on the original CIFAR10, showing that the input structure indeed



(a) $\alpha = 0.25$ (b) $\alpha = 0.50$ (c) $\alpha = 0.75$

Figure A.20: Test accuracy at different steps for varying mixture α of Gaussian inputs with CIFAR10.

has a significant impact on the learning. Furthermore, the network learns well over the CIFAR10 part (with accuracy similar to that on the original data) but learns slower with worse accuracy on the Gaussian part. This suggests that the CIFAR10 input structure is still helping the network to learn effective features. While the results on MNIST+Gaussian do not show a significant trend (possibly because the tasks there are simpler), the results on SVHN+Gaussian show similar significant trends as CIFAR10+Gaussian.

Figure A.20 shows the results when we vary the fraction of the Gaussian data α. We observe that the test accuracy curve on the original part and that on the synthetic part have roughly the same trend for different α as before, further verifying our insights.

Figure A.19 shows the results when mixed with Tiny ImageNet data instead of Gaussians. It shows a similar trend, while the performance on the Tiny ImageNet part is higher than that on the Gaussian part. This suggests that compared to Gaussians, the Tiny ImageNet data has helpful input structures, though not as helpful as that on the original data for learning the particular labeling.

Larger Network on MNIST for Checking The Effect of Input Structure

Here we perform the experiment on MNIST as in A.5.5, but for a network with $m = 50$ hidden neurons rather than $m = 9$. Figure A.21 shows similar results as those for $m = 9$: the learning on the MNIST input part is faster and better than that on the Gaussian input part. The separation between the two is actually more significant than that for $m = 9$.

This then also supports our insight about the effect of input structures.

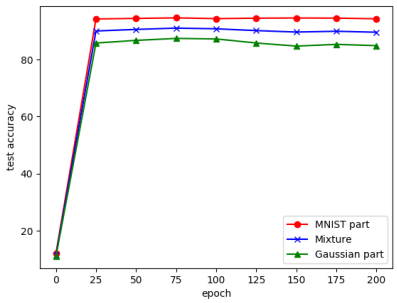
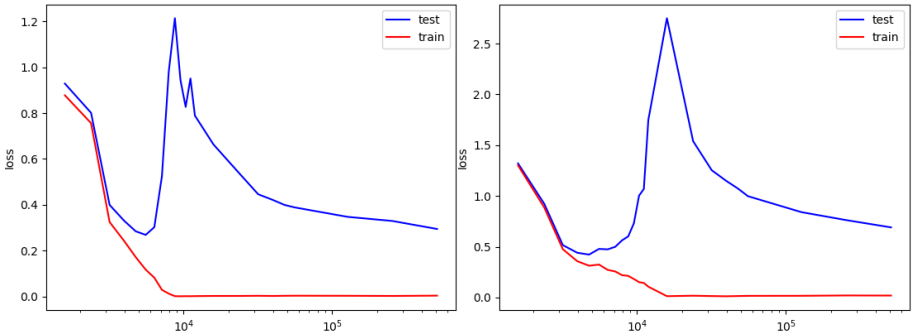
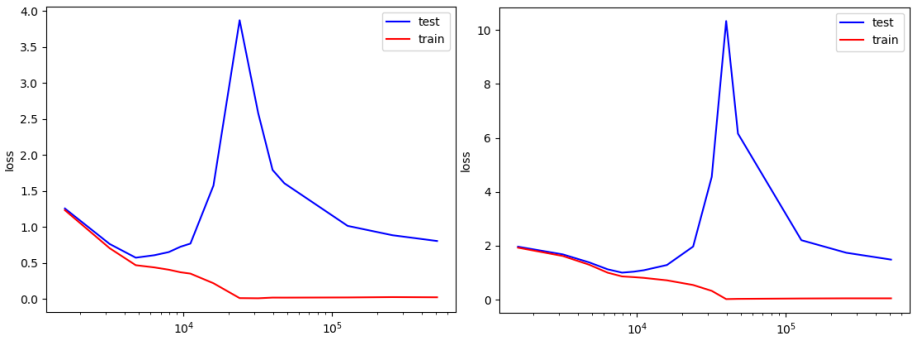


Figure A.21: Test accuracy at different steps for an equal mixture $\alpha = 0.5$ of Gaussian inputs with MNIST, where $m = 50$.

Empirical Verification of Our Method



(a) Teacher's double descent curve (b) Student's curve when Teacher=FC(9)



(c) Student's curve when Teacher=FC(50) (d) Student's curve when Teacher=FC(500)

Figure A.22: Double descent curves of the students trained on data with synthetic labels (Loss v.s. Parameter number).

We also perform experiments to verify the intuition behind our methodology, i.e., the method gives a synthetic labeling function with roughly the same complexity on the original inputs and the injected inputs. We first use our method on MNIST and samples (of the same size as MNIST) from a Gaussian to get the teacher FC(9); the double descent curve is in Figure A.22(a). Then we train students on the Gaussian data with synthetic labels from the teacher, and plot the double descent curve for the students in Figure A.22(b). The local minimums of the two U-shapes are roughly the same, matching our reasoning. Then we also train larger teachers and plot the double descent curve for students on Gaussian data. Figure A.22(c) Teacher size 50. Figure A.22(d) Teacher size 500. The local minimum of the U-shape becomes larger when the teacher gets larger, again matching our reasoning.

A.6 Provable Guarantees for Neural Networks in A More General Setting

This section provides the analysis in a more general setting. We first describe the learning problems, and then provide the proofs following similar intuitions as for the simpler settings in the main text.

A.6.1 Problem Setup

Let $\mathbf{X} = \mathbb{R}^d$ be the input space, and $\mathcal{Y} = \{\pm 1\}$ be the label space. Suppose $\mathbf{M} \in \mathbb{R}^{d \times D}$ is a dictionary with D elements, where each element $\mathbf{M}_j$ can be regarded as a pattern. We assume quite general incoherent dictionary:

(D) $\mathbf{M}$ is μ-incoherent, i.e., the columns of $\mathbf{M}$ are unit vectors, and for any $i \neq j, |\langle \mathbf{M}_i, \mathbf{M}_j \rangle| \leq \mu/\sqrt{d}$.

Note that the setting in the main text corresponds to $\mu = 0$.

Let $\tilde{\phi} \in \{0,1\}^D$ be a hidden vector that indicates the presence of each pattern, and $\mathcal{D}_{\tilde{\phi}}$ a distribution for $\tilde{\phi}$. Let $\mathbf{A} \subseteq [D]$ be a subset of size k corresponding to the class relevant patterns. Let $P \subseteq [k]$. We first sample $\tilde{\phi}$ from $\mathcal{D}_{\tilde{\phi}}$, and then generate the input $\tilde{\mathbf{x}}$ and the class label y from $\tilde{\phi}, \mathbf{A}, P$ by:

$$\tilde{\mathbf{x}} = \mathbf{M}\tilde{\phi} + \zeta, \quad y = \begin{cases} +1, & \text{if } \sum_{i \in \mathbf{A}} \tilde{\phi}_i \in P, \\ -1, & \text{otherwise} \end{cases} \tag{A.290}$$

where the Gaussian noise $\zeta \sim \mathcal{N}(0, \sigma_\zeta^2 I_{d \times d})$ is independent from $\tilde{\phi}$. Note that the setting in the main text corresponds to $\sigma_\zeta = 0$.

We allow general $\mathcal{D}_{\tilde{\phi}}$ with the following assumptions:

(A1) The patterns in $\mathbf{A}$ are correlated with the labels: for any $i \in A$, for $v \in \{\pm 1\}$ let $\gamma_v = \mathbb{E}[y\tilde{\phi}_i | y = v]$, then $\gamma := (\gamma_{+1} + \gamma_{-1})/2 > 0$.

(A2) The patterns outside $\mathbf{A}$ are independent of the patterns in $\mathbf{A}$.

Note that we allow imbalanced classes. Let $p_{\min} := \min(\Pr[y=-1], \Pr[y=+1])$. If the classes are balanced, then the assumption **(A1)** implies the assumption **(A1)** in the main text, so the setting here is more general. **(A2)** is also more general, in particular, allowing dependence between irrelevant patterns and non-identical distributions for them.

Let $\mathcal{D}(\mathbf{A}, P, \mathcal{D}_{\tilde{\phi}})$ denote the distribution on $(\tilde{\mathbf{x}}, y)$ corresponding to some $\mathbf{A}, P$, and $\mathcal{D}_{\tilde{\phi}}$. Given parameters $\Xi = (d, D, k, \gamma, p_{\mathrm{o}}, \mu, \sigma_\zeta)$, the family $\mathcal{F}_\Xi$ of distributions for learning is the set of all $\mathcal{D}(\mathbf{A}, P, \mathcal{D}_{\tilde{\phi}})$ with $\mathbf{A} \subseteq [D]$, $P \subseteq [k]$, and $\mathcal{D}_{\tilde{\phi}}$ satisfying the above assumptions.

One special case is the mixture of two Gaussians.

Example. Suppose M has one single column v, and $y = +1$ if $\tilde{\phi} = 1$ and $y = -1$ otherwise. Then the data distribution is simply a mixture of two Gaussians: $\tilde{\mathbf{x}} \sim \frac{v}{2} + \mathcal{N}(y\frac{v}{2}, \sigma_\zeta^2 I_{d\times d})$.

Neural Network Learning

Again, we will normalize the data for learning: we first compute $\mathbf{x} = (\tilde{\mathbf{x}} - \mathbb{E}[\tilde{\mathbf{x}}])/\tilde{\sigma}$ where $\tilde{\sigma}^2 := \sum_{i=1}^d (\tilde{\mathbf{x}}_i - \mathbb{E}[\tilde{\mathbf{x}}_i])^2 = \sum_{j\in[D]} \mathrm{Var}(\tilde{\phi}_j) + d\sigma_\zeta^2$ is the variance of the data, and then train on $(\mathbf{x}, y)$. This is equivalent to setting $\phi = (\tilde{\phi} - \mathbb{E}[\tilde{\phi}])/\tilde{\sigma}$ and generating $\mathbf{x} = \mathbf{M}\phi + \zeta/\sigma_\zeta$. For $(\tilde{\mathbf{x}}, y)$ from $\mathcal{D}$ and the normalized $(\mathbf{x}, y)$, we will simply say $(\mathbf{x}, y) \sim \mathcal{D}$.

The learning will be the same as that in the main text, except the following. We will use a small $\sigma_{\mathbf{w}}^2 = \tilde{\sigma}^2/\mathrm{poly}(Dm)$. And we will use a weighted loss to handle the imbalanced classes in the first two steps for feature learning, and then use the unweighted loss in the remaining steps. Formally, the weighted loss is:

$$L_{\mathcal{D}}^{\alpha}(g; \sigma_\xi) = \mathbb{E}_{(\mathbf{x},y)}[\alpha_y \ell(y, g(\mathbf{x}; \xi))], \tag{A.291}$$

where the class weights $\alpha_v = \frac{1}{2\Pr[y=v]}$ for $v \in \{\pm 1\}$.

A.6.2 Main Result

In this setting, we have the following theorem:

Theorem A.6.1. *Set*

$$\eta^{(1)} = \frac{\gamma^2 p_{\min}\tilde{\sigma}}{km^3}, \lambda_{\mathbf{a}}^{(1)} = 0, \lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \sigma_\xi^{(1)} = 1/k^{3/2}, \tag{A.292}$$

$$\eta^{(2)} = 1, \lambda_{\mathbf{a}}^{(2)} = \lambda_{\mathbf{w}}^{(2)} = 1/(2\eta^{(2)}), \sigma_\xi^{(2)} = 1/k^{3/2}, \tag{A.293}$$

$$\eta^{(t)} = \eta = \frac{k^2}{Tm^{1/3}}, \lambda_{\mathbf{a}}^{(t)} = \lambda_{\mathbf{w}}^{(t)} = \lambda \le \frac{k^3}{\tilde{\sigma}m^{1/3}}, \sigma_\xi^{(t)} = 0, \text{ for } 2 < t \le T. \tag{A.294}$$

For any $\delta \in (0, O(1/k^3))$*, if* $\mu \le O(\sqrt{d}/D)$*,* $\sigma_\zeta \le O(\min\{1/\tilde{\sigma}, \tilde{\sigma}/\sqrt{d}\})$*,* $k = \Omega\left(\log^2\left(\frac{Dmd}{\delta\gamma p_{\min}}\right)\right)$*,* $m \ge \max\{\Omega(k^4), D, d\}$*, then we have for any* $\mathcal{D} \in \mathcal{F}_\Xi$*, with probability at least* $1-\delta$*, there exists* $t \in [T]$ *such that*

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \ne y] \le L_{\mathcal{D}}(g^{(t)}) = O\left(\frac{k^8}{m^{2/3}} + \frac{k^3 T}{m^2} + \frac{k^2 m^{2/3}}{T}\right). \tag{A.295}$$

Consequently, for any $\epsilon \in (0,1)$*, if* $T = m^{4/3}$*, and* $m \ge \max\{\Omega(k^{12}/\epsilon^{3/2}), D\}$*, then*

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \ne y] \le L_{\mathcal{D}}(g^{(t)}) \le \epsilon. \tag{A.296}$$

The rest of the section is devoted to the proof of this theorem.

A.6.3 Notations

Recall some notations that we will use throughout the analysis.

For a vector v and an index set I, let v_I denote the vector containing the entries of v indexed by I, and v_{-I} denote the vector containing the entries of v with indices outside I.

Let $\rho := \mathbf{M}^\top \mathbf{M}$. Then we have $\rho_{jj} = 1$ for any j, and $|\rho_{j\ell}| \le \mu/\sqrt{d}$ for any $j \ne \ell$.

By initialization, $\mathbf{w}_i^{(0)}$ for $i \in [m]$ are i.i.d. copies of the same random variable $\mathbf{w}^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$; similar for $\mathbf{a}^{(0)}$ and $\mathbf{b}^{(0)}$. Let $\sigma_{\phi_j}^2 := p_{\text{o}j}(1-p_{\text{o}j})/\tilde{\sigma}^2$ denote the variance of ϕ_ℓ for $\ell \notin \mathbf{A}$, where $p_{\text{o}j} = \Pr[\tilde{\phi}_j = 1]$. Let p_{o} be the value such that with probability $1 - \exp(-\Omega(k))$, $\sum_{j\notin\mathbf{A}} \tilde{\phi}_j \le p_{\text{o}}(D-k)$ for some $p_{\text{o}} \in [0,1]$. That is, p_{o} is an upper bound on the density of $\tilde{\phi}_j$ with high probability.

Let $q_\ell := \langle \mathbf{w}^{(0)}, \mathbf{M}_\ell \rangle$. Similarly, define $q_{i,\ell}^{(t)} := \langle \mathbf{w}_i^{(t)}, \mathbf{M}_\ell \rangle$.

We also define the following sets to denote typical initialization. For a fixed $\delta \in (0,1)$, define

$$\mathcal{G}_{\mathbf{w}}(\delta) := \Bigg\{ \mathbf{w} \in \mathbb{R}^d : q_\ell = \langle \mathbf{w}, \mathbf{M}_\ell \rangle, \frac{\sigma_{\mathbf{w}}^2 d}{2} \le \|\mathbf{w}^{(0)}\|_2^2 \le \frac{3\sigma_{\mathbf{w}}^2 d}{2}, \tag{A.297}$$

$$\frac{\sigma_{\mathbf{w}}^2 (D-k)}{2} \le \sum_{\ell \notin \mathbf{A}} q_\ell^2 \le \frac{3\sigma_{\mathbf{w}}^2 (D-k)}{2},$$

$$\max_\ell |q_\ell| \le \sigma_{\mathbf{w}} \sqrt{2\log(Dm/\delta)} \Bigg\}, \tag{A.298}$$

$$\mathcal{G}_{\mathbf{a}}(\delta) := \{ \mathbf{a} \in \mathbb{R} : |\mathbf{a}| \le \sigma_{\mathbf{a}} \sqrt{2\log(m/\delta)} \}. \tag{A.299}$$

$$\mathcal{G}_{\mathbf{b}}(\delta) := \{ \mathbf{b} \in \mathbb{R} : |\mathbf{b}| \le \sigma_{\mathbf{b}} \sqrt{2\log(m/\delta)} \}. \tag{A.300}$$

A.6.4 Existence of A Good Network

We first show that there exists a network that can fit the data distribution.

Lemma A.6.2. *Suppose* $\frac{k\mu}{\sqrt{d}} \frac{p_o D}{\tilde{\sigma}} \le \frac{1}{16}$. *For any* $\mathcal{D} \in \mathcal{F}_\Xi$, *there exists a network* $g^*(\mathbf{x}) = \sum_{i=1}^n \mathbf{a}_i^* \sigma(\langle \mathbf{w}_i^*, \mathbf{x} \rangle + \mathbf{b}_i^*)$ *which satisfies*

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}[y g^*(x) \le 1] \le \exp(-\Omega(k)) + \exp\left(-\Omega\left(\frac{1}{\sigma_\zeta^2 (k + k^2\mu/\sqrt{d})} \right) \right).$$

Furthermore, the number of neurons $n = 3(k+1)$, $|\mathbf{a}_i^*| \le 64k, 1/(64k) \le |\mathbf{b}_i^*| \le 1/4$, $\mathbf{w}_i^* = \tilde{\sigma} \sum_{j \in \mathbf{A}} \mathbf{M}_j / (8k)$, *and* $|\langle \mathbf{w}_i^*, \mathbf{x} \rangle + \mathbf{b}_i^*| \le 1$ *for any* $i \in [n]$ *and* $(\mathbf{x}, y) \sim \mathcal{D}$.

Consequently, if furthermore we have $k\mu/\sqrt{d} < 1$ *and* $\sigma_\zeta < 1/k$, *then*

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}[y g^*(x) \le 1] \le \exp(-\Omega(k)).$$

Proof of Lemma A.6.2. Let $\mathbf{w} = \tilde{\sigma} \sum_{j \in \mathbf{A}} \mathbf{M}_j$ and let $u = \sum_{j \in \mathbf{A}} \mathbb{E}[\tilde{\phi}_j]$. We have

$$\langle \mathbf{w}, \mathbf{x} \rangle = \tilde{\sigma} \sum_{j \in \mathbf{A}} \langle \mathbf{M}_j, \mathbf{M}\phi \rangle + \langle \mathbf{w}, \zeta/\tilde{\sigma} \rangle \tag{A.301}$$

$$= \sum_{j \in \mathbf{A}} \phi_j + \sum_{j \in \mathbf{A}, \ell \neq j} \rho_{j\ell} \phi_\ell + \langle \mathbf{w}, \zeta/\tilde{\sigma} \rangle \tag{A.302}$$

$$= \sum_{j \in \mathbf{A}} \tilde{\phi}_j - u + \underbrace{\sum_{j \in \mathbf{A}, \ell \neq j} \rho_{j\ell} \phi_\ell + \langle \mathbf{w}, \zeta/\tilde{\sigma} \rangle}_{:= \epsilon_{\mathbf{x}}} . \tag{A.303}$$

With probability $\geq 1 - \exp(-\Omega(k))$, among all $j \notin \mathbf{A}$, we have that at most $p_{\mathrm{o}}(D-k)$ of ϕ_j are $(1-p_{\mathrm{o}})/\tilde{\sigma}$, while the others are $-p_{\mathrm{o}}/\tilde{\sigma}$, and thus

$$\left| \sum_{j \in \mathbf{A}, \ell \neq j} \rho_{j\ell} \phi_\ell \right| \leq \frac{k\mu}{\sqrt{d}} \frac{p_{\mathrm{o}} D}{\tilde{\sigma}} \leq \frac{1}{16}. \tag{A.304}$$

Furthermore, $\langle \mathbf{w}, \zeta \rangle \sim \mathcal{N}(0, \sigma_\zeta^2 \|w\|_2^2)$ and $\|w\|_2^2 \leq \tilde{\sigma}^2 (k + k^2\mu/\sqrt{d})$, we have

$$\Pr[|\langle \mathbf{w}, \zeta/\tilde{\sigma} \rangle| \leq 1/16] \geq 1 - \exp\left(-\Theta\left(\frac{1}{\sigma_\zeta^2 \|w\|_2^2 / \tilde{\sigma}^2} \right) \right) \tag{A.305}$$

$$\geq 1 - \exp\left(-\Theta\left(\frac{1}{\sigma_\zeta^2 (k + k^2\mu/\sqrt{d})} \right) \right). \tag{A.306}$$

For good data points with ϕ and ζ satisfying the above, we have $|\epsilon_{\mathbf{x}}| \leq 1/8$. By Lemma A.2.1,

$$g_1^*(x) := \sum_{p \in P} \delta_{p-\mu,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) - \sum_{p \notin P, 0 \leq p \leq k} \delta_{p-\mu,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle) \tag{A.307}$$

$$= \sum_{p \in P} \delta_{p,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + u) - \sum_{p \notin P, 0 \leq p \leq k} \delta_{p,4,1/2}(\langle \mathbf{w}, \mathbf{x} \rangle + u) \tag{A.308}$$

$$= \sum_{p \in P} \delta_{p,4,1/2}\left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j + \epsilon_{\mathbf{x}} \right) - \sum_{p \notin P, 0 \leq p \leq k} \delta_{p,4,1/2}\left(\sum_{j \in \mathbf{A}} \tilde{\phi}_j + \epsilon_{\mathbf{x}} \right). \tag{A.309}$$

Then for good data points, we have $yg_1^*(x) \geq 1$. Similarly,

$$g_2^*(x) := \sum_{p\in P} \delta_{p-\mu+1/4,8,1/2}(\langle \mathbf{w}, \mathbf{x}\rangle) - \sum_{p\notin P, 0\leq p\leq k} \delta_{p-\mu+1/4,8,1/2}(\langle \mathbf{w}, \mathbf{x}\rangle) \tag{A.310}$$

$$= \sum_{p\in P} \delta_{p+1/4,8,1/2}(\langle \mathbf{w}, \mathbf{x}\rangle + u) - \sum_{p\notin P, 0\leq p\leq k} \delta_{p+1/4,8,1/2}(\langle \mathbf{w}, \mathbf{x}\rangle + u) \tag{A.311}$$

$$= \sum_{p\in P} \delta_{p+1/4,8,1/2}\left(\sum_{j\in \mathbf{A}} \tilde{\phi}_j + \epsilon_{\mathbf{x}}\right) - \sum_{p\notin P, 0\leq p\leq k} \delta_{p+1/4,8,1/2}\left(\sum_{j\in \mathbf{A}} \tilde{\phi}_j + \epsilon_{\mathbf{x}}\right). \tag{A.312}$$

Then for good data points, we have $yg_2^*(x) \geq 1$.

Note that the bias terms in g_1^* and g_2^* have distance at least $1/4$, then at least one of them satisfies that all its bias terms have absolute value $\geq 1/8$. Pick that one and denote it as $g(\mathbf{x}) = \sum_{i=1}^n \mathbf{a}_i \sigma_{\mathrm{r}}(\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i)$. By the positive homogeneity of σ_{r}, we have

$$g(\mathbf{x}) = \sum_{i=1}^n 8k\mathbf{a}_i \sigma_{\mathrm{r}}(\langle \mathbf{w}_i, \mathbf{x}\rangle/(8k) + \mathbf{b}_i/(8k)). \tag{A.313}$$

Since for any good data points, $|\langle \mathbf{w}_i, \mathbf{x}\rangle/(8k) + \mathbf{b}_i/(8k)| \leq 1$, then

$$g(\mathbf{x}) = \sum_{i=1}^n 8k\mathbf{a}_i \sigma(\langle \mathbf{w}_i, \mathbf{x}\rangle/(8k) + \mathbf{b}_i/(8k)) \tag{A.314}$$

where σ is the truncated ReLU. Now we can set $\mathbf{a}_i^* = 8k\mathbf{a}_i, \mathbf{w}_i^* = \mathbf{w}_i/(8k), \mathbf{b}_i^* = \mathbf{b}_i/(8k)$, to get our final g^*. □

A.6.5 Initialization

We first show that with high probability, the initial weights are in typical positions.

Lemma A.6.3. *Suppose $D\mu/\sqrt{d} \leq 1/16$. For any $\delta \in (0,1)$, with probability at least $1 - \delta - 2\exp(-\Theta(D-k))$ over $\mathbf{w}^{(0)}$,*

$$\sigma_{\mathbf{w}}^2 d/2 \leq \|\mathbf{w}^{(0)}\|_2^2 \leq 3\sigma_{\mathbf{w}}^2 d/2,$$

$$\sigma_{\mathbf{w}}^2(D-k)/2 \le \sum_{\ell \notin \mathbf{A}} q_\ell^2 \le 3\sigma_{\mathbf{w}}^2(D-k)/2,$$

$$\max_\ell |q_\ell| \le \sigma_{\mathbf{w}} \sqrt{2\log(D/\delta)}.$$

With probability at least $1-\delta$ *over* $\mathbf{b}^{(0)}$,

$$|\mathbf{b}^{(0)}| \le \sigma_{\mathbf{b}} \sqrt{2\log(1/\delta)}.$$

With probability at least $1-\delta$ *over* $\mathbf{a}^{(0)}$,

$$|\mathbf{a}^{(0)}| \le \sigma_{\mathbf{a}} \sqrt{2\log(1/\delta)}.$$

Proof of Lemma A.6.3. The bound on $\|\mathbf{w}^{(0)}\|_2^2$ follows from the property of Gaussians.

Note that $q = \mathbf{M}^\top \mathbf{w}^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 \rho)$ for the matrix $\rho = \mathbf{M}^\top \mathbf{M}$. We have with probability $\ge 1-\delta/2$, $\max_\ell |q_\ell| \le \sqrt{2\sigma_{\mathbf{w}}^2 \log \frac{D}{\delta}}$.

For any subset $S \subseteq [D]$, let ρ_S denote the submatrix of ρ containing the rows and columns indexed by S. Then $q_S = \mathbf{M}^\top \mathbf{w}^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 \rho_S)$. By diagonalizing ρ_S and then applying Bernstein's inequality, we have with probability $\ge 1 - 2\exp\left(-\Theta(|S|/\|\rho\|_2\right)$, $\|q_S\|_2^2 \in \left((\|\rho_S\|_F^2 - \frac{|S|}{4})\sigma_{\mathbf{w}}^2, (\|\rho_S\|_F^2 + \frac{|S|}{4})\sigma_{\mathbf{w}}^2\right)$. By Gershgorin circle theorem, we have

$$\|\rho\|_2 \le 1 + (|S|-1)\mu/\sqrt{d} \le 17/16.$$

Similarly, we have

$$\frac{3}{4}|S| \le \left(\frac{15}{16}\right)^2 |S| \le \|\rho_S\|_F^2 \le \left(\frac{17}{16}\right)^2 |S| \le \frac{5}{4}|S|.$$

The bounds on q then follow.

The bounds on $\mathbf{b}^{(0)}$ and $\mathbf{a}^{(0)}$ follow from the property of Gaussians. □

Lemma A.6.4. *Suppose* $D\mu/\sqrt{d} \le 1/16$. *We have:*

- *With probability* $\ge 1 - \delta - 2m\exp(-\Theta(D-k))$ *over* $\mathbf{w}_i^{(0)}$*'s, for all* $i \in [2m]$, $\mathbf{w}_i^{(0)} \in$

$\mathcal{G}_{\mathbf{w}}(\delta)$.

- *With probability* $\geq 1-\delta$ *over* $\mathbf{b}_i^{(0)}$*'s, for all* $i \in [2m]$, $\mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta)$.
- *With probability* $\geq 1-\delta$ *over* $\mathbf{a}_i^{(0)}$*'s, for all* $i \in [2m]$, $\mathbf{a}_i^{(0)} \in \mathcal{G}_{\mathbf{a}}(\delta)$.

Proof of Lemma A.6.4. This follows from Lemma A.6.3 by union bound. □

A.6.6 Some Auxiliary Lemmas

The expression of the gradients will be used frequently.

Lemma A.6.5.

$$\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}^{\alpha}(g;\sigma_\xi) = -\mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ \alpha_y y \mathbb{I}[yg(\mathbf{x};\xi) \leq 1] \mathbb{E}_{\xi_i} \mathbb{I}[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i + \xi_i \in (0,1)] \mathbf{x} \right\}, \tag{A.315}$$

$$\frac{\partial}{\partial \mathbf{b}_i} L_{\mathcal{D}}^{\alpha}(g;\sigma_\xi) = -\mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ \alpha_y y \mathbb{I}[yg(\mathbf{x};\xi) \leq 1] \mathbb{E}_{\xi_i} \mathbb{I}[\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i \in (0,1)] \right\}, \tag{A.316}$$

$$\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}^{\alpha}(g;\sigma_\xi) = -\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ \alpha_y y \mathbb{I}[yg(\mathbf{x};\xi) \leq 1] \mathbb{E}_{\xi_i} \sigma(\langle \mathbf{w}_i, \mathbf{x}\rangle + \mathbf{b}_i + \xi_i) \right\}. \tag{A.317}$$

Proof of Lemma A.6.5. It follows from straightforward calculation. □

We also have the following auxiliary lemma for later calculations.

Lemma A.6.6.

$$\mathbb{E}_{\phi_{\mathbf{A}}} \{\alpha_y y\} = 0, \tag{A.318}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}} \{|\alpha_y y|\} = 1, \tag{A.319}$$

$$\mathbb{E}_{\phi_j} \{|\phi_j|\} = 2\sigma_{\phi_j}^2 \tilde{\sigma}, \text{ for } j \notin \mathbf{A}, \tag{A.320}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}} \{\alpha_y y \phi_j\} = \frac{\gamma}{\tilde{\sigma}}, \text{ for } j \in \mathbf{A}, \tag{A.321}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}} \{|\alpha_y y \phi_j|\} \leq \frac{1}{\tilde{\sigma}}, \text{ for all } j \in [D]. \tag{A.322}$$

Proof of Lemma A.6.6.

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y | y = v\} \Pr[y = v] \tag{A.323}$$

$$= \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{y | y = v\} \tag{A.324}$$

$$= 0. \tag{A.325}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|\alpha_y y|\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|\alpha_y y|\, | y = v\} \Pr[y = v] \tag{A.326}$$

$$= \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|y|\, | y = v\} \tag{A.327}$$

$$= 1. \tag{A.328}$$

$$\mathbb{E}_{\phi_j}\{|\phi_j|\} = \frac{|-p_{\mathrm{o}j}|(1-p_{\mathrm{o}j}) + |1-p_{\mathrm{o}j}|p_{\mathrm{o}j}}{\tilde{\sigma}} = 2\sigma_{\phi_j}^2\tilde{\sigma}. \tag{A.329}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y\phi_j\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y\phi_j |\, y = v\} \Pr[y = v] \tag{A.330}$$

$$= \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{y\phi_j |\, y = v\} \tag{A.331}$$

$$= \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\left\{ y\frac{\tilde{\phi}_j - \mathbb{E}[\tilde{\phi}_j]}{\tilde{\sigma}} \middle| y = v \right\} \tag{A.332}$$

$$= \frac{1}{2\tilde{\sigma}}(\gamma_{+1} + \gamma_{-1}) = \frac{\gamma}{\tilde{\sigma}}. \tag{A.333}$$

$$\mathbb{E}_{\phi_{\mathbf{A}}}\{|\alpha_y y\phi_j|\} = \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|\alpha_v y\phi_j|\, | y = v\} \Pr[y = v] \tag{A.334}$$

$$\leq \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|y\phi_j|\, | y = v\} \tag{A.335}$$

$$\leq \frac{1}{2} \sum_{v\in\{\pm 1\}} \mathbb{E}_{\phi_{\mathbf{A}}}\{|y\phi_j|\, | y = v\} \tag{A.336}$$

$$\leq \frac{1}{\tilde{\sigma}}. \tag{A.337}$$

□

A.6.7 Feature Emergence: First Gradient Step

We will show that w.h.p. over the initialization, after the first gradient step, there are neurons that represent good features.

We begin with analyzing the gradients.

Lemma A.6.7. *Fix $\delta \in (0,1)$ and suppose $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta)$ for all $i \in [2m]$. Let*

$$\epsilon_e := \frac{D\sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}}{\tilde{\sigma}^2\sigma_\xi^{(1)}} + \frac{\sqrt{d}\sigma_\xi\sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}}{\tilde{\sigma}\sigma_\xi^{(1)}}, \epsilon_\nu := \epsilon_e.$$

If $\sigma_\zeta^2\sigma_{\mathbf{w}}^2 d/\tilde{\sigma}^2 = O(1/k)$, $p_o = \Omega(k^2/D)$, $k = \Omega(\log^2(Dmd/\delta))$, and $\sigma_\xi^{(1)} = O(1/k)$, then

$$\frac{\partial}{\partial \mathbf{w}_i} L_{\mathcal{D}}^{\alpha}(g^{(0)}; \sigma_\xi^{(1)}) = -\mathbf{a}_i^{(0)} \left(\sum_{j=1}^{D} \mathbf{M}_j T_j + \nu \right) \tag{A.338}$$

where T_j satisfies:

- *if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \le O(\epsilon_e/\tilde{\sigma})$, where $\beta \in [\Omega(1), 1]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;*
- *if $j \notin \mathbf{A}$, then $|T_j| \le O(\sigma_{\phi_j}^2 \epsilon_e \tilde{\sigma})$;*
- *$|\nu_j| \le O\left(\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\epsilon_\nu\right) + \frac{\sigma_\zeta d}{\tilde{\sigma}} e^{-\Theta(k)}$.*

Proof of Lemma A.6.7. Consider one neuron index i and omit the subscript i in the pa-

rameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x};\xi^{(1)})=0$, we have

$$\frac{\partial}{\partial \mathbf{w}} L_{\mathcal{D}}^{\alpha}(g^{(0)};\sigma_{\xi}^{(1)}) \tag{A.339}$$

$$= -\mathbf{a}^{(0)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y \mathbb{I}[yg^{(0)}(\mathbf{x};\xi^{(1)})\le 1]\mathbb{E}_{\xi^{(1)}}\mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\mathbf{x}\right\} \tag{A.340}$$

$$= -\mathbf{a}^{(0)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\alpha_y y \mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\mathbf{x}\right\} \tag{A.341}$$

$$= -\mathbf{a}^{(0)}\sum_{j=1}^{D}\mathbf{M}_j \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\alpha_y y \phi_j \mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\}}_{:=T_j} \tag{A.342}$$

$$- \mathbf{a}^{(0)}\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\frac{\alpha_y y \zeta}{\tilde{\sigma}}\mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\}}_{:=\nu} \tag{A.343}$$

First, consider $j\in\mathbf{A}$.

$$T_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\alpha_y y \phi_j \mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\} \tag{A.344}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\zeta}\left\{\alpha_y y \phi_j \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\}. \tag{A.345}$$

where $\iota := \langle \mathbf{w}^{(0)}, \zeta/\tilde{\sigma}\rangle$.

Let

$$I_a := \Pr_{\xi^{(1)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.346}$$

$$I_a' := \Pr_{\xi^{(1)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.347}$$

Note that $|\langle \phi_{\mathbf{A}}, q_{\mathbf{A}}\rangle| = O(\frac{k\sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}}{\tilde{\sigma}^2})$, and that $|\iota| = |\langle \mathbf{w}_i^{(0)}, \zeta/\tilde{\sigma}\rangle| = O(\frac{\sqrt{d}\sigma_\xi\sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}}{\tilde{\sigma}})$, and that $|\langle \phi, q\rangle|, |\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle|$ are $O(\frac{D\sigma_{\mathbf{w}}\sqrt{2\log(D/\delta)}}{\tilde{\sigma}^2})$. When $\sigma_{\mathbf{w}}$ is sufficiently small, by the

property of the Gaussian $\xi^{(1)}$, we have

$$
\begin{aligned}
&|I_a - I'_a| && \text{(A.348)}\\
&\leq \left| \Pr_{\xi^{(1)}} \left[\langle \phi, q \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0 \right] - \Pr_{\xi^{(1)}} \left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \geq 0 \right] \right| && \text{(A.349)}\\
&+ \Pr_{\xi^{(1)}} \left[\langle \phi, q \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \geq 1 \right] + \Pr_{\xi^{(1)}} \left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \geq 1 \right] && \text{(A.350)}\\
&= O(\epsilon_{\mathrm{e}}). && \text{(A.351)}
\end{aligned}
$$

In summary,

$$
|\mathbb{E}_{\zeta, \phi_{-\mathbf{A}}} (I_a - I'_a)| = O(\epsilon_{\mathrm{e}}). \tag{A.352}
$$

Then we have

$$
\begin{aligned}
&\left| T_j - \mathbb{E}_{\phi_{\mathbf{A}}, \zeta, \phi_{-\mathbf{A}}} \left\{ \alpha_y y \phi_j I'_a \right\} \right| && \text{(A.353)}\\
&\leq \mathbb{E}_{\phi_{\mathbf{A}}} \left\{ |\alpha_y y \phi_j| \left| \mathbb{E}_{\zeta, \phi_{-\mathbf{A}}} (I_a - I'_a) \right| \right\} && \text{(A.354)}\\
&\leq O(\epsilon_{\mathrm{e}}) \mathbb{E}_{\phi_{\mathbf{A}}} \left\{ |\alpha_y y \phi_j| \right\} && \text{(A.355)}\\
&\leq O(\epsilon_{\mathrm{e}} / \tilde{\sigma}) && \text{(A.356)}
\end{aligned}
$$

where the last step is from Lemma A.6.6. Furthermore,

$$
\begin{aligned}
&\mathbb{E}_{\phi_{\mathbf{A}}, \zeta, \phi_{-\mathbf{A}}} \left\{ \alpha_y y \phi_j I'_a \right\} && \text{(A.357)}\\
&= \mathbb{E}_{\phi_{\mathbf{A}}} \left\{ \alpha_y y \phi_j \right\} \mathbb{E}_{\zeta, \phi_{-\mathbf{A}}} [I'_a] && \text{(A.358)}\\
&= \mathbb{E}_{\phi_{\mathbf{A}}} \left\{ \alpha_y y \phi_j \right\} \Pr_{\phi_{-\mathbf{A}}, \zeta, \phi_{-\mathbf{A}}} \left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0, 1) \right] && \text{(A.359)}
\end{aligned}
$$

When $\sigma_{\mathbf{w}}$ is sufficiently small, we have

$$
\begin{aligned}
&\Pr_{\phi_{-\mathbf{A}}} \left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \mathbf{b}^{(0)} \in (0, 1/2) \right] \geq \Omega(1), && \text{(A.360)}\\
&\Pr_{\zeta, \xi^{(1)}} \left[\iota + \xi^{(1)} \in (0, 1/2) \right] = 1/2 - \exp(-\Omega(k)), && \text{(A.361)}
\end{aligned}
$$

This leads to

$$\beta := \mathbb{E}_{\zeta,\phi_{-\mathbf{A}}}[I'_a] = \Pr_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right] \geq \Omega(1). \tag{A.362}$$

By Lemma A.6.6, $\mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y \phi_j\} = \gamma/\tilde{\sigma}$. Therefore,

$$|T_j - \beta\gamma/\tilde{\sigma}| \leq O(\epsilon_{\mathrm{e}}/\tilde{\sigma}). \tag{A.363}$$

Now, consider $j \notin \mathbf{A}$. Let B denote $\mathbf{A} \cup \{j\}$.

$$T_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\zeta,\xi^{(1)}}\left\{\alpha_y y \phi_j \mathbb{I}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\} \tag{A.364}$$

$$= \mathbb{E}_{\phi_B}\mathbb{E}_{\phi_{-B},\zeta,\xi^{(1)}}\left\{\alpha_y y \phi_j \mathbb{I}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\} \tag{A.365}$$

$$= \mathbb{E}_{\phi_B,\zeta}\left\{\alpha_y y \phi_j \Pr_{\phi_{-B},\xi^{(1)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]\right\}. \tag{A.366}$$

Let

$$I_b := \Pr_{\xi^{(1)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.367}$$

$$I'_b := \Pr_{\xi^{(1)}}\left[\langle \phi_{-B}, q_{-B}\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.368}$$

Similar as above, we have $|\mathbb{E}_{\zeta,\xi^{(1)}}(I_b - I'_b)| \leq O(\epsilon_{\mathrm{e}})$. Then by Lemma A.6.6,

$$\left|T_j - \mathbb{E}_{\phi_B,\zeta,\phi_{-B}}\left\{\alpha_y y \phi_j I'_b\right\}\right| \tag{A.369}$$

$$\leq \mathbb{E}_{\phi_B}\left\{|\alpha_y y \phi_j||\mathbb{E}_{\zeta,\phi_{-B}}(I_b - I'_b)|\right\} \tag{A.370}$$

$$\leq O(\epsilon_{\mathrm{e}})\mathbb{E}_{\phi_{\mathbf{A}}}\{|\alpha_y y|\}\,\mathbb{E}_{\phi_j}\{|\phi_j|\} \tag{A.371}$$

$$\leq O(\epsilon_{\mathrm{e}}) \times 1 \times O(\sigma^2_{\phi_j}\tilde{\sigma}) \tag{A.372}$$

$$= O(\sigma^2_{\phi_j}\epsilon_{\mathrm{e}}\tilde{\sigma}). \tag{A.373}$$

Furthermore,

$$\mathbb{E}_{\phi_B,\zeta,\phi_{-B}}\left\{\alpha_y y\phi_j I_b'\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\right\}\mathbb{E}_{\phi_j}\left\{\phi_j\right\}\mathbb{E}_{\zeta,\phi_{-B}}[I_b'] = 0. \tag{A.374}$$

Therefore,

$$|T_j| \leq O(\sigma_\phi^2 \epsilon_{\mathrm{e}} \tilde{\sigma}). \tag{A.375}$$

Finally, consider ν_j.

$$\nu_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}\mathbb{I}[\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\} \tag{A.376}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}\mathbb{I}[\langle \phi,q\rangle + \iota_j + \iota_{-j} + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\} \tag{A.377}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\zeta}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}\Pr_{\phi_{-\mathbf{A}},\xi^{(1)}}[\langle \phi,q\rangle + \iota_j + \iota_{-j} + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)]\right\} \tag{A.378}$$

where $\iota_j := \mathbf{w}_j^{(0)}\zeta_j/\tilde{\sigma}$ and $\iota_{-j} := \langle \mathbf{w}^{(0)},\zeta/\tilde{\sigma}\rangle - \iota_j$.

With probability $\geq 1 - d\exp(-\Theta(k))$ over ζ, for any j, $|\zeta_j| \leq O(\sigma_\zeta\sqrt{\log(k)})$. Let $\mathcal{G}_\zeta$ denote this event.

Let

$$I_j := \Pr_{\xi^{(1)}}\left[\langle \phi,q\rangle + \iota_j + \iota_{-j} + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.379}$$

$$I_j' := \Pr_{\xi^{(1)}}\left[\langle \phi,q\rangle + \iota_{-j} + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.380}$$

Similar as above, we have $|\mathbb{E}_\zeta[I_j - I_j'|\mathcal{G}_\zeta]| \leq O(\epsilon_\nu)$. Then

$$|\mathbb{E}_{\zeta,\phi_{-\mathbf{A}}}(I_j - I_j')| \leq |\mathbb{E}_{\zeta,\phi_{-\mathbf{A}}}[(I_j - I_j')|\mathcal{G}_\zeta]| + \Pr[-\mathcal{G}_\zeta] \tag{A.381}$$

$$\leq O(\epsilon_\nu + d\exp(-\Theta(k))). \tag{A.382}$$

$$\left|\nu_j - \mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}} I'_j\right\}\right| \tag{A.383}$$

$$= \left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)\right\}\right| \tag{A.384}$$

$$\leq \left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)|\mathcal{G}_\zeta\right\}\right| + \left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)| - \mathcal{G}_\zeta\right\}\right| \Pr[-\mathcal{G}_\zeta]. \tag{A.385}$$

The first term is bounded by

$$\left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)|\mathcal{G}_\zeta\right\}\right| \tag{A.386}$$

$$\leq \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\frac{\alpha_y y\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}|\mathbb{E}_{\zeta,\phi_{-\mathbf{A}}}[I_b - I'_b|\mathcal{G}_\zeta]|\right\} \tag{A.387}$$

$$\leq O(\epsilon_\nu)\mathbb{E}_{\phi_{\mathbf{A}}}\left\{|\alpha_y y|\right\}\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}} \tag{A.388}$$

$$\leq O(\epsilon_\nu) \times 1 \times \frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}} \tag{A.389}$$

$$= O\left(\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\epsilon_\nu\right). \tag{A.390}$$

The second term is bounded by

$$\left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)| - \mathcal{G}_\zeta\right\}\right| \Pr[-\mathcal{G}_\zeta] \tag{A.391}$$

$$\leq \left|\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}}(I_j - I'_j)| - \mathcal{G}_\zeta\right\}\right| \times de^{-\Theta(k)} \tag{A.392}$$

$$\leq \mathbb{E}_{\phi_{\mathbf{A}}}\left|\frac{\alpha_y y}{\tilde{\sigma}}\right| \times \mathbb{E}_\zeta\left\{|\zeta_j|| - \mathcal{G}_\zeta\right\} \times de^{-\Theta(k)} \tag{A.393}$$

$$\leq \frac{\sigma_\zeta}{\tilde{\sigma}} \times de^{-\Theta(k)} \tag{A.394}$$

$$\leq \frac{\sigma_\zeta d}{\tilde{\sigma}}e^{-\Theta(k)}. \tag{A.395}$$

Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\zeta,\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y\zeta_j}{\tilde{\sigma}} I'_j\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\right\}\mathbb{E}_{\zeta_j}\left\{\frac{\zeta_j}{\tilde{\sigma}}\right\}\mathbb{E}_{\zeta_{-j}}[I'_j] = 0. \tag{A.396}$$

Therefore,

$$|\nu_j| \leq O\left(\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\epsilon_\nu\right) + \frac{\sigma_\zeta d}{\tilde{\sigma}}e^{-\Theta(k)}. \tag{A.397}$$

□

Lemma A.6.8. *Under the same assumptions as in Lemma A.6.7,*

$$\frac{\partial}{\partial \mathbf{b}_i} L_{\mathcal{D}}^{\alpha}(g^{(0)}; \sigma_\xi^{(1)}) = -\mathbf{a}_i^{(0)} T_b \tag{A.398}$$

where $|T_b| \leq O(\epsilon_e)$.

Proof of Lemma A.6.8. Consider one neuron index i and omit the subscript i in the parameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x}; \xi^{(1)}) = 0$, we have

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}^{\alpha}(g^{(0)}; \sigma_\xi^{(1)}) && \text{(A.399)}\\
&= -\mathbf{a}^{(0)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ \alpha_y y \mathbb{I}[y g^{(0)}(\mathbf{x}; \xi^{(1)}) \leq 1] \mathbb{E}_{\xi^{(1)}} \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\} && \text{(A.400)}\\
&= -\mathbf{a}^{(0)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}, \xi^{(1)}} \left\{ \alpha_y y \mathbb{I}[\langle \mathbf{w}^{(0)}, \mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\} && \text{(A.401)}\\
&= -\mathbf{a}^{(0)} \underbrace{\mathbb{E}_{\phi_{\mathbf{A}}, \zeta, \xi^{(1)}} \left\{ \alpha_y y \Pr_{\phi_{-\mathbf{A}}} \left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1) \right] \right\}}_{:= T_b}. && \text{(A.402)}
\end{aligned}$$

where $\iota := \langle \mathbf{w}^{(0)}, \zeta/\tilde{\sigma}\rangle$. Similar to the proof in Lemma A.6.7,

$$\begin{aligned}
&\left| \mathbb{E}_\zeta \left(\Pr_{\phi_{-\mathbf{A}}, \xi^{(1)}} [\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right. \right. && \text{(A.403)}\\
&\qquad \left. \left. - \Pr_{\phi_{-\mathbf{A}}, \xi^{(1)}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right) \right| = O(\epsilon_{\mathrm{e}}). && \text{(A.404)}
\end{aligned}$$

Then

$$\left| T_b - \mathbb{E}_{\phi_{\mathbf{A}},\zeta} \left\{ \alpha_y y \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\} \right| \tag{A.405}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\zeta} \left\{ |\alpha_y y| \left| \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}} [\langle \phi, q \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right. \right. \tag{A.406}$$

$$\left. \left. - \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right| \right\} \tag{A.407}$$

$$\leq O(\epsilon_{\mathrm{e}}) \mathbb{E}_{\phi_{\mathbf{A}}} \{ |\alpha_y y| \} \tag{A.408}$$

$$\leq O(\epsilon_{\mathrm{e}}). \tag{A.409}$$

Also,

$$\mathbb{E}_{\phi_{\mathbf{A}},\zeta} \left\{ \alpha_y y \Pr_{\phi_{-\mathbf{A}},\xi^{(1)}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \right\} \tag{A.410}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}} \{ \alpha_y y \} \Pr_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}} [\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}} \rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)] \tag{A.411}$$

$$= 0. \tag{A.412}$$

Therefore, $|T_b| \leq O(\epsilon_{\mathrm{e}})$. □

Lemma A.6.9. *We have*

$$\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}^{\alpha}(g^{(0)}; \sigma_{\xi}^{(1)}) = -T_a \tag{A.413}$$

where $|T_a| \leq O(\max_{\ell} q_{i,\ell}^{(0)})$. *So if* $w_i^{(0)} \in \mathcal{G}(\delta)$, $|T_a| \leq O(\sigma_{\mathbf{w}} \sqrt{\log(Dm/\delta)})$.

Proof of Lemma A.6.9. Consider one neuron index i and omit the subscript i in the pa-

rameters. Since the unbiased initialization leads to $g^{(0)}(\mathbf{x};\xi^{(1)})=0$, we have

$$\frac{\partial}{\partial \mathbf{a}} L_{\mathcal{D}}^{\alpha}(g^{(0)};\sigma_{\xi}^{(1)}) \tag{A.414}$$

$$= -\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y \mathbb{I}[y g^{(0)}(\mathbf{x};\xi^{(1)})\le 1]\mathbb{E}_{\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\} \tag{A.415}$$

$$= -\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(1)}}\left\{\alpha_y y \sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\}}_{:=T_a}. \tag{A.416}$$

Let $\phi'_{\mathbf{A}}$ be an independent copy of $\phi_{\mathbf{A}}$, ϕ' be the vector obtained by replacing in ϕ the entries $\phi_{\mathbf{A}}$ with $\phi'_{\mathbf{A}}$, and let $x'=\mathbf{M}\phi'+\zeta/\tilde{\sigma}$ and its label is y'. Then

$$|T_a| = \left|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y \mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})\right\}\right| \tag{A.417}$$

$$\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=1\right\} \tag{A.418}$$

$$- \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=-1\right\}\Bigg| \tag{A.419}$$

$$\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y=1\right\} \tag{A.420}$$

$$- \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(1)}}\sigma(\langle \mathbf{w}^{(0)},\mathbf{x}'\rangle + \mathbf{b}^{(0)} + \xi^{(1)})|y'=-1\right\}\Bigg|. \tag{A.421}$$

Since σ is 1-Lipschitz,

$$|T_a| \le \frac{1}{2}\mathbb{E}_{\phi_{\mathbf{A}},\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}}}\left|\langle \mathbf{w}^{(0)},\mathbf{M}\phi\rangle - \langle \mathbf{w}^{(0)},\mathbf{M}\phi'\rangle\right| |y=1,y'=-1\right\} \tag{A.422}$$

$$\le \frac{1}{2}\mathbb{E}_{\phi_{-\mathbf{A}}}\left(\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(0)},\mathbf{M}\phi\rangle\right| |y=1\right\} + \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(0)},\mathbf{M}\phi'\rangle\right| |y'=-1\right\}\right) \tag{A.423}$$

$$\le \max_{\ell} q_{i,\ell}^{(0)} \sqrt{\mathbb{E}_{\phi}\left(\sum_{\ell\in[D]}\phi_{\ell}^2 + \sum_{j\neq\ell:j,\ell\in\mathbf{A}}|\phi_j\phi_\ell|\right)} \tag{A.424}$$

$$\le \max_{\ell} q_{i,\ell}^{(0)} \sqrt{\mathbb{E}_{\phi}\left(1+O(1)\right)} \tag{A.425}$$

$$= \Theta(\max_{\ell} q_{i,\ell}^{(0)}). \tag{A.426}$$

□

With the bounds on the gradient, we now summarize the results for the weights after the first gradient step.

Lemma A.6.10. *Set*

$$\lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \lambda_{\mathbf{a}}^{(1)} = \lambda_{\mathbf{b}}^{(1)} = 0, \sigma_\xi^{(1)} = 1/k^{3/2}.$$

Fix $\delta \in (0,1)$ *and suppose* $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta)$ *for all* $i \in [2m]$. *If* $k = \Omega(\log^2(Dm/\delta))$, *then for all* $i \in [m]$, $\mathbf{w}_i^{(1)} = \sum_{\ell=1}^D q_{i,\ell}^{(1)} \mathbf{M}_\ell + \upsilon$ *satisfying*

- *if* $\ell \in A$, *then* $|q_{i,\ell}^{(1)} - \eta^{(1)}\mathbf{a}_i^{(0)}\beta\gamma/\tilde{\sigma}| \le O\left(\frac{|\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_e}{\tilde{\sigma}}\right)$, *where* $\beta \in [\Omega(1), 1]$ *and depends only on* $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;
- *if* $\ell \notin \mathbf{A}$, *then* $|q_{i,\ell}^{(1)}| \le O\left(|\eta^{(1)}\mathbf{a}_i^{(0)}|\sigma_{\phi_\ell}^2\epsilon_e\tilde{\sigma}\right)$;
- $|\upsilon_j| \le O\left(|\eta^{(1)}\mathbf{a}_i^{(0)}|\left(\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\epsilon_\nu + \frac{\sigma_\zeta d}{\tilde{\sigma}}e^{-\Theta(k)}\right)\right)$.

and

- $\mathbf{b}_i^{(1)} = \mathbf{b}_i^{(0)} + \eta^{(1)}\mathbf{a}_i^{(0)}T_b$ *where* $|T_b| = O(\epsilon_e)$;
- $\mathbf{a}_i^{(1)} = \mathbf{a}_i^{(0)} + \eta^{(1)}T_a$ *where* $|T_a| = O(\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$.

Proof of Lemma A.6.10. This follows from Lemma A.6.4 and Lemma A.6.7-A.6.9. □

A.6.8 Feature Improvement: Second Gradient Step

We first show that with properly set $\eta^{(1)}$, for most $\mathbf{x}$, $|g^{(1)}(\mathbf{x}; \sigma_\xi^{(2)})| < 1$ and thus $yg^{(1)}(\mathbf{x}; \sigma_\xi^{(2)}) < 1$.

Lemma A.6.11. *Fix* $\delta \in (0,1)$ *and suppose* $\mathbf{w}_i^{(0)} \in \mathcal{G}_{\mathbf{w}}(\delta), \mathbf{b}_i^{(0)} \in \mathcal{G}_{\mathbf{b}}(\delta), \mathbf{a}_i^{(0)} \in \mathcal{G}_{\mathbf{a}}(\delta)$ *for all* $i \in [2m]$. *If* $D\mu/\sqrt{d} \le 1/16$, $\sigma_\zeta\tilde{\sigma} = O(1)$, $\sigma_\zeta^2 d/\tilde{\sigma}^2 = O(1)$, $k = \Omega(\log^2(Dm/\delta))$, $\sigma_{\mathbf{a}} \le \tilde{\sigma}^2/(\gamma k^2)$, $\eta^{(1)} = O\left(\frac{\gamma}{km\sigma_{\mathbf{a}}\tilde{\sigma}}\right)$, *and* $\sigma_\xi^{(2)} \le 1/k$, *then with probability* $\ge 1 - (d + D)\exp(-\Omega(k))$ *over* $(\mathbf{x}, y)$, *we have* $yg^{(1)}(\mathbf{x}; \sigma_\xi^{(2)}) < 1$. *Furthermore, for any* $i \in [2m]$, $\left|\langle \mathbf{w}_i^{(1)}, \zeta/\tilde{\sigma}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, $\left|\langle q_i^{(1)}, \phi\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, *and* $\left|\langle (q_i^{(1)})_{-\mathbf{A}}, \phi_{-\mathbf{A}}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, *and for any* $j \in [d], \ell \in [D]$, $|\zeta_j| \le O(\sigma_\zeta\sqrt{\log(k)})$ *and* $|\langle \zeta, D_\ell\rangle| \le O(\sigma_\zeta\sqrt{\log(k)})$.

Proof of Lemma A.6.11. Note that $\mathbf{w}_i^{(0)} = \mathbf{w}_{m+i}^{(0)}$, $\mathbf{b}_i^{(0)} = \mathbf{b}_{m+i}^{(0)}$, and $\mathbf{a}_i^{(0)} = \mathbf{a}_{m+i}^{(0)}$. Then the gradient for $\mathbf{w}_{m+i}$ is the negation of that for $\mathbf{w}_{m+i}$, the gradient for $\mathbf{b}_{m+i}$ is the negation of that for $\mathbf{b}_{m+i}$, and the gradient for $\mathbf{a}_{m+i}$ is the same as that for $\mathbf{a}_{m+i}$.

$$\left|g^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right| \tag{A.427}$$

$$= \left|\sum_{i=1}^{2m} \mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right| \tag{A.428}$$

$$= \left|\sum_{i=1}^{m} \left(\mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle + \mathbf{b}_{m+i}^{(1)} + \xi_{m+i}^{(2)})\right)\right| \tag{A.429}$$

$$\leq \left|\sum_{i=1}^{m} \left(\mathbf{a}_i^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right)\right| \tag{A.430}$$

$$+ \left|\sum_{i=1}^{m} \left(-\mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)}) + \mathbf{a}_{m+i}^{(1)} \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle + \mathbf{b}_{m+i}^{(1)} + \xi_i^{(2)})\right)\right|. \tag{A.431}$$

Then we have

$$\left|g^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right| \leq \sum_{i=1}^{m} \left|2\eta^{(1)} T_a \mathbb{E}_{\xi^{(2)}} \sigma(\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)} + \xi_i^{(2)})\right| \tag{A.432}$$

$$+ \sum_{i=1}^{m} \left|\mathbf{a}_{m+i}^{(1)}\right| \left(\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| + \left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right|\right) \tag{A.433}$$

$$\leq \sum_{i=1}^{m} \left|2\eta^{(1)} T_a\right| \left(\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle + \mathbf{b}_i^{(1)}\right| + \mathbb{E}_{\xi^{(2)}} \left|\xi_i^{(2)}\right|\right) \tag{A.434}$$

$$+ \sum_{i=1}^{m} \left|\mathbf{a}_{m+i}^{(1)}\right| \left(\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| + \left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right|\right). \tag{A.435}$$

With probability $\geq 1 - \exp(-\Omega(k))$, among all $j \notin \mathbf{A}$, we have that at most $p_{\mathrm{o}}(D-k)$ of ϕ_j are $(1-p_{\mathrm{o}j})/\tilde{\sigma}$, while the others are $-p_{\mathrm{o}j}/\tilde{\sigma}$. With probability $\geq 1 - (d+D)\exp(-\Omega(k))$ over ζ, for any j, $|\zeta_j| \leq O(\sigma_\zeta \sqrt{\log(k)})$ and $|\langle \zeta, D_\ell \rangle| \leq O(\sigma_\zeta \sqrt{\log(k)})$. For data points with ϕ and ζ satisfying these, we have:

Claim A.6.1. $\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| \leq O(\eta^{(1)}/\gamma)(1+\tilde{\sigma}+\tilde{\sigma}/\sqrt{k})$.

Proof of Claim A.6.1.

$$\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| = \left|\langle \sum_{\ell=1}^{D} q_{i,\ell}^{(1)} \mathbf{M}_\ell + \upsilon, \sum_{j=1}^{D} \phi_j \mathbf{M}_j + \zeta/\tilde{\sigma}\rangle\right| \tag{A.436}$$

$$\leq \left|\langle \sum_{\ell=1}^{D} q_{i,\ell}^{(1)} \mathbf{M}_\ell, \sum_{j=1}^{D} \phi_j \mathbf{M}_j\rangle\right| + \left|\langle \sum_{\ell=1}^{D} q_{i,\ell}^{(1)} \mathbf{M}_\ell, \zeta/\tilde{\sigma}\rangle\right| + \left|\langle \upsilon, \sum_{j=1}^{D} \phi_j \mathbf{M}_j\rangle\right| + \left|\langle \upsilon, \zeta/\tilde{\sigma}\rangle\right|. \tag{A.437}$$

For each term above we bound as follows. Note that when $\sigma_{\mathbf{w}}$ is sufficiently small, $\epsilon_{\mathrm{e}} = O(k\log^{1/2}(Dm/\delta)/\sqrt{D})$. Let

$$B_1 := \beta\gamma/\tilde{\sigma} + \epsilon_{\mathrm{e}}/\tilde{\sigma}, \tag{A.438}$$

$$B_2 := \sigma_\phi^2 \epsilon_{\mathrm{e}} \tilde{\sigma} = O(\epsilon_{\mathrm{e}}/\sqrt{D}), \tag{A.439}$$

$$C_1 = \frac{k}{\tilde{\sigma}}, \tag{A.440}$$

$$C_2 := p_{\mathrm{o}} D/\tilde{\sigma} = O(D/\tilde{\sigma}). \tag{A.441}$$

Then

$$|\mathbf{a}_i^{(0)}| B_1 C_1 = O(\log(Dm/\delta)/k + \log(Dm/\delta)\epsilon_{\mathrm{e}}/(\gamma k)) = O(1/\gamma), \tag{A.442}$$

$$|\mathbf{a}_i^{(0)}| B_2 C_2 = O(\tilde{\sigma}/\gamma), \tag{A.443}$$

$$|\mathbf{a}_i^{(0)}| B_1 C_2 = O(D/k + \sqrt{D}/\gamma), \tag{A.444}$$

$$|\mathbf{a}_i^{(0)}| B_2 C_1 = O(\epsilon_{\mathrm{e}}/(\gamma\sqrt{k})) = O(1/\gamma), \tag{A.445}$$

Then by the assumption on μ,

$$\left|\langle\sum_{\ell=1}^{D} q_{i,\ell}^{(1)}\mathbf{M}_\ell, \sum_{j=1}^{D}\phi_j\mathbf{M}_j\rangle\right| \tag{A.446}$$

$$= \left|\sum_{\ell\in\mathbf{A}}\langle q_{i,\ell}^{(1)}\mathbf{M}_\ell, \mathbf{M}_\ell\phi_\ell\rangle\right| + \left|\sum_{\ell\notin\mathbf{A}}\langle q_{i,\ell}^{(1)}\mathbf{M}_\ell, \mathbf{M}_\ell\phi_\ell\rangle\right| + \left|\sum_{\ell\neq j}\langle q_{i,\ell}^{(1)}\mathbf{M}_\ell, \mathbf{M}_j\phi_j\rangle\right| \tag{A.447}$$

$$\leq O(|\eta^{(1)}\mathbf{a}_i^{(0)}|)\left(B_1C_1 + B_2C_2 + \frac{\mu}{\sqrt{d}}(kB_1(C_1+C_2) + DB_2(C_1+C_2))\right) \tag{A.448}$$

$$\leq O(|\eta^{(1)}\mathbf{a}_i^{(0)}|)\left(B_1C_1 + B_2C_2 + \frac{k\mu}{\sqrt{d}}B_1C_2 + B_2C_1\right) \tag{A.449}$$

$$\leq O(\eta^{(1)})(1/\gamma + \tilde{\sigma}/\gamma + 1/\gamma + 1/\gamma) \tag{A.450}$$

$$\leq O(\eta^{(1)}/\gamma)(1+\tilde{\sigma}). \tag{A.451}$$

By the assumption on σ_ζ,

$$\left|\langle\sum_{\ell=1}^{D} q_{i,\ell}^{(1)}\mathbf{M}_\ell, \zeta/\tilde{\sigma}\rangle\right| \tag{A.452}$$

$$\leq O(|\eta^{(1)}\mathbf{a}_i^{(0)}|)(kB_1 + DB_2)\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}} \tag{A.453}$$

$$\leq O(\eta^{(1)})(kB_1 + DB_2)\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\frac{\tilde{\sigma}^2\sqrt{\log(Dm/\delta)}}{\gamma k^2} \tag{A.454}$$

$$\leq O(\eta^{(1)})\left(\sigma_\zeta\log(Dm/\delta)\left(\frac{1}{k} + \frac{\epsilon_{\mathrm{e}}}{\gamma k}\right) + \sigma_\zeta\tilde{\sigma}\frac{\log(k)\log(Dm/\delta)}{\gamma k}\right) \tag{A.455}$$

$$\leq O(\eta^{(1)}/\gamma). \tag{A.456}$$

Also note that $|\nu_j| \leq O\left(\frac{\sigma_\zeta\log^2(Dm/\delta)}{\tilde{\sigma}\sqrt{D}}\right)$. Then by the assumption on σ_ζ,

$$\left|\langle\upsilon, \sum_{j=1}^{D}\phi_j\mathbf{M}_j\rangle\right| \tag{A.457}$$

$$\leq O(|\eta^{(1)}\mathbf{a}_i^{(0)}|) \times \sqrt{d} \times O\left(\frac{\sigma_\zeta\log^2(Dm/\delta)}{\tilde{\sigma}\sqrt{D}}\right) \times (C_1+C_2) \tag{A.458}$$

$$\leq O(|\eta^{(1)}/\gamma). \tag{A.459}$$

Finally, we have

$$|\langle \upsilon, \zeta/\tilde{\sigma}\rangle| \leq \sum_{j=1}^{d} |\upsilon_j||\zeta_j/\tilde{\sigma}| \tag{A.460}$$

$$\leq O(|\eta^{(1)}\mathbf{a}_i^{(0)}|) \times d \times \frac{\sigma_\zeta \log^2(Dm/\delta)}{\tilde{\sigma}\sqrt{D}} \frac{\tilde{\sigma}\sqrt{\log(k)}}{\tilde{\sigma}} \tag{A.461}$$

$$\leq O(\eta^{(1)}/\gamma)\frac{\tilde{\sigma}}{\sqrt{k}}. \tag{A.462}$$

□

We also have:

$$|T_a| = O(\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}) \tag{A.463}$$

$$\left|\mathbf{b}_i^{(1)}\right| \leq \left|\mathbf{b}_i^{(0)}\right| + \left|\eta^{(1)}\mathbf{a}_i^{(0)}T_b\right| \tag{A.464}$$

$$\leq \frac{\sqrt{\log(m/\delta)}}{k^2} + \left|\eta^{(1)}\mathbf{a}_i^{(0)}\epsilon_{\mathrm{e}}\right|. \tag{A.465}$$

$$\mathbb{E}_{\xi^{(2)}}\left|\xi_i^{(2)}\right| \leq O(\sigma_\xi^{(2)}). \tag{A.466}$$

$$|\mathbf{a}_{m+i}^{(1)}| \leq |\mathbf{a}_i^{(0)}| + |\eta^{(1)}T_a| \leq |\mathbf{a}_i^{(0)}| + O(\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}). \tag{A.467}$$

$$\left|\langle \mathbf{w}_i^{(1)} - \mathbf{w}_{m+i}^{(1)}, \mathbf{x}\rangle\right| = 2\left|\langle \mathbf{w}_i^{(1)}, \mathbf{x}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma). \tag{A.468}$$

$$\left|\mathbf{b}_i^{(1)} - \mathbf{b}_{m+i}^{(1)}\right| = 2|\eta^{(1)}\mathbf{a}_i^{(0)}T_b| = O(|\eta^{(1)}\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}). \tag{A.469}$$

Then we have

$$\left|g^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right| \le O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\right)\left(\frac{\eta^{(1)}\tilde{\sigma}}{\gamma}+\frac{\sqrt{\log(m/\delta)}}{k^2}+\left|\eta^{(1)}\mathbf{a}_i^{(0)}\epsilon_{\mathrm{e}}\right|+\sigma_\xi^{(2)}\right) \tag{A.470}$$

$$+O\left(m(|\mathbf{a}_i^{(0)}|+\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})\right)\left(\frac{\eta^{(1)}\tilde{\sigma}}{\gamma}+\left|\eta^{(1)}\mathbf{a}_i^{(0)}\epsilon_{\mathrm{e}}\right|\right) \tag{A.471}$$

$$=O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\frac{\log(Dm/\delta)}{k}+m|\mathbf{a}_i^{(0)}|\left(\frac{\eta^{(1)}\tilde{\sigma}}{\gamma}+\left|\eta^{(1)}\mathbf{a}_i^{(0)}\epsilon_{\mathrm{e}}\right|\right)\right) \tag{A.472}$$

$$=O\left(m\eta^{(1)}\sigma_{\mathbf{w}}\frac{\log(Dm/\delta)}{k}+m|\mathbf{a}_i^{(0)}|\frac{\eta^{(1)}\tilde{\sigma}}{\gamma}+m|\mathbf{a}_i^{(0)}|\frac{\eta^{(1)}\tilde{\sigma}}{\gamma\sqrt{k}}\right) \tag{A.473}$$

$$<1. \tag{A.474}$$

Then $\left|yg^{(1)}(\mathbf{x};\sigma_\xi^{(2)})\right|<1$. Finally, the statement on $\left|\langle(q_i^{(1)})_{-\mathbf{A}},\phi_{-\mathbf{A}}\rangle\right|$ follows from a similar calculation on $\left|\langle\mathbf{w}_i^{(1)},\mathbf{x}\rangle\right|=\left|\langle q_i^{(1)},\phi\rangle\right|$. □

We are now ready to analyze the gradients in the second gradient step.

Lemma A.6.12. *Fix $\delta\in(0,1)$ and suppose $\mathbf{w}_i^{(0)}\in\mathcal{G}_{\mathbf{w}}(\delta),\mathbf{b}_i^{(0)}\in\mathcal{G}_{\mathbf{b}}(\delta),\mathbf{a}_i^{(0)}\in\mathcal{G}_{\mathbf{a}}(\delta)$ for all $i\in[2m]$. Let $\epsilon_{e2}:=O\left(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_e)}{\tilde{\sigma}^2\sigma_\xi^{(2)}}\right)+\exp(-\Theta(k))$. If $D\mu/\sqrt{d}\le 1/16$, $\sigma_\zeta\tilde{\sigma}=O(1)$, $\sigma_\zeta^2 d/\tilde{\sigma}^2=O(1)$, $k=\Omega(\log^2(Dm/\delta))$, $\sigma_{\mathbf{a}}\le\tilde{\sigma}^2/(\gamma k^2)$, $\eta^{(1)}=O\left(\frac{\gamma}{km\sigma_{\mathbf{a}}\tilde{\sigma}}\right)$, and $\sigma_\xi^{(2)}=1/k^{3/2}$, then*

$$\frac{\partial}{\partial\mathbf{w}_i}L_{\mathcal{D}}(g^{(1)};\sigma_\xi^{(2)})=-\mathbf{a}_i^{(1)}\left(\sum_{j=1}^{D}\mathbf{M}_jT_j+\nu\right) \tag{A.475}$$

where T_j satisfies:

- *if $j\in A$, then $|T_j-\beta\gamma/\tilde{\sigma}|\le O(\epsilon_{e2}/\tilde{\sigma}+\eta^{(1)}/\sigma_\xi^{(2)}+\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_e/(\tilde{\sigma}\sigma_\xi^{(2)}))$, where $\beta\in[\Omega(1),1]$ and depends only on $\mathbf{w}_i^{(0)},\mathbf{b}_i^{(0)}$;*
- *if $j\notin\mathbf{A}$, then $|T_j|\le\frac{1}{\tilde{\sigma}}\exp(-\Omega(k))+O(\sigma_\phi^2\tilde{\sigma}\epsilon_{e2})$;*
- *$|\nu_j|\le O\left(\frac{\eta^{(1)}\sigma_\zeta}{\gamma\sigma_\xi^{(2)}}\right)+\exp(-\Omega(k))$.*

Proof of Lemma A.6.12. Consider one neuron index i and omit the subscript i in the parameters. By Lemma A.6.11, with probability at least $1-(d+D)\exp(-\Omega(k))=1-\exp(-\Omega(k))$ over $(\mathbf{x},y)$, $yg^{(1)}(\mathbf{x};\xi^{(2)})>1$ and furthermore, for any $i\in[2m]$, $\left|\langle \mathbf{w}_i^{(1)},\zeta/\tilde{\sigma}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, $\left|\langle q_i^{(1)},\phi\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, and $\left|\langle (q_i^{(1)})_{-\mathbf{A}},\phi_{-\mathbf{A}}\rangle\right| = O(\eta^{(1)}\tilde{\sigma}/\gamma)$, and for any $j\in[d],\ell\in[D]$, $|\zeta_j|\le O(\sigma_\zeta\sqrt{\log(k)})$ and $|\langle\zeta,D_\ell\rangle|\le O(\sigma_\zeta\sqrt{\log(k)})$. Let $\mathbb{I}_{\mathbf{x}}$ be the indicator of this event.

$$\frac{\partial}{\partial \mathbf{w}} L_{\mathcal{D}}^{\alpha}(g^{(1)};\sigma_\xi^{(2)}) \tag{A.476}$$

$$= -\mathbf{a}^{(1)}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y \mathbb{I}_{\mathbf{x}}\mathbb{E}_{\xi^{(2)}}\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\mathbf{x}\right\} \tag{A.477}$$

$$= -\mathbf{a}^{(1)}\sum_{j=1}^{D}\mathbf{M}_j \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\alpha_y y \mathbb{I}_{\mathbf{x}}\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\phi_j\right\}}_{:=T_j} \tag{A.478}$$

$$- \mathbf{a}^{(1)}\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\frac{\alpha_y y\zeta}{\tilde{\sigma}}\mathbb{I}_{\mathbf{x}}\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\right\}}_{:=\nu}. \tag{A.479}$$

Let $T_{j1}:=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\alpha_y y\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\phi_j\right\}$. We have

$$|T_j - T_{j1}| \tag{A.480}$$

$$= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\alpha_y y(1-\mathbb{I}_{\mathbf{x}})\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\phi_j\right\}\right| \tag{A.481}$$

$$\le \frac{1}{\tilde{\sigma}}\exp(-\Omega(k)). \tag{A.482}$$

Similarly, let $\nu':=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\frac{\alpha_y y\zeta}{\tilde{\sigma}}\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\right\}$. We have

$$\left|\nu-\nu'\right| \tag{A.483}$$

$$= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\frac{\alpha_y y\zeta}{\tilde{\sigma}}(1-\mathbb{I}_{\mathbf{x}})\mathbb{I}[\langle \mathbf{w}^{(1)},\mathbf{x}\rangle+\mathbf{b}^{(1)}+\xi^{(2)}\in(0,1)]\right\}\right| \tag{A.484}$$

$$\le \frac{\sigma_\zeta}{\tilde{\sigma}}\exp(-\Omega(k)). \tag{A.485}$$

So it is sufficient to bound T_{j1} and ν'. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

First, consider $j \in \mathbf{A}$.

$$T_{j1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\alpha_y y \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\phi_j\right\} \tag{A.486}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y \phi_j \Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\} \tag{A.487}$$

where $\iota := \langle \mathbf{w}^{(1)}, \zeta/\tilde{\sigma}\rangle$. Let

$$I_a := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right], \tag{A.488}$$

$$I'_a := \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.489}$$

By the property of the Gaussian $\xi^{(2)}$, that $|\langle \phi_{\mathbf{A}}, q_{\mathbf{A}}\rangle| = O(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_{\mathrm{e}})}{\tilde{\sigma}^2})$, and that $|\iota| = |\langle \mathbf{w}_i^{(1)}, \zeta/\tilde{\sigma}\rangle|, |\langle \phi, q\rangle|, |\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle|$ are all $O(\eta^{(1)}\tilde{\sigma}/\gamma) < O(1/k)$, we have

$$|I_a - I'_a| \tag{A.490}$$

$$\leq \left|\Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \geq 0\right] - \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \geq 0\right]\right| \tag{A.491}$$

$$+ \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \geq 1\right] + \Pr_{\xi^{(2)}}\left[\langle \phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \geq 1\right] \tag{A.492}$$

$$= O\left(\frac{\eta^{(1)}|\mathbf{a}_i^{(0)}|k(\gamma+\epsilon_{\mathrm{e}})}{\tilde{\sigma}^2\sigma_\xi^{(2)}}\right) + \exp(-\Theta(k)) = O(\epsilon_{\mathrm{e}2}). \tag{A.493}$$

This leads to

$$\left|T_{j1} - \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\alpha_y y \phi_j I'_a\right\}\right| \tag{A.494}$$

$$\leq \mathbb{E}_{\phi_{\mathbf{A}}}\left\{|\alpha_y y \phi_j|\left|\mathbb{E}_{\phi_{-\mathbf{A}}}(I_a - I'_a)\right|\right\} \tag{A.495}$$

$$\leq O(\epsilon_{\mathrm{e}2})\mathbb{E}_{\phi_{\mathbf{A}}}\left\{|\alpha_y y \phi_j|\right\} \tag{A.496}$$

$$\leq O(\epsilon_{\mathrm{e}2}/\tilde{\sigma}) \tag{A.497}$$

where the last step is from Lemma A.6.6. Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\alpha_y y\phi_j I'_a\right\} \tag{A.498}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\phi_j\right\}\mathbb{E}_{\phi_{-\mathbf{A}}}[I'_a] \tag{A.499}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\phi_j\right\}\Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.500}$$

By Lemma A.2.13, we have $|\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle+\iota| \leq O(\eta^{(1)}\tilde{\sigma}/\gamma)$. Also, $|b^{(1)}-b^{(0)}| \leq O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}})$. By the property of $\xi^{(2)}$,

$$\left|\Pr_{\xi^{(2)}}\left[\langle\phi_{-\mathbf{A}}, q_{-\mathbf{A}}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right] - \Pr_{\xi^{(2)}}\left[\mathbf{b}^{(0)} + \xi^{(2)} \in (0,1)\right]\right| \tag{A.501}$$

$$\leq O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)})) + O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/\sigma_\xi^{(2)}). \tag{A.502}$$

On the other hand,

$$\beta := \Pr_{\phi_{-\mathbf{A}},\xi^{(2)}}\left[\mathbf{b}^{(0)} + \xi^{(2)} \in (0,1)\right] = \Pr_{\xi^{(2)}}\left[\xi^{(2)} \in (-\mathbf{b}^{(0)}, 1-\mathbf{b}^{(0)})\right] \tag{A.503}$$

$$= \Omega(1) \tag{A.504}$$

and β only depends on $\mathbf{b}^{(0)}$. By Lemma A.6.6, $\mathbb{E}_{\phi_{\mathbf{A}}}\{\alpha_y y\phi_j\} = \gamma/\tilde{\sigma}$. Therefore,

$$|T_{j1} - \beta\gamma/\tilde{\sigma}| \leq O(\epsilon_{\mathrm{e}2}/\tilde{\sigma}) + O(\eta^{(1)}/\sigma_\xi^{(2)}) + O(\eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/(\tilde{\sigma}\sigma_\xi^{(2)})). \tag{A.505}$$

Now, consider $j \notin \mathbf{A}$. Let B denote $\mathbf{A} \cup \{j\}$.

$$T_{j1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\alpha_y y\phi_j \mathbb{I}\left[\langle\phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\} \tag{A.506}$$

$$= \mathbb{E}_{\phi_B}\mathbb{E}_{\phi_{-B},\zeta,\xi^{(2)}}\left\{\alpha_y y\phi_j \mathbb{I}\left[\langle\phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\} \tag{A.507}$$

$$= \mathbb{E}_{\phi_B}\left\{\alpha_y y\phi_j \Pr_{\phi_{-B},\zeta,\xi^{(2)}}\left[\langle\phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]\right\}. \tag{A.508}$$

Let

$$I_b := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right], \tag{A.509}$$

$$I'_b := \Pr_{\xi^{(2)}}\left[\langle \phi_{-B}, q_{-B}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)\right]. \tag{A.510}$$

Similar as above, we have $|I_b - I'_b| \le \epsilon_{\mathrm{e}2}$. Then by Lemma A.6.6,

$$\left|T_{j1} - \mathbb{E}_{\phi_B,\phi_{-B},\zeta}\left\{\alpha_y y \phi_j I'_b\right\}\right| \tag{A.511}$$

$$\le \mathbb{E}_{\phi_B}\left\{|\alpha_y y \phi_j| |\mathbb{E}_{\phi_{-B},\zeta}(I_b - I'_b)|\right\} \tag{A.512}$$

$$\le O(\epsilon_{\mathrm{e}2})\mathbb{E}_{\phi_{\mathbf{A}}}\left\{|\alpha_y y|\right\}\mathbb{E}_{\phi_j}\left\{|\phi_j|\right\} \tag{A.513}$$

$$\le O(\epsilon_{\mathrm{e}2}) \times O(\sigma_\phi^2 \tilde{\sigma}) \tag{A.514}$$

$$= O(\sigma_\phi^2 \tilde{\sigma} \epsilon_{\mathrm{e}2}). \tag{A.515}$$

Furthermore,

$$\mathbb{E}_{\phi_B,\phi_{-B},\zeta}\left\{\alpha_y y \phi_j I'_b\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\right\}\mathbb{E}_{\phi_j}\left\{\phi_j\right\}\mathbb{E}_{\phi_{-B}}[I'_b] = 0. \tag{A.516}$$

Therefore,

$$|T_{j1}| \le O(\sigma_\phi^2 \tilde{\sigma} \epsilon_{\mathrm{e}2}). \tag{A.517}$$

Finally, consider ν'_j.

$$\nu'_j = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}}\mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\right\} \tag{A.518}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}}\mathbb{I}[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\right\} \tag{A.519}$$

$$= \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}}\Pr_{\xi^{(2)}}[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)]\right\} \tag{A.520}$$

Let

$$I_j := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right], \tag{A.521}$$

$$I'_j := \Pr_{\xi^{(2)}}\left[\langle \phi, q\rangle + \mathbf{b}^{(0)} + \xi^{(1)} \in (0,1)\right]. \tag{A.522}$$

Since $|\iota| \le O(\eta^{(1)}\tilde{\sigma}/\gamma)$, we have $|I_j - I'_j| \le O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)}))$. Then

$$\left|\nu'_j - \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}} I'_j\right\}\right| \tag{A.523}$$

$$= \left|\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}}(I_j - I'_j)\right\}\right| \tag{A.524}$$

$$\le O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)}))\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta}\left|\frac{\alpha_y y \zeta_j}{\tilde{\sigma}}\right| \tag{A.525}$$

$$\le O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)}))\mathbb{E}_{\phi_{\mathbf{A}}}|\alpha_y y|\,\mathbb{E}_\zeta\left|\frac{\zeta_j}{\tilde{\sigma}}\right| \tag{A.526}$$

$$\le O(\eta^{(1)}\tilde{\sigma}/(\gamma\sigma_\xi^{(2)})) \times 1 \times \frac{\sigma_\zeta}{\tilde{\sigma}} \tag{A.527}$$

$$\le O(\eta^{(1)}\sigma_\zeta/(\gamma\sigma_\xi^{(2)})). \tag{A.528}$$

Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}},\zeta}\left\{\frac{\alpha_y y \zeta_j}{\tilde{\sigma}} I'_j\right\} = \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\frac{\alpha_y y}{\tilde{\sigma}} I'_j\right\}\mathbb{E}_\zeta\{\zeta_j\} = 0. \tag{A.529}$$

Therefore,

$$|\nu_j| \le O\left(\frac{\eta^{(1)}\sigma_\zeta}{\gamma\sigma_\xi^{(2)}}\right) + \exp(-\Omega(k)). \tag{A.530}$$

□

Lemma A.6.13. *Under the same assumptions as in Lemma A.6.12,*

$$\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) = -\mathbf{a}_i^{(1)} T_b \tag{A.531}$$

where $|T_b| \le \exp(-\Omega(k)) + O(\epsilon_{e2})$.

Proof of Lemma A.6.13. Consider one neuron index i and omit the subscript i in the parameters. By Lemma A.6.11, $\Pr[yg^{(1)}(\mathbf{x};\xi^{(2)}) > 1] \le \exp(-\Omega(k))$. Let $\mathbb{I}_{\mathbf{x}} = \mathbb{I}[yg^{(1)}(\mathbf{x};\xi^{(2)}) \le 1]$.

$$\frac{\partial}{\partial \mathbf{b}} L_{\mathcal{D}}^{\alpha}(g^{(1)}; \sigma_{\xi}^{(2)}) \tag{A.532}$$

$$= -\mathbf{a}^{(1)} \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left\{ \alpha_y y \mathbb{I}_{\mathbf{x}} \mathbb{E}_{\xi^{(2)}} \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\}}_{:=T_b}. \tag{A.533}$$

Let $T_{b1} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ \alpha_y y \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\}$. We have

$$|T_b - T_{b1}| \tag{A.534}$$

$$= \left| \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ \alpha_y y (1 - \mathbb{I}_{\mathbf{x}}) \mathbb{I}[\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1)] \right\} \right| \tag{A.535}$$

$$\le \exp(-\Omega(k)). \tag{A.536}$$

So it is sufficient to bound T_{b1}. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

$$T_{b1} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D},\xi^{(2)}} \left\{ \alpha_y y \mathbb{I}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\} \tag{A.537}$$

$$= \mathbb{E}_{\phi_A} \mathbb{E}_{\phi_{-A},\xi^{(2)}} \left\{ \alpha_y y \mathbb{I}\left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\} \tag{A.538}$$

$$= \mathbb{E}_{\phi_A} \left\{ \alpha_y y \Pr_{\phi_{-A},\xi^{(2)}} \left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right] \right\}. \tag{A.539}$$

Let

$$I_b := \Pr_{\xi^{(2)}} \left[\langle \phi, q\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right], \tag{A.540}$$

$$I_b' := \Pr_{\xi^{(2)}} \left[\langle \phi_{-\mathbf{A}}, q_{-A}\rangle + \iota + \mathbf{b}^{(1)} + \xi^{(2)} \in (0,1) \right]. \tag{A.541}$$

Similar as in Lemma A.2.14, we have $|I_b - I_b'| \leq \epsilon_{\text{e2}}$. Then by Lemma A.6.6,

$$\left|T_{b1} - \mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\alpha_y y I_b'\right\}\right| \tag{A.542}$$

$$= \left|\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\alpha_y y (I_b - I_b')\right\}\right| \tag{A.543}$$

$$= O(\epsilon_{\text{e2}})\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left|\alpha_y y\right| \tag{A.544}$$

$$\leq O(\epsilon_{\text{e2}}). \tag{A.545}$$

Furthermore,

$$\mathbb{E}_{\phi_{\mathbf{A}},\phi_{-\mathbf{A}}}\left\{\alpha_y y I_b'\right\} = \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y\right\}\mathbb{E}_{\phi_{-\mathbf{A}}}[I_b'] = 0. \tag{A.546}$$

Therefore, $|T_{b1}| \leq O(\epsilon_{\text{e2}})$ and the statement follows. □

Lemma A.6.14. *Under the same assumptions as in Lemma A.6.12,*

$$\frac{\partial}{\partial \mathbf{a}_i} L_{\mathcal{D}}(g^{(1)}; \sigma_\xi^{(2)}) = -T_a \tag{A.547}$$

where $|T_a| = O\left(\frac{\eta^{(1)}\tilde{\sigma}}{\gamma p_{\min}}\right) + \exp(-\Omega(k))\text{poly}\left(\frac{dD}{p_{\min}}\right)$.

Proof of Lemma A.6.14. Consider one neuron index i and omit the subscript i in the parameters. By Lemma A.6.11, $\Pr[yg^{(1)}(\mathbf{x}; \xi^{(2)}) > 1] \leq \exp(-\Theta(k))$. Let $\mathbb{I}_{\mathbf{x}} = \mathbb{I}[yg^{(1)}(\mathbf{x}; \xi^{(2)}) \leq 1]$.

$$\frac{\partial}{\partial \mathbf{a}} L_{\mathcal{D}}^{\alpha}(g^{(1)}; \sigma_\xi^{(2)}) \tag{A.548}$$

$$= -\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y \mathbb{I}_{\mathbf{x}} \mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}}_{:=T_a}. \tag{A.549}$$

Let $T_{a1} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y \mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}$. We have

$$|T_a - T_{a1}| \tag{A.550}$$

$$= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left\{\alpha_y y (1 - \mathbb{I}_{\mathbf{x}}) \mathbb{E}_{\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}\right| \tag{A.551}$$

$$\leq \exp(-\Omega(k)). \tag{A.552}$$

So it is sufficient to bound T_{a1}. For simplicity, we use q as a shorthand for $q_i^{(1)}$.

Let $\phi'_{\mathbf{A}}$ be an independent copy of $\phi_{\mathbf{A}}$, ϕ' be the vector obtained by replacing in ϕ the entries $\phi_{\mathbf{A}}$ with $\phi'_{\mathbf{A}}$, and let $x' = \mathbf{M}\phi' + \zeta$ and its label is y'. Then

$$
\begin{aligned}
|T_{a1}| &:= \left|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\alpha_y y \mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})\right\}\right| && \text{(A.553)}\\
&\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(1)})|y=1\right\} && \text{(A.554)}\\
&\quad - \mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y=-1\right\}\Bigg| && \text{(A.555)}\\
&\le \frac{1}{2}\Bigg|\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y=1\right\} && \text{(A.556)}\\
&\quad - \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}},\zeta,\xi^{(2)}}\sigma(\langle \mathbf{w}^{(1)}, \mathbf{x}'\rangle + \mathbf{b}^{(1)} + \xi^{(2)})|y'=-1\right\}\Bigg| && \text{(A.557)}\\
&\le \frac{1}{2}\mathbb{E}_{\phi_{\mathbf{A}},\phi'_{\mathbf{A}}}\left\{\mathbb{E}_{\phi_{-\mathbf{A}}}\left|\langle \mathbf{w}^{(1)}, \mathbf{x}\rangle - \langle \mathbf{w}^{(1)}, \mathbf{x}'\rangle\right| |y=1, y'=-1\right\} && \text{(A.558)}\\
&\le \frac{1}{2}\mathbb{E}_{\phi_{-\mathbf{A}}}\left(\mathbb{E}_{\phi_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(1)}, \mathbf{M}\phi\rangle\right| |y=1\right\} + \mathbb{E}_{\phi'_{\mathbf{A}}}\left\{\left|\langle \mathbf{w}^{(1)}, \mathbf{M}\phi'\rangle\right| |y'=-1\right\}\right) && \text{(A.559)}\\
&\le \mathbb{E}_{\phi_{-\mathbf{A}},\phi_{\mathbf{A}}}\left|\alpha_y\langle \mathbf{w}^{(1)}, \mathbf{M}\phi\rangle\right| && \text{(A.560)}\\
&= \mathbb{E}_{\phi}\left|\alpha_y\langle \mathbf{w}^{(1)}, \mathbf{M}\phi\rangle\right| && \text{(A.561)}\\
&= O(\eta^{(1)}\tilde{\sigma}/\gamma) + \exp(-\Omega(k)) \times \frac{\sqrt{dD}}{\tilde{\sigma}} \times \|\mathbf{w}^{(1)}\|_\infty && \text{(A.562)}\\
&= O\left(\frac{\eta^{(1)}\tilde{\sigma}}{\gamma p_{\min}}\right) + \exp(-\Omega(k))\mathrm{poly}(dD/p_{\min}) && \text{(A.563)}
\end{aligned}
$$

where the fourth step follows from that σ is 1-Lipschitz, and the second to the last line from Lemma A.6.11 and that $\left|\langle \mathbf{w}^{(1)}, \mathbf{M}\phi\rangle\right| \le \|\mathbf{w}^{(1)}\|_\infty\sqrt{d\|\mathbf{M}\phi\|_2^2}$. □

With the above lemmas about the gradients, we are now ready to show that at the end of the second step, we get a good set of features for accurate prediction.

Lemma A.6.15. *Set*

$$
\begin{aligned}
&\eta^{(1)} = \frac{\gamma^2 p_{\min}\tilde{\sigma}}{km^3}, \lambda_{\mathbf{a}}^{(1)} = 0, \lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \sigma_\xi^{(1)} = 1/k^{3/2}, && \text{(A.564)}\\
&\eta^{(2)} = 1, \lambda_{\mathbf{a}}^{(2)} = \lambda_{\mathbf{w}}^{(2)} = 1/(2\eta^{(2)}), \sigma_\xi^{(2)} = 1/k^{3/2}. && \text{(A.565)}
\end{aligned}
$$

Fix $\delta \in (0, O(1/k^3))$. If $D\mu/\sqrt{d} \le 1/16$, $\sigma_\zeta \tilde{\sigma} = O(1)$, $\sigma_\zeta^2 d/\tilde{\sigma}^2 = O(1)$, $k = \Omega\left(\log^2\left(\frac{Dmd}{\delta\gamma p_{\min}}\right)\right)$, and $m \ge \max\{\Omega(k^4), D, d\}$, then with probability at least $1-\delta$ over the initialization, there exist $\tilde{\mathbf{a}}_i$'s such that $\tilde{g}(\mathbf{x}) := \sum_{i=1}^{2m} \tilde{\mathbf{a}}_i \sigma(\langle \mathbf{w}_i^{(2)}, \mathbf{x}\rangle + \mathbf{b}_i^{(2)})$ satisfies $L_{\mathcal{D}}(\tilde{g}) \le \exp(-\Omega(k))$. Furthermore, $\|\tilde{\mathbf{a}}\|_0 = O(m/k)$, $\|\tilde{\mathbf{a}}\|_\infty = O(k^5/m)$, and $\|\tilde{\mathbf{a}}\|_2^2 = O(k^9/m)$. Finally, $\|\mathbf{a}^{(2)}\|_\infty = O\left(\frac{1}{km^2}\right)$, $\|\mathbf{w}_i^{(2)}\|_2 = O(\tilde{\sigma}/k)$, and $|\mathbf{b}_i^{(2)}| = O(1/k^2)$ for all $i \in [2m]$.

Proof of Lemma A.6.15. By Lemma A.6.2, there exists a network $g^*(\mathbf{x}) = \sum_{\ell=1}^{3(k+1)} \mathbf{a}_\ell^* \sigma(\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*)$ satisfying

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}[yg^*(x) \le 1] \le \exp(-\Omega(k)).$$

Furthermore, the number of neurons $n = 3(k+1)$, $|\mathbf{a}_i^*| \le 64k, 1/(64k) \le |\mathbf{b}_i^*| \le 1/4$, $\mathbf{w}_i^* = \tilde{\sigma}\sum_{j\in\mathbf{A}} \mathbf{M}_j/(8k)$, and $|\langle \mathbf{w}_i^*, \mathbf{x}\rangle + \mathbf{b}_i^*| \le 1$ for any $i \in [n]$ and $(\mathbf{x}, y) \sim \mathcal{D}$. Now we fix an ℓ, and show that with high probability there is a neuron in $g^{(2)}$ that can approximate the ℓ-th neuron in g^*.

With probability $\ge 1 - \exp(-\Omega(\max\{2p_{\mathrm{o}}(D-k), k\}))$, among all $j \notin \mathbf{A}$, we have that at most $2p_{\mathrm{o}}(D-k)+k$ of ϕ_j are $(1-p_{\mathrm{o}})/\tilde{\sigma}$, while the others are $-p_{\mathrm{o}}/\tilde{\sigma}$. With probability $\ge 1-(d+D)\exp(-\Omega(k))$ over ζ, for any j, $|\zeta_j| \le O(\sigma_\zeta\sqrt{\log(k)})$ and $|\langle \zeta, D_\ell\rangle| \le O(\sigma_\zeta\sqrt{\log(k)})$. Below we consider data points with ϕ and ζ satisfying these.

By Lemma A.6.4, with probability $1-2\delta$ over $\mathbf{w}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{w}}(\delta)$; with probability $1-\delta$ over $\mathbf{a}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{a}}(\delta)$; with probability $1-\delta$ over $\mathbf{b}_i^{(0)}$'s, they are all in $\mathcal{G}_{\mathbf{b}}(\delta)$. Under these events, by Lemma A.6.10, Lemma A.6.12 and A.6.13, for any neuron $i \in [2m]$, we have

$$\mathbf{w}_i^{(2)} = \mathbf{a}_i^{(1)}\left(\sum_{j=1}^{D} \mathbf{M}_j T_j + \nu\right), \tag{A.566}$$

$$\mathbf{b}_i^{(2)} = \mathbf{b}_i^{(1)} + \mathbf{a}_i^{(1)} T_b. \tag{A.567}$$

where

- if $j \in A$, then $|T_j - \beta\gamma/\tilde{\sigma}| \le \epsilon_{\mathbf{w}1} := O(\epsilon_{\mathrm{e}2}/\tilde{\sigma} + \eta^{(1)}/\sigma_\xi^{(2)} + \eta^{(1)}|\mathbf{a}_i^{(0)}|\epsilon_{\mathrm{e}}/(\tilde{\sigma}\sigma_\xi^{(2)}))$, where $\beta \in [\Omega(1), 1]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$;

- if $j \notin \mathbf{A}$, then $|T_j| \le \epsilon_{\mathbf{w}2} := \frac{1}{\tilde{\sigma}} \exp(-\Omega(k)) + O(\sigma_\phi^2 \tilde{\sigma} \epsilon_{\mathrm{e}2})$;
- $|\nu_j| \le \epsilon_\nu := O\left(\frac{\eta^{(1)} \sigma_\zeta}{\gamma \sigma_\xi^{(2)}}\right) + \exp(-\Omega(k))$.
- $|T_b| \le \epsilon_b := \exp(-\Omega(k)) + O(\epsilon_{\mathrm{e}2})$.

Given the initialization, with probability $\Omega(1)$ over $\mathbf{b}_i^{(0)}$, we have

$$|\mathbf{b}_i^{(0)}| \in \left[\frac{1}{2k^2}, \frac{2}{k^2}\right], \mathrm{sign}(\mathbf{b}_i^{(0)}) = \mathrm{sign}(\mathbf{b}_\ell^*). \tag{A.568}$$

Finally, since $\frac{8k|\mathbf{b}_\ell^*|\beta\gamma}{|\mathbf{b}_i^{(0)}|\tilde{\sigma}^2} \in [\Omega(k^2\gamma/\tilde{\sigma}^2), O(k^3\gamma/\tilde{\sigma}^2)]$ and depends only on $\mathbf{w}_i^{(0)}, \mathbf{b}_i^{(0)}$, we have that for $\epsilon_{\mathbf{a}} = \Theta(1/k^2)$, with probability $\Omega(\epsilon_{\mathbf{a}}) > \delta$ over $\mathbf{a}_i^{(0)}$,

$$\left|\frac{8k|\mathbf{b}_\ell^*|\beta\gamma}{|\mathbf{b}_i^{(0)}|\tilde{\sigma}^2}\mathbf{a}_i^{(0)} - 1\right| \le \epsilon_{\mathbf{a}}, \quad |\mathbf{a}_i^{(0)}| = O\left(\frac{\tilde{\sigma}^2}{k^2\gamma}\right). \tag{A.569}$$

Let $n_a = \epsilon_{\mathbf{a}} m/4$. For the given value of m, by (A.566)-(A.569) we have with probability $\ge 1 - 5\delta$ over the initialization, for each ℓ there is a different set of neurons $I_\ell \subseteq [m]$ with $|I_\ell| = n_a$ and such that for each $i_\ell \in I_\ell$,

$$|\mathbf{b}_{i_\ell}^{(0)}| \in \left[\frac{1}{2k^2}, \frac{2}{k^2}\right], \quad \mathrm{sign}(\mathbf{b}_{i_\ell}^{(0)}) = \mathrm{sign}(\mathbf{b}_\ell^*), \tag{A.570}$$

$$\left|\frac{8k|\mathbf{b}_\ell^*|\beta\gamma}{|b_{i_\ell}^{(0)}|\tilde{\sigma}^2}\mathbf{a}_{i_\ell}^{(0)} - 1\right| \le \epsilon_{\mathbf{a}}, \quad |\mathbf{a}_{i_\ell}^{(0)}| = O\left(\frac{\tilde{\sigma}^2}{k^2\gamma}\right). \tag{A.571}$$

Now, construct $\tilde{\mathbf{a}}$ such that $\tilde{\mathbf{a}}_{i_\ell} = \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a}$ for each ℓ and each $i_\ell \in I_\ell$, and $\tilde{\mathbf{a}}_i = 0$ elsewhere. To show that it gives accurate predictions, we first consider bounding some error terms.

For the given values of parameters, we have

$$\epsilon_{\text{e}2} = O\left(\frac{\gamma}{m^2}\right), \tag{A.572}$$

$$\epsilon_{\mathbf{w}1} = O\left(\frac{k\gamma}{m^2\tilde{\sigma}} + \frac{\gamma\epsilon_{\text{e}}}{km^2}\right), \tag{A.573}$$

$$\epsilon_{\mathbf{w}2} = O\left(\frac{\gamma}{m^2\tilde{\sigma}}\right), \tag{A.574}$$

$$\epsilon_{\nu} = O\left(\frac{\gamma k}{m^3}\right), \tag{A.575}$$

$$\epsilon_{b} = O\left(\frac{\gamma}{m^2}\right). \tag{A.576}$$

We also have the following useful claims.

Claim A.6.2. $\sum_{\ell\in\mathbf{A}} |\langle \mathbf{M}_\ell, \mathbf{x}\rangle| \leq O\left(\frac{k}{\tilde{\sigma}}\right)$.

Proof of Claim A.6.2.

$$\sum_{\ell\in\mathbf{A}} |\langle \mathbf{M}_\ell, \mathbf{x}\rangle| \tag{A.577}$$

$$\leq \sum_{\ell\in\mathbf{A}} \left(|\phi_j| + \left| \sum_{j\neq\ell} \mathbf{M}_\ell^\top \mathbf{M}_j \phi_j \right| + \left| \mathbf{M}_\ell^\top \zeta/\tilde{\sigma} \right| \right) \tag{A.578}$$

$$\leq O\left(\frac{k}{\tilde{\sigma}}\right) + O\left(kD\frac{\mu}{\sqrt{d}\tilde{\sigma}}\right) + O\left(k\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\right) \tag{A.579}$$

$$\leq O\left(\frac{k}{\tilde{\sigma}}\right). \tag{A.580}$$

□

Claim A.6.3.

$$\left| \langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle \right| \leq O\left(\frac{1}{m}\right). \tag{A.581}$$

Proof of Claim A.6.3.

$$\left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle\right| = \left|\langle \sum_{\ell=1}^{D} \mathbf{a}_{i_\ell}^{(1)} T_\ell \mathbf{M}_\ell + \upsilon, \mathbf{x}\rangle\right| \tag{A.582}$$

$$\leq \left|\langle \sum_{\ell\in\mathbf{A}} \mathbf{a}_{i_\ell}^{(1)} T_\ell \mathbf{M}_\ell, \mathbf{x}\rangle\right| + \left|\langle \sum_{\ell\notin\mathbf{A}} \mathbf{a}_{i_\ell}^{(1)} T_\ell \mathbf{M}_\ell, \mathbf{x}\rangle\right| + |\langle \upsilon, \mathbf{x}\rangle| . \tag{A.583}$$

Then

$$\left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.584}$$

$$\leq \left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| + \left|\frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.585}$$

$$\leq \left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| + \left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| \left|\frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| . \tag{A.586}$$

The first term is

$$\left|\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(1)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \tag{A.587}$$

$$\leq \left|\mathbf{a}_{i_\ell}^{(1)}\right| \left(\left|\langle \sum_{\ell\in\mathbf{A}} \left(T_\ell - \frac{\beta\gamma}{\tilde{\sigma}}\right) \mathbf{M}_\ell, \mathbf{x}\rangle\right| + \left|\langle \sum_{\ell\notin\mathbf{A}} T_\ell \mathbf{M}_\ell, \mathbf{x}\rangle\right| + |\langle \nu, \mathbf{x}\rangle| \right) . \tag{A.588}$$

By Claim A.6.2,

$$\left|\langle \sum_{\ell\in\mathbf{A}} \left(T_\ell - \frac{\beta\gamma}{\tilde{\sigma}}\right) \mathbf{M}_\ell, \mathbf{x}\rangle\right| \leq \sum_{\ell\in\mathbf{A}} \left|T_\ell - \frac{\beta\gamma}{\tilde{\sigma}}\right| |\langle \mathbf{M}_\ell, \mathbf{x}\rangle| \tag{A.589}$$

$$\leq \epsilon_{w1} \sum_{\ell\in\mathbf{A}} |\langle \mathbf{M}_\ell, \mathbf{x}\rangle| \tag{A.590}$$

$$\leq O\left(\frac{k\epsilon_{\mathbf{w}1}}{\tilde{\sigma}}\right) . \tag{A.591}$$

$$\left|\langle\sum_{\ell\notin\mathbf{A}} T_\ell\mathbf{M}_\ell,\mathbf{x}\rangle\right| \le \left|\sum_{\ell\notin\mathbf{A}} T_\ell\phi_j\right| + \left|\sum_{\ell\notin\mathbf{A},j\neq\ell} T_\ell\mathbf{M}_\ell^\top\mathbf{M}_j\phi_j\right| + \left|\sum_{\ell\notin\mathbf{A}} T_\ell\mathbf{M}_\ell^\top\zeta/\tilde{\sigma}\right| \tag{A.592}$$

$$\le O\left(\frac{D\epsilon_{\mathbf{w}2}}{\tilde{\sigma}}\right) + O\left(D^2\epsilon_{\mathbf{w}2}\frac{\mu}{\sqrt{d}\tilde{\sigma}}\right) + O\left(D\epsilon_{\mathbf{w}2}\frac{\sigma_\zeta\sqrt{\log(k)}}{\tilde{\sigma}}\right) \tag{A.593}$$

$$\le O\left(\frac{D\epsilon_{\mathbf{w}2}}{\tilde{\sigma}}\right). \tag{A.594}$$

$$|\langle\nu,\mathbf{x}\rangle| \le \left|\langle\nu,\sum_{\ell\in[D]}\mathbf{M}_\ell\phi_\ell\rangle\right| + |\langle\nu,\zeta/\tilde{\sigma}\rangle| \tag{A.595}$$

$$\le \sum_{\ell\in[D]}|\phi_\ell\langle\nu,\mathbf{M}_\ell\rangle| + |\langle\nu,\zeta/\tilde{\sigma}\rangle| \tag{A.596}$$

$$\le O\left(\frac{p_{\mathrm{o}}D+k}{\tilde{\sigma}}\right)\epsilon_\nu\sqrt{d} + d\epsilon_\nu\frac{O(\sigma_\zeta\sqrt{\log(k)})}{\tilde{\sigma}} \tag{A.597}$$

$$\le O\left(\frac{\epsilon_\nu\sqrt{d}}{\tilde{\sigma}}\right)\left(p_{\mathrm{o}}D + \sqrt{d}\epsilon_\nu\sqrt{\log(k)}\right) \tag{A.598}$$

$$\le O\left(\frac{\epsilon_\nu\sqrt{d}}{\tilde{\sigma}}\right)\left(p_{\mathrm{o}}D + \tilde{\sigma}\sqrt{\log(k)}\right) \tag{A.599}$$

$$\le O\left(\frac{p_{\mathrm{o}}D\epsilon_\nu\sqrt{d}}{\tilde{\sigma}}\right). \tag{A.600}$$

Then by (A.572)-(A.576),

$$\epsilon_\phi := \left(\frac{k\epsilon_{\mathbf{w}1}}{\tilde{\sigma}} + \frac{D\epsilon_{\mathbf{w}2}}{\tilde{\sigma}} + \frac{p_{\mathrm{o}}D\epsilon_\nu\sqrt{d}}{\tilde{\sigma}}\right) = O\left(\frac{k^2\gamma}{m^2\tilde{\sigma}^2} + \frac{\gamma\epsilon_{\mathrm{e}}}{m^2\tilde{\sigma}} + \frac{\gamma}{m\tilde{\sigma}^2} + \frac{\gamma k}{m^{3/2}\tilde{\sigma}}\right). \tag{A.601}$$

We have $\left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| = O(\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)})$. So the first term is bounded by

$$\left|\langle\mathbf{w}_{i_\ell}^{(2)},\mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}}\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j,\mathbf{x}\rangle\right| \le \left|\mathbf{a}_{i_\ell}^{(1)}\right|\epsilon_\phi \tag{A.602}$$

$$\le O\left(\frac{\tilde{\sigma}^2}{k^2\gamma} + \eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\right)\left(\frac{k^2\gamma}{m^2\tilde{\sigma}^2} + \frac{\gamma\epsilon_{\mathrm{e}}}{m^2\tilde{\sigma}} + \frac{\gamma}{m\tilde{\sigma}^2} + \frac{\gamma k}{m^{3/2}\tilde{\sigma}}\right) \le O\left(\frac{1}{m}\right). \tag{A.603}$$

By Claim A.6.2, the second term is bounded by

$$\left|\mathbf{a}_{i_\ell}^{(1)} - \mathbf{a}_{i_\ell}^{(0)}\right| \left|\frac{\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle\right| \leq O\left(\frac{k\eta^{(1)}\sigma_{\mathbf{w}}\sqrt{\log(Dm/\delta)}\gamma}{\tilde{\sigma}^2}\right) \leq O\left(\frac{\gamma^3}{m^3}\right). \tag{A.604}$$

Combining the bounds on the two terms leads to the claim. □

Claim A.6.4.

$$|\mathbf{b}_{i_\ell}^{(2)} - \mathbf{b}_{i_\ell}^{(0)}| \leq O\left(\frac{1}{k^2 m}\right). \tag{A.605}$$

Proof of Claim A.6.4. By Lemma A.6.10 and A.6.13:

$$|\mathbf{b}_{i_\ell}^{(2)} - \mathbf{b}_{i_\ell}^{(0)}| \leq |\mathbf{b}_{i_\ell}^{(2)} - \mathbf{b}_{i_\ell}^{(1)}| + |\mathbf{b}_{i_\ell}^{(1)} - \mathbf{b}_{i_\ell}^{(0)}| \tag{A.606}$$

$$\leq O\left(\eta^{(1)}|\mathbf{a}_{i_\ell}^{(0)}|\epsilon_{\mathrm{e}} + |\mathbf{a}_{i_\ell}^{(1)}|\left(\exp(-\Omega(k)) + \epsilon_{\mathrm{e}2}\right)\right) \tag{A.607}$$

$$\leq O\left(\frac{\gamma}{km^2} + \frac{1}{k^2 m}\right) \leq O\left(\frac{1}{k^2 m}\right). \tag{A.608}$$

□

We are now ready to show $\tilde{g}$ is close to $2g^*$.

$$|\tilde{g}(\mathbf{x}) - 2g^*(\mathbf{x})| \tag{A.609}$$

$$= \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \tilde{\mathbf{a}}_{i_\ell} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \sum_{\ell=1}^{3(k+1)} 2\mathbf{a}_\ell^* \sigma\left(\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*\right) \right| \tag{A.610}$$

$$= \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|n_a} \sigma\left(\frac{|b_{i_\ell}^{(0)}|}{|\mathbf{b}_\ell^*|}\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + b_{i_\ell}^{(0)}\right) \right| \tag{A.611}$$

$$\leq \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{1}{n_a} \left(\frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(2)}\right) - \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(0)} \right) \right) \right| \tag{A.612}$$

$$+ \left| \sum_{\ell=1}^{3(k+1)} \sum_{i_\ell \in I_\ell} \frac{1}{n_a} \left(\frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle + \mathbf{b}_{i_\ell}^{(0)} \right) - \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|b_{i_\ell}^{(0)}|} \sigma\left(\frac{|b_{i_\ell}^{(0)}|}{|\mathbf{b}_\ell^*|}\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + b_{i_\ell}^{(0)} \right) \right) \right|. \tag{A.613}$$

Here the second equation follows from that σ is positive-homogeneous in $[0,1]$, $|\langle \mathbf{w}_\ell^*, \mathbf{x}\rangle + \mathbf{b}_\ell^*| \leq 1$, $|\mathbf{b}_{i_\ell}^{(0)}|/|\mathbf{b}_\ell^*| \leq 1$.

By Claim A.6.3 and A.6.4, the first term is bounded by:

$$3(k+1)\max_\ell \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|\mathbf{b}_{i_\ell}^{(0)}|} \left(\left| \langle \mathbf{w}_{i_\ell}^{(2)}, \mathbf{x}\rangle - \frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}} \sum_{j\in\mathbf{A}} \langle \mathbf{M}_j, \mathbf{x}\rangle \right| + |\mathbf{b}_{i_\ell}^{(2)} - \mathbf{b}_{i_\ell}^{(0)}| \right) \tag{A.614}$$

$$\leq 3(k+1)\max_\ell \frac{2\mathbf{a}_\ell^*|\mathbf{b}_\ell^*|}{|\mathbf{b}_{i_\ell}^{(0)}|} O\left(\frac{1}{m}\right) \tag{A.615}$$

$$\leq O\left(\frac{k^4}{m}\right). \tag{A.616}$$

By Claim A.6.2, the second term is bounded by:

$$3(k+1)\max_{\ell}\frac{2\mathbf{a}_{\ell}^{*}|\mathbf{b}_{\ell}^{*}|}{|\mathbf{b}_{i_\ell}^{(0)}|}\left|\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}}\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j,\mathbf{x}\rangle-\frac{|b_{i_\ell}^{(0)}|}{|\mathbf{b}_{\ell}^{*}|}\langle\mathbf{w}_{\ell}^{*},\mathbf{x}\rangle\right| \tag{A.617}$$

$$\leq 3(k+1)\max_{\ell}\frac{2\mathbf{a}_{\ell}^{*}|\mathbf{b}_{\ell}^{*}|}{|\mathbf{b}_{i_\ell}^{(0)}|}\left|\frac{\mathbf{a}_{i_\ell}^{(0)}\beta\gamma}{\tilde{\sigma}}\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j,\mathbf{x}\rangle-\frac{|b_{i_\ell}^{(0)}|\tilde{\sigma}}{8k|\mathbf{b}_{\ell}^{*}|}\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j,\mathbf{x}\rangle\right| \tag{A.618}$$

$$\leq 3(k+1)\max_{\ell}\frac{2\mathbf{a}_{\ell}^{*}|\mathbf{b}_{\ell}^{*}|}{|\mathbf{b}_{i_\ell}^{(0)}|}\frac{\tilde{\sigma}|\mathbf{b}_{i_\ell}^{(0)}|}{8k|\mathbf{b}_{\ell}^{*}|}\left|\frac{8k\mathbf{a}_{i_\ell}\beta\gamma|\mathbf{b}_{\ell}^{*}|}{\tilde{\sigma}^2|\mathbf{b}_{i_\ell}^{(0)}|}-1\right|\left|\sum_{j\in\mathbf{A}}\langle\mathbf{M}_j,\mathbf{x}\rangle\right| \tag{A.619}$$

$$\leq 3(k+1)\max_{\ell}O(\mathbf{a}_{\ell}^{*}\epsilon_{\mathbf{a}}) \tag{A.620}$$

$$\leq O\left(k^2\epsilon_{\mathbf{a}}\right). \tag{A.621}$$

Then

$$|\tilde{g}(\mathbf{x})-2g^{*}(\mathbf{x})|=O\left(\frac{k^4}{m}+k^2\epsilon_{\mathbf{a}}\right)\leq 1. \tag{A.622}$$

This guarantees $y\tilde{g}(\mathbf{x})\geq 1$. Changing the scaling of δ leads to the statement.

Finally, the bounds on $\tilde{\mathbf{a}}$ follow from the above calculation. The bound on $\|\mathbf{a}^{(2)}\|_2$ follows from Lemma A.6.14, and those on $\|\mathbf{w}_i^{(2)}\|_2$ and $\|\mathbf{b}_i^{(2)}\|_2$ follow from (A.566)(A.567) and the bounds on $\mathbf{a}_i^{(1)}$ and $\mathbf{b}_i^{(1)}$ in Lemma A.6.10. □

A.6.9 Classifier Learning Stage and Main Theorem

Once we have a good set of features in Lemma A.6.15, we can follow exactly the same argument as in Section A.2.6 and A.2.7 for the simplified setting, and arrive at the main theorem for the general setting:

Theorem A.6.16 (Restatement of Theorem A.6.1)**.** *Set*

$$\eta^{(1)} = \frac{\gamma^2 p_{\min}\tilde{\sigma}}{km^3}, \lambda_{\mathbf{a}}^{(1)} = 0, \lambda_{\mathbf{w}}^{(1)} = 1/(2\eta^{(1)}), \sigma_{\xi}^{(1)} = 1/k^{3/2}, \tag{A.623}$$

$$\eta^{(2)} = 1, \lambda_{\mathbf{a}}^{(2)} = \lambda_{\mathbf{w}}^{(2)} = 1/(2\eta^{(2)}), \sigma_{\xi}^{(2)} = 1/k^{3/2}, \tag{A.624}$$

$$\eta^{(t)} = \eta = \frac{k^2}{Tm^{1/3}}, \lambda_{\mathbf{a}}^{(t)} = \lambda_{\mathbf{w}}^{(t)} = \lambda \leq \frac{k^3}{\tilde{\sigma}m^{1/3}}, \sigma_{\xi}^{(t)} = 0, \text{ for } 2 < t \leq T. \tag{A.625}$$

For any $\delta \in (0, O(1/k^3))$*, if* $\mu \leq O(\sqrt{d}/D)$*,* $\sigma_\zeta \leq O(\min\{1/\tilde{\sigma}, \tilde{\sigma}/\sqrt{d}\})$*,* $k = \Omega\left(\log^2\left(\frac{Dmd}{\delta\gamma p_{\min}}\right)\right)$*,* $m \geq \max\{\Omega(k^4), D, d\}$*, then we have for any* $\mathcal{D} \in \mathcal{F}_\Xi$*, with probability at least* $1-\delta$*, there exists* $t \in [T]$ *such that*

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \neq y] \leq L_{\mathcal{D}}(g^{(t)}) = O\left(\frac{k^8}{m^{2/3}} + \frac{k^3 T}{m^2} + \frac{k^2 m^{2/3}}{T}\right). \tag{A.626}$$

Consequently, for any $\epsilon \in (0,1)$*, if* $T = m^{4/3}$*, and* $m \geq \max\{\Omega(k^{12}/\epsilon^{3/2}), D\}$*, then*

$$\Pr[\text{sign}(g^{(t)}(\mathbf{x})) \neq y] \leq L_{\mathcal{D}}(g^{(t)}) \leq \epsilon. \tag{A.627}$$

Appendix B

Discussions, Complete Proofs and Additional Experiments in Chapter 3: A Theoretical Framework Towards Provable Guarantees for Neural Networks via Gradient Feature Learning

Appendix B.1 describes the limitations of our work. In Appendix B.2, we present our framework implications about simplicity bias. The complete proof of our main results is given in Appendix B.3. We present the case study of linear data in Appendix B.4.1, mixtures of Gaussians in Appendix B.4.2 and Appendix B.4.3, parity functions in Appendix B.4.4, Appendix B.4.5 and Appendix B.4.6, and multiple-index models in Appendix B.4.7. We put the auxiliary lemmas in Appendix B.5.

B.1 Limitations

Recover Existing Results. The framework may or may not recover the width or sample complexity bounds in existing work.

1. The framework can give matching bounds as the existing work in some cases, like parities over uniform inputs (Appendix B.4.5).
2. In some other cases, it gives polynomial error bounds not the same as those in the existing work (e.g., for parities over structured inputs). This is because our work is analyzing general cases, and thus may not give better than or the same bounds as those in special cases, since special cases have more properties that can be exploited to get potentially better bounds. On the other hand, our bounds can already show the advantage over kernel methods (e.g., Proposition 3.3.9).

We would like to emphasize that our contribution is providing an analysis framework that can (1) formalize the unifying principles of learning features from gradients in network training, and (2) give polynomial error bounds for prototypical problems. Our focus is not to recover the guarantees in existing work.

Failure Cases. There are some failure cases that gradient feature learning framework cannot cover:

1. In [232], they constructed a function that is easy to approximate using a 3-layer network but not approximable by any 2-layer network. Since the function is not approximable by any 2-layer network, it cannot be approximated by the gradient-induced networks as well, so OPT will be large. As a result, the final error will be large.
2. In uniform parity data distribution, considering an odd number of features rather than even, i.e., k is an odd number in Assumption B.4.28, we can show that our gradient feature set is empty even when p in Equation (3.6) is exponentially small, thus the OPT is a positive constant since the gradient induced network can only be constants. Meanwhile, the neural network won't be able to learn this data distribution

because its gradient is always 0 through the training, and the final error equals OPT. The first case corresponds to the approximation hardness of 2-layer networks, while the second case gives a learning hardness example. The above two cases show that if there is an approximation or learning hardness, our gradient feature learning framework may be vacuous because the optimal model in the gradient feature class has a large risk, then the ground-truth mapping from inputs to labels is not learnable by gradient descent. These analyses are consistent with previous works [232, 26].

B.2 More Further Implications

Our general framework also sheds some light on several interesting phenomena in neural network (NN) learning observed in practice. Feature learning beyond the kernel regime has been discussed in Section 3.3.1 and Section 3.3.2. The lottery ticket hypothesis (LTH) has been discussed in Section 3.4. Below we discuss other implications.

Implicit Regularization/Simplicity Bias. It is now well known that practical NN are overparameterized and traditional uniform convergence bounds cannot adequately explain their generalization performance [312, 195, 133]. It is generally believed that the optimization has some *implicit regularization* effect that restricts learning dynamics to a subset of the whole hypothesis class, which is not of high capacity so can lead to good generalization [199, 106]. Furthermore, learning dynamics tend to first learn simple functions and then learn more and more sophisticated ones (referred to as simplicity bias) [197, 235]. However, it remains elusive to formalize such simplicity bias.

Our framework provides a candidate explanation: the learning dynamics first learn to approximate the best network in a smaller family of gradient feature induced networks $\mathcal{F}_{d,r,B_F,S}$ and then learn to approximate the best in a larger family. Consider the number of neurons r for illustration. Let $r_1 \ll r_2$, and let T_1 and T_2 be their corresponding runtime bounds for T in the main Theorem 3.2.12. Clearly, $T_1 \ll T_2$. Then, at time T_1, the theorem guarantees the learning dynamics learn to approximate the best in the family

$\mathcal{F}_{d,r_1,B_F,S}$ with r_1 neurons, but not for the larger family $\mathcal{F}_{d,r_2,B_F,S}$. Later, at time T_2, the learning dynamics learn to approximate the best in the larger family $\mathcal{F}_{d,r_2,B_F,S}$. That is, the learning first learns simpler functions and then more sophisticated ones where the simplicity bias is measured by the size of the family of gradient feature-induced networks. The implicit regularization is then restricting to networks approximating smaller families of gradient feature-induced networks. Furthermore, we can also conclude that for an SGD-optimized NN, its actual representation power is from the subset of NN based on gradient features, instead of the whole set of NN. This view helps explain the simplicity bias/implicit regularization phenomenon of NN learning in practice.

Learning over Different Data Distributions. Our framework articulates the following key principles (pointed out for specific problems in existing work but not articulated more generally):

- Role of gradient: the gradient leads to the emergence of good features, which is useful for the learning of upper layers in later stages.
- From features to solutions: learned features in early steps will not be distorted, if not improved, in later stages. The training dynamic for upper layers will eventually learn a good combination of hidden neurons based on gradient features, giving a good solution.

Then, more interesting insights are obtained from the generality of the framework. To build a general framework, the meaningful error guarantees should be data-dependent, since NN learning on general data distributions is hard and data-independent guarantees will be vacuous [66, 65]. Comparing the optimal in a family of "ground-truth" functions (inspired by agnostic learning in learning theory) is a useful method to obtain the data-dependent bound. We further construct the "ground-truth" functions using properties of the training dynamics, i.e., gradient features. This greatly facilitates the analysis of the training dynamics and is the key to obtaining the final guarantees. On the other hand, the framework can also be viewed as using the optimal by gradient-induced NN to measure or quantify the "complexity" of the problem. For easier problems, this quantity is smaller, and our framework can give a

better error bound. So this provides a united way to derive guarantees for specific problems.

New Perspectives about Roadmaps Forward. We argue a new perspective about the connection between the strong representation power and the successful learning of NN. Traditionally, the strong representation power of NN is the key reason for hardness results of NN learning: NN has strong representation power and can encode hard learning questions, so they are hard to learn. See the proof in SQ bound from [63] or NP-hardness from [34]. The strong representation power also causes trouble for the statistical aspect: it leads to vacuous generalization bounds when traditional uniform convergence tools are used.

Our framework suggests a perspective in sharp contrast: the strong representation power of NN with gradient features is actually the key to successful learning. More concretely, the optimal error of the gradient feature-induced NN being small (i.e., strong representation power for a given data distribution) can lead to a small guarantee, which is the key to successful learning. The above new perspective suggests a different analysis road than traditional ones. Traditional analysis typically first reasons about the optimal based on the whole function class, i.e. the ground truth, then analyze how NN learns proper features and reaches the optimal. In contrast, our framework defines feature family first, and then reasons about the optimal based on it.

Our framework provides the foundation for future work on analyzing gradient-based NN learning, which may inspire future directions including but not limited to (1) defining a new feature family for 2-layer NN rather than gradient feature, (2) considering deep NN and introducing new gradient features (e.g., gradient feature notion for upper layers), (3) defining different gradient feature family at different training stages (e.g., gradient feature notion for later stages). In particular, the challenges in the later-stage analysis are: (a) the weights in the later stage will not be as normal as the initialization, and we need new tools to analyze their properties; (b) to show that the later-stage features eventually lead to a good solution, we may need new analysis tools for the non-convex optimization due to the changes in the first layer weights.

B.3 Gradient Feature Learning Framework

We first prove a Simplified Gradient Feature Learning Framework in Appendix B.3.1, which only considers one-step gradient feature learning. Then, we prove our Gradient Feature Learning Framework, e.g., no freezing of the first layer. In Appendix B.3.2, we consider population loss to simplify the proof. Then, we provide more discussion about our problem setup and our core concept in Appendix B.3.3. Finally, we prove our Gradient Feature Learning Framework under empirical loss considering sample complexity in Appendix B.3.4.

B.3.1 Simplified Gradient Feature Learning Framework

Algorithm 4 Training by Algorithm 1 with no updates for the first layer after the first gradient step

Initialize $f_{(\mathbf{a}^{(0)},\mathbf{W}^{(0)},\mathbf{b})} \in \mathcal{F}_{d,m}$; Sample $\mathcal{Z} \sim \mathcal{D}^n$
Get $(\mathbf{a}^{(1)}, \mathbf{W}^{(1)}, \mathbf{b})$ by one gradient step update and fix $\mathbf{W}^{(1)}, \mathbf{b}$
for $t = 2$ **to** T **do**
  $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} - \eta^{(t)} \nabla_{\mathbf{a}} \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{\Xi^{(t-1)}})$
end for

Theorem 3.2.4 (Simple Setting)**.** *Assume* $\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a},\mathbf{W}^{(1)},\mathbf{b})}\right)$ *is L-smooth to* $\mathbf{a}$*. Let* $\eta^{(t)} = \frac{1}{L}, \lambda^{(t)} = 0$*, for all* $t \in \{2, 3, \ldots, T\}$*. Training by Algorithm 1 with no updates for the first layer after the first gradient step, w.h.p., there exists* $t \in [T]$ *such that* $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) \leq \mathrm{OPT}_{\mathbf{W}^{(1)},\mathbf{b},B_{a2}} + O\Big(\frac{L(\|\mathbf{a}^{(1)}\|_2^2 + B_{a2}^2)}{T} + \sqrt{\frac{B_{a2}^2(\|\mathbf{W}^{(1)}\|_F^2 B_x^2 + \|\mathbf{b}\|_2^2)}{n}}\Big)$.

Proof of Theorem 3.2.4. Recall that

$$\mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}} := \big\{ f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \in \mathcal{F}_{d,m} \;\big|\; \|\mathbf{a}\|_2 \leq B_{a2} \big\}, \qquad \mathrm{OPT}_{\mathbf{W},\mathbf{b},B_{a2}} := \min_{g \in \mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}}} \mathcal{L}_{\mathcal{D}}(f). \tag{B.1}$$

We denote $f^* = \arg\min_{g \in \mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}}} \mathcal{L}_{\mathcal{D}}(f)$ and $\tilde{f}^* = \arg\min_{g \in \mathcal{F}_{\mathbf{W},\mathbf{b},B_{a2}}} \widetilde{\mathcal{L}}_{\mathcal{Z}}(f)$. We use $\mathbf{a}^*$

and $\tilde{\mathbf{a}}^*$ to denote their second layer weights respectively. Then, we have

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) =& \mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) - \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) && \text{(B.2)}\\
&+ \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) - \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\tilde{\mathbf{a}}^*,\mathbf{W}^{(1)},\mathbf{b})}) && \text{(B.3)}\\
&+ \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\tilde{\mathbf{a}}^*,\mathbf{W}^{(1)},\mathbf{b})}) - \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}) && \text{(B.4)}\\
&+ \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}) && \text{(B.5)}\\
&+ \mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}) && \text{(B.6)}\\
\leq& \left|\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) - \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})})\right| && \text{(B.7)}\\
&+ \left|\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}) - \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\tilde{\mathbf{a}}^*,\mathbf{W}^{(1)},\mathbf{b})})\right| && \text{(B.8)}\\
&+ 0 && \text{(B.9)}\\
&+ \left|\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})})\right| && \text{(B.10)}\\
&+ \mathrm{OPT}_{\mathbf{W}^{(1)},\mathbf{b},B_{a2}}. && \text{(B.11)}
\end{aligned}$$

Fixing $\mathbf{W}^{(1)}$, $\mathbf{b}$ and optimizing $\mathbf{a}$ only is a convex optimization problem. Note that $\eta \leq \frac{1}{L}$, where $\widetilde{\mathcal{L}}_{\mathcal{Z}}$ is L-smooth to $\mathbf{a}$. Thus with gradient descent, we have

$$\frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}\right) - \widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}^*,\mathbf{W}^{(1)},\mathbf{b})}\right) \leq \frac{\|\mathbf{a}^{(1)} - \mathbf{a}^*\|_2^2}{2T\eta}. \tag{B.12}$$

Then our theorem gets proved by Lemma B.5.9 and generalization bounds based on Rademacher complexity. □

B.3.2 Gradient Feature Learning Framework under Expected Risk

We consider the following training process under population loss to simplify the proof. We prove our Gradient Feature Learning Framework under empirical loss considering sample complexity in Appendix B.3.4.

Given an input distribution, we can get a Gradient Feature set S_{p,γ,B_G} and $g^*(\mathbf{x}) = \sum_{j=1}^{r} \mathbf{a}_j^* \sigma(\left\langle \mathbf{w}_j^*, \mathbf{x} \right\rangle - \mathbf{b}_j^*)$, where $f^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$ is a Gradient Feature Induced networks defined in Definition 3.2.11. Considering training by Algorithm 5, we have the following

Algorithm 5 Network Training via Gradient Descent

Initialize $(\mathbf{a}^{(0)}, \mathbf{W}^{(0)}, \mathbf{b})$ as in Equation (3.8)
for $t = 1$ **to** T **do**
  $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} - \eta^{(t)} \nabla_{\mathbf{a}} \mathcal{L}_{\mathcal{D}}^{\lambda^{(t)}}(f_{\Xi^{(t-1)}})$
  $\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \eta^{(t)} \nabla_{\mathbf{W}} \mathcal{L}_{\mathcal{D}}^{\lambda^{(t)}}(f_{\Xi^{(t-1)}})$
end for

results.

Theorem B.3.1 (Gradient Feature Learning Framework under Expected Risk)**.** *Assume Assumption 3.2.1. For any $\epsilon, \delta \in (0,1)$, if $m \leq e^d$ and*

$$m = \Omega\left(\frac{1}{p}\left(\frac{rB_{a1}B_{x1}}{\epsilon}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right), \tag{B.13}$$

$$T = \Omega\left(\frac{1}{\epsilon}\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}}{\sqrt{B_bB_G}} + \frac{1}{B_{x1}(mp)^{\frac{1}{4}}}\right)\right), \tag{B.14}$$

then with proper hyper-parameter values, we have with probability $\geq 1-\delta$, there exists $t \in [T]$ in Algorithm 5 with

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \leq \text{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma} + \epsilon. \tag{B.15}$$

See the full statement and proof in Theorem B.3.9. Below, we show some lemmas used in the analysis of population loss.

Feature Learning

We first show that a large subset of neurons has gradients at the first step as good features.

Definition B.3.2 (Nice Gradients Set. Equivalent to Equation (3.9))**.** We define

$$G_{(D,+1),Nice} := \left\{i \in [m] : \left\langle \mathbf{w}_i^{(1)}, D \right\rangle > (1-\gamma)\left\|\mathbf{w}_i^{(1)}\right\|_2,\ \left\|\mathbf{w}_i^{(1)}\right\|_2 \geq \left|\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}\right| B_G\right\}$$
$$G_{(D,-1),Nice} := \left\{i \in [2m] \setminus [m] : \left\langle \mathbf{w}_i^{(1)}, D \right\rangle > (1-\gamma)\left\|\mathbf{w}_i^{(1)}\right\|_2,\ \left\|\mathbf{w}_i^{(1)}\right\|_2 \geq \left|\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}\right| B_G\right\}$$

where γ, B_G is the same in the Definition 3.2.7.

Lemma B.3.3 (Feature Emergence. Full Statement of Lemma 3.2.13)**.** *Let* $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$*. For any* r *size subset* $\{(D_1, s_1), \ldots, (D_r, s_r)\} \subseteq S_{p,\gamma,B_G}$*, with probability at least* $1 - 2re^{-cmp}$ *where* $c > 0$ *is a universal constant, we have that for all* $j \in [r]$*,* $|G_{(D_j,s_j),Nice}| \geq \frac{mp}{4}$*.*

Proof of Lemma B.3.3. By symmetric initialization and Lemma B.5.1, we have for all $i \in [2m]$

$$\mathbf{w}_i^{(1)} = -\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right]\mathbf{x}\right] \tag{B.16}$$

$$= -\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}G(\mathbf{w}_i^{(0)}, \mathbf{b}_i). \tag{B.17}$$

For all $j \in [r]$, as $(D_j, s_j) \in S_{p,\gamma,B_G}$, by Lemma B.5.3,

(1) if $s_j = +1$, for all $i \in [m]$, we have

$$\Pr\left[i \in G_{(D_j,s_j),Nice}\right] \tag{B.18}$$

$$= \Pr\left[\frac{\left\langle \mathbf{w}_i^{(1)}, D_j\right\rangle}{\left\|\mathbf{w}_i^{(1)}\right\|_2} > (1-\gamma),\ \left\|\mathbf{w}_i^{(1)}\right\|_2 \geq \left|\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}\right| B_G\right] \tag{B.19}$$

$$= \Pr\left[\frac{\left\langle \mathbf{w}_i^{(1)}, D_j\right\rangle}{\left\|\mathbf{w}_i^{(1)}\right\|_2} > (1-\gamma),\ \left\|\mathbf{w}_i^{(1)}\right\|_2 \geq \left|\eta^{(1)}\ell'(0)\mathbf{a}_i^{(0)}\right| B_G,\ \frac{\mathbf{b}_i}{|\mathbf{b}_i|} = s_j\right] \tag{B.20}$$

$$\geq \Pr\left[G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \in \mathcal{C}_{D_j,\gamma},\ \|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \geq B_G,\ \frac{\mathbf{b}_i}{|\mathbf{b}_i|} = s_j,\ \mathbf{a}_i^{(0)}\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle > 0\right]$$

$$\geq \frac{p}{2}, \tag{B.21}$$

(2) if $s_j = -1$, for all $i \in [2m] \setminus [m]$, similarly we have

$$\Pr\left[i \in G_{(D_j,s_j),Nice}\right] \geq \frac{p}{2}. \tag{B.22}$$

By concentration inequality, (Chernoff's inequality under small deviations), we have

$$\Pr\left[|G_{(D_j,s_j),Nice}| < \frac{mp}{4}\right] \leq 2e^{-cmp}. \tag{B.23}$$

We complete the proof by union bound. □

Good Network Exists

Then, the gradients allow for obtaining a set of neurons approximating the "ground-truth" network with comparable loss.

Lemma B.3.4 (Existence of Good Networks. Full Statement of Lemma 3.2.14)**.** *Let* $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$. *For any* $B_\epsilon \in (0, B_b)$, *let* $\sigma_a = \Theta\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)}B_G B_\epsilon}\right)$ *and* $\delta = 2re^{-\sqrt{mp}}$. *Then, with probability at least* $1-\delta$ *over the initialization, there exists* $\tilde{\mathbf{a}}_i$*'s such that* $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) = \sum_{i=1}^{4m} \tilde{\mathbf{a}}_i \sigma\left(\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)$ *satisfies*

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \le rB_{a1}\left(\frac{B_{x1}^2 B_b}{\sqrt{mp}B_G B_\epsilon} + B_{x1}\sqrt{2\gamma} + B_\epsilon\right) + \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}, \tag{B.24}$$

and $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Proof of Lemma B.3.4. Recall $g^*(\mathbf{x}) = \sum_{j=1}^r \mathbf{a}_j^* \sigma(\left\langle \mathbf{w}_j^*, \mathbf{x}\right\rangle - \mathbf{b}_j^*)$, where $f^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$ is defined in Definition 3.2.11 and let $s_j^* = \frac{\mathbf{b}_j^*}{|\mathbf{b}_j^*|}$. By Lemma B.3.3, with probability at least $1-\delta_1$, $\delta_1 = 2re^{-cmp}$, for all $j \in [r]$, we have $|G_{(\mathbf{w}_j^*, s_j^*),Nice}| \ge \frac{mp}{4}$. Then for all $i \in G_{(\mathbf{w}_j^*, s_j^*),Nice} \subseteq [2m]$, we have $-\ell'(0)\eta^{(1)}G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\frac{\mathbf{b}_j^*}{\tilde{b}}$ only depend on $\mathbf{w}_i^{(0)}$ and $\mathbf{b}_i$, which is independent of $\mathbf{a}_i^{(0)}$. Given Definition 3.2.7, we have

$$-\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \frac{\mathbf{b}_j^*}{\tilde{b}} \in \left[\ell'(0)\eta^{(1)}B_{x1}\frac{B_b}{\tilde{b}}, -\ell'(0)\eta^{(1)}B_{x1}\frac{B_b}{\tilde{b}}\right]. \tag{B.25}$$

We split $[r]$ into $\Gamma = \{j \in [r] : |\mathbf{b}_j^*| < B_\epsilon\}$, $\Gamma_- = \{j \in [r] : \mathbf{b}_j^* \le -B_\epsilon\}$ and $\Gamma_+ = \{j \in [r] : \mathbf{b}_j^* \ge B_\epsilon\}$. Let $\epsilon_a = \frac{B_{x1}B_b}{\sqrt{mp}B_G B_\epsilon}$. Then we know that for all $j \in \Gamma_+ \cup \Gamma_-$, for all

$i \in G_{(\mathbf{w}_j^*, s_j^*), Nice}$, we have

$$\Pr_{\mathbf{a}_i^{(0)} \sim \mathcal{N}(0, \sigma_a^2)} \left[\left| -\mathbf{a}_i^{(0)} \ell'(0) \eta^{(1)} \| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 \frac{|\mathbf{b}_j^*|}{\tilde{b}} - 1 \right| \leq \epsilon_a \right] \tag{B.26}$$

$$= \Pr_{\mathbf{a}_i^{(0)} \sim \mathcal{N}(0, \sigma_a^2)} \left[1 - \epsilon_a \leq -\mathbf{a}_i^{(0)} \ell'(0) \eta^{(1)} \| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 \frac{|\mathbf{b}_j^*|}{\tilde{b}} \leq 1 + \epsilon_a \right] \tag{B.27}$$

$$= \Pr_{g \sim \mathcal{N}(0,1)} \left[1 - \epsilon_a \leq g \Theta \left(\frac{\| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 |\mathbf{b}_j^*|}{B_G B_\epsilon} \right) \leq 1 + \epsilon_a \right] \tag{B.28}$$

$$= \Pr_{g \sim \mathcal{N}(0,1)} \left[(1 - \epsilon_a) \Theta \left(\frac{B_G B_\epsilon}{\| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 |\mathbf{b}_j^*|} \right) \leq g \leq (1 + \epsilon_a) \Theta \left(\frac{B_G B_\epsilon}{\| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 |\mathbf{b}_j^*|} \right) \right]$$

$$= \Theta \left(\frac{\epsilon_a B_G B_\epsilon}{\| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 |\mathbf{b}_j^*|} \right) \tag{B.29}$$

$$\geq \Omega \left(\frac{\epsilon_a B_G B_\epsilon}{B_{x1} B_b} \right) \tag{B.30}$$

$$= \Omega \left(\frac{1}{\sqrt{mp}} \right). \tag{B.31}$$

Thus, with probability $\Omega\left(\frac{1}{\sqrt{mp}}\right)$ over $\mathbf{a}_i^{(0)}$, we have

$$\left| -\mathbf{a}_i^{(0)} \ell'(0) \eta^{(1)} \| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 \frac{|\mathbf{b}_j^*|}{\tilde{b}} - 1 \right| \leq \epsilon_a, \quad \left| \mathbf{a}_i^{(0)} \right| = O\left(\frac{\tilde{b}}{-\ell'(0) \eta^{(1)} B_G B_\epsilon} \right). \tag{B.32}$$

Similarly, for $j \in \Gamma$, for all $i \in G_{(\mathbf{w}_j^*, s_j^*), Nice}$, with probability $\Omega\left(\frac{1}{\sqrt{mp}}\right)$ over $\mathbf{a}_i^{(0)}$, we have

$$\left| -\mathbf{a}_i^{(0)} \ell'(0) \eta^{(1)} \| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 \frac{B_\epsilon}{\tilde{b}} - 1 \right| \leq \epsilon_a, \quad \left| \mathbf{a}_i^{(0)} \right| = O\left(\frac{\tilde{b}}{-\ell'(0) \eta^{(1)} B_G B_\epsilon} \right). \tag{B.33}$$

For all $j \in [r]$, let $\Lambda_j \subseteq G_{(\mathbf{w}_j^*, s_j^*), Nice}$ be the set of i's such that condition Equation (B.32) or Equation (B.33) are satisfied. By Chernoff bound and union bound, with probability at least $1 - \delta_2$, $\delta_2 = re^{-\sqrt{mp}}$, for all $j \in [r]$ we have $|\Lambda_j| \geq \Omega(\sqrt{mp})$.

We have for $\forall j \in \Gamma_+ \cup \Gamma_-, \forall i \in \Lambda_j$,

$$\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \left\langle \mathbf{w}_j^*, \mathbf{x}\right\rangle\right| \tag{B.34}$$

$$\leq \left\| -\mathbf{a}_i^{(0)}\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \frac{|\mathbf{b}_j^*|}{\tilde{b}} \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2} - \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2} + \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2} - \mathbf{w}_j^* \right\| \|\mathbf{x}\|_2 \tag{B.35}$$

$$\leq (\epsilon_a + \sqrt{2\gamma})\|\mathbf{x}\|_2. \tag{B.36}$$

Similarly, for $\forall j \in \Gamma, \forall i \in \Lambda_j$,

$$\left|\frac{B_\epsilon}{\tilde{b}}\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \left\langle \mathbf{w}_j^*, \mathbf{x}\right\rangle\right| \leq (\epsilon_a + \sqrt{2\gamma})\|\mathbf{x}\|_2. \tag{B.37}$$

If $i \in \Lambda_j$, $j \in \Gamma_+ \cup \Gamma_-$, set $\tilde{\mathbf{a}}_i = \mathbf{a}_j^* \frac{|\mathbf{b}_j^*|}{|\Lambda_j|\tilde{b}}$, if $i \in \Lambda_j$, $j \in \Gamma$, set $\tilde{\mathbf{a}}_i = \mathbf{a}_j^* \frac{B_\epsilon}{|\Lambda_j|\tilde{b}}$, otherwise set $\tilde{\mathbf{a}}_i = 0$, we have $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Finally, we have

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \tag{B.38}$$

$$=\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(g^*) + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.39}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) - g^*(\mathbf{x})\right|\right] + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.40}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{i=1}^{m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle - \tilde{b}\right) + \sum_{i=m+1}^{2m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle + \tilde{b}\right) - \sum_{j=1}^{r}\mathbf{a}_j^*\sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle - \mathbf{b}_j^*)\right|\right]$$

$$+ \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.41}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_+}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle - \tilde{b}\right) - \sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle - \mathbf{b}_j^*)\right|\right|\right] \tag{B.42}$$

$$+ \mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_-}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle + \tilde{b}\right) - \sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle - \mathbf{b}_j^*)\right|\right|\right] \tag{B.43}$$

$$+ \mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{B_\epsilon}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle - \tilde{b}\right) - \sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle - \mathbf{b}_j^*)\right|\right|\right] + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.44}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_+}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle - \left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right] \tag{B.45}$$

$$+ \mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_-}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle - \left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right] \tag{B.46}$$

$$+ \mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{B_\epsilon}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle + B_\epsilon - \left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right] + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.47}$$

$$\leq r\|\mathbf{a}^*\|_\infty(\epsilon_a + \sqrt{2\gamma})\mathbb{E}_{(\mathbf{x},y)}\|\mathbf{x}\|_2 + |\Gamma|\|\mathbf{a}^*\|_\infty B_\epsilon + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.48}$$

$$\leq rB_{x1}B_{a1}(\epsilon_a + \sqrt{2\gamma}) + |\Gamma|B_{a1}B_\epsilon + \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}. \tag{B.49}$$

We finish the proof by union bound and $\delta \geq \delta_1 + \delta_2$. □

Learning an Accurate Classifier

We will use the following theorem from existing work to prove that gradient descent learns a good classifier (Theorem B.3.9). Theorem B.3.1 is simply a direct corollary of

Theorem B.3.9.

Theorem B.3.5 (Theorem 13 in [63]). *Fix some η, and let $f_1, \ldots, f_T$ be some sequence of convex functions. Fix some θ_1, and assume we update $\theta_{t+1} = \theta_t - \eta \nabla f_t(\theta_t)$. Then for every θ^* the following holds:*

$$\frac{1}{T}\sum_{t=1}^{T} f_t(\theta_t) \le \frac{1}{T}\sum_{t=1}^{T} f_t(\theta^*) + \frac{1}{2\eta T}\|\theta^*\|_2^2 + \|\theta_1\|_2 \frac{1}{T}\sum_{t=1}^{T} \|\nabla f_t(\theta_t)\|_2 + \eta \frac{1}{T}\sum_{t=1}^{T} \|\nabla f_t(\theta_t)\|_2^2.$$

To apply the theorem we first present a few lemmas bounding the change in the network during steps.

Lemma B.3.6 (Bound of $\Xi^{(0)}, \Xi^{(1)}$). *Assume the same conditions as in Lemma B.3.4, and $d \ge \log m$, with probability at least $1 - \delta - \frac{1}{m^2}$ over the initialization, $\|\mathbf{a}^{(0)}\|_\infty = O\left(\frac{\tilde{b}\sqrt{\log m}}{-\ell'(0)\eta^{(1)} B_G B_\epsilon}\right)$, and for all $i \in [4m]$, we have $\|\mathbf{w}_i^{(0)}\|_2 = O\left(\sigma_{\mathbf{w}}\sqrt{d}\right)$. Finally, $\|\mathbf{a}^{(1)}\|_\infty = O\left(-\eta^{(1)}\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d} + \tilde{b})\right)$, and for all $i \in [4m]$, $\|\mathbf{w}_i^{(1)}\|_2 = O\left(\frac{\tilde{b}\sqrt{\log m} B_{x1}}{B_G B_\epsilon}\right)$.*

Proof of Lemma B.3.6. By Lemma B.5.4, we have $\|\mathbf{a}^{(0)}\|_\infty = O\left(\frac{\tilde{b}\sqrt{\log m}}{-\ell'(0)\eta^{(1)} B_G B_\epsilon}\right)$ with probability at least $1 - \frac{1}{2m^2}$ by property of maximum i.i.d Gaussians. For any $i \in [4m]$, by Lemma B.5.5 and $d \ge \log m$, we have

$$\Pr\left(\frac{1}{\sigma_{\mathbf{w}}^2}\left\|\mathbf{w}_i^{(0)}\right\|_2^2 \ge d + 2\sqrt{4d\log(m)} + 8\log(m)\right) \le O\left(\frac{1}{m^4}\right). \tag{B.50}$$

Thus, by union bound, with probability at least $1 - \frac{1}{2m^2}$, for all $i \in [4m]$, we have $\|\mathbf{w}_i^{(0)}\|_2 = O\left(\sigma_{\mathbf{w}}\sqrt{d}\right)$.

For all $i \in [4m]$, we have

$$\begin{aligned}
|\mathbf{a}_i^{(1)}| =& -\eta^{(1)}\ell'(0)\left|\mathbb{E}_{(\mathbf{x},y)}\left[y\left[\sigma\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right]\right| && \text{(B.51)}\\
\le& -\eta^{(1)}\ell'(0)(\|\mathbf{w}_i^{(0)}\|_2 \mathbb{E}_{(\mathbf{x},y)}[\|\mathbf{x}\|_2] + \tilde{b}) && \text{(B.52)}\\
\le& O\left(-\eta^{(1)}\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d} + \tilde{b})\right). && \text{(B.53)}
\end{aligned}$$

$$\|\mathbf{w}_i^{(1)}\|_2 = -\eta^{(1)}\ell'(0)\left\|\mathbf{a}_i^{(0)}\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle - \mathbf{b}_i\right]\mathbf{x}\right]\right\|_2 \tag{B.54}$$

$$\leq O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_G B_\epsilon}\right). \tag{B.55}$$

□

Lemma B.3.7 (Bound of $\Xi^{(t)}$)**.** *Assume the same conditions as in Lemma B.3.6, and let $\eta = \eta^{(t)}$ for all $t \in \{2,3,\ldots,T\}$, $0 < T\eta B_{x1} \leq o(1)$, and $0 = \lambda = \lambda^{(t)}$ for all $t \in \{2,3,\ldots,T\}$, for all $i \in [4m]$, we have*

$$|\mathbf{a}_i^{(t)}| \leq O\left(|\mathbf{a}_i^{(1)}| + \|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}} + \eta\tilde{b}\right) \tag{B.56}$$

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \leq O\left(t\eta B_{x1}|\mathbf{a}_i^{(1)}| + t\eta^2 B_{x1}^2\|\mathbf{w}_i^{(1)}\|_2 + t\eta^2 B_{x1}\tilde{b}\right). \tag{B.57}$$

Proof of Lemma B.3.7. For all $i \in [4m]$, by Lemma B.3.6,

$$|\mathbf{a}_i^{(t)}| = \left|(1-\eta\lambda)\mathbf{a}_i^{(t-1)} - \eta\mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_{\Xi^{(t-1)}(\mathbf{x})})y\left[\sigma\left(\left\langle \mathbf{w}_i^{(t-1)},\mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right]\right| \tag{B.58}$$

$$\leq \left|(1-\eta\lambda)\mathbf{a}_i^{(t-1)}\right| + \eta\left|\mathbb{E}_{(\mathbf{x},y)}\left[\left[\sigma\left(\left\langle \mathbf{w}_i^{(t-1)},\mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right]\right| \tag{B.59}$$

$$\leq \left|\mathbf{a}_i^{(t-1)}\right| + \eta(B_{x1}\|\mathbf{w}_i^{(t-1)}\|_2 + \tilde{b}) \tag{B.60}$$

$$\leq \left|\mathbf{a}_i^{(t-1)}\right| + \eta B_{x1}\|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(1)}\|_2 + \eta B_{x1}\|\mathbf{w}_i^{(1)}\|_2 + \eta\tilde{b} \tag{B.61}$$

$$= \left|\mathbf{a}_i^{(t-1)}\right| + \eta B_{x1}\|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(1)}\|_2 + \eta Z_i, \tag{B.62}$$

where we denote $Z_i = B_{x1}\|\mathbf{w}_i^{(1)}\|_2 + \tilde{b}$. Then we give a bound of the first layer's weights change,

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \tag{B.63}$$

$$= \left\|(1-\eta\lambda)\mathbf{w}_i^{(t-1)} - \eta\mathbf{a}_i^{(t-1)}\mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_{\Xi^{(t-1)}(\mathbf{x})})y\sigma'\left[\left\langle \mathbf{w}_i^{(t-1)},\mathbf{x}\right\rangle - \mathbf{b}_i\right]\mathbf{x}\right] - \mathbf{w}_i^{(1)}\right\|_2 \tag{B.64}$$

$$\leq \|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(1)}\|_2 + \eta B_{x1}|\mathbf{a}_i^{(t-1)}|. \tag{B.65}$$

Combine two bounds, we can get

$$|\mathbf{a}_i^{(t)}| \leq |\mathbf{a}_i^{(t-1)}| + \eta Z_i + (\eta B_{x1})^2 \sum_{l=1}^{t-2} |\mathbf{a}_i^{(l)}| \tag{B.66}$$

$$\Leftrightarrow \sum_{l=1}^{t} |\mathbf{a}_i^{(l)}| \leq 2\left(\sum_{l=1}^{t-1} |\mathbf{a}_i^{(l)}|\right) - (1-(\eta B_{x1})^2)\left(\sum_{l=1}^{t-2} |\mathbf{a}_i^{(l)}|\right) + \eta Z_i. \tag{B.67}$$

Let $h(1) = |\mathbf{a}_i^{(1)}|,\ h(2) = 2|\mathbf{a}_i^{(1)}| + \eta Z_i$ and $h(t+2) = 2h(t+1) - (1-(\eta B_{x1})^2)h(t) + \eta Z_i$ for $n \in \mathbb{N}_+$, by Lemma B.5.8, we have

$$h(t) = -\frac{Z_i}{\eta B_{x1}^2} + c_1(1-\eta B_{x1})^{(t-1)} + c_2(1+\eta B_{x1})^{(t-1)} \tag{B.68}$$

$$c_1 = \frac{1}{2}\left(|\mathbf{a}_i^{(1)}| + \frac{Z_i}{\eta B_{x1}^2} - \frac{|\mathbf{a}_i^{(1)}| + \eta Z_i}{\eta B_{x1}}\right) \tag{B.69}$$

$$c_2 = \frac{1}{2}\left(|\mathbf{a}_i^{(1)}| + \frac{Z_i}{\eta B_{x1}^2} + \frac{|\mathbf{a}_i^{(1)}| + \eta Z_i}{\eta B_{x1}}\right). \tag{B.70}$$

Thus, by $|c_1| \leq c_2$, and $0 < T\eta B_{x1} \leq o(1)$, we have

$$|\mathbf{a}_i^{(t)}| \leq h(t) - h(t-1) \tag{B.71}$$

$$= -\eta B_{x1} c_1 (1-\eta B_{x1})^{(t-2)} + \eta B_{x1} c_2 (1+\eta B_{x1})^{(t-2)} \tag{B.72}$$

$$\leq 2\eta B_{x1} c_2 (1+\eta B_{x1})^t \tag{B.73}$$

$$\leq O(2\eta B_{x1} c_2). \tag{B.74}$$

Similarly, by binomial approximation, we also have

$$\begin{aligned}
\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \leq & \eta B_{x1} h(t-1) && \text{(B.75)}\\
= & \eta B_{x1}\left(-\frac{Z_i}{\eta B_{x1}^2} + c_1(1-\eta B_{x1})^{(t-2)} + c_2(1+\eta B_{x1})^{(t-2)}\right) && \text{(B.76)}\\
\leq & \eta B_{x1} O\left(-\frac{Z_i}{\eta B_{x1}^2} + c_1(1-(t-2)\eta B_{x1}) + c_2(1+(t-2)\eta B_{x1})\right) && \text{(B.77)}\\
\leq & \eta B_{x1} O\left(-\frac{Z_i}{\eta B_{x1}^2} + c_1 + c_2 + (c_2-c_1)t\eta B_{x1}\right) && \text{(B.78)}\\
\leq & \eta B_{x1} O\left(|\mathbf{a}_i^{(1)}| + \frac{|\mathbf{a}_i^{(1)}| + \eta Z_i}{\eta B_{x1}} t\eta B_{x1}\right) && \text{(B.79)}\\
\leq & O\left((\eta|\mathbf{a}_i^{(1)}| + \eta^2 Z_i) t B_{x1}\right). && \text{(B.80)}
\end{aligned}$$

We finish the proof by plugging Z_i, c_2 into the bound. □

Lemma B.3.8 (Bound of Loss Gap and Gradient)**.** *Assume the same conditions as in Lemma B.3.7, for all $t \in [T]$, we have*

$$|\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})| \leq B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \tag{B.81}$$

and for all $t \in [T]$, for all $i \in [4m]$, we have

$$\left|\frac{\partial \mathcal{L}_{\mathcal{D}}(f_{\Xi^{(t)}})}{\partial \mathbf{a}_i^{(t)}}\right| \leq B_{x1}(\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 + \|\mathbf{w}_i^{(1)}\|_2) + \tilde{b}. \tag{B.82}$$

Proof of Lemma B.3.8. It follows from that

$$\begin{aligned}
& |\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})| && \text{(B.83)}\\
& \leq \mathbb{E}_{(\mathbf{x},y)}|f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}(\mathbf{x}) - f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x})| && \text{(B.84)}\\
& \leq \mathbb{E}_{(\mathbf{x},y)}\left[\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\left|\sigma\left[\left\langle \mathbf{w}_i^{(t)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right] - \sigma\left[\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right]\right|\right] && \text{(B.85)}\\
& \leq B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2. && \text{(B.86)}
\end{aligned}$$

Also, we have

$$\left|\frac{\partial \mathcal{L}_{\mathcal{D}}(f_{\Xi^{(t)}})}{\partial \mathbf{a}_i^{(t)}}\right| = \left|\mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_{\Xi^{(t)}}(\mathbf{x}))y\left[\sigma\left(\left\langle \mathbf{w}_i^{(t)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right]\right| \tag{B.87}$$

$$\leq B_{x1}\|\mathbf{w}_i^{(t)}\|_2 + \tilde{b} \tag{B.88}$$

$$\leq B_{x1}(\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 + \|\mathbf{w}_i^{(1)}\|_2) + \tilde{b}. \tag{B.89}$$

□

We are now ready to prove the main theorem.

Theorem B.3.9 (Online Convex Optimization. Full Statement of Theorem B.3.1)**.** *Consider training by Algorithm 5, and any $\delta \in (0,1)$. Assume $d \geq \log m$. Set*

$$\sigma_{\mathbf{w}} > 0, \quad \tilde{b} > 0, \quad \eta^{(t)} = \eta,\ \lambda^{(t)} = 0 \text{ for all } t \in \{2,3,\ldots,T\},$$
$$\eta^{(1)} = \Theta\left(\frac{\min\{O(\eta), O(\eta\tilde{b})\}}{-\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b})}\right),\ \lambda^{(1)} = \frac{1}{\eta^{(1)}}, \quad \sigma_a = \Theta\left(\frac{\tilde{b}(mp)^{\frac{1}{4}}}{-\ell'(0)\eta^{(1)}B_{x1}\sqrt{B_G B_b}}\right).$$

Let $0 < T\eta B_{x1} \leq o(1)$, $m = \Omega\left(\frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right)$. With probability at least $1-\delta$ over the initialization, there exists $t \in [T]$ such that

$$\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma}\right) \tag{B.90}$$
$$+ \eta\left(\sqrt{r}B_{a2}B_b T\eta B_{x1}^2 + m\tilde{b}\right) O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + 1\right) + O\left(\frac{B_{a2}^2 B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right).$$

Furthermore, for any $\epsilon \in (0,1)$*, set*

$$\tilde{b} =\Theta\left(\frac{B_G^{\frac{1}{4}} B_{a2} B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right), \quad m = \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right), \tag{B.91}$$

$$\eta =\Theta\left(\frac{\epsilon}{\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right)}\right), \quad T = \Theta\left(\frac{1}{\eta B_{x1}(mp)^{\frac{1}{4}}}\right), \tag{B.92}$$

we have there exists $t \in [T]$ *with*

$$\Pr[\operatorname{sign}(f_{\Xi^{(t)}})(\mathbf{x}) \neq y] \leq \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma} + \epsilon. \tag{B.93}$$

Proof of Theorem B.3.9. By $m = \Omega\left(\frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right)$ we have $2re^{-\sqrt{mp}} + \frac{1}{m^2} \leq \delta$. For any $B_\epsilon \in (0, B_b)$, when $\sigma_a = \Theta\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)}B_GB_\epsilon}\right)$, by Theorem B.3.5, Lemma B.3.4, Lemma B.3.8, with probability at least $1-\delta$ over the initialization, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.94}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\left|\left(\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right| + \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right) \tag{B.95}$$

$$+ \frac{\|\tilde{\mathbf{a}}\|_2^2}{2\eta T} + (2\|\mathbf{a}^{(1)}\|_2\sqrt{m} + 4\eta m)\max_{i\in[4m]}\left|\frac{\partial\mathcal{L}_{\mathcal{D}}(f_{\Xi^{(T)}})}{\partial\mathbf{a}_i^{(T)}}\right| \tag{B.96}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{B_{x1}^2B_b}{\sqrt{mp}B_GB_\epsilon} + B_{x1}\sqrt{2\gamma} + B_\epsilon\right) \tag{B.97}$$

$$+ B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 \tag{B.98}$$

$$+ \frac{\|\tilde{\mathbf{a}}\|_2^2}{2\eta T} + 4mB_{x1}(\|\mathbf{a}^{(1)}\|_\infty + \eta)\left(\max_{i\in[4m]}\|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 + \max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}}\right). \tag{B.99}$$

By Lemma B.3.4, Lemma B.3.6, Lemma B.3.7, when $\eta^{(1)} = \Theta\left(\frac{\min\{O(\eta), O(\eta\tilde{b})\}}{-\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b})}\right)$, we have

$$\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right), \quad \|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right) \tag{B.100}$$

$$\|\mathbf{a}^{(1)}\|_\infty = O\left(-\eta^{(1)}\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b})\right) \tag{B.101}$$

$$= \min\{O(\eta), O(\eta\tilde{b})\} \tag{B.102}$$

$$\max_{i\in[4m]} \|\mathbf{w}_i^{(1)}\|_2 = O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_G B_\epsilon}\right) \tag{B.103}$$

$$\max_{i\in[4m]} \|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 = O\left(T\eta B_{x1}\|\mathbf{a}^{(1)}\|_\infty + T\eta^2 B_{x1}^2 \max_{i\in[4m]} \|\mathbf{w}_i^{(1)}\|_2 + T\eta^2 B_{x1}\tilde{b}\right) \tag{B.104}$$

$$= O\left(T\eta^2 B_{x1}^2\left(\max_{i\in[4m]} \|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}}\right)\right). \tag{B.105}$$

Set $B_\epsilon = \frac{B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}}$, we have $\sigma_a = \Theta\left(\frac{\tilde{b}(mp)^{\frac{1}{4}}}{-\ell'(0)\eta^{(1)}B_{x1}\sqrt{B_G B_b}}\right)$ which satisfy the requirements. Then,

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.106}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma}\right) \tag{B.107}$$

$$+ \left(\sqrt{r}B_{a2}B_b T\eta^2 B_{x1}^2\frac{B_{x1}}{\tilde{b}} + m\eta B_{x1}\right) O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_G B_\epsilon} + \frac{\tilde{b}}{B_{x1}}\right) + O\left(\frac{B_{a2}^2 B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right)$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma}\right) \tag{B.108}$$

$$+ \eta\left(\sqrt{r}B_{a2}B_b T\eta B_{x1}^2 + m\tilde{b}\right) O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + 1\right) + O\left(\frac{B_{a2}^2 B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right). \tag{B.109}$$

Furthermore, for any $\epsilon \in (0,1)$, set

$$\tilde{b} =\Theta\left(\frac{B_G^{\frac{1}{4}} B_{a2} B_b^{\frac{3}{4}}}{\sqrt{r B_{a1}}}\right), \quad m = \Omega\left(\frac{1}{p\epsilon^4}\left(r B_{a1} B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right), \tag{B.110}$$

$$\eta =\Theta\left(\frac{\epsilon}{\left(\frac{\sqrt{r} B_{a2} B_b B_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m} B_{x1} (mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + 1\right)}\right), \quad T = \Theta\left(\frac{1}{\eta B_{x1} (mp)^{\frac{1}{4}}}\right), \tag{B.111}$$

we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \leq & \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + r B_{a1}\left(\frac{2B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma}\right) + \frac{\epsilon}{2} && \text{(B.112)}\\
& + O\left(\frac{B_{x1} B_{a2}^2 B_b^2}{\tilde{b}^2 (mp)^{\frac{1}{4}}}\right) && \text{(B.113)}\\
\leq & \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + r B_{a1} B_{x1}\sqrt{2\gamma} + \epsilon. && \text{(B.114)}
\end{aligned}$$

We finish the proof as the 0-1 classification error is bounded by the loss function, e.g., $\mathbb{I}[\mathrm{sign}(g(\mathbf{x})) \neq y] \leq \frac{\ell(yg(\mathbf{x}))}{\ell(0)}$, where $\ell(0) = 1$. □

B.3.3 More Discussion about Setting

Range of $\sigma_{\mathbf{w}}$. In practice, the value of $\sigma_{\mathbf{w}}$ cannot be arbitrary, because its choice will have an effect on the Gradient Feature set S_{p,γ,B_G}. On the other hand, $d \geq \log m$ is a natural assumption, otherwise, the two-layer neural networks may fall in the NTK regime.

Parameter Choice. We use $\lambda = 1/\eta$ in the first step so that the neural network will totally forget its initialization, leading to the feature emergence here. This is a common setting for analysis convenience in previous work, e.g., [63, 239, 62]. We can extend this to other choices (e.g., small initialization and large step size for the first few steps), as long as after the gradient update, the gradient dominates the neuron weights. We use $\lambda = 0$ afterward as the regularization effect is weak in our analysis. We can extend our analysis

to λ being a small value.

Early Stopping. Our analysis divides network learning into two stages: the feature learning stage, and then classifier learning over the good features. The feature learning stage is simplified to one gradient step for the convenience of analysis, while in practice feature learning can happen in multiple steps. The current framework focuses on the gradient features in the early gradient steps, while feature learning can also happen in later steps, in particular for more complicated data. It is an interesting direction to extend the analysis to a longer training horizon.

Role of s. The s encodes the sign of the bias term, which is important. Recall that we do not update the bias term for simplicity. Let's consider a simple toy example. Assume we have $f_1(x) = a_1\sigma(w_1^\top x + 1)$, $f_2(x) = a_2\sigma(w_2^\top x - 1)$ and $f_3(x) = a_3\sigma(w_3^\top x + 2)$, where σ is ReLU activation function which is a homogeneous function.

1. The sign of the bias term is important. We can see that we always have $a_1\sigma(w_1^\top x+1) \neq a_2\sigma(w_2^\top x - 1)$ for any a_1, w_1, a_2, w_2. This means that $f_1(x)$ and $f_2(x)$ are intrinsically different and have different active patterns. Thus, we need to handle the sign of the bias term carefully.
2. The scaling of the bias is absorbed. On the other hand, we can see that $a_1\sigma(w_1^\top x+1) = a_3\sigma(w_3^\top x + 2)$ when $a_1 = 2a_3, 2w_1 = w_3$. It means that the scale of the bias term is less important, which can be absorbed into other terms.

Thus, we only need to handle bias with different signs carefully.

Gradient Feature Distribution. We may define a gradient feature distribution rather than a gradient feature set. However, we find that the technical tools used in this continuous setting are pretty different from the discrete version.

Activation Functions. We can change the ReLU activation function to a sublinear activation function, e.g. leaky ReLU, sigmoid, to get a similar conclusion. First, we need

to introduce a corresponding gradient feature set, and then we can make it by following the same analysis pipeline. For simplicity, we present ReLU only.

B.3.4 Gradient Feature Learning Framework under Empirical Risk with Sample Complexity

In this section, we consider training with empirical risk. Intuitively, the proof is straightforward from the proof for population loss. We can simply replace the population loss with the empirical loss, which will introduce an error term in the gradient analysis. We use concentration inequality to control the error term and show that the error term depends inverse-polynomially on the sample size n.

Definition B.3.10 (Empirical Simplified Gradient Vector)**.** Recall $\mathcal{Z} = \{(\mathbf{x}^{(l)}, y^{(l)})\}_{l\in[n]}$, for any $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, an Empirical Simplified Gradient Vector is defined as

$$\widetilde{G}(\mathbf{w}, b) := \frac{1}{n}\sum_{l\in[n]}[y^{(l)}\mathbf{x}^{(l)}\mathbb{I}[\mathbf{w}^\top\mathbf{x}^{(l)} > b]]. \tag{B.115}$$

Definition B.3.11 (Empirical Gradient Feature)**.** Recall $\mathcal{Z} = \{(\mathbf{x}^{(l)}, y^{(l)})\}_{l\in[n]}$, let $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ be random variables drawn from some distribution $\mathcal{W}, \mathcal{B}$. An Empirical Gradient Feature set with parameters p, γ, B_G is defined as:

$$\widetilde{S}_{p,\gamma,B_G}(\mathcal{W}, \mathcal{B}) := \left\{(D, s) \;\middle|\; \Pr_{\mathbf{w},b}\left[\widetilde{G}(\mathbf{w}, b) \in \mathcal{C}_{D,\gamma} \text{ and } \|\widetilde{G}(\mathbf{w}, b)\|_2 \geq B_G \text{ and } s = \frac{b}{|b|}\right] \geq p\right\}.$$

When clear from context, write it as $\widetilde{S}_{p,\gamma,B_G}$.

Considering training by Algorithm 1, we have the following results.

Theorem 3.2.12 (Main Result)**.** *Assume Assumption 3.2.1. For any $\epsilon, \delta \in (0, 1)$, if*

$m \leq e^d$ *and*

$$
\begin{aligned}
m =& \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right), \\
T =& \Omega\left(\frac{1}{\epsilon}\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}}{\sqrt{B_bB_G}} + \frac{1}{B_{x1}(mp)^{\frac{1}{4}}}\right)\right), \\
\frac{n}{\log n} =& \tilde{\Omega}\left(\frac{m^3pB_x^2B_{a2}^4B_b}{\epsilon^2r^2B_{a1}^2B_G} + \frac{(mp)^{\frac{1}{2}}B_{x2}}{B_bB_G} + \frac{B_x^2}{B_{x2}} + \frac{1}{p} + \left(\frac{1}{B_G^2} + \frac{1}{B_{x1}^2}\right)\frac{B_{x2}}{|\ell'(0)|^2} + \frac{Tm}{\delta}\right),
\end{aligned}
$$

then with initialization (3.8) and proper hyper-parameter values, we have with probability $\geq 1-\delta$ *over the initialization and training samples, there exists* $t \in [T]$ *in Algorithm 1 with:*

$$
\begin{aligned}
&\Pr[\operatorname{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \\
&\qquad \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)} + \epsilon.
\end{aligned}
$$

See the full statement and proof in Theorem B.3.17. Below, we show some lemmas used in the analysis under empirical loss.

Lemma B.3.12 (Empirical Gradient Concentration Bound)**.** *When* $\frac{n}{\log n} > \frac{B_x^2}{B_{x2}}$*, with probability at least* $1 - O\left(\frac{1}{n}\right)$ *over training samples, for all* $i \in [4m]$*, we have*

$$
\left\|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_i}\right\|_2 \leq O\left(\frac{|\mathbf{a}_i|\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right), \tag{B.116}
$$

$$
\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{a}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{a}_i}\right| \leq O\left(\frac{\|\mathbf{w}_i\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right), \tag{B.117}
$$

$$
\left|\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_\Xi\right) - \mathcal{L}_{\mathcal{D}}\left(f_\Xi\right)\right| \leq O\left(\frac{\left(\|\mathbf{a}\|_0\|\mathbf{a}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i\|_2B_x + \tilde{b}) + 1\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right). \tag{B.118}
$$

Proof of Lemma B.3.12. First, we define,

$$
\mathbf{z}^{(l)} = \ell'(y^{(l)}f_\Xi(\mathbf{x}^{(l)}))y^{(l)}\left[\sigma'\left(\left\langle\mathbf{w}_i,\mathbf{x}^{(l)}\right\rangle - \mathbf{b}_i\right)\mathbf{x}^{(l)}\right] \tag{B.119}
$$

$$
- \mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_\Xi(\mathbf{x}))y\left[\sigma'\left(\langle\mathbf{w}_i,\mathbf{x}\rangle - \mathbf{b}_i\right)\right]\mathbf{x}\right]. \tag{B.120}
$$

As $|\ell'(z)| \leq 1, |y| \leq 1, |\sigma'(z)| \leq 1$, we have $\mathbf{z}^{(l)}$ is zero-mean random vector with $\left\|\mathbf{z}^{(l)}\right\|_2 \leq 2B_x$ as well as $\mathbb{E}\left[\left\|\mathbf{z}^{(l)}\right\|_2^2\right] \leq B_{x2}$. Then by Vector Bernstein Inequality, Lemma 18 in [145], for $0 < z < \frac{B_{x2}}{B_x}$ we have

$$\Pr\left(\left\|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_i}\right\|_2 \geq |\mathbf{a}_i|z\right) = \Pr\left(\left\|\frac{1}{n}\sum_{l\in[n]}\mathbf{z}^{(l)}\right\|_2 \geq z\right) \tag{B.121}$$

$$\leq \exp\left(-n\cdot\frac{z^2}{8B_{x2}} + \frac{1}{4}\right). \tag{B.122}$$

Thus, let $z = n^{-\frac{1}{2}}\sqrt{B_{x2}\log n}$, with probability at least $1 - O\left(\frac{1}{n}\right)$, we have

$$\left\|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_i}\right\|_2 \leq O\left(\frac{|\mathbf{a}_i|\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right). \tag{B.123}$$

On the other hand, by Bernstein Inequality, for $z > 0$ we have

$$\Pr\left(\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{a}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{a}_i}\right| > z\|\mathbf{w}_i\|_2\right) \tag{B.124}$$

$$= \Pr\left(\left|\frac{1}{n}\sum_{l\in[n]}\left(\ell'(y^{(l)}f_\Xi(\mathbf{x}^{(l)}))y^{(l)}\left[\sigma\left(\left\langle\mathbf{w}_i,\mathbf{x}^{(l)}\right\rangle - \mathbf{b}_i\right)\right]\right.\right.\right. \tag{B.125}$$

$$\left.\left.\left. - \mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_\Xi(\mathbf{x}))y\left[\sigma\left(\langle\mathbf{w}_i,\mathbf{x}\rangle - \mathbf{b}_i\right)\right]\right]\right)\right| > z\|\mathbf{w}_i\|_2\right) \tag{B.126}$$

$$\leq 2\exp\left(-\frac{\frac{1}{2}nz^2}{B_{x2} + \frac{1}{3}B_x z}\right). \tag{B.127}$$

Thus, when $\frac{n}{\log n} > \frac{B_x^2}{B_{x2}}$, let $z = n^{-\frac{1}{2}}\sqrt{B_{x2}\log n}$, with probability at least $1 - O\left(\frac{1}{n}\right)$, we have

$$\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{a}_i} - \frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{a}_i}\right| \leq O\left(\frac{\|\mathbf{w}_i\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right). \tag{B.128}$$

Finally, we have

$$\left|\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{\Xi}\right)-\mathcal{L}_{\mathcal{D}}\left(f_{\Xi}\right)\right| \tag{B.129}$$

$$=\left|\frac{1}{n}\sum_{l=1}^{n}\left(\ell\left(y^{(l)}\mathbf{a}^{\top}\left[\sigma(\mathbf{W}^{\top}\mathbf{x}^{(l)}-\mathbf{b})\right]\right)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell\left(y\mathbf{a}^{\top}\left[\sigma(\mathbf{W}^{\top}\mathbf{x}-\mathbf{b})\right]\right)\right]\right)\right|. \tag{B.130}$$

By Assumption 3.2.1, we have $\ell\left(y^{(l)}\mathbf{a}^{\top}\left[\sigma(\mathbf{W}^{\top}\mathbf{x}^{(l)}-\mathbf{b})\right]\right)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell\left(y\mathbf{a}^{\top}\left[\sigma(\mathbf{W}^{\top}\mathbf{x}-\mathbf{b})\right]\right)\right]$ is a zero-mean random variable, with bound $2\|\mathbf{a}\|_0\|\mathbf{a}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i\|_2 B_x+\tilde{b})+2$. By Hoeffding's inequality, for all $z>0$, we have

$$\Pr\left(\left|\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{\Xi}\right)-\mathcal{L}_{\mathcal{D}}\left(f_{\Xi}\right)\right|\geq z\right)\leq 2\exp\left(-\frac{z^2 n}{(\|\mathbf{a}\|_0\|\mathbf{a}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i\|_2 B_x+\tilde{b})+1)^2}\right).$$

Thus, with probability at least $1-O\left(\frac{1}{n}\right)$, we have

$$\left|\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{\Xi}\right)-\mathcal{L}_{\mathcal{D}}\left(f_{\Xi}\right)\right|\leq O\left(\frac{\left(\|\mathbf{a}\|_0\|\mathbf{a}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i\|_2 B_x+\tilde{b})+1\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right). \tag{B.131}$$

□

The gradients allow for obtaining a set of neurons approximating the "ground-truth" network with comparable loss.

Lemma B.3.13 (Existence of Good Networks under Empirical Risk)**.** *Suppose* $\frac{n}{\log n}>\Omega\left(\frac{B_x^2}{B_{x2}}+\frac{1}{p}+\frac{B_{x2}}{B_G^2|\ell'(0)|^2}\right)$*. Let* $\lambda^{(1)}=\frac{1}{\eta^{(1)}}$*. For any* $B_\epsilon\in(0,B_b)$*, let* $\sigma_a=\Theta\left(\frac{\tilde{b}}{-|\ell'(0)|\eta^{(1)}B_G B_\epsilon}\right)$ *and* $\delta=2re^{-\sqrt{\frac{mp}{2}}}$*. Then, with probability at least* $1-\delta$ *over the initialization and training samples, there exists* $\tilde{\mathbf{a}}_i$*'s such that* $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x})=\sum_{i=1}^{4m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)$ *satisfies*

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \tag{B.132}$$

$$\leq rB_{a1}\left(\frac{2B_{x1}^2 B_b}{\sqrt{mp}B_G B_\epsilon}+B_{x1}\sqrt{2\gamma+O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)}+B_\epsilon\right)+\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}, \tag{B.133}$$

and $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Proof of Lemma B.3.13. Denote $\rho = O\left(\frac{1}{n}\right)$ and $\beta = O\left(\frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)$. Note that by symmetric initialization, we have $\ell'(yf_{\Xi^{(0)}}(\mathbf{x})) = |\ell'(0)|$ for any $\mathbf{x} \in \mathcal{X}$, so that, by Lemma B.3.12, we have $\left\|\widetilde{G}(\mathbf{w}_i^{(0)}, \mathbf{b}_i) - G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\right\|_2 \leq \frac{\beta}{|\ell'(0)|}$ with probability at least $1-\rho$. Thus, by union bound, we can see that $S_{p,\gamma,B_G} \subseteq \widetilde{S}_{p-\rho,\gamma+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|}}$. Consequently, we have $\mathrm{OPT}_{d,r,B_F,\widetilde{S}_{p-\rho,\gamma+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|}}} \leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}$. Exactly follow the proof in Lemma B.3.4 by replacing S_{p,γ,B_G} to $\widetilde{S}_{p-\rho,\gamma+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|}}$. Then, we finish the proof by $\rho \leq \frac{p}{2}$, $\frac{\beta}{|\ell'(0)|} \leq (1-1/\sqrt{2})B_G$. □

We will use Theorem B.3.5 to prove that gradient descent learns a good classifier (Theorem B.3.17). Theorem 3.2.12 is simply a direct corollary of Theorem B.3.17. To apply the theorem we first present a few lemmas bounding the change in the network during steps.

Lemma B.3.14 (Bound of $\Xi^{(0)}, \Xi^{(1)}$ under Empirical Risk)**.** *Assume the same conditions as in Lemma B.3.13, and $d \geq \log m$, with probability at least $1-\delta-\frac{1}{m^2}-O\left(\frac{m}{n}\right)$ over the initialization and training samples, $\|\mathbf{a}^{(0)}\|_\infty = O\left(\frac{\tilde{b}\sqrt{\log m}}{|\ell'(0)|\eta^{(1)}B_GB_\epsilon}\right)$, and for all $i \in [4m]$, we have $\|\mathbf{w}_i^{(0)}\|_2 = O\left(\sigma_{\mathbf{w}}\sqrt{d}\right)$. Finally, $\|\mathbf{a}^{(1)}\|_\infty = O\left(\eta^{(1)}|\ell'(0)|(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b}) + \eta^{(1)}\frac{\sigma_{\mathbf{w}}\sqrt{dB_{x2}\log n}}{n^{\frac{1}{2}}}\right)$, and for all $i \in [4m]$, $\|\mathbf{w}_i^{(1)}\|_2 = O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_GB_\epsilon} + \frac{\tilde{b}\sqrt{\log m B_{x2}\log n}}{|\ell'(0)|B_GB_\epsilon n^{\frac{1}{2}}}\right)$.*

Proof of Lemma B.3.14. The proof exactly follows the proof of Lemma B.3.6 with Lemma B.3.12. □

Lemma B.3.15 (Bound of $\Xi^{(t)}$ under Empirical Risk)**.** *Assume the same conditions as in Lemma B.3.14, and let $\eta = \eta^{(t)}$ for all $t \in \{2,3,\ldots,T\}$, $0 < T\eta B_{x1} \leq o(1)$, and $0 = \lambda = \lambda^{(t)}$ for all $t \in \{2,3,\ldots,T\}$. With probability at least $1-O\left(\frac{Tm}{n}\right)$ over training*

samples, for all $i \in [4m]$*, for all* $t \in \{2, 3, \ldots, T\}$*, we have*

$$|\mathbf{a}_i^{(t)}| \leq O\left(|\mathbf{a}_i^{(1)}| + \|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)} + \eta\tilde{b}\right) \tag{B.134}$$

$$\begin{aligned}\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \leq & O\bigg(t\eta\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)|\mathbf{a}_i^{(1)}| + t\eta^2\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)^2\|\mathbf{w}_i^{(1)}\|_2 \\ & + t\eta^2\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)\tilde{b}\bigg).\end{aligned} \tag{B.135}$$

Proof of Lemma B.3.15. The proof exactly follows the proof of Lemma B.3.7 with Lemma B.3.12. Note that, we have

$$\begin{aligned}|\mathbf{a}_i^{(t)}| &\leq \left|\mathbf{a}_i^{(t-1)}\right| + \eta(B_{x1}\|\mathbf{w}_i^{(t-1)}\|_2 + \tilde{b}) + \eta\frac{\|\mathbf{w}_i^{(t-1)}\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}} && \text{(B.136)} \\ &\leq \left|\mathbf{a}_i^{(t-1)}\right| + \eta\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)\|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(1)}\|_2 + \eta Z_i, && \text{(B.137)}\end{aligned}$$

where we denote $Z_i = \left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)\|\mathbf{w}_i^{(1)}\|_2 + \tilde{b}$. Similarly, we have

$$\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 \leq \|\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(1)}\|_2 + \eta\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)|\mathbf{a}_i^{(t-1)}|. \tag{B.138}$$

We finish the proof by following the same arguments in the proof of Lemma B.3.7 and union bound. □

Lemma B.3.16 (Bound of Loss Gap and Gradient under Empirical Risk)**.** *Assume the same conditions as in Lemma B.3.15. With probability at least* $1 - O\left(\frac{T}{n}\right)$*, for all* $t \in [T]$*, we have*

$$\begin{aligned}&\left|\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}\right) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right| && \text{(B.139)} \\ \leq & O\left(\frac{\left(\|\tilde{\mathbf{a}}\|_0\|\tilde{\mathbf{a}}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i^{(t)}\|_2 B_x + \tilde{b}) + 1\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right) && \text{(B.140)} \\ & + B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2. && \text{(B.141)}\end{aligned}$$

With probability at least $1 - O\left(\frac{T}{n}\right)$*, for all* $t \in [T]$*,* $i \in [4m]$ *we have*

$$\left|\frac{\partial \widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}(f_{\Xi^{(t)}})}{\partial \mathbf{a}_i^{(t)}}\right| \leq B_{x1}(\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 + \|\mathbf{w}_i^{(1)}\|_2) + \tilde{b} + O\left(\frac{\|\mathbf{w}_i^{(t)}\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right). \quad \text{(B.142)}$$

Proof of Lemma B.3.16. By Lemma B.3.8 and Lemma B.3.12, with probability at least $1 - O\left(\frac{T}{n}\right)$, for all $t \in [T]$, we have

$$\left|\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}\right) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right| \quad \text{(B.143)}$$

$$\leq \left|\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}\right) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})})\right| + \left|\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right| \quad \text{(B.144)}$$

$$\leq O\left(\frac{\left(\|\tilde{\mathbf{a}}\|_0\|\tilde{\mathbf{a}}\|_\infty(\max_{i\in[4m]}\|\mathbf{w}_i^{(t)}\|_2 B_x + \tilde{b}) + 1\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right) \quad \text{(B.145)}$$

$$+ B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2. \quad \text{(B.146)}$$

By Lemma B.3.8 and Lemma B.3.12, with probability at least $1 - O\left(\frac{T}{n}\right)$, for all $t \in [T]$, $i \in [4m]$ we have

$$\left|\frac{\partial \widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}(f_{\Xi^{(t)}})}{\partial \mathbf{a}_i^{(t)}}\right| \leq B_{x1}(\|\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(1)}\|_2 + \|\mathbf{w}_i^{(1)}\|_2) + \tilde{b} + O\left(\frac{\|\mathbf{w}_i^{(t)}\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right). \quad \text{(B.147)}$$

□

We are now ready to prove the main theorem.

Theorem B.3.17 (Online Convex Optimization under Empirical Risk. Full Statement of Theorem 3.2.12)**.** *Consider training by Algorithm 1, and any* $\delta \in (0,1)$*. Assume* $d \geq \log m$*. Set*

$$\sigma_{\mathbf{w}} > 0, \quad \tilde{b} > 0, \quad \eta^{(t)} = \eta,\ \lambda^{(t)} = 0 \text{ for all } t \in \{2,3,\ldots,T\},$$
$$\eta^{(1)} = \Theta\left(\frac{\min\{O(\eta), O(\eta\tilde{b})\}}{-\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d} + \tilde{b})}\right),\ \lambda^{(1)} = \frac{1}{\eta^{(1)}}, \quad \sigma_a = \Theta\left(\frac{\tilde{b}(mp)^{\frac{1}{4}}}{-\ell'(0)\eta^{(1)}B_{x1}\sqrt{B_G B_b}}\right).$$

Let $0 < T\eta B_{x1} \leq o(1)$*,* $m = \Omega\left(\frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right)$ *and* $\frac{n}{\log n} > \Omega\left(\frac{B_x^2}{B_{x2}} + \frac{1}{p} + \left(\frac{1}{B_G^2} + \frac{1}{B_{x1}^2}\right)\frac{B_{x2}}{|\ell'(0)|^2} + \frac{Tm}{\delta}\right)$.

With probability at least $1-\delta$ over the initialization and training samples, there exists $t \in [T]$ such that

$$\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.148}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2\sqrt{2}B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)}\right) \tag{B.149}$$

$$+ \eta\left(\sqrt{r}B_{a2}B_b T\eta B_{x1}^2 + m\tilde{b}\right)O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + 1\right) + O\left(\frac{B_{a2}^2 B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right) \tag{B.150}$$

$$+ \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\left(\left(\frac{rB_{a1}B_b}{\tilde{b}} + m\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + \frac{\tilde{b}}{B_{x1}}\right)\right)\right. \tag{B.151}$$

$$\left.\cdot\left(\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + T\eta^2 B_{x1}\tilde{b}\right)B_x + \tilde{b}\right) + 2\right) \tag{B.152}$$

$$+ \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\left(m\eta\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + T\eta^2 B_{x1}\tilde{b}\right)\sqrt{B_{x2}}\right). \tag{B.153}$$

Furthermore, for any $\epsilon \in (0,1)$, set

$$\tilde{b} = \Theta\left(\frac{B_G^{\frac{1}{4}}B_{a2}B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right), \quad m = \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right),$$

$$\eta = \Theta\left(\frac{\epsilon}{\left(\frac{\sqrt{r}B_{a2}B_b B_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}} + 1\right)}\right), \quad T = \Theta\left(\frac{1}{\eta B_{x1}(mp)^{\frac{1}{4}}}\right),$$

$$\frac{n}{\log n} = \Omega\left(\frac{m^3 p B_x^2 B_{a2}^4 B_b(\log m)^2}{\epsilon^2 r^2 B_{a1}^2 B_G} + \frac{(mp)^{\frac{1}{2}}B_{x2}\log m}{B_b B_G} + \frac{B_x^2}{B_{x2}} + \frac{1}{p} + \left(\frac{1}{B_G^2} + \frac{1}{B_{x1}^2}\right)\frac{B_{x2}}{|\ell'(0)|^2} + \frac{Tm}{\delta}\right),$$

we have there exists $t \in [T]$ with

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}})(\mathbf{x}) \neq y] \leq \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.154}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)} + \epsilon. \tag{B.155}$$

Proof of Theorem B.3.17. We follow the proof in Theorem B.3.9. By $m = \Omega\left(\frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right)$ and $\frac{n}{\log n} > \Omega\left(\frac{B_x^2}{B_{x2}} + \frac{1}{p} + \frac{B_{x2}}{B_G^2|\ell'(0)|^2} + \frac{Tm}{\delta}\right)$, we have $2re^{-\sqrt{\frac{mp}{2}}} + \frac{1}{m^2} + O\left(\frac{Tm}{n}\right) \leq \delta$. For any $B_\epsilon \in (0, B_b)$, when $\sigma_a = \Theta\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)}B_G B_\epsilon}\right)$, by Theorem B.3.5, Lemma B.3.12, Lemma B.3.13, Lemma B.3.16, with probability at least $1-\delta$ over the initialization and training samples, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.156}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\left|\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) - \widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{\Xi^{(t)}}\right)\right| + \frac{1}{T}\sum_{t=1}^{T}\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{\Xi^{(t)}}\right) \tag{B.157}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\left|\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) - \widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{\Xi^{(t)}}\right)\right| + \frac{1}{T}\sum_{t=1}^{T}\left|\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(t)},\mathbf{b})}\right) - \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})})\right| \tag{B.158}$$

$$+ \mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) + \frac{\|\tilde{\mathbf{a}}\|_2^2}{2\eta T} + (2\|\mathbf{a}^{(1)}\|_2\sqrt{m} + 4\eta m)\max_{t\in[T],i\in[4m]}\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}^{(t)}}(f_{\Xi^{(t)}})}{\partial\mathbf{a}_i^{(t)}}\right| \tag{B.159}$$

$$\leq B_{x1}\|\tilde{\mathbf{a}}\|_2\sqrt{\|\tilde{\mathbf{a}}\|_0}\max_{i\in[4m]}\|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 \tag{B.160}$$

$$+ O\left(\frac{\left(\|\tilde{\mathbf{a}}\|_0\|\tilde{\mathbf{a}}\|_\infty + m\|\mathbf{a}^{(T)}\|_\infty)(\max_{i\in[4m]}\|\mathbf{w}_i^{(T)}\|_2 B_x + \tilde{b}) + 2\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right) \tag{B.161}$$

$$+ \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2B_{x1}^2 B_b}{\sqrt{mp}B_G B_\epsilon} + B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)} + B_\epsilon\right) \tag{B.162}$$

$$+ \frac{\|\tilde{\mathbf{a}}\|_2^2}{2\eta T} + 4mB_{x1}(\|\mathbf{a}^{(1)}\|_\infty + \eta) \tag{B.163}$$

$$\cdot\left(\max_{i\in[4m]}\|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 + \max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}} + O\left(\frac{\max_{i\in[4m]}\|\mathbf{w}_i^{(T)}\|_2\sqrt{B_{x2}\log n}}{B_{x1}n^{\frac{1}{2}}}\right)\right).$$

Set $B_\epsilon = \frac{B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{2B_b}{B_G}}$, we have $\sigma_a = \Theta\left(\frac{\tilde{b}(mp)^{\frac{1}{4}}}{-\ell'(0)\eta^{(1)}B_{x1}\sqrt{B_G B_b}}\right)$ which satisfy the requirements. By Lemma B.3.13, Lemma B.3.14, Lemma B.3.15, $\frac{n}{\log n} > \Omega\left(\frac{B_{x2}}{B_{x1}^2|\ell'(0)|^2}\right)$, when

$\eta^{(1)} = \Theta\left(\frac{\min\{O(\eta), O(\eta\tilde{b})\}}{-\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b})}\right)$, we have

$$\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right), \quad \|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right), \quad \|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right) \tag{B.164}$$

$$\|\mathbf{a}^{(1)}\|_\infty = O\left(\eta^{(1)}|\ell'(0)|(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b}) + \eta^{(1)}\frac{\sigma_{\mathbf{w}}\sqrt{dB_{x2}\log n}}{n^{\frac{1}{2}}}\right) \tag{B.165}$$

$$= \min\{O(\eta), O(\eta\tilde{b})\} \tag{B.166}$$

$$\|\mathbf{a}^{(T)}\|_\infty \leq O\left(\|\mathbf{a}^{(1)}\|_\infty + \max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)} + \eta\tilde{b}\right) \tag{B.167}$$

$$\leq O\left(\max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}}\right) \tag{B.168}$$

$$\max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 = O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_G B_\epsilon} + \frac{\tilde{b}\sqrt{\log m}\sqrt{B_{x2}\log n}}{|\ell'(0)|B_G B_\epsilon n^{\frac{1}{2}}}\right) \tag{B.169}$$

$$= O\left(\frac{\tilde{b}\sqrt{\log m}B_{x1}}{B_G B_\epsilon}\right) \tag{B.170}$$

$$= O\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_b B_G}}\right) \tag{B.171}$$

$$\max_{i\in[4m]}\|\mathbf{w}_i^{(T)} - \mathbf{w}_i^{(1)}\|_2 = O\Bigg(T\eta\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)|\mathbf{a}_i^{(1)}| \tag{B.172}$$

$$+ T\eta^2\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)^2\|\mathbf{w}_i^{(1)}\|_2 \tag{B.173}$$

$$+ T\eta^2\left(B_{x1} + \frac{\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right)\tilde{b}\Bigg) \tag{B.174}$$

$$= O\left(T\eta^2 B_{x1}^2\left(\max_{i\in[4m]}\|\mathbf{w}_i^{(1)}\|_2 + \frac{\tilde{b}}{B_{x1}}\right)\right). \tag{B.175}$$

Then, following the proof in Theorem B.3.9, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.176}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2\sqrt{2}B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)}\right) \tag{B.177}$$

$$+ \eta\left(\sqrt{r}B_{a2}B_bT\eta B_{x1}^2 + m\tilde{b}\right)O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right) + O\left(\frac{B_{a2}^2B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right) \tag{B.178}$$

$$+ O\left(\frac{\left((\|\tilde{\mathbf{a}}\|_0\|\tilde{\mathbf{a}}\|_\infty + m\|\mathbf{a}^{(T)}\|_\infty)(\max_{i\in[4m]}\|\mathbf{w}_i^{(T)}\|_2B_x + \tilde{b}) + 2\right)\sqrt{\log n}}{n^{\frac{1}{2}}}\right) \tag{B.179}$$

$$+ O\left(\frac{m\eta\max_{i\in[4m]}\|\mathbf{w}_i^{(T)}\|_2\sqrt{B_{x2}\log n}}{n^{\frac{1}{2}}}\right) \tag{B.180}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2\sqrt{2}B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)}\right) \tag{B.181}$$

$$+ \eta\left(\sqrt{r}B_{a2}B_bT\eta B_{x1}^2 + m\tilde{b}\right)O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right) + O\left(\frac{B_{a2}^2B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right) \tag{B.182}$$

$$+ \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\Bigg(\left(\frac{rB_{a1}B_b}{\tilde{b}} + m\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + \frac{\tilde{b}}{B_{x1}}\right)\right) \tag{B.183}$$

$$\cdot\left(\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + T\eta^2B_{x1}\tilde{b}\right)B_x + \tilde{b}\right) + 2\Bigg) \tag{B.184}$$

$$+ \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\left(m\eta\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + T\eta^2B_{x1}\tilde{b}\right)\sqrt{B_{x2}}\right). \tag{B.185}$$

Furthermore, for any $\epsilon \in (0,1)$, set

$$\tilde{b} = \Theta\left(\frac{B_G^{\frac{1}{4}}B_{a2}B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right), \quad m = \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}B_{x1}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right),$$

$$\eta = \Theta\left(\frac{\epsilon}{\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right)}\right), \quad T = \Theta\left(\frac{1}{\eta B_{x1}(mp)^{\frac{1}{4}}}\right),$$

$$\frac{n}{\log n} = \Omega\left(\frac{m^3pB_x^2B_{a2}^4B_b(\log m)^2}{\epsilon^2r^2B_{a1}^2B_G} + \frac{(mp)^{\frac{1}{2}}B_{x2}\log m}{B_bB_G} + \frac{B_x^2}{B_{x2}} + \frac{1}{p} + \left(\frac{1}{B_G^2} + \frac{1}{B_{x1}^2}\right)\frac{B_{x2}}{|\ell'(0)|^2} + \frac{Tm}{\delta}\right),$$

and note that $B_G \leq B_{x_1} \leq B_x$ and $\sqrt{B_{x_2}} \leq B_x$ naturally, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.186}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2\sqrt{2}B_{x1}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)}\right) \tag{B.187}$$

$$+ \frac{\epsilon}{2} + O\left(\frac{B_{x1}B_{a2}^2B_b^2}{\tilde{b}^2(mp)^{\frac{1}{4}}}\right) + \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\left(\frac{mB_xB_{a2}^2\sqrt{B_b}(mp)^{\frac{1}{2}}\log m}{rB_{a1}\sqrt{B_G}}\right) \tag{B.188}$$

$$+ \frac{\sqrt{\log n}}{n^{\frac{1}{2}}}O\left(\frac{\epsilon\sqrt{B_{x2}\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}}\right) \tag{B.189}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma + O\left(\frac{\sqrt{B_{x2}\log n}}{B_G|\ell'(0)|n^{\frac{1}{2}}}\right)} + \epsilon. \tag{B.190}$$

We finish the proof as the 0-1 classification error is bounded by the loss function, e.g., $\mathbb{I}[\mathrm{sign}(g(\mathbf{x})) \neq y] \leq \frac{\ell(yg(\mathbf{x}))}{\ell(0)}$, where $\ell(0) = 1$.

□

B.4 Applications in Special Cases

We present the case study of linear data in Appendix B.4.1, mixtures of Gaussians in Appendix B.4.2 and Appendix B.4.3, parity functions in Appendix B.4.4, Appendix B.4.5 and Appendix B.4.6, and multiple-index models in Appendix B.4.7.

In special case applications, we consider binary classification with hinge loss, e.g., $\ell(z) = \max\{1 - z, 0\}$. Let $\mathbf{X} = \mathbb{R}^d$ be the input space, and $\mathcal{Y} = \{\pm 1\}$ be the label space.

Remark B.4.1 (Hinge Loss and Logistic Loss). Both hinge loss and logistic loss can be used in special cases and general cases. For convenience, we use hinge loss in special cases, where we can directly get the ground-truth NN close form of the optimal solution which has zero loss. For logistic loss, there is no zero-loss solution. We can still show that the OPT value has an exponentially small upper bound at the cost of more computation.

B.4.1 Linear Data

Data Distributions. Suppose two labels are equiprobable, i.e., $\mathbb{E}[y=-1]=\mathbb{E}[y=+1]=\frac{1}{2}$. The input data are linearly separable and there is a ground truth direction $\mathbf{w}^*$, where $\|\mathbf{w}^*\|_2=1$, such that $y\langle\mathbf{w}^*,\mathbf{x}\rangle>0$. We also assume $\mathbb{E}[yP_{\mathbf{w}^{*\perp}}\mathbf{x}]=0$, where $P_{\mathbf{w}^{*\perp}}$ is the projection operator on the complementary space of the ground truth, i.e., the components of input data being orthogonal with the ground truth are independent of the label y. We define the input data signal level as $\rho:=\mathbb{E}[y\langle\mathbf{w}^*,\mathbf{x}\rangle]>0$ and the margin as $\beta:=\min_{(\mathbf{x},y)}y\langle\mathbf{w}^*,\mathbf{x}\rangle>0$.

We call this data distribution $\mathcal{D}_{linear}$.

Lemma B.4.2 (Linear Data: Gradient Feature Set)**.** *Let $\tilde{b}=d^{\tau}B_{x1}\sigma_{\mathbf{w}}$, where τ is any number large enough to satisfy $d^{\tau/2-\frac{1}{4}}>\Omega\left(\frac{\sqrt{B_{x2}}}{\rho}\right)$. For $\mathcal{D}_{linear}$ setting, we have $(\mathbf{w}^*,-1)\in S_{p,\gamma,B_G}$ where*

$$p=\frac{1}{2},\quad \gamma=\Theta\left(\frac{\sqrt{B_{x2}}}{\rho d^{\tau/2-\frac{1}{4}}}\right),\quad B_G=\rho-\Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right). \tag{B.191}$$

Proof of Lemma B.4.2. By data distribution, we have

$$\mathbb{E}_{(\mathbf{x},y)}[y\mathbf{x}]=\rho\mathbf{w}^*. \tag{B.192}$$

Define $S_{Sure}:\{i\in[m]:\|\mathbf{w}_i^{(0)}\|_2\le 2\sqrt{d}\sigma_{\mathbf{w}}\}$. For all $i\in[m]$, we have

$$\Pr[i\in S_{Sure}]=\Pr[\|\mathbf{w}_i^{(0)}\|_2\le 2\sqrt{d}\sigma_{\mathbf{w}}]\ge\frac{1}{2}. \tag{B.193}$$

For all $i \in S_{Sure}$, by Markov's inequality and considering neuron $i+m$, we have

$$\begin{aligned}
\Pr_{\mathbf{x}}\left[\left\langle \mathbf{w}_{i+m}^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_{i+m} < 0\right] =& \Pr_{\mathbf{x}}\left[\left\langle \mathbf{w}_{i}^{(0)}, \mathbf{x}\right\rangle + \mathbf{b}_{i} < 0\right] && \text{(B.194)}\\
\leq& \Pr_{\mathbf{x}}\left[\|\mathbf{w}_{i}^{(0)}\|_2\|\mathbf{x}\|_2 \geq \mathbf{b}_i\right] && \text{(B.195)}\\
\leq& \Pr_{\mathbf{x}}\left[\|\mathbf{x}\|_2 \geq \frac{d^{\tau-\frac{1}{2}}B_{x1}}{2}\right] && \text{(B.196)}\\
\leq& \Theta\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right). && \text{(B.197)}
\end{aligned}$$

For all $i \in S_{Sure}$, by Hölder's inequality, we have

$$\begin{aligned}
&\left\|\mathbb{E}_{(\mathbf{x},y)}\left[y\left(1-\sigma'\left[\left\langle \mathbf{w}_{i+m}^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_{i+m}\right]\right)\mathbf{x}\right]\right\|_2 && \text{(B.198)}\\
=&\left\|\mathbb{E}_{(\mathbf{x},y)}\left[y\left(1-\sigma'\left[\left\langle \mathbf{w}_{i}^{(0)}, \mathbf{x}\right\rangle + \mathbf{b}_{i}\right]\right)\mathbf{x}\right]\right\|_2 && \text{(B.199)}\\
\leq&\sqrt{\mathbb{E}[\|\mathbf{x}\|_2^2]\mathbb{E}\left[\left(1-\sigma'\left[\left\langle \mathbf{w}_{i}^{(0)}, \mathbf{x}\right\rangle + \mathbf{b}_{i}\right]\right)^2\right]} && \text{(B.200)}\\
\leq&\Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right). && \text{(B.201)}
\end{aligned}$$

We have

$$\begin{aligned}
1-\frac{\left|\left\langle G(\mathbf{w}_{i+m}^{(0)}, \mathbf{b}_{i+m}), \mathbf{w}^*\right\rangle\right|}{\|G(\mathbf{w}_{i+m}^{(0)}, \mathbf{b}_{i+m})\|_2} =& 1-\frac{\left|\left\langle G(\mathbf{w}_{i}^{(0)}, -\mathbf{b}_{i}), \mathbf{w}^*\right\rangle\right|}{\|G(\mathbf{w}_{i}^{(0)}, -\mathbf{b}_{i})\|_2} && \text{(B.202)}\\
\leq& 1-\frac{\rho-\Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right)}{\rho+\Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right)} && \text{(B.203)}\\
=& \Theta\left(\frac{\sqrt{B_{x2}}}{\rho d^{\tau/2-\frac{1}{4}}}\right) = \gamma. && \text{(B.204)}
\end{aligned}$$

We finish the proof by $\frac{\mathbf{b}_{i+m}}{|\mathbf{b}_{i+m}|} = -1$. □

Lemma B.4.3 (Linear Data: Existence of Good Networks)**.** *Assume the same conditions*

as in Lemma B.4.2. Define

$$g^*(\mathbf{x}) = \frac{1}{\beta}\sigma(\langle \mathbf{w}^*, \mathbf{x}\rangle) - \frac{1}{\beta}\sigma(\langle -\mathbf{w}^*, \mathbf{x}\rangle). \tag{B.205}$$

For $\mathcal{D}_{linear}$ setting, we have $g^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$, where $r = 2, B_F = (B_{a1}, B_{a2}, B_b) = \left(\frac{1}{\beta}, \frac{\sqrt{2}}{\beta}, \frac{1}{B_{x1}^2}\right)$, $p = \frac{1}{2}$, $\gamma = \Theta\left(\frac{\sqrt{B_{x2}}}{\rho d^{\tau/2-\frac{1}{4}}}\right)$, $B_G = \rho - \Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right)$. We also have $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} = 0$.

Proof of Lemma B.4.3. By Lemma B.4.2 and Lemma B.5.3, we have $g^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$. We also have

$$\begin{aligned}
\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} \leq & \mathcal{L}_{\mathcal{D}_{linear}}(g^*) && \text{(B.206)}\\
= & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{linear}}\mathcal{L}_{(\mathbf{x},y)}(g^*) && \text{(B.207)}\\
= & 0. && \text{(B.208)}
\end{aligned}$$

□

Theorem B.4.4 (Linear Data: Main Result)**.** *For $\mathcal{D}_{linear}$ setting, for any $\delta \in (0,1)$ and for any $\epsilon \in (0,1)$ when*

$$m = \mathrm{poly}\left(\frac{1}{\delta}, \frac{1}{\epsilon}, \frac{1}{\beta}, \frac{1}{\rho}\right) \leq e^d, \quad T = \mathrm{poly}\left(m, B_{x1}\right), \quad n = \mathrm{poly}\left(m, B_x, \frac{1}{\delta}, \frac{1}{\epsilon}, \frac{1}{\beta}, \frac{1}{\rho}\right), \tag{B.209}$$

trained by Algorithm 1 with hinge loss, with probability at least $1-\delta$ over the initialization, with proper hyper-parameters, there exists $t \in [T]$ such that

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \epsilon. \tag{B.210}$$

Proof of Theorem B.4.4. Let $\tilde{b} = d^\tau B_{x1}\sigma_{\mathbf{w}}$, where τ is a number large enough to satisfy $d^{\tau/2-\frac{1}{4}} > \Omega\left(\frac{\sqrt{B_{x2}}}{\rho}\right)$ and $O\left(\frac{B_{x1}B_{x2}^{\frac{1}{4}}}{\beta\sqrt{\rho}d^{\tau/4-\frac{1}{8}}}\right) \leq \frac{\epsilon}{2}$. By Lemma B.4.3, we have $g^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$, where $r = 2, B_F = (B_{a1}, B_{a2}, B_b) = \left(\frac{1}{\beta}, \frac{\sqrt{2}}{\beta}, \frac{1}{B_{x1}^2}\right)$, $p = \frac{1}{2}$, $\gamma = \Theta\left(\frac{\sqrt{B_{x2}}}{\rho d^{\tau/2-\frac{1}{4}}}\right)$,

$B_G = \rho - \Theta\left(\frac{\sqrt{B_{x2}}}{d^{\tau/2-\frac{1}{4}}}\right)$. We also have $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma},B_G} = 0$.

Adjust $\sigma_{\mathbf{w}}$ such that $\tilde{b} = d^{\tau} B_{x1}\sigma_{\mathbf{w}} = \Theta\left(\frac{B_G^{\frac{1}{4}} B_{a2} B_b^{\frac{3}{4}}}{\sqrt{r B_{a1}}}\right)$. Injecting above parameters into Theorem 3.2.12, we have with probability at least $1-\delta$ over the initialization, with proper hyper-parameters, there exists $t \in [T]$ such that

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq O\left(\frac{B_{x1} B_{x2}^{\frac{1}{4}}}{\beta\sqrt{\rho} d^{\tau/4-\frac{1}{8}}}\right) + O\left(\frac{B_{x1} B_{x2}{}^{\frac{1}{4}} (\log n)^{\frac{1}{4}}}{\beta\sqrt{\rho} n^{\frac{1}{4}}}\right) + \epsilon/2 \leq \epsilon. \quad \text{(B.211)}$$

□

B.4.2 Mixture of Gaussians

We recap the problem setup in Section 3.3.1 for readers' convenience.

Problem Setup

Data Distributions. We follow the notations from [227]. The data are from a mixture of r high-dimensional Gaussians, and each Gaussian is assigned to one of two possible labels in $\mathcal{Y} = \{\pm 1\}$. Let $\mathcal{S}(y) \subseteq [r]$ denote the set of indices of the Gaussians associated with the label y. The data distribution is then:

$$q(\mathbf{x}, y) = q(y)q(\mathbf{x}|y), \quad q(\mathbf{x}|y) = \sum_{j \in \mathcal{S}(y)} p_j \mathcal{N}_j(\mathbf{x}), \quad \text{(B.212)}$$

where $\mathcal{N}_j(\mathbf{x})$ is a multivariate normal distribution with mean μ_j and covariance Σ_j, and p_j are chosen such that $q(\mathbf{x}, y)$ is correctly normalized.

We call this data distribution $\mathcal{D}_{mixture}$.

We will make some assumptions about the Gaussians, for which we first introduce some notations. For all $j \in [r]$, let $y_{(j)} \in \{+1, -1\}$ be the label for $\mathcal{N}_j(\mathbf{x})$.

$$D_j := \frac{\mu_j}{\|\mu_j\|_2}, \quad \tilde{\mu}_j := \mu_j/\sqrt{d}, \quad B_{\mu 1} := \min_{j\in[r]} \|\tilde{\mu}_j\|_2, \quad B_{\mu 2} := \max_{j\in[r]} \|\tilde{\mu}_j\|_2, \quad p_B := \min_{j\in[r]} p_j.$$

Assumption B.4.5 (Mixture of Gaussians. Recap of Assumption 3.3.1)**.** Let $8 \leq \tau \leq d$ be a parameter that will control our final error guarantee. Assume

- Equiprobable labels: $q(-1) = q(+1) = 1/2$.
- For all $j \in [r]$, $\Sigma_j = \sigma_j I_{d\times d}$. Let $\sigma_B := \max_{j\in[r]} \sigma_j$ and $\sigma_{B+} := \max\{\sigma_B, B_{\mu 2}\}$.
- $r \leq 2d, \quad p_B \geq \frac{1}{2d}, \quad \Omega\left(\frac{1}{d} + \sqrt{\frac{\tau {\sigma_{B+}}^2 \log d}{d}}\right) \leq B_{\mu 1} \leq B_{\mu 2} \leq d.$
- The Gaussians are well-separated: for all $i \neq j \in [r]$, we have $-1 \leq \langle D_i, D_j\rangle \leq \theta$, where $0 \leq \theta \leq \min\left\{\frac{1}{2r}, \frac{\sigma_{B+}}{B_{\mu 2}}\sqrt{\frac{\tau \log d}{d}}\right\}$.

Below, we define a sufficient condition that randomly initialized weights will fall in nice gradients set after the first gradient step update.

Definition B.4.6 (Mixture of Gaussians: Subset of Nice Gradients Set)**.** Recall $\mathbf{w}_i^{(0)}$ is the weight for the i-th neuron at initialization. For all $j \in [r]$, let $S_{D_j,Sure} \subseteq [m]$ be those neurons that satisfy

- $\left\langle \mathbf{w}_i^{(0)}, \mu_j \right\rangle \geq C_{Sure,1}\mathbf{b}_i$,
- $\left\langle \mathbf{w}_i^{(0)}, \mu_{j'} \right\rangle \leq C_{Sure,2}\mathbf{b}_i$, for all $j' \neq j, j' \in [r]$.
- $\left\|\mathbf{w}_i^{(0)}\right\|_2 \leq \Theta(\sqrt{d}\sigma_{\mathbf{w}})$.

Mixture of Gaussians: Feature Learning

We show the important Lemma B.4.7 first and defer other Lemmas after it.

Lemma B.4.7 (Mixture of Gaussians: Gradient Feature Set. Part statement of Lemma 3.3.3)**.** *Let $C_{Sure,1} = \frac{3}{2}$, $C_{Sure,2} = \frac{1}{2}$, $\tilde{b} = C_b\sqrt{\tau d \log d}\sigma_{\mathbf{w}}\sigma_{B+}$, where C_b is a large enough universal constant. For $\mathcal{D}_{mixture}$ setting, we have $(D_j, +1) \in S_{p,\gamma,B_G}$ for all $j \in [r]$, where*

$$p = \Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\left(9C_b^2 \tau {\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right), \quad \gamma = \frac{1}{d^{0.9\tau - 1.5}}, \tag{B.213}$$

$$B_G = p_B B_{\mu 1}\sqrt{d} - O\left(\frac{\sigma_{B+}}{d^{0.9\tau}}\right). \tag{B.214}$$

Proof of Lemma B.4.7. For all $j \in [r]$, by Lemma B.4.10, for all $i \in S_{D_j,Sure}$,

$$1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right|}{\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2} \tag{B.215}$$

$$\leq 1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right|}{\sqrt{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right|^2 + \max_{D_j^\top D_j^\perp = 0, \|D_j^\perp\|_2 = 1} \left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j^\perp \right\rangle\right|^2}} \tag{B.216}$$

$$\leq 1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right|}{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right| + \max_{D_j^\top D_j^\perp = 0, \|D_j^\perp\|_2 = 1} \left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j^\perp \right\rangle\right|} \tag{B.217}$$

$$\leq 1 - \frac{1}{1 + \frac{B_{\mu 2} O\left(\frac{1}{d^{\tau - \frac{1}{2}}}\right) + \sigma_{B+} O\left(\frac{1}{d^{0.9\tau}}\right)}{p_j B_{\mu 1} \sqrt{d}\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - B_{\mu 2} O\left(\frac{1}{d^{\tau - \frac{1}{2}}}\right) - \sigma_{B+} O\left(\frac{1}{d^{0.9\tau}}\right)}} \tag{B.218}$$

$$\leq \frac{\sigma_{B+} O\left(\frac{1}{d^{0.9\tau}}\right)}{p_j B_{\mu 1} \sqrt{d} - \sigma_{B+} O\left(\frac{1}{d^{0.9\tau}}\right)} \tag{B.219}$$

$$< \frac{1}{d^{0.9\tau - 1.5}} = \gamma, \tag{B.220}$$

where the last inequality follows $B_{\mu 1} \geq \Omega\left(\sigma_{B+} \sqrt{\frac{\tau \log d}{d}}\right)$.

Thus, we have $G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \in \mathcal{C}_{D_j,\gamma}$ and $\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right| \leq \|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \leq B_{x1}$, $\frac{\mathbf{b}_i}{|\mathbf{b}_i|} = +1$. Thus, by Lemma B.4.8, we have

$$\Pr_{\mathbf{w},b}\left[G(\mathbf{w}, b) \in \mathcal{C}_{D_j,\gamma} \text{ and } \|G(\mathbf{w}, b)\|_2 \geq B_G \text{ and } \frac{b}{|b|} = +1\right] \tag{B.221}$$

$$\geq \Pr\left[i \in S_{D_j,Sure}\right] \tag{B.222}$$

$$\geq p. \tag{B.223}$$

Thus, $(D_j, +1) \in S_{p,\gamma,B_G}$. We finish the proof. □

Below are Lemmas used in the proof of Lemma B.4.7. In Lemma B.4.8, we calculate p used in S_{p,γ,B_G}.

Lemma B.4.8 (Mixture of Gaussians: Geometry at Initialization. Lemma B.2 in [10])**.** *Assume the same conditions as in Lemma B.4.7, recall for all $i \in [m]$, $\mathbf{w}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d \times d})$,*

over the random initialization, we have for all $i \in [m], j \in [r]$,

$$\Pr\left[i \in S_{D_j,Sure}\right] \geq \Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\left(9C_b^2 \tau {\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right). \tag{B.224}$$

Proof of Lemma B.4.8. Recall for all $l \in [r]$, $\tilde{\mu}_l = \mu_l/\sqrt{d}$.

WLOG, let $j = r$. For all $l \in [r-1]$. We define $Z_1 = \{l \in [r-1] : \langle D_l, D_r\rangle \geq -\theta\}$ and $Z_2 = \{l \in [r-1] : -1 < \langle D_l, D_r\rangle < -\theta\}$. WLOG, let $Z_1 = [r_1]$, $Z_2 = \{r_1+1, \ldots, r_2\}$, where $0 \leq r_1 \leq r_2 \leq r-1$. We define the following events

$$\zeta_l = \left\{\left\langle \mathbf{w}_i^{(0)}, \mu_l\right\rangle \leq C_{Sure,2}\mathbf{b}_i\right\}, \hat{\zeta}_l = \left\{\left|\left\langle \mathbf{w}_i^{(0)}, \mu_l\right\rangle\right| \leq C_{Sure,2}\mathbf{b}_i\right\}. \tag{B.225}$$

We define space $A = \text{span}(\mu_1, \ldots, \mu_{r_1})$ and $\hat{\mu}_r = P_{A^\perp}\mu_r$, where $P_{A^\perp}$ is the projection operator on the complementary space of A. For $l \in Z_2$, we also define $\dot{\mu}_l = \mu_l - \frac{\langle \mu_l, \mu_r\rangle \mu_r}{\|\mu_r\|_2^2}$, and the event

$$\dot{\zeta}_l = \left\{\left\langle \mathbf{w}_i^{(0)}, \dot{\mu}_l\right\rangle \leq C_{Sure,2}\mathbf{b}_i\right\}, \hat{\dot{\zeta}}_l = \left\{\left|\left\langle \mathbf{w}_i^{(0)}, \dot{\mu}_l\right\rangle\right| \leq C_{Sure,2}\mathbf{b}_i\right\}. \tag{B.226}$$

For $l \in Z_2$, we have $\mu_l = \dot{\mu}_l - \rho\mu_r$, where $\rho \geq 0$. So $\langle \mathbf{w}, \mu_l\rangle = \langle \mathbf{w}, \dot{\mu}_l\rangle - \rho\langle \mathbf{w}, \mu_r\rangle \leq \langle \mathbf{w}, \dot{\mu}_l\rangle$ when $\langle \mathbf{w}, \mu_r\rangle \geq 0$. As a result, we have

$$\dot{\zeta}_l \cap \left\{\left\langle \mathbf{w}_i^{(0)}, \mu_r\right\rangle \geq C_{Sure,1}\mathbf{b}_i\right\} \subseteq \zeta_l \cap \left\{\left\langle \mathbf{w}_i^{(0)}, \mu_r\right\rangle \geq C_{Sure,1}\mathbf{b}_i\right\}. \tag{B.227}$$

By Assumption 3.3.1, we have

$$\frac{1}{2} \leq 1 - r\theta \leq 1 - r_1\theta \leq \frac{\|\hat{\mu}_r\|_2}{\|\mu_r\|_2} \leq 1. \tag{B.228}$$

We also have,

$$
\begin{aligned}
&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i, \zeta_1, \ldots, \zeta_{r-1}\right] && \text{(B.229)}\\
=&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i, \zeta_1, \ldots, \zeta_{r_2}\right] && \text{(B.230)}\\
\geq&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i, \zeta_1, \ldots, \zeta_{r_1}, \dot{\zeta}_{r_1+1}, \ldots, \dot{\zeta}_{r_2}\right] && \text{(B.231)}\\
\geq&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i, \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}, \hat{\dot{\zeta}}_{r_1+1}, \ldots, \hat{\dot{\zeta}}_{r_2}\right] && \text{(B.232)}\\
=&\underbrace{\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}, \hat{\dot{\zeta}}_{r_1+1}, \ldots, \hat{\dot{\zeta}}_{r_2}\right]}_{p_r} \underbrace{\Pr\left[\hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}, \hat{\dot{\zeta}}_{r_1+1}, \ldots, \hat{\dot{\zeta}}_{r_2}\right]}_{\Pi_{l\in[r_2]}p_l}.
\end{aligned}
$$

For the first condition in Definition B.4.6, we have,

$$
\begin{aligned}
p_r =&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}, \hat{\dot{\zeta}}_{r_1+1}, \ldots, \hat{\dot{\zeta}}_{r_2}\right] && \text{(B.233)}\\
=&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r + \mu_r - \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.234)}\\
\geq&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r + \mu_r - \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i,\ \left\langle \mathbf{w}_i^{(0)}, \mu_r - \hat{\mu}_r \right\rangle \geq 0 \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.235)}\\
=&\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r + \mu_r - \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle|\ \left\langle \mathbf{w}_i^{(0)}, \mu_r - \hat{\mu}_r \right\rangle \geq 0, \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.236)}\\
&\cdot \Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_r - \hat{\mu}_r \right\rangle \geq 0 \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.237)}\\
=&\frac{1}{2}\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r + \mu_r - \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle| \left\langle \mathbf{w}_i^{(0)}, \mu_r - \hat{\mu}_r \right\rangle \geq 0, \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.238)}\\
\geq&\frac{1}{2}\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i \middle| \left\langle \mathbf{w}_i^{(0)}, \mu_r - \hat{\mu}_r \right\rangle \geq 0, \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}\right] && \text{(B.239)}\\
=&\frac{1}{2}\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \hat{\mu}_r \right\rangle \geq C_{Sure,1}\mathbf{b}_i\right] && \text{(B.240)}\\
\geq&\Theta\left(\frac{\|\tilde{\mu}_r\|_2}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(9C_b^2{\tau\sigma_{B+}}^2/(2\|\tilde{\mu}_r\|_2^2)\right)}}\right), && \text{(B.241)}
\end{aligned}
$$

where the last equality following that $\hat{\mu}_r$ is orthogonal with $\mu_1, \ldots, \mu_{r_1}$ and the property of the standard Gaussian vector, and the last inequality follows Lemma B.5.6.

For the second condition in Definition B.4.6, by Lemma B.5.6, we have,

$$p_1 = \Pr\left[\hat{\zeta}_1\right] = 1 - \Theta\left(\frac{\|\tilde{\mu}_1\|_2}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\left(C_b^2 \tau {\sigma_{B+}}^2/(8\|\tilde{\mu}_1\|_2^2)\right)}}\right) \tag{B.242}$$

$$p_2 = \Pr\left[\hat{\zeta}_2 \middle| \hat{\zeta}_1\right] \geq \Pr\left[\hat{\zeta}_2\right] \geq 1 - \Theta\left(\frac{\|\tilde{\mu}_2\|_2}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\left(C_b^2 \tau {\sigma_{B+}}^2/(8\|\tilde{\mu}_2\|_2^2)\right)}}\right) \tag{B.243}$$

$$\vdots \tag{B.244}$$

$$p_{r-1} = \Pr\left[\hat{\zeta}_{r_2} \middle| \hat{\zeta}_1, \ldots, \hat{\zeta}_{r_1}, \hat{\zeta}_{r_1+1}, \ldots, \hat{\zeta}_{r_2}\right] \geq \Pr\left[\hat{\zeta}_{r_2}\right] \geq \Pr\left[\hat{\zeta}_{r_2}\right] \tag{B.245}$$

$$\geq 1 - \Theta\left(\frac{\|\tilde{\mu}_{r-1}\|_2}{\sqrt{\tau \log d}\sigma_{B+} \cdot d^{\left(C_b^2 \tau {\sigma_{B+}}^2/(8\|\tilde{\mu}_{r_2}\|_2^2)\right)}}\right). \tag{B.246}$$

On the other hand, if X is a $\chi^2(k)$ random variable. Then we have

$$\Pr(X \geq k + 2\sqrt{kx} + 2x) \leq e^{-x}. \tag{B.247}$$

Therefore, by assumption $B_{\mu 1} \geq \Omega\left(\sigma_{B+}\sqrt{\frac{\tau \log d}{d}}\right)$, we have

$$\Pr\left(\frac{1}{\sigma_{\mathbf{w}}^2}\left\|\mathbf{w}_i^{(0)}\right\|_2^2 \geq d + 2\sqrt{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2) + 2\right) d \log d}\right. \tag{B.248}$$

$$\left. + 2\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2) + 2\right)\log d\right) \tag{B.249}$$

$$\leq O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right). \tag{B.250}$$

Recall $B_{\mu 1} = \min_{j\in[r]} \|\tilde{\mu}_j\|_2, B_{\mu 2} = \max_{j\in[r]} \|\tilde{\mu}_j\|_2$. Thus, by union bound, we have

$$\begin{aligned}
&\Pr\left[i \in S_{D_j,Sure}\right] && \text{(B.251)}\\
\geq&\Pi_{l\in[r]} p_l - O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right) && \text{(B.252)}\\
\geq&\Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\cdot\left(1-\frac{rB_{\mu 2}}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(C_b^2\tau{\sigma_{B+}}^2/(8B_{\mu 2}^2)\right)}}\right)\right) && \text{(B.253)}\\
&- O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right) && \text{(B.254)}\\
\geq&\Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right). && \text{(B.255)}
\end{aligned}$$

□

In Lemma B.4.9, we compute the activation pattern for the neurons in $S_{D_j,Sure}$.

Lemma B.4.9 (Mixture of Gaussians: Activation Pattern)**.** *Assume the same conditions as in Lemma B.4.7, for all $j \in [r], i \in S_{D_j,Sure}$, we have*

(1) When $\mathbf{x} \sim \mathcal{N}_j(\mu_j, \sigma_j I_{d\times d})$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i \geq 0\right] \geq 1 - O\left(\frac{1}{d^\tau}\right). \tag{B.256}$$

(2) For all $j' \neq j, j' \in [r]$, when $\mathbf{x} \sim \mathcal{N}_{j'}(\mu_{j'}, \Sigma_{j'})$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathcal{N}_{j'}(\mu_{j'},\sigma_{j'} I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i \geq 0\right] \leq O\left(\frac{1}{d^\tau}\right). \tag{B.257}$$

Proof of Lemma B.4.9. In the proof, we need $\tilde{b} = C_b\sqrt{\tau d\log d}\sigma_{\mathbf{w}}\sigma_{B+}$, where C_b is a large enough universal constant. For the first statement, when $\mathbf{x} \sim \mathcal{N}_j(\mu_j, \sigma_j I_{d\times d})$, by $C_{Sure,1} \geq \frac{3}{2}$,

we have

$$\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle - \mathbf{b}_i \geq 0\right] \geq \Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle \geq (1-C_{Sure,1})\mathbf{b}_i\right] \tag{B.258}$$

$$\geq \Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle \geq -\frac{\mathbf{b}_i}{2}\right] \tag{B.259}$$

$$=1-\Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle \leq -\frac{\mathbf{b}_i}{2}\right] \tag{B.260}$$

$$\geq 1-\exp\left(-\frac{{\mathbf{b}_i}^2}{\Theta(d\sigma_{\mathbf{w}}^2\sigma_j^2)}\right) \tag{B.261}$$

$$\geq 1-O\left(\frac{1}{d^\tau}\right), \tag{B.262}$$

where the third inequality follows the Chernoff bound and symmetricity of the Gaussian vector.

For the second statement, we prove similarly by $0 < C_{Sure,2} \leq \frac{1}{2}$. □

Then, Lemma B.4.10 gives gradients of neurons in $S_{D_j,Sure}$. It shows that these gradients are highly aligned with D_j.

Lemma B.4.10 (Mixture of Gaussians: Feature Emergence)**.** *Assume the same conditions as in Lemma B.4.7, for all $j \in [r]$, $i \in S_{D_j,Sure}$, we have*

$$\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], y_{(j)}D_j\right\rangle \tag{B.263}$$

$$\geq p_j B_{\mu 1}\sqrt{d}\left(1-O\left(\frac{1}{d^\tau}\right)\right) - B_{\mu 2}O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) - \sigma_{B+}O\left(\frac{1}{d^{0.9\tau}}\right). \tag{B.264}$$

For any unit vector $D_j^\perp$ which is orthogonal with D_j, we have

$$\left|\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], D_j^\perp\right\rangle\right| \leq B_{\mu 2}O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) + \sigma_{B+}O\left(\frac{1}{d^{0.9\tau}}\right). \tag{B.265}$$

Proof of Lemma B.4.10. For all $j \in [r]$, $i \in S_{D_j,Sure}$, we have

$$\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right] \tag{B.266}$$

$$= \sum_{l\in[r]} p_l \mathbb{E}_{\mathbf{x}\sim\mathcal{N}_l(\mathbf{x})}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right] \tag{B.267}$$

$$= \sum_{l\in[r]} p_l y_{(l)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_l)\right]. \tag{B.268}$$

Thus, by Lemma B.5.7 and Lemma B.4.9,

$$\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], y_{(j)}D_j\right\rangle \tag{B.269}$$

$$= p_j \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_j)^\top D_j\right] \tag{B.270}$$

$$+ \sum_{l\in[r],l\neq j} p_l y_{(l)} y_{(j)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_l)^\top D_j\right] \tag{B.271}$$

$$\geq p_j \mu_j^\top D_j\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - \sum_{l\in[r],l\neq j} p_l |\mu_l^\top D_j| O\left(\frac{1}{d^\tau}\right) \tag{B.272}$$

$$- p_j\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.273}$$

$$- \sum_{l\in[r],l\neq j} p_l\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.274}$$

$$\geq p_j B_{\mu 1}\sqrt{d}\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - B_{\mu 2} O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) \tag{B.275}$$

$$- p_j\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I)}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right) - 1\right)\mathbf{x}^\top D_j\right]\right| \tag{B.276}$$

$$- \sum_{l\in[r],l\neq j} p_l\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.277}$$

$$= p_j B_{\mu 1}\sqrt{d}\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - B_{\mu 2} O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) \tag{B.278}$$

$$- p_j\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I)}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)\right)\mathbf{x}^\top D_j\right]\right| \tag{B.279}$$

$$- \sum_{l\in[r],l\neq j} p_l\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.280}$$

$$\geq p_j B_{\mu 1}\sqrt{d}\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - B_{\mu 2} O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) - \sigma_{B+} O\left(\frac{1}{d^{0.9\tau}}\right). \tag{B.281}$$

For any unit vector $D_j^{\perp}$ which is orthogonal with D_j, similarly, we have

$$\begin{aligned}
&\left|\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\mathbf{x}\right],D_j^{\perp}\right\rangle\right| && \text{(B.282)}\\
\leq& p_j\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_j\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| && \text{(B.283)}\\
&+\sum_{l\in[r],l\neq j} p_l\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_l\right\rangle-\mathbf{b}_i\right)(\mathbf{x}+\mu_l)^\top D_j^{\perp}\right]\right| && \text{(B.284)}\\
\leq& B_{\mu 2}O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right)+p_j\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_j I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_j\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| && \text{(B.285)}\\
&+\sum_{l\in[r],l\neq j} p_l\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_l\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| && \text{(B.286)}\\
\leq& B_{\mu 2}O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right)+\sigma_{B+}O\left(\frac{1}{d^{0.9\tau}}\right). && \text{(B.287)}
\end{aligned}$$

□

Mixture of Gaussians: Final Guarantee

Lemma B.4.11 (Mixture of Gaussians: Existence of Good Networks. Part statement of Lemma 3.3.3)**.** *Assume the same conditions as in Lemma B.4.7. Define*

$$g^*(\mathbf{x})=\sum_{j=1}^{r}\frac{y_{(j)}}{\sqrt{\tau\log d}\sigma_{B+}}\left[\sigma\left(\langle D_j,\mathbf{x}\rangle-2\sqrt{\tau\log d}\sigma_{B+}\right)\right]. \tag{B.288}$$

*For $\mathcal{D}_{mixture}$ setting, we have $g^*\in\mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$, where $B_F=(B_{a1},B_{a2},B_b)=\left(\frac{1}{\sqrt{\tau\log d}\sigma_{B+}},\frac{\sqrt{r}}{\sqrt{\tau\log d}\sigma_{B+}},2\sqrt{\tau\log$
$p=\Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(9C_b^2{\tau\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right)$, $\gamma=\frac{1}{d^{0.9\tau-1.5}}$, $B_G=p_B B_{\mu 1}\sqrt{d}-O\left(\frac{\sigma_{B+}}{d^{0.9\tau}}\right)$ and $B_{x1}=$
$(B_{\mu 2}+\sigma_{B+})\sqrt{d}, B_{x2}=(B_{\mu 2}+\sigma_{B+})^2 d$. We also have $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}\leq\frac{3}{d^{\tau}}+\frac{4}{d^{0.9\tau-1}\sqrt{\tau\log d}}$.*

Proof of Lemma B.4.11. We can check $B_{x1}=(B_{\mu 2}+\sigma_{B+})\sqrt{d}, B_{x2}=(B_{\mu 2}+\sigma_{B+})^2 d$ by direct calculation. By Lemma B.4.7, we have $g^*\in\mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$.

For any $j \in [r]$, by $B_{\mu 1} \geq \Omega\left(\sigma_{B+}\sqrt{\frac{\tau \log d}{d}}\right) \geq 4\sigma_{B+}\sqrt{\frac{\tau \log d}{d}}$, we have

$$\begin{aligned}
&\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\langle D_j, \mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} \geq \sqrt{\tau\log d}\sigma_{B+}\right] && \text{(B.289)}\\
=&\Pr_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})}\left[\langle D_j, \mathbf{x}\rangle + \|\mu_j\|_2 - 2\sqrt{\tau\log d}\sigma_{B+} \geq \sqrt{\tau\log d}\sigma_{B+}\right] && \text{(B.290)}\\
\geq&\Pr_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})}\left[\langle D_j, \mathbf{x}\rangle + \sqrt{d}B_{\mu 1} - 2\sqrt{\tau\log d}\sigma_{B+} \geq \sqrt{\tau\log d}\sigma_{B+}\right] && \text{(B.291)}\\
\geq&\Pr_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})}\left[\langle D_j, \mathbf{x}\rangle \geq -\sqrt{\tau\log d}\sigma_{B+}\right] && \text{(B.292)}\\
\geq&1-\frac{1}{d^\tau}, && \text{(B.293)}
\end{aligned}$$

where the last inequality follows Chernoff bound.

For any $l \neq j, l \in [r]$, by $\theta \leq \frac{\sigma_{B+}}{B_{\mu 2}}\sqrt{\frac{\tau\log d}{d}}$, we have

$$\begin{aligned}
&\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\langle D_l, \mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} \geq 0\right] && \text{(B.294)}\\
\leq&\Pr_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})}\left[\langle D_l, \mathbf{x}\rangle + \theta B_{\mu 2}\sqrt{d} - 2\sqrt{\tau\log d}\sigma_{B+} \geq 0\right] && \text{(B.295)}\\
\leq&\Pr_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})}\left[\langle D_l, \mathbf{x}\rangle \geq \sqrt{\tau\log d}\sigma_{B+}\right] && \text{(B.296)}\\
\leq&\frac{1}{d^\tau}. && \text{(B.297)}
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\Pr_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[yg^*(\mathbf{x}) > 1] && \text{(B.298)}\\
\geq&\sum_{j\in[r]} p_j\left(\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\langle D_j, \mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} \geq \sqrt{\tau\log d}\sigma_{B+}\right]\right) && \text{(B.299)}\\
&-\sum_{j\in[r]} p_j\left(\sum_{l\neq j, l\in[r]}\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})}\left[\langle D_l, \mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} < 0\right]\right) && \text{(B.300)}\\
\geq&1-\frac{2}{d^\tau}. && \text{(B.301)}
\end{aligned}$$

We also have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[\mathbb{I}[yg^*(\mathbf{x})\le 1]|yg^*(\mathbf{x})|] \tag{B.302}$$

$$\le \sum_{j\in[r]} p_j \left(\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})} \left[\langle D_j,\mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} < \sqrt{\tau\log d}\sigma_{B+}\right] \frac{y_{(j)}^2\sqrt{\tau\log d}\sigma_{B+}}{\sqrt{\tau\log d}\sigma_{B+}} \right)$$

$$+ \sum_{j\in[r]} p_j \left(\sum_{l\neq j, l\in[r]} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_j I_{d\times d})} \left[\sigma'\left[\langle D_l,\mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+} > 0\right] \frac{\langle D_l,\mathbf{x}\rangle - 2\sqrt{\tau\log d}\sigma_{B+}}{\sqrt{\tau\log d}\sigma_{B+}}\right] \right)$$

$$\le \frac{1}{d^\tau} + \sum_{j\in[r]} p_j \left(\sum_{l\neq j, l\in[r]} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}_j(0,\sigma_j I_{d\times d})} \left[\sigma'\left[\langle D_l,\mathbf{x}\rangle > \sqrt{\tau\log d}\sigma_{B+}\right] \frac{\langle D_l,\mathbf{x}\rangle - \sqrt{\tau\log d}\sigma_{B+}}{\sqrt{\tau\log d}\sigma_{B+}}\right] \right)$$

$$\le \frac{1}{d^\tau} + \frac{1}{\sqrt{\tau\log d}} \sum_{j\in[r]} p_j \left(\sum_{l\neq j, l\in[r]} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}_j(0,I_{d\times d})} \left[\sigma'\left[\langle D_l,\mathbf{x}\rangle > \sqrt{\tau\log d}\right] \langle D_l,\mathbf{x}\rangle\right] \right) \tag{B.303}$$

$$\le \frac{1}{d^\tau} + \frac{4}{d^{0.9\tau-1}\sqrt{\tau\log d}}, \tag{B.304}$$

where the second last inequality follows Lemma B.5.7 and $r \le 2d$. Thus, we have

$$\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} \le \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[\ell(yg^*(\mathbf{x}))] \tag{B.305}$$

$$= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[\mathbb{I}[yg^*(\mathbf{x})\le 1](1-yg^*(\mathbf{x}))] \tag{B.306}$$

$$\le \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[\mathbb{I}[yg^*(\mathbf{x})\le 1]|yg^*(\mathbf{x})|] + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{mixture}}[\mathbb{I}[yg^*(\mathbf{x})\le 1]]$$

$$\le \frac{3}{d^\tau} + \frac{4}{d^{0.9\tau-1}\sqrt{\tau\log d}}. \tag{B.307}$$

□

Theorem 3.3.4 (Mixtures of Gaussians: Main Result)**.** *Assume Assumption 3.3.1. For any $\epsilon,\delta\in(0,1)$, when Algorithm 1 uses hinge loss with*

$$m = \mathrm{poly}\left(\frac{1}{\delta},\frac{1}{\epsilon},d^{\Theta\left(\tau{\sigma_{B+}}^2/B_{\mu 1}^2\right)},r,\frac{1}{p_B}\right) \le e^d, \quad T = \mathrm{poly}\left(m\right), \quad n = \mathrm{poly}\left(m\right)$$

and proper hyper-parameters, then with probability at least $1-\delta$, there exists $t\in[T]$ such

that

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{\sqrt{2}r}{d^{0.4\tau-0.8}} + \epsilon.$$

Proof of Theorem 3.3.4. Let $\tilde{b} = C_b\sqrt{\tau d \log d}\sigma_{\mathbf{w}}\sigma_{B+}$, where C_b is a large enough universal constant.

By Lemma B.4.11, we have $g^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$, where $B_F = (B_{a1}, B_{a2}, B_b) = \left(\frac{1}{\sqrt{\tau\log d}\sigma_{B+}}, \frac{\sqrt{r}}{\sqrt{\tau\log d}\sigma_{B+}}, 2\sqrt{\tau}\right.$ $p = \Theta\left(\frac{B_{\mu 1}}{\sqrt{\tau\log d}\sigma_{B+}\cdot d^{\left(9C_b^2\tau{\sigma_{B+}}^2/(2B_{\mu 1}^2)\right)}}\right)$, $\gamma = \frac{1}{d^{0.9\tau-1.5}}$, $B_G = p_B B_{\mu 1}\sqrt{d} - O\left(\frac{\sigma_{B+}}{d^{0.9\tau}}\right)$ and $B_{x1} = (B_{\mu 2}+\sigma_{B+})\sqrt{d}$, $B_{x2} = (B_{\mu 2}+\sigma_{B+})^2 d$. We also have $\text{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} \leq \frac{3}{d^\tau} + \frac{4}{d^{0.9\tau-1}\sqrt{\tau\log d}}$.

Adjust $\sigma_{\mathbf{w}}$ such that $\tilde{b} = C_b\sqrt{\tau d\log d}\sigma_{\mathbf{w}}\sigma_{B+} = \Theta\left(\frac{B_G^{\frac{1}{4}}B_{a2}B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right)$. Injecting above parameters into Theorem 3.2.12, we have with probability at least $1-\delta$ over the initialization, with proper hyper-parameters, there exists $t \in [T]$ such that

$$\begin{aligned}
&\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] && \text{(B.308)}\\
\leq&\frac{3}{d^\tau} + \frac{4}{d^{0.9\tau-1}\sqrt{\tau\log d}} + \frac{\sqrt{2}rB_{\mu 2}}{d^{(0.9\tau-1.5)/2}\sqrt{\tau\log d}\sigma_{B+}} + O\left(\frac{rB_{a1}B_{x1}B_{x2}{}^{\frac{1}{4}}(\log n)^{\frac{1}{4}}}{\sqrt{B_G}n^{\frac{1}{4}}}\right) + \epsilon/2\\
\leq&\frac{\sqrt{2}r}{d^{0.4\tau-0.8}} + \epsilon. && \text{(B.309)}
\end{aligned}$$

□

B.4.3 Mixture of Gaussians - XOR

We consider a special Mixture of Gaussians distribution studied in [227]. Consider the same data distribution in Appendix B.4.2 and Definition B.4.6 with the following assumptions.

Assumption B.4.12 (Mixture of Gaussians in [227])**.** Assume four Gaussians cluster with XOR-like pattern, for any $\tau > 0$,

- $r = 4$ and $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$.
- $\mu_1 = -\mu_2$, $\mu_3 = -\mu_4$ and $\|\mu_1\|_2 = \|\mu_2\|_2 = \|\mu_3\|_2 = \|\mu_4\|_2 = \sqrt{d}$ and $\langle\mu_1, \mu_3\rangle = 0$.
- For all $j \in [4]$, $\Sigma_j = \sigma_B I_{d\times d}$ and $1 \leq \sigma_B \leq \sqrt{\frac{d}{\tau\log\log d}}$.

- $y_{(1)} = y_{(2)} = 1$ and $y_{(3)} = y_{(4)} = -1$.

We denote this data distribution as $\mathcal{D}_{mixture-xor}$ setting.

Mixture of Gaussians - XOR: Feature Learning

Lemma B.4.13 (Mixture of Gaussians in [227]: Gradient Feature Set)**.** *Let* $C_{Sure,1} = \frac{6}{5}$, $C_{Sure,2} = \frac{\sqrt{2}}{\sqrt{\tau \log\log d}}$, $\tilde{b} = \sqrt{\tau d \log\log d}\sigma_{\mathbf{w}}\sigma_B$ *and* d *is large enough. For* $\mathcal{D}_{mixture-xor}$ *setting, we have* $(D_j, +1) \in S_{p,\gamma,B_G}$ *for all* $j \in [4]$*, where*

$$p = \Theta\left(\frac{1}{\sqrt{\tau\log\log d}\sigma_B \cdot (\log d)^{\frac{18\tau\sigma_B^2}{25}}}\right), \quad \gamma = \frac{\sigma_B}{\sqrt{d}}, \tag{B.310}$$

$$B_G = \frac{\sqrt{d}}{4}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right). \tag{B.311}$$

Proof of Lemma B.4.13. For all $j \in [r]$, by Lemma B.4.16, for all $i \in S_{D_j,Sure}$,

$$1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right|}{\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2} \tag{B.312}$$

$$\leq 1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right|}{\sqrt{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right|^2 + \max_{D_j^\top D_j^\perp = 0, \|D_j^\perp\|_2 = 1}\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j^\perp\right\rangle\right|^2}} \tag{B.313}$$

$$\leq 1 - \frac{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right|}{\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right| + \max_{D_j^\top D_j^\perp = 0, \|D_j^\perp\|_2 = 1}\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j^\perp\right\rangle\right|} \tag{B.314}$$

$$\leq 1 - \frac{1}{1 + \frac{\sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right)}{\frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right)}} \tag{B.315}$$

$$\leq \frac{\sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right)}{\frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right)} \tag{B.316}$$

$$< \frac{\sigma_B}{\sqrt{d}} = \gamma. \tag{B.317}$$

Thus, we have $G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \in \mathcal{C}_{D_j,\gamma}$ and $\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j\right\rangle\right| \leq \|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \leq B_{x1}$,

$\frac{\mathbf{b}_i}{|\mathbf{b}_i|} = +1$. Thus, by Lemma B.4.14, we have

$$\Pr_{\mathbf{w},b}\left[G(\mathbf{w},b) \in \mathcal{C}_{D_j,\gamma} \text{ and } \|G(\mathbf{w},b)\|_2 \geq B_G \text{ and } \frac{b}{|b|} = +1\right] \tag{B.318}$$

$$\geq \Pr\left[i \in S_{D_j,Sure}\right] \tag{B.319}$$

$$\geq p. \tag{B.320}$$

Thus, $(D_j, +1) \in S_{p,\gamma,B_G}$. We finish the proof. □

Lemma B.4.14 (Mixture of Gaussians in [227]: Geometry at Initialization)**.** *Assume the same conditions as in Lemma B.4.13. Recall for all $i \in [m]$, $\mathbf{w}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$, over the random initialization, we have for all $i \in [m], j \in [4]$,*

$$\Pr\left[i \in S_{D_j,Sure}\right] \geq \Theta\left(\frac{1}{\sqrt{\tau \log\log d}\sigma_B \cdot (\log d)^{\frac{18\tau\sigma_B^2}{25}}}\right). \tag{B.321}$$

Proof of Lemma B.4.14. WLOG, let $j = 1$. By Assumption B.4.12, for the first condition in Definition B.4.6, we have,

$$\Pr\left[\left\langle \mathbf{w}_i^{(0)}, \mu_1 \right\rangle \geq C_{Sure,1}\mathbf{b}_i\right] \geq \Theta\left(\frac{1}{\sqrt{\tau \log\log d}\sigma_B \cdot (\log d)^{\frac{18\tau\sigma_B^2}{25}}}\right), \tag{B.322}$$

where the the last inequality follows Lemma B.5.6.

For the second condition in Definition B.4.6, by Lemma B.5.6, we have,

$$\Pr\left[\left|\left\langle \mathbf{w}_i^{(0)}, \mu_2 \right\rangle\right| \leq C_{Sure,2}\mathbf{b}_i\right] \geq 1 - \frac{1}{2\sqrt{\pi}}\frac{1}{\sigma_B \cdot e^{\sigma_B^2}}, \tag{B.323}$$

On the other hand, if X is a $\chi^2(k)$ random variable. Then we have

$$\Pr(X \geq k + 2\sqrt{kx} + 2x) \leq e^{-x}. \tag{B.324}$$

Therefore, we have

$$\Pr\left(\frac{1}{\sigma_{\mathbf{w}}^2}\left\|\mathbf{w}_i^{(0)}\right\|_2^2 \geq d+2\sqrt{\left(\frac{18\tau\sigma_B^2}{25}+2\right)d\log\log d}+2\left(\frac{18\tau\sigma_B^2}{25}+2\right)\log\log d\right) \tag{B.325}$$

$$\leq O\left(\frac{1}{(\log d)^2\cdot(\log d)^{\frac{18\tau\sigma_B^2}{25}}}\right). \tag{B.326}$$

Thus, by union bound, we have

$$\Pr\left[i\in S_{D_j,Sure}\right] \geq \Theta\left(\frac{1}{\sqrt{\tau\log\log d}\sigma_B\cdot(\log d)^{\frac{18\tau\sigma_B^2}{25}}}\right). \tag{B.327}$$

□

Lemma B.4.15 (Mixture of Gaussians in [227]: Activation Pattern)**.** *Assume the same conditions as in Lemma B.4.13, for all $j\in[4], i\in S_{D_j,Sure}$, we have*

(1) When $\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_B I_{d\times d})$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_B I_{d\times d})}\left[\left\langle\mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\geq 0\right]\geq 1-\frac{1}{(\log d)^{\frac{\tau}{50}}}. \tag{B.328}$$

(2) For all $j'\neq j, j'\in[4]$, when $\mathbf{x}\sim\mathcal{N}_{j'}(\mu_{j'},\sigma_B I_{d\times d})$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathcal{N}_{j'}(\mu_{j'},\sigma_B I_{d\times d})}\left[\left\langle\mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\geq 0\right]\leq O\left(\frac{1}{(\log d)^{\frac{\tau}{2}}}\right). \tag{B.329}$$

Proof of Lemma B.4.15. In the proof, we need $\tilde{b}=\sqrt{\tau d\log\log d}\sigma_{\mathbf{w}}\sigma_B$. For the first state-

ment, when $\mathbf{x} \sim \mathcal{N}_j(\mu_j, \sigma_B I_{d\times d})$, by $C_{Sure,1} \geq \frac{6}{5}$, we have

$$\Pr_{\mathbf{x}\sim\mathcal{N}_j(\mu_j,\sigma_B I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i \geq 0\right] \geq \Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle \geq (1-C_{Sure,1})\mathbf{b}_i\right] \tag{B.330}$$

$$\geq \Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle \geq -\frac{\mathbf{b}_i}{5}\right] \tag{B.331}$$

$$= 1 - \Pr_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I_{d\times d})}\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle \leq -\frac{\mathbf{b}_i}{5}\right] \tag{B.332}$$

$$\geq 1 - \exp\left(-\frac{{\mathbf{b}_i}^2}{50d\sigma_{\mathbf{w}}^2\sigma_B^2}\right) \tag{B.333}$$

$$\geq 1 - \frac{1}{(\log d)^{\frac{\tau}{50}}}, \tag{B.334}$$

where the third inequality follows the Chernoff bound and symmetricity of the Gaussian vector.

For the second statement, we prove similarly by $0 < C_{Sure,2} \leq \frac{\sqrt{2}}{\sqrt{\tau\log\log d}}$. □

Then, Lemma B.4.16 gives gradients of neurons in $S_{D_j,Sure}$. It shows that these gradients are highly aligned with D_j.

Lemma B.4.16 (Mixture of Gaussians in [227]: Feature Emergence)**.** *Assume the same conditions as in Lemma B.4.13, for all $j \in [4]$, $i \in S_{D_j,Sure}$, we have*

$$\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], y_{(j)}D_j\right\rangle \tag{B.335}$$

$$\geq \frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right). \tag{B.336}$$

For any unit vector $D_j^{\perp}$ which is orthogonal with D_j, we have

$$\left|\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], D_j^{\perp}\right\rangle\right| \leq \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right). \tag{B.337}$$

Proof of Lemma B.4.16. For all $j \in [4]$, $i \in S_{D_j,Sure}$, we have

$$\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right] \tag{B.338}$$

$$= \sum_{l\in[4]} \frac{1}{4}\mathbb{E}_{\mathbf{x}\sim\mathcal{N}_l(\mathbf{x})}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right] \tag{B.339}$$

$$= \sum_{l\in[4]} \frac{1}{4}y_{(l)}\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_l)\right]. \tag{B.340}$$

Thus, by Lemma B.5.7 and Lemma B.4.15,

$$\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\mathbf{x}\right], y_{(j)}D_j\right\rangle \tag{B.341}$$

$$= \frac{1}{4}\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_j)^\top D_j\right] \tag{B.342}$$

$$+ \sum_{l\in[4],l\neq j} \frac{1}{4}y_{(l)}y_{(j)}\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I_{d\times d})}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)(\mathbf{x}+\mu_l)^\top D_j\right] \tag{B.343}$$

$$\geq \frac{1}{4}\mu_j^\top D_j\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sum_{l\in[4],l\neq j}\frac{1}{4}|\mu_l^\top D_j| O\left(\frac{1}{d^{\frac{\tau}{2}}}\right) \tag{B.344}$$

$$- \frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.345}$$

$$- \sum_{l\in[4],l\neq j}\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.346}$$

$$\geq \frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I)}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)-1\right)\mathbf{x}^\top D_j\right]\right|$$

$$- \sum_{l\in[4],l\neq j}\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.347}$$

$$= \frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I)}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_j\right\rangle - \mathbf{b}_i\right)\right)\mathbf{x}^\top D_j\right]\right|$$

$$- \sum_{l\in[4],l\neq j}\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}+\mu_l\right\rangle - \mathbf{b}_i\right)\mathbf{x}^\top D_j\right]\right| \tag{B.348}$$

$$\geq \frac{1}{4}\sqrt{d}\left(1 - O\left(\frac{1}{(\log d)^{\frac{\tau}{50}}}\right)\right) - \sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right). \tag{B.349}$$

For any unit vector $D_j^{\perp}$ which is orthogonal with D_j, similarly, we have

$$\left|\left\langle \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\mathbf{x}\right],D_j^{\perp}\right\rangle\right| \tag{B.350}$$

$$\leq\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_j\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| \tag{B.351}$$

$$+\sum_{l\in[4],l\neq j}\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_l\right\rangle-\mathbf{b}_i\right)(\mathbf{x}+\mu_l)^\top D_j^{\perp}\right]\right| \tag{B.352}$$

$$\leq\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_B I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_j\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| \tag{B.353}$$

$$+\sum_{l\in[4],l\neq j}\frac{1}{4}\left|\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma_l I)}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}+\mu_l\right\rangle-\mathbf{b}_i\right)\mathbf{x}^\top D_j^{\perp}\right]\right| \tag{B.354}$$

$$\leq\sigma_B O\left(\frac{1}{(\log d)^{0.018\tau}}\right). \tag{B.355}$$

□

Mixture of Gaussians - XOR: Final Guarantee

Lemma B.4.17 (Mixture of Gaussians in [227]: Existence of Good Networks)**.** *Assume the same conditions as in Lemma B.4.13 and let* $\tau=1$ *and when* $0<\tilde{\tau}\leq O\left(\frac{d}{\sigma_B^2\log d}\right)$*. Define*

$$g^*(\mathbf{x})=\sum_{j=1}^{4}\frac{y_{(j)}}{\sqrt{\tilde{\tau}\log d}\sigma_B}\left[\sigma\left(\langle D_j,\mathbf{x}\rangle-2\sqrt{\tilde{\tau}\log d}\sigma_B\right)\right]. \tag{B.356}$$

For $\mathcal{D}_{mixture-xor}$ *setting, we have* $g^*\in\mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$*, where* $B_F=(B_{a1},B_{a2},B_b)=\left(\frac{1}{\sqrt{\tilde{\tau}\log d}\sigma_B},\frac{2}{\sqrt{\tilde{\tau}\log d}\sigma_B},2\sqrt{\tilde{\tau}\log d}\sigma_B\right)$*,* $p=\Omega\left(\frac{1}{\sigma_B\cdot(\log d)^{\sigma_B^2}}\right)$*,* $\gamma=\frac{\sigma_B}{\sqrt{d}}$*,* $r=4$*,* $B_G=\frac{1}{5}\sqrt{d}$ *and* $B_{x1}=(1+\sigma_B)\sqrt{d}, B_{x2}=(1+\sigma_B)^2 d$*. We also have* $\mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}\leq\frac{3}{d^{\tilde{\tau}}}+\frac{4}{d^{0.9\tilde{\tau}-1}\sqrt{\tilde{\tau}\log d}}$.

Proof of Lemma B.4.17. We finish the proof by following the proof of Lemma B.4.11 □

Theorem B.4.18 (Mixture of Gaussians in [227]: Main Result)**.** *For* $\mathcal{D}_{mixture-xor}$ *setting with Assumption B.4.12, when* d *is large enough, for any* $\delta\in(0,1)$ *and for any* $\epsilon\in(0,1)$

when

$$m =\Omega\left(\sigma_B(\log d)^{\sigma_B^2}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^2+\frac{1+\sigma_B}{\epsilon^4}\right)+\frac{1}{\sqrt{\delta}}\right)\leq e^d, \tag{B.357}$$

$$T =\text{poly}(\sigma_B, 1/\epsilon, 1/\delta, \log d), \tag{B.358}$$

$$n =\tilde{\Omega}\left(\frac{m^3(1+\sigma_B^2)}{\epsilon^2\max\left\{\sigma_B\cdot(\log d)^{\sigma_B^2}, 1\right\}}+\sigma_B\cdot(\log d)^{\sigma_B^2}+\frac{Tm}{\delta}\right), \tag{B.359}$$

trained by Algorithm 1 with hinge loss, with probability at least $1-\delta$ over the initialization and training samples, with proper hyper-parameters, there exists $t\in[T]$ such that

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x}))\neq y]\leq O\left(\left(1+\sigma_B^{\frac{3}{2}}\right)\left(\frac{1}{d^{\frac{1}{4}}}+\frac{(\log n)^{\frac{1}{4}}}{n^{\frac{1}{4}}}\right)\right)+\epsilon. \tag{B.360}$$

Proof of Theorem B.4.18. Let $\tilde{b}=\sqrt{d\log\log d}\sigma_{\mathbf{w}}\sigma_B$. By Lemma B.4.17, let $\tau=1$ and when $\tilde{\tau}=O\left(\frac{d}{\sigma_B^2\log d}\right)$, we have $g^*\in\mathcal{F}_{d,r,B_F,S_{p,\gamma,B_G}}$, where $B_F=(B_{a1},B_{a2},B_b)=\left(\frac{1}{\sqrt{\tilde{\tau}\log d}\sigma_B},\frac{2}{\sqrt{\tilde{\tau}\log d}\sigma_B},2\sqrt{\tilde{\tau}\log d}\sigma_B\right)$, $p=\Omega\left(\frac{1}{\sigma_B\cdot(\log d)^{\sigma_B^2}}\right)$, $\gamma=\frac{\sigma_B}{\sqrt{d}}$, $r=4$, $B_G=\frac{1}{5}\sqrt{d}$ and $B_{x1}=(1+\sigma_B)\sqrt{d}, B_{x2}=(1+\sigma_B)^2d$. We also have $\text{OPT}_{d,r,B_F,S_{p,\gamma,B_G}}\leq\frac{3}{d^{\tilde{\tau}}}+\frac{4}{d^{0.9\tilde{\tau}-1}\sqrt{\tilde{\tau}\log d}}$.

Adjust $\sigma_{\mathbf{w}}$ such that $\tilde{b}=\sqrt{d\log\log d}\sigma_{\mathbf{w}}\sigma_B=\Theta\left(\frac{B_G^{\frac{1}{4}}B_{a2}B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right)$. Injecting above parameters into Theorem 3.2.12, we have with probability at least $1-\delta$ over the initialization, with proper hyper-parameters, there exists $t\in[T]$ such that

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x}))\neq y]\leq O\left(\left(1+\sigma_B^{\frac{3}{2}}\right)\left(\frac{1}{d^{\frac{1}{4}}}+\frac{(\log n)^{\frac{1}{4}}}{n^{\frac{1}{4}}}\right)\right)+\epsilon. \tag{B.361}$$

□

B.4.4 Parity Functions

We recap the problem setup in Section 3.3.2 for readers' convenience.

Problem Setup

Data Distributions. Suppose $\mathbf{M} \in \mathbb{R}^{d \times D}$ is an unknown dictionary with D columns that can be regarded as patterns. For simplicity, assume $d = D$ and $\mathbf{M}$ is orthonormal. Let $\phi \in \mathbb{R}^d$ be a hidden representation vector. Let $\mathbf{A} \subseteq [D]$ be a subset of size rk corresponding to the class relevant patterns and r is an odd number. Then the input is generated by $\mathbf{M}\phi$, and some function on $\phi_{\mathbf{A}}$ generates the label. WLOG, let $\mathbf{A} = \{1, \ldots, rk\}$, $\mathbf{A}^\perp = \{rk+1, \ldots, d\}$. Also, we split $\mathbf{A}$ such that for all $j \in [r]$, $\mathbf{A}_j = \{(j-1)k+1, \ldots, jk\}$. Then the input $\mathbf{x}$ and the class label y are given by:

$$\mathbf{x} = \mathbf{M}\phi, \; y = g^*(\phi_{\mathbf{A}}) = \text{sign}\left(\sum_{j=1}^{r} \text{XOR}(\phi_{\mathbf{A}_j})\right), \tag{B.362}$$

where g^* is the ground-truth labeling function mapping from $\mathbb{R}^{rk}$ to $\mathcal{Y} = \{\pm 1\}$, $\phi_{\mathbf{A}}$ is the sub-vector of ϕ with indices in $\mathbf{A}$, and $\text{XOR}(\phi_{\mathbf{A}_j}) = \prod_{l \in \mathbf{A}_j} \phi_l$ is the parity function.

We still need to specify the distribution of ϕ, which determines the structure of the input distribution:

$$\mathbf{X} := (1 - 2rp_A)\mathbf{X}_U + \sum_{j \in [r]} p_A(\mathbf{X}_{j,+} + \mathbf{X}_{j,-}). \tag{B.363}$$

For all corresponding $\phi_{\mathbf{A}^\perp}$ in $\mathbf{X}$, we have $\forall l \in \mathbf{A}^\perp$, independently:

$$\phi_l = \begin{cases} +1, & \text{w.p. } p_o \\ -1, & \text{w.p. } p_o \\ 0, & \text{w.p. } 1 - 2p_o \end{cases}$$

where p_o controls the signal noise ratio: if p_o is large, then there are many nonzero entries in $\mathbf{A}^\perp$ which are noise interfering with the learning of the ground-truth labeling function on $\mathbf{A}$.

For corresponding $\phi_{\mathbf{A}}$, any $j \in [r]$, we have

- In $\mathbf{X}_{j,+}$, $\phi_{\mathbf{A}_j} = [+1, +1, \ldots, +1]^\top$ and $\phi_{\mathbf{A}\setminus\mathbf{A}_j}$ only have zero elements.
- In $\mathbf{X}_{j,-}$, $\phi_{\mathbf{A}_j} = [-1, -1, \ldots, -1]^\top$ and $\phi_{\mathbf{A}\setminus\mathbf{A}_j}$ only have zero elements.
- In $\mathbf{X}_U$, we have $\phi_{\mathbf{A}}$ draw from $\{+1, -1\}^{rk}$ uniformly.

We call this data distribution $\mathcal{D}_{parity}$.

Assumption B.4.19 (Parity Functions. Recap of Assumption 3.3.5)**.** Let $8 \leq \tau \leq d$ be a parameter that will control our final error guarantee. Assume k is an odd number and:

$$k \geq \Omega(\tau \log d), \quad d \geq rk + \Omega(\tau r \log d), \quad p_o = O\left(\frac{rk}{d - rk}\right), \quad p_A \geq \frac{1}{d}. \tag{B.364}$$

Remark B.4.20. The assumptions require k, d, and p_A to be sufficiently large so as to provide enough large signals for learning. When $p_o = \Theta(\frac{rk}{d-rk})$ means that the signal noise ratio is constant: the expected norm of $\phi_{\mathbf{A}}$ and that of $\phi_{\mathbf{A}^\perp}$ are comparable.

To apply our framework, again we only need to compute the parameters in the Gradient Feature set and the corresponding optimal approximation loss. To this end, we first define the gradient features: For all $j \in [r]$, let

$$D_j = \frac{\sum_{l \in \mathbf{A}_j} \mathbf{M}_l}{\| \sum_{l \in \mathbf{A}_j} \mathbf{M}_l \|_2}. \tag{B.365}$$

Remark B.4.21. Our data distribution is symmetric, which means for any $\phi \in \mathbb{R}^d$:

- $-y = g^*(-\phi_{\mathbf{A}})$ and $-x = \mathbf{M}(-\phi)$,
- $\mathbb{P}(\phi) = \mathbb{P}(-\phi)$,
- $\mathbb{E}_{(\mathbf{x},y)}[y\mathbf{x}] = \mathbf{0}$.

Below, we define a sufficient condition that randomly initialized weights will fall in nice gradients set after the first gradient step update.

Definition B.4.22 (Parity Functions: Subset of Nice Gradients Set)**.** Recall $\mathbf{w}_i^{(0)}$ is the weight for the i-th neuron at initialization. For all $j \in [r]$, let $S_{D_j,Sure} \subseteq [m]$ be those neurons that satisfy

- $\left\langle \mathbf{w}_i^{(0)}, D_j \right\rangle \geq \frac{C_{Sure,1}}{\sqrt{k}} \mathbf{b}_i$,

- $\left|\left\langle \mathbf{w}_i^{(0)}, D_{j'} \right\rangle\right| \leq \frac{C_{Sure,2}}{\sqrt{k}} \mathbf{b}_i$, for all $j' \neq j, j' \in [r]$,
- $\left\|P_{\mathbf{A}} \mathbf{w}_i^{(0)}\right\|_2 \leq \Theta(\sqrt{rk}\sigma_{\mathbf{w}})$,
- $\left\|P_{\mathbf{A}^\perp} \mathbf{w}_i^{(0)}\right\|_2 \leq \Theta(\sqrt{d-rk}\sigma_{\mathbf{w}})$,

where $P_{\mathbf{A}}, P_{\mathbf{A}^\perp}$ are the projection operator on the space $\mathbf{M}_{\mathbf{A}}$ and $\mathbf{M}_{\mathbf{A}^\perp}$.

Parity Functions: Feature Learning

We show the important Lemma B.4.23 first and defer other Lemmas after it.

Lemma B.4.23 (Parity Functions: Gradient Feature Set. Part statement of Lemma 3.3.7)**.** *Let $C_{Sure,1} = \frac{3}{2}$, $C_{Sure,2} = \frac{1}{2}$, $\tilde{b} = C_b\sqrt{\tau r k \log d}\sigma_{\mathbf{w}}$, where C_b is a large enough universal constant. For $\mathcal{D}_{parity}$ setting, we have $(D_j, +1), (D_j, -1) \in S_{p,\gamma,B_G}$ for all $j \in [r]$, where*

$$p = \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(9C_b^2\tau r/8\right)}}\right), \quad \gamma = \frac{1}{d^{\tau-2}}, \quad B_G = \sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right). \tag{B.366}$$

Proof of Lemma B.4.23. Note that for all $l \in [d]$, we have $\mathbf{M}_l^\top \mathbf{x} = \phi_l$. For all $j \in [r]$, by Lemma B.4.26, for all $i \in S_{D_j,Sure}$, when $\gamma = \frac{1}{d^{\tau-2}}$,

$$\begin{aligned}
&\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), D_j \right\rangle\right| - (1-\gamma)\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 && \text{(B.367)}\\
=&\left|\left\langle G(\mathbf{w}_i^{(0)}, \mathbf{b}_i), \frac{\sum_{l\in\mathbf{A}_j}\mathbf{M}_l}{\sqrt{k}} \right\rangle\right| - (1-\gamma)\|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 && \text{(B.368)}\\
\geq&\sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right) - \left(1-\frac{1}{d^{\tau-2}}\right)\sqrt{kp_A^2 + \sum_{l\in[d]} O\left(\frac{1}{d^\tau}\right)^2} && \text{(B.369)}\\
\geq&\sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right) - \left(1-\frac{1}{d^{\tau-2}}\right)\left(\sqrt{k}p_A + O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right)\right) && \text{(B.370)}\\
\geq&\frac{\sqrt{k}p_A}{d^{\tau-2}} - O\left(\frac{\sqrt{k}}{d^\tau}\right) - O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right) && \text{(B.371)}\\
>&0. && \text{(B.372)}
\end{aligned}$$

Thus, we have $G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \in \mathcal{C}_{D_j,\gamma}$ and $\sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right) \leq \|G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\|_2 \leq \sqrt{k}p_A +$

$O\left(\frac{1}{d^{\tau-\frac{1}{2}}}\right)$, $\frac{\mathbf{b}_i}{|\mathbf{b}_i|} = +1$. Thus, by Lemma B.4.24, we have

$$\Pr_{\mathbf{w},b}\left[G(\mathbf{w},b) \in \mathcal{C}_{D_j,\gamma} \text{ and } \|G(\mathbf{w},b)\|_2 \geq B_G \text{ and } \frac{b}{|b|} = +1\right] \tag{B.373}$$

$$\geq \Pr\left[i \in S_{D_j,Sure}\right] \tag{B.374}$$

$$\geq p. \tag{B.375}$$

Thus, $(D_j, +1) \in S_{p,\gamma,B_G}$. Since $\mathbb{E}_{(\mathbf{x},y)}[y\mathbf{x}] = \mathbf{0}$, by Lemma B.5.2 and considering $i \in [2m] \setminus [m]$, we have $(D_j, -1) \in S_{p,\gamma,B_G}$. We finish the proof. □

Below are Lemmas used in the proof of Lemma B.4.23. In Lemma B.4.24, we calculate p used in S_{p,γ,B_G}.

Lemma B.4.24 (Parity Functions: Geometry at Initialization. Lemma B.2 in [10])**.** *Assume the same conditions as in Lemma B.4.23, recall for all $i \in [m]$, $\mathbf{w}_i^{(0)} \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2 I_{d\times d})$, over the random initialization, we have for all $i \in [m], j \in [r]$,*

$$\Pr\left[i \in S_{D_j,Sure}\right] \geq \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(9C_b^2 \tau r/8\right)}}\right). \tag{B.376}$$

Proof of Lemma B.4.24. For every $i \in [m]$, $j, j' \in [r]$, $j \neq j'$, by Lemma B.5.6,

$$p_1 = \Pr\left[\left\langle \mathbf{w}_i^{(0)}, D_j \right\rangle \geq \frac{C_{Sure,1}}{\sqrt{k}}\mathbf{b}_i\right] = \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(9C_b^2 \tau r/8\right)}}\right) \tag{B.377}$$

$$p_2 = \Pr\left[\left|\left\langle \mathbf{w}_i^{(0)}, D_{j'} \right\rangle\right| \geq \frac{C_{Sure,2}}{\sqrt{k}}\mathbf{b}_i\right] = \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(C_b^2 \tau r/8\right)}}\right). \tag{B.378}$$

On the other hand, if X is a $\chi^2(k)$ random variable, by Lemma B.5.5, we have

$$\Pr(X \geq k + 2\sqrt{kx} + 2x) \leq e^{-x}. \tag{B.379}$$

Therefore, by assumption $rk \geq \Omega(\tau r \log d), d - rk \geq \Omega(\tau r \log d)$, we have

$$\Pr\left(\frac{1}{\sigma_{\mathbf{w}}^2}\left\|P_A \mathbf{w}_i^{(0)}\right\|_2^2 \geq rk + 2\sqrt{\left(9C_b^2\tau r/8 + 2\right)rk\log d} + 2\left(9C_b^2\tau r/8 + 2\right)\log d\right) \tag{B.380}$$

$$\leq O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau r/8\right)}}\right), \tag{B.381}$$

$$\Pr\left(\frac{1}{\sigma_{\mathbf{w}}^2}\left\|P_A \mathbf{w}_i^{(0)}\right\|_2^2 \geq (d-rk) + 2\sqrt{\left(9C_b^2\tau r/8 + 2\right)(d-rk)\log d} + 2\left(9C_b^2\tau r/8 + 2\right)\log d\right)$$

$$\leq O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau r/8\right)}}\right). \tag{B.382}$$

Thus, by union bound, and $D_1, \ldots, D_r$ being orthogonal with each other, we have

$$\Pr\left[i \in S_{D_j,Sure}\right] \geq p_1(1-p_2)^{r-1} - O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau r/8\right)}}\right) \tag{B.383}$$

$$= \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(9C_b^2\tau r/8\right)}} \cdot \left(1 - \frac{r}{\sqrt{\tau r \log d} \cdot d^{\left(C_b^2\tau r/8\right)}}\right)\right) \tag{B.384}$$

$$- O\left(\frac{1}{d^2 \cdot d^{\left(9C_b^2\tau r/8\right)}}\right) \tag{B.385}$$

$$= \Theta\left(\frac{1}{\sqrt{\tau r \log d} \cdot d^{\left(9C_b^2\tau r/8\right)}}\right). \tag{B.386}$$

□

In Lemma B.4.25, we compute the activation pattern for the neurons in $S_{D_j,Sure}$.

Lemma B.4.25 (Parity Functions: Activation Pattern)**.** *Assume the same conditions as in Lemma B.4.23, for all $j \in [r], i \in S_{D_j,Sure}$, we have*

(1) When $\mathbf{x} \sim \mathbf{X}$, we have

$$\Pr_{\mathbf{x}\sim\mathbf{X}}\left[\left|\sum_{l\in A^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle\right| \geq t\right] \leq \exp\left(-\frac{t^2}{\Theta\left(rk\sigma_{\mathbf{w}}^2\right)}\right). \tag{B.387}$$

(2) When $\mathbf{x} \sim \mathbf{X}_U$, we have

$$\Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\left|\sum_{l\in A}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle\right| \geq t\right] \leq \exp\left(-\frac{t^2}{\Theta(rk\sigma_{\mathbf{w}}^2)}\right). \tag{B.388}$$

(3) When $\mathbf{x} \sim \mathbf{X}_U$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\sum_{l\in[d]}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle - \mathbf{b}_i \geq 0\right] \leq O\left(\frac{1}{d^\tau}\right). \tag{B.389}$$

(4) When $\mathbf{x} \sim \mathbf{X}_{j,+}$, the activation probability satisfies,

$$\Pr_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sum_{l\in[d]}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle - \mathbf{b}_i \geq 0\right] \geq 1 - O\left(\frac{1}{d^\tau}\right). \tag{B.390}$$

(5) For all $j' \neq j, j' \in [r]$, $s \in \{+,-\}$, when $\mathbf{x} \sim \mathbf{X}_{j',s}$, or $\mathbf{x} \sim \mathbf{X}_{j,-}$, the activation probability satisfies,

$$\Pr\left[\sum_{l\in[d]}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle - \mathbf{b}_i \geq 0\right] \leq O\left(\frac{1}{d^\tau}\right). \tag{B.391}$$

Proof of Lemma B.4.25. For the first statement, when $\mathbf{x} \sim \mathbf{X}$, note that $\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle \phi_l$ is a mean-zero sub-Gaussian random variable with sub-Gaussion norm $\Theta\left(\left|\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle\right| \sqrt{p_o}\right)$.

$$\Pr_{\mathbf{x}\sim\mathbf{X}}\left[\left|\sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle\right| \geq t\right] = \Pr_{\mathbf{x}\sim\mathbf{X}}\left[\left|\sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle \phi_l\right| \geq t\right] \tag{B.392}$$

$$\leq \exp\left(-\frac{t^2}{\sum_{l\in\mathbf{A}^\perp}\Theta\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle^2 p_o\right)}\right) \tag{B.393}$$

$$\leq \exp\left(-\frac{t^2}{\Theta\left((d-rk)\sigma_{\mathbf{w}}^2 p_o\right)}\right) \tag{B.394}$$

$$\leq \exp\left(-\frac{t^2}{\Theta(rk\sigma_{\mathbf{w}}^2)}\right), \tag{B.395}$$

where the inequality follows general Hoeffding's inequality.

For the second statement, when $\mathbf{x} \sim \mathbf{X}_U$, by Hoeffding's inequality,

$$\Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\left|\sum_{l\in\mathbf{A}}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle\right| \geq t\right] = \Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\left|\sum_{l\in\mathbf{A}}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle \phi_l\right| \geq t\right] \tag{B.396}$$

$$\leq 2\exp\left(-\frac{t^2}{2\sum_{l\in\mathbf{A}}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\right\rangle^2}\right) \tag{B.397}$$

$$\leq \exp\left(-\frac{t^2}{\Theta(rk\sigma_{\mathbf{w}}^2)}\right). \tag{B.398}$$

In the proof of the third to the last statement, we need $\tilde{b} = C_b\sqrt{\tau rk\log d}\sigma_{\mathbf{w}}$, where C_b is a large enough universal constant.

For the third statement, when $\mathbf{x} \sim \mathbf{X}_U$, by union bound and previous statements,

$$\Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\sum_{l\in[d]}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle - \mathbf{b}_i \geq 0\right] \tag{B.399}$$

$$\leq \Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\sum_{l\in\mathbf{A}}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle \geq \frac{\mathbf{b}_i}{2}\right] + \Pr_{\mathbf{x}\sim\mathbf{X}_U}\left[\sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle \geq \frac{\mathbf{b}_i}{2}\right] \tag{B.400}$$

$$\leq O\left(\frac{1}{d^\tau}\right). \tag{B.401}$$

For the forth statement, when $\mathbf{x} \sim \mathbf{X}_{j,+}$, by $C_{Sure,1} \geq \frac{3}{2}$ and previous statements,

$$\Pr_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sum_{l\in[d]}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle - \mathbf{b}_i \geq 0\right] \tag{B.402}$$

$$= \Pr_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sum_{l\in\mathbf{A}_j}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle + \sum_{l\in A\setminus\mathbf{A}_j}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle + \sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle \geq \mathbf{b}_i\right] \tag{B.403}$$

$$\geq \Pr_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle \geq (1 - C_{Sure,1})\mathbf{b}_i\right] \tag{B.404}$$

$$\geq \Pr_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sum_{l\in\mathbf{A}^\perp}\left\langle \mathbf{w}_i^{(0)}, \mathbf{M}_l\phi_l\right\rangle \geq -\frac{\mathbf{b}_i}{2}\right] \tag{B.405}$$

$$\geq 1 - O\left(\frac{1}{d^\tau}\right). \tag{B.406}$$

For the last statement, we prove similarly by $0 < C_{Sure,2} \leq \frac{1}{2}$. $\square$

Then, Lemma B.4.26 gives gradients of neurons in $S_{D_j,Sure}$. It shows that these gradients are highly aligned with D_j.

Lemma B.4.26 (Parity Functions: Feature Emergence)**.** *Assume the same conditions as in Lemma B.4.23, for all $j \in [r]$, $i \in S_{D_j,Sure}$, we have the following holds:*

(1) For all $l \in \mathbf{A}_j$, we have

$$p_A - O\left(\frac{1}{d^\tau}\right) \leq \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] \leq p_A + O\left(\frac{1}{d^\tau}\right). \tag{B.407}$$

(2) For all $l \in \mathbf{A}_{j'}$, any $j' \neq j, j' \in [r]$, we have

$$\left|\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right]\right| \leq O\left(\frac{1}{d^\tau}\right). \tag{B.408}$$

(3) For all $l \in \mathbf{A}^\perp$, we have

$$\left|\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right]\right| \leq O\left(\frac{1}{d^\tau}\right). \tag{B.409}$$

Proof of Lemma B.4.26. For all $l \in [d]$, we have

$$\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] \tag{B.410}$$

$$= p_A \sum_{l\in[r]}\left(\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{l,+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] - \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{l,-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right]\right) \tag{B.411}$$

$$+ (1 - 2rp_A)\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_U}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right]. \tag{B.412}$$

For the first statement, for all $l \in \mathbf{A}_j$, by Lemma B.4.25 (3) and (4), we have

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] && \text{(B.413)}\\
=&p_A\left(\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right] + \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right) && \text{(B.414)}\\
&+(1-2rp_A)\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_U}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] && \text{(B.415)}\\
\geq&p_A\left(1 - O\left(\frac{1}{d^\tau}\right)\right) - O\left(\frac{1}{d^\tau}\right) && \text{(B.416)}\\
\geq&p_A - O\left(\frac{1}{d^\tau}\right), && \text{(B.417)}
\end{aligned}$$

and we also have

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] && \text{(B.418)}\\
=&p_A\left(\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right] + \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right) && \text{(B.419)}\\
&+(1-2rp_A)\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_U}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right] && \text{(B.420)}\\
\leq&p_A + O\left(\frac{1}{d^\tau}\right). && \text{(B.421)}
\end{aligned}$$

Similarly, for the second statement, for all $l \in \mathbf{A}_{j'}$, any $j' \neq j, j' \in [r]$, by Lemma B.4.25 (3) and (5), we have

$$\begin{aligned}
&\left|\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\phi_l\right]\right| && \text{(B.422)}\\
\leq&\left|p_A\left(\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j',+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right] + \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j',-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)\right]\right)\right| + O\left(\frac{1}{d^\tau}\right)\\
\leq&O\left(\frac{1}{d^\tau}\right). && \text{(B.423)}
\end{aligned}$$

For the third statement, for all $l \in \mathbf{A}^{\perp}$, by Lemma B.4.25 (3), (4), (5), we have

$$\begin{aligned}
&\left|\mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\phi_l\right]\right| && \text{(B.424)}\\
\leq& p_A\sum_{l\in[r]}\left|\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{l,+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\phi_l\right]-\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{l,-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\phi_l\right]\right|+O\left(\frac{1}{d^{\tau}}\right)\\
\leq& p_A\left|\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\phi_l\right]-\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\phi_l\right]\right|+O\left(\frac{1}{d^{\tau}}\right)\\
\leq& p_A\left|\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\right)\phi_l\right]-\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\right)\phi_l\right]\right|\\
&+p_A\left|\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\phi_l\right]-\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\phi_l\right]\right|+O\left(\frac{1}{d^{\tau}}\right) && \text{(B.425)}\\
=& p_A\left|\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,+}}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\right)\phi_l\right]-\mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{j,-}}\left[\left(1-\sigma'\left(\left\langle \mathbf{w}_i^{(0)},\mathbf{x}\right\rangle-\mathbf{b}_i\right)\right)\phi_l\right]\right|\\
&+O\left(\frac{1}{d^{\tau}}\right) && \text{(B.426)}\\
\leq& O\left(\frac{1}{d^{\tau}}\right), && \text{(B.427)}
\end{aligned}$$

where the second inequality follows $2rp_A \leq 1$ and the third inequality follows the triangle inequality. □

Parity Functions: Final Guarantee

Lemma B.4.27 (Parity Functions: Existence of Good Networks. Part statement of Lemma 3.3.7)**.** *Assume the same conditions as in Lemma B.4.23. Define*

$$\begin{aligned}
g^*(\mathbf{x}) =& \sum_{j=1}^{r}\sum_{i=0}^{k}(-1)^{i+1}\sqrt{k} && \text{(B.428)}\\
&\cdot\left[\sigma\left(\langle D_j,\mathbf{x}\rangle-\frac{2i-k-1}{\sqrt{k}}\right)-2\sigma\left(\langle D_j,\mathbf{x}\rangle-\frac{2i-k}{\sqrt{k}}\right)+\sigma\left(\langle D_j,\mathbf{x}\rangle-\frac{2i-k+1}{\sqrt{k}}\right)\right].
\end{aligned}$$

For $\mathcal{D}_{parity}$ setting, we have $g^* \in \mathcal{F}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}}$, where $B_F = (B_{a1},B_{a2},B_b) = \left(2\sqrt{k},2\sqrt{rk(k+1)},\frac{k+1}{\sqrt{k}}\right)$, $p=\Theta\left(\frac{1}{\sqrt{\tau r\log d}\cdot d^{\left(9C_b^2\tau r/8\right)}}\right)$, $\gamma=\frac{1}{d^{\tau-2}}$, $B_G=\sqrt{k}p_A-O\left(\frac{\sqrt{k}}{d^{\tau}}\right)$ and $B_{x1}=\sqrt{d}, B_{x2}=d$. We also have $\mathrm{OPT}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}}=0$.

Proof of Lemma B.4.27. We can check $B_{x1}=\sqrt{d}, B_{x2}=d$ by direct calculation. By

Lemma B.4.23, we have $g^* \in \mathcal{F}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}}$. We note that

$$\sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k-1}{\sqrt{k}}\right) - 2\sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k}{\sqrt{k}}\right) + \sigma\left(\langle D_j, \mathbf{x}\rangle - \frac{2i-k+1}{\sqrt{k}}\right) \quad \text{(B.429)}$$

is a bump function for $\langle D_j, \mathbf{x}\rangle$ at $\frac{2i-k}{\sqrt{k}}$. We can check that $yg^*(\mathbf{x}) \geq 1$. Thus, we have

$$\begin{aligned}
\mathrm{OPT}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}} \leq & \mathcal{L}_{\mathcal{D}_{parity}}(g^*) && \text{(B.430)}\\
= & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{parity}} \mathcal{L}_{(\mathbf{x},y)}(g^*) && \text{(B.431)}\\
= & 0. && \text{(B.432)}
\end{aligned}$$

□

Theorem 3.3.8 (Parity Functions: Main Result)**.** *Assume Assumption 3.3.5. For any $\epsilon, \delta \in (0,1)$, when Algorithm 1 uses hinge loss with*

$$m = \mathrm{poly}\left(\frac{1}{\delta}, \frac{1}{\epsilon}, d^{\Theta(\tau r)}, k, \frac{1}{p_A}\right) \leq e^d, \quad T = \mathrm{poly}\left(m\right), \quad n = \mathrm{poly}\left(m\right)$$

and proper hyper-parameters, then with probability at least $1-\delta$, there exists $t \in [T]$ such that

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{3r\sqrt{k}}{d^{(\tau-3)/2}} + \epsilon.$$

Proof of Theorem 3.3.8. Let $\tilde{b} = C_b\sqrt{\tau r k \log d}\sigma_{\mathbf{w}}$, where C_b is a large enough universal constant. By Lemma B.4.27, we have $g^* \in \mathcal{F}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}}$, where $B_F = (B_{a1}, B_{a2}, B_b) = \left(2\sqrt{k}, 2\sqrt{rk(k+1)}, \frac{k+1}{\sqrt{k}}\right)$, $p = \Theta\left(\frac{1}{\sqrt{\tau r \log d}\cdot d^{\left(9C_b^2\tau r/8\right)}}\right)$, $\gamma = \frac{1}{d^{\tau-2}}$, $B_G = \sqrt{k}p_A - O\left(\frac{\sqrt{k}}{d^\tau}\right)$ and $B_{x1} = \sqrt{d}, B_{x2} = d$. We also have $\mathrm{OPT}_{d,3r(k+1),B_F,S_{p,\gamma,B_G}} = 0$.

Adjust $\sigma_{\mathbf{w}}$ such that $\tilde{b} = C_b\sqrt{\tau r k \log d}\sigma_{\mathbf{w}} = \Theta\left(\frac{B_G^{\frac{1}{4}} B_{a2} B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right)$. Injecting above parameters into Theorem 3.2.12, we have with probability at least $1-\delta$ over the initialization, with

proper hyper-parameters, there exists $t \in [T]$ such that

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{2\sqrt{2}r\sqrt{k}}{d^{(\tau-3)/2}} + O\left(\frac{rB_{a1}B_{x1}B_{x2}{}^{\frac{1}{4}}(\log n)^{\frac{1}{4}}}{\sqrt{B_G}n^{\frac{1}{4}}}\right) + \epsilon/2 \leq \frac{3r\sqrt{k}}{d^{(\tau-3)/2}} + \epsilon.$$

□

B.4.5 Uniform Parity Functions

We consider the sparse parity problem over the uniform data distribution studied in [26]. We use the properties of the problem to prove the key lemma (i.e., the existence of good networks) in our framework and then derive the final guarantee from our theorem of the simple setting (Theorem 3.2.4). We provide Theorem B.4.31 as (1) use it as a warm-up and (2) follow the original analysis in [26] to give a comparison. We will provide Theorem B.4.40 as an alternative version that trains both layers.

Consider the same data distribution in Appendix B.4.4 and Definition B.4.22 with the following assumptions.

Assumption B.4.28 (Uniform Parity Functions)**.** We follow the data distribution in Appendix B.4.4. Let $r = 1, p_A = 0, p_o = \frac{1}{2}$, $\mathbf{M} = I_{d\times d}$ and $d \geq 2k^2$, and k is an even number.

We denote this data distribution as $\mathcal{D}_{parity-uniform}$ setting.

To apply our framework, again we only need to compute the parameters in the Gradient Feature set and the corresponding optimal approximation loss. To this end, we first define the gradient features: let

$$D = \frac{\sum_{l\in\mathbf{A}} \mathbf{M}_l}{\|\sum_{l\in\mathbf{A}} \mathbf{M}_l\|_2}. \tag{B.433}$$

We follow the initialization and training dynamic in [26].

Initialization and Loss. We use hinge loss and we have unbiased initialization, for all $i \in [m]$,

$$\mathbf{a}_i^{(0)} \sim \text{Unif}(\{\pm 1\}), \mathbf{w}_i^{(0)} \sim \text{Unif}(\{\pm 1\}^d), \mathbf{b}_i = \text{Unif}(\{-1 + 1/k, \ldots, 1 - 1/k\}). \quad \text{(B.434)}$$

Training Process. We use the following one-step training algorithm for this specific data distribution.

Algorithm 6 Network Training via Gradient Descent [26]. Special case of Algorithm 4

Initialize $(\mathbf{a}^{(0)}, \mathbf{W}^{(0)}, \mathbf{b})$ as in Equation (3.8) and Equation (B.434); Sample $\mathcal{Z} \sim \mathcal{D}_{parity-uniform}^n$
$\mathbf{W}^{(1)} = \mathbf{W}^{(0)} - \eta^{(1)}(\nabla_{\mathbf{W}} \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{\Xi^{(0)}}) + \lambda^{(1)}\mathbf{W}^{(0)})$
$\mathbf{a}^{(1)} = \mathbf{a}^{(0)} - \eta^{(1)}(\nabla_{\mathbf{a}} \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{\Xi^{(0)}}) + \lambda^{(1)}\mathbf{a}^{(0)})$
for $t = 2$ **to** T **do**
  $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} - \eta^{(t)} \nabla_{\mathbf{a}} \widetilde{\mathcal{L}}_{\mathcal{Z}}(f_{\Xi^{(t-1)}})$
end for

Use the notation in Section 5.3 of [203], for every $S \in [n]$, s.t. $|S| = k$, we define

$$\xi_k := \widehat{\text{Maj}}(S) = (-1)^{\frac{k-1}{2}} \frac{\binom{\frac{d-1}{2}}{\frac{d-1}{2}}}{\binom{d-1}{k-1}} \cdot 2^{-(d-1)} \binom{d-1}{\frac{d-1}{2}}. \quad \text{(B.435)}$$

Lemma B.4.29 (Uniform Parity Functions: Existence of Good Networks. Rephrase of Lemma 5 in [26])**.** *For every $\epsilon, \delta \in (0, 1/2)$, denoting $\tau = \frac{|\xi_{k-1}|}{16k\sqrt{2d\log(32k^3 d/\epsilon)}}$, let $\eta^{(1)} = \frac{1}{k|\xi_{k-1}|}$, $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$, $m \geq k \cdot 2^k \log(k/\delta)$, $n \geq \frac{2}{\tau^2}\log(4dm/\delta)$ and $d \geq \Omega\left(k^4 \log(kd/\epsilon)\right)$, w.p. at least $1 - 2\delta$ over the initialization and the training samples, there exists $\tilde{\mathbf{a}} \in \mathbb{R}^m$ with $\|\tilde{\mathbf{a}}\|_\infty \leq 8k$ and $\|\tilde{\mathbf{a}}\|_2 \leq 8k\sqrt{k}$ such that $f_{(\tilde{\mathbf{a}}, \mathbf{W}^{(1)}, \mathbf{b})}$ satisfies*

$$\mathcal{L}_{\mathcal{D}_{parity-uniform}}\left(f_{(\tilde{\mathbf{a}}, \mathbf{W}^{(1)}, \mathbf{b})}\right) \leq \epsilon. \quad \text{(B.436)}$$

Additionally, it holds that $\|\sigma({\mathbf{W}^{(1)}}^\top \mathbf{x} - \mathbf{b})\|_\infty \leq d + 1$.

Remark B.4.30. In [26], they update the bias term in the first gradient step. However, if we check the proof carefully, we can see that the fixed bias still goes through all their analysis.

Uniform Parity Functions: Final Guarantee

Considering training by Algorithm 6, we have the following results.

Theorem B.4.31 (Uniform Parity Functions: Main Result)**.** *Fix* $\epsilon \in (0, 1/2)$ *and let* $m \geq \Omega\left(k \cdot 2^k \log(k/\epsilon)\right)$, $n \geq \Omega\left(k^{7/6} d \binom{d}{k-1} \log(kd/\epsilon) \log(dm/\epsilon) + \frac{k^3 m d^2}{\epsilon^2}\right)$, $d \geq \Omega\left(k^4 \log(kd/\epsilon)\right)$. *Let* $\eta^{(1)} = \frac{1}{k|\xi_{k-1}|}$, $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$, *and* $\eta = \eta^{(t)} = \Theta\left(\frac{1}{d^2 m}\right)$, *for all* $t \in \{2, 3, \ldots, T\}$. *If* $T \geq \Omega\left(\frac{k^3 m d^2}{\epsilon}\right)$, *then training by Algorithm 6 with hinge loss, w.h.p. over the initialization and the training samples, there exists* $t \in [T]$ *such that*

$$\Pr[\text{sign}(f_{\Xi^{(t)}})(\mathbf{x}) \neq y] \leq \mathcal{L}_{\mathcal{D}_{parity-uniform}} f_{(\mathbf{a}^{(t)}, \mathbf{W}^{(1)}, \mathbf{b})} \leq \epsilon. \tag{B.437}$$

Proof of Theorem B.4.31. By Lemma B.4.29, w.h.p., we have for properly chosen hyper-parameters,

$$\text{OPT}_{\mathbf{W}^{(1)}, \mathbf{b}, B_{a2}} \leq \mathcal{L}_{\mathcal{D}_{parity-uniform}} \left(f_{(\tilde{\mathbf{a}}, \mathbf{W}^{(1)}, \mathbf{b})}\right) \leq \frac{\epsilon}{3}. \tag{B.438}$$

We compute the L-smooth constant of $\widetilde{\mathcal{L}}_{\mathcal{Z}} \left(f_{(\mathbf{a}, \mathbf{W}^{(1)}, \mathbf{b})}\right)$ to $\mathbf{a}$.

$$\begin{aligned}
&\left\|\nabla_{\mathbf{a}} \widetilde{\mathcal{L}}_{\mathcal{Z}} \left(f_{(\mathbf{a}_1, \mathbf{W}^{(1)}, \mathbf{b})}\right) - \nabla_{\mathbf{a}} \widetilde{\mathcal{L}}_{\mathcal{Z}} \left(f_{(\mathbf{a}_2, \mathbf{W}^{(1)}, \mathbf{b})}\right)\right\|_2 && \text{(B.439)}\\
=&\left\|\frac{1}{n} \sum_{\mathbf{x} \in \mathcal{Z}} \left[\left(\ell'\left(y f_{(\mathbf{a}_1, \mathbf{W}^{(1)}, \mathbf{b})}(\mathbf{x})\right) - \ell'\left(y f_{(\mathbf{a}_2, \mathbf{W}^{(1)}, \mathbf{b})}(\mathbf{x})\right)\right) \sigma(\mathbf{W}^{(1)\top} \mathbf{x} - \mathbf{b})\right]\right\|_2 && \text{(B.440)}\\
\leq&\left\|\frac{1}{n} \sum_{\mathbf{x} \in \mathcal{Z}} \left[\left|f_{(\mathbf{a}_1, \mathbf{W}^{(1)}, \mathbf{b})}(\mathbf{x}) - f_{(\mathbf{a}_2, \mathbf{W}^{(1)}, \mathbf{b})}(\mathbf{x})\right| \sigma(\mathbf{W}^{(1)\top} \mathbf{x} - \mathbf{b})\right]\right\|_2 && \text{(B.441)}\\
\leq&\frac{1}{n} \sum_{\mathbf{x} \in \mathcal{Z}} \left[\|\mathbf{a}_1 - \mathbf{a}_2\|_2 \left\|\sigma(\mathbf{W}^{(1)\top} \mathbf{x} - \mathbf{b})\right\|_2^2\right]. && \text{(B.442)}
\end{aligned}$$

By the Lemma B.4.29, we have $\|\sigma({\mathbf{W}^{(1)}}^\top \mathbf{x} - \mathbf{b})\|_\infty \leq d + 1$. Thus, we have,

$$\begin{aligned}
L &= O\left(\frac{1}{n} \sum_{\mathbf{x} \in \mathcal{Z}} \left\|\sigma(\mathbf{W}^{(1)\top} \mathbf{x} - \mathbf{b})\right\|_2^2\right) && \text{(B.443)}\\
&\leq O(d^2 m). && \text{(B.444)}
\end{aligned}$$

This means that we can let $\eta = \Theta\left(\frac{1}{d^2 m}\right)$ and we will get our convergence result. Note that we have $\mathbf{a}^{(1)} = \mathbf{0}$ and $\|\tilde{\mathbf{a}}\|_2 = O\left(k\sqrt{k}\right)$. So, if we choose $T \geq \Omega\left(\frac{k^3}{\epsilon\eta}\right)$, there exists $t \in [T]$ such that $\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}\right) - \widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}\right) \leq O\left(\frac{L\|\mathbf{a}^{(1)}-\tilde{\mathbf{a}}\|_2^2}{T}\right) \leq \epsilon/3$.

We also have $\sqrt{\frac{\|\tilde{\mathbf{a}}\|_2^2(\|\mathbf{W}^{(1)}\|_F^2 B_x^2 + \|\mathbf{b}\|_2^2)}{n}} \leq \frac{\epsilon}{3}$. Then our theorem gets proved by Theorem 3.2.4. □

B.4.6 Uniform Parity Functions: Alternative Analysis

It is also possible to unify [26] into our general Gradient Feature Learning Framework by mildly modifying the framework in Theorem 3.2.12. In order to do that, we first need to use a different metric in the definition of gradient features.

Modified General Feature Learning Framework for Uniform Parity Functions

Definition B.4.32 (Gradient Feature with Infinity Norm)**.** For a unit vector $D \in \mathbb{R}^d$ with $\|D\|_2 = 1$, and a $\gamma_\infty \in (0,1)$, a direction neighborhood (cone) $\mathcal{C}^\infty_{D,\gamma_\infty}$ is defined as: $\mathcal{C}^\infty_{D,\gamma_\infty} := \left\{\mathbf{w} \;\middle|\; \left\|\frac{\mathbf{w}}{\|\mathbf{w}\|} - D\right\|_\infty < \gamma_\infty\right\}$. Let $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ be random variables drawn from some distribution $\mathcal{W}, \mathcal{B}$. A Gradient Feature set with parameters $p, \gamma_\infty, B_G, B_{G1}$ is defined as:

$$S^\infty_{p,\gamma_\infty,B_G,B_{G1}}(\mathcal{W},\mathcal{B}) := \left\{(D,s) \;\middle|\; \Pr_{\mathbf{w},b}\left[G(\mathbf{w},b) \in \mathcal{C}^\infty_{D,\gamma_\infty}\ ,\ B_{G1} \geq \|G(\mathbf{w},b)\|_2 \geq B_G\ ,\ s = \frac{b}{|b|}\right] \geq p\right\}.$$

When clear from context, write it as $S^\infty_{p,\gamma_\infty,B_G,B_{G1}}$.

Definition B.4.33 (Optimal Approximation via Gradient Features with Infinity Norm)**.** The Optimal Approximation network and loss using gradient feature induced networks $\mathcal{F}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}}$ are defined as:

$$g^* := \operatorname*{arg\,min}_{g \in \mathcal{F}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}}} \mathcal{L}_{\mathcal{D}}(f), \tag{B.445}$$

$$\mathrm{OPT}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}} := \min_{g \in \mathcal{F}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}}} \mathcal{L}_{\mathcal{D}}(f). \tag{B.446}$$

We consider the data distribution in Appendix B.4.4 with Assumption B.4.28, i.e., $\mathcal{D}_{parity-uniform}$ in Appendix B.4.5. Note that with this dataset, we have $\|\mathbf{x}\|_\infty \le B_{x\infty} = 1$. We use the following unbiased initialization:

$$\begin{aligned}
&\text{for } i \in \{1,\dots,m\}: \quad \mathbf{a}_i^{(0)} \sim \mathcal{N}(0,\sigma_a^2), \mathbf{w}_i^{(0)} \sim \{\pm 1\}^d, \mathbf{b}_i = \tilde{b} \le 1,\\
&\text{for } i \in \{m+1,\dots,2m\}: \quad \mathbf{a}_i^{(0)} = -\mathbf{a}_{i-m}^{(0)}, \mathbf{w}_i^{(0)} = -\mathbf{w}_{i-m}^{(0)}, \mathbf{b}_i = -\mathbf{b}_{i-m},\\
&\text{for } i \in \{2m+1,\dots,4m\}: \quad \mathbf{a}_i^{(0)} = -\mathbf{a}_{i-2m}^{(0)}, \mathbf{w}_i^{(0)} = \mathbf{w}_{i-2m}^{(0)}, \mathbf{b}_i = \mathbf{b}_{i-2m}
\end{aligned} \tag{B.447}$$

Let ∇_i denote the gradient of the i-th neuron $\nabla_{\mathbf{w}_i}\mathcal{L}_{\mathcal{D}}(f_{\Xi^{(0)}})$. Denote the subset of neurons with nice gradients approximating feature (D,s) as:

$$G_{(D,s),Nice}^{\infty} := \left\{ i \in [2m] : s = \frac{\mathbf{b}_i}{|\mathbf{b}_i|}, \left\| \frac{\nabla_i}{\|\nabla_i\|} - D \right\|_\infty \le \gamma_\infty, \left|\mathbf{a}_i^{(0)}\right| B_{G1} \ge \|\nabla_i\|_2 \ge \left|\mathbf{a}_i^{(0)}\right| B_G \right\}.$$

Lemma B.4.34 (Existence of Good Networks. Modified Version of Lemma 3.2.14 Under Uniform Parity Setting)**.** *Let $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$. For any $B_\epsilon \in (0, B_b)$, let $\sigma_a = \Theta\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)} B_G B_\epsilon}\right)$ and $\delta = 2re^{-\sqrt{mp}} + \frac{1}{d^2}$. Then, with probability at least $1-\delta$ over the initialization, there exists $\tilde{\mathbf{a}}_i$'s such that $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) = \sum_{i=1}^{4m} \tilde{\mathbf{a}}_i \sigma\left(\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)$ satisfies*

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \le r B_{a1}\left(\frac{B_{x1} B_{G1} B_b}{\sqrt{mp} B_G B_\epsilon} + \sqrt{2\log(d)d}\gamma_\infty + B_\epsilon\right) + \mathrm{OPT}_{d,r,B_F,S_{p,\gamma_\infty,B_G,B_{G1}}^{\infty}},$$

and $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Proof of Lemma B.4.34. Recall $g^*(\mathbf{x}) = \sum_{j=1}^r \mathbf{a}_j^* \sigma(\left\langle \mathbf{w}_j^*, \mathbf{x}\right\rangle - \mathbf{b}_j^*)$, where $f^* \in \mathcal{F}_{d,r,B_F,S_{p,\gamma_\infty,B_G,B_{G1}}^{\infty}}$ is defined in Definition B.4.33 and let $s_j^* = \frac{\mathbf{b}_j^*}{|\mathbf{b}_j^*|}$. By Lemma B.3.3, with probability at least $1-\delta_1$, $\delta_1 = 2re^{-cmp}$, for all $j \in [r]$, we have $|G_{(\mathbf{w}_j^*,s_j^*),Nice}^{\infty}| \ge \frac{mp}{4}$. Then for all $i \in G_{(\mathbf{w}_j^*,s_j^*),Nice}^{\infty} \subseteq [2m]$, we have $-\ell'(0)\eta^{(1)} G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)\frac{\mathbf{b}_j^*}{\tilde{b}}$ only depend on $\mathbf{w}_i^{(0)}$ and $\mathbf{b}_i$,

which is independent of $\mathbf{a}_i^{(0)}$. Given Definition B.4.32, we have

$$-\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2\frac{\mathbf{b}_j^*}{\tilde{b}}\in\left[\ell'(0)\eta^{(1)}B_{x1}\frac{B_b}{\tilde{b}},-\ell'(0)\eta^{(1)}B_{x1}\frac{B_b}{\tilde{b}}\right]. \tag{B.448}$$

We split $[r]$ into $\Gamma=\{j\in[r]:|\mathbf{b}_j^*|<B_\epsilon\}$, $\Gamma_-=\{j\in[r]:\mathbf{b}_j^*\le -B_\epsilon\}$ and $\Gamma_+=\{j\in[r]:\mathbf{b}_j^*\ge B_\epsilon\}$. Let $\epsilon_a=\frac{B_{G1}B_b}{\sqrt{mp}B_GB_\epsilon}$. Then we know that for all $j\in\Gamma_+\cup\Gamma_-$, for all $i\in G^\infty_{(\mathbf{w}_j^*,s_j^*),Nice}$, we have

$$\Pr_{\mathbf{a}_i^{(0)}\sim\mathcal{N}(0,\sigma_a^2)}\left[\left|-\mathbf{a}_i^{(0)}\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2\frac{|\mathbf{b}_j^*|}{\tilde{b}}-1\right|\le\epsilon_a\right] \tag{B.449}$$

$$=\Pr_{\mathbf{a}_i^{(0)}\sim\mathcal{N}(0,\sigma_a^2)}\left[1-\epsilon_a\le-\mathbf{a}_i^{(0)}\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2\frac{|\mathbf{b}_j^*|}{\tilde{b}}\le1+\epsilon_a\right] \tag{B.450}$$

$$=\Pr_{g\sim\mathcal{N}(0,1)}\left[1-\epsilon_a\le g\Theta\left(\frac{\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2|\mathbf{b}_j^*|}{B_GB_\epsilon}\right)\le1+\epsilon_a\right] \tag{B.451}$$

$$=\Pr_{g\sim\mathcal{N}(0,1)}\left[(1-\epsilon_a)\Theta\left(\frac{B_GB_\epsilon}{\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2|\mathbf{b}_j^*|}\right)\le g\le(1+\epsilon_a)\Theta\left(\frac{B_GB_\epsilon}{\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2|\mathbf{b}_j^*|}\right)\right]$$

$$=\Theta\left(\frac{\epsilon_aB_GB_\epsilon}{\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2|\mathbf{b}_j^*|}\right) \tag{B.452}$$

$$\ge\Omega\left(\frac{\epsilon_aB_GB_\epsilon}{B_{G1}B_b}\right) \tag{B.453}$$

$$=\Omega\left(\frac{1}{\sqrt{mp}}\right). \tag{B.454}$$

Thus, with probability $\Omega\left(\frac{1}{\sqrt{mp}}\right)$ over $\mathbf{a}_i^{(0)}$, we have

$$\left|-\mathbf{a}_i^{(0)}\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2\frac{|\mathbf{b}_j^*|}{\tilde{b}}-1\right|\le\epsilon_a,\quad\left|\mathbf{a}_i^{(0)}\right|=O\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)}B_GB_\epsilon}\right). \tag{B.455}$$

Similarly, for $j\in\Gamma$, for all $i\in G^\infty_{(\mathbf{w}_j^*,s_j^*),Nice}$, with probability $\Omega\left(\frac{1}{\sqrt{mp}}\right)$ over $\mathbf{a}_i^{(0)}$, we have

$$\left|-\mathbf{a}_i^{(0)}\ell'(0)\eta^{(1)}\|G(\mathbf{w}_i^{(0)},\mathbf{b}_i)\|_2\frac{B_\epsilon}{\tilde{b}}-1\right|\le\epsilon_a,\quad\left|\mathbf{a}_i^{(0)}\right|=O\left(\frac{\tilde{b}}{-\ell'(0)\eta^{(1)}B_GB_\epsilon}\right). \tag{B.456}$$

For all $j\in[r]$, let $\Lambda_j\subseteq G^\infty_{(\mathbf{w}_j^*,s_j^*),Nice}$ be the set of i's such that condition Equation (B.455)

or Equation (B.456) are satisfied. By Chernoff bound and union bound, with probability at least $1-\delta_2$, $\delta_2 = re^{-\sqrt{mp}}$, for all $j \in [r]$ we have $|\Lambda_j| \geq \Omega(\sqrt{mp})$. We have for $\forall j \in \Gamma_+ \cup \Gamma_-, \forall i \in \Lambda_j$,

$$\left| \frac{|\mathbf{b}_j^*|}{\tilde{b}} \left\langle \mathbf{w}_i^{(1)}, \mathbf{x} \right\rangle - \left\langle \mathbf{w}_j^*, \mathbf{x} \right\rangle \right| \tag{B.457}$$

$$\leq \left| \left\langle -\mathbf{a}_i^{(0)} \ell'(0) \eta^{(1)} \| G(\mathbf{w}_i^{(0)}, \mathbf{b}_i) \|_2 \frac{|\mathbf{b}_j^*|}{\tilde{b}} \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2} - \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2}, \mathbf{x} \right\rangle + \left\langle \frac{\mathbf{w}_i^{(1)}}{\|\mathbf{w}_i^{(1)}\|_2} - \mathbf{w}_j^*, \mathbf{x} \right\rangle \right|$$

$$\leq \epsilon_a \|\mathbf{x}\|_2 + \sqrt{2\log(d)d}\gamma_\infty. \tag{B.458}$$

With probability $1 - \frac{1}{d^2}$ by Hoeffding's inequality. Similarly, for $\forall j \in \Gamma, \forall i \in \Lambda_j$,

$$\left| \frac{B_\epsilon}{\tilde{b}} \left\langle \mathbf{w}_i^{(1)}, \mathbf{x} \right\rangle - \left\langle \mathbf{w}_j^*, \mathbf{x} \right\rangle \right| \leq \epsilon_a \|\mathbf{x}\|_2 + \sqrt{2\log(d)d}\gamma_\infty. \tag{B.459}$$

If $i \in \Lambda_j$, $j \in \Gamma_+ \cup \Gamma_-$, set $\tilde{\mathbf{a}}_i = \mathbf{a}_j^* \frac{|\mathbf{b}_j^*|}{|\Lambda_j|\tilde{b}}$, if $i \in \Lambda_j$, $j \in \Gamma$, set $\tilde{\mathbf{a}}_i = \mathbf{a}_j^* \frac{B_\epsilon}{|\Lambda_j|\tilde{b}}$, otherwise set $\tilde{\mathbf{a}}_i = 0$, we have $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Finally, we have

$$\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \tag{B.460}$$

$$=\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) - \mathcal{L}_{\mathcal{D}}(g^*) + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.461}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) - g^*(\mathbf{x})\right|\right] + \mathcal{L}_{\mathcal{D}}(g^*) \tag{B.462}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{i=1}^{m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\tilde{b}\right)+\sum_{i=m+1}^{2m}\tilde{\mathbf{a}}_i\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle+\tilde{b}\right)-\sum_{j=1}^{r}\mathbf{a}_j^*\sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle-\mathbf{b}_j^*)\right|\right]$$

$$+\mathcal{L}_{\mathcal{D}}(g^*) \tag{B.463}$$

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_+}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\tilde{b}\right)-\sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle-\mathbf{b}_j^*)\right|\right|\right] \tag{B.464}$$

$$+\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_-}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle+\tilde{b}\right)-\sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle-\mathbf{b}_j^*)\right|\right|\right] \tag{B.465}$$

$$+\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{B_\epsilon}{\tilde{b}}\sigma\left(\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\tilde{b}\right)-\sigma(\langle\mathbf{w}_j^*,\mathbf{x}\rangle-\mathbf{b}_j^*)\right|\right|\right]+\mathcal{L}_{\mathcal{D}}(g^*)$$

(B.466)

$$\leq\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_+}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right] \tag{B.467}$$

$$+\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma_-}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{|\mathbf{b}_j^*|}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle-\left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right] \tag{B.468}$$

$$+\mathbb{E}_{(\mathbf{x},y)}\left[\left|\sum_{j\in\Gamma}\sum_{i\in\Lambda_j}\mathbf{a}_j^*\frac{1}{|\Lambda_j|}\left|\frac{B_\epsilon}{\tilde{b}}\left\langle\mathbf{w}_i^{(1)},\mathbf{x}\right\rangle+B_\epsilon-\left\langle\mathbf{w}_j^*,\mathbf{x}\right\rangle\right|\right|\right]+\mathcal{L}_{\mathcal{D}}(g^*) \tag{B.469}$$

$$\leq r\|\mathbf{a}^*\|_\infty(\epsilon_a\mathbb{E}_{(\mathbf{x},y)}\|\mathbf{x}\|_2+\sqrt{2\log(d)d}\gamma_\infty)+|\Gamma|\|\mathbf{a}^*\|_\infty B_\epsilon+\mathcal{L}_{\mathcal{D}}(g^*) \tag{B.470}$$

$$\leq rB_{a1}(\epsilon_a B_{x1}+\sqrt{2\log(d)d}\gamma_\infty)+|\Gamma|B_{a1}B_\epsilon+\mathrm{OPT}_{d,r,B_F,S^\infty_{p,\gamma,B_G,B_{G1}}}. \tag{B.471}$$

We finish the proof by union bound and $\delta\geq\delta_1+\delta_2+\frac{1}{d^2}$. □

Lemma B.4.35 (Empirical Gradient Concentration Bound for Single Coordinate)**.** *For $i\in[m]$, when $n\geq(\log(d))^6$, with probability at least $1-O\left(\exp\left(-n^{\frac{1}{3}}\right)\right)$ over training*

samples, we have

$$\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}-\frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}\right|\leq O\left(\frac{|\mathbf{a}_i|B_{x\infty}}{n^{\frac{1}{3}}}\right),\quad\forall j\in[d]. \tag{B.472}$$

Proof of Lemma B.4.35. First, we define,

$$z_{i,j}^{(l)}=\ell'(y^{(l)}f_\Xi(\mathbf{x}^{(l)}))y^{(l)}\left[\sigma'\left(\left\langle\mathbf{w}_i,\mathbf{x}^{(l)}\right\rangle-\mathbf{b}_i\right)\mathbf{x}_j^{(l)}\right] \tag{B.473}$$

$$-\mathbb{E}_{(\mathbf{x},y)}\left[\ell'(yf_\Xi(\mathbf{x}))y\left[\sigma'\left(\langle\mathbf{w}_i,\mathbf{x}\rangle-\mathbf{b}_i\right)\right]\mathbf{x}_j\right]. \tag{B.474}$$

As $|\ell'(z)|\leq 1,|y|\leq 1,|\sigma'(z)|\leq 1$, we have $z_{i,j}^{(l)}$ is zero-mean random variable with $\left|z_{i,j}^{(l)}\right|\leq 2B_{x\infty}$ as well as $\mathbb{E}\left[\left|z_{i,j}^{(l)}\right|_2^2\right]\leq 4B_{x\infty}^2$. Then by Bernstein Inequality, for $0<z<2B_{x\infty}$, we have

$$\Pr\left(\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}-\frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}\right|\geq|\mathbf{a}_i|z\right)=\Pr\left(\left|\frac{1}{n}\sum_{l\in[n]}z_{i,j}^{(l)}\right|\geq z\right) \tag{B.475}$$

$$\leq\exp\left(-n\cdot\frac{z^2}{8B_{x\infty}}\right). \tag{B.476}$$

Thus, for some $i\in[m]$, when $n\geq(\log(d))^6$, with probability at least $1-O\left(\exp\Theta\left(-n^{\frac{1}{3}}\right)\right)$, from a union bound over $j\in[d]$, we have, for $\forall j\in[d]$,

$$\left|\frac{\partial\widetilde{\mathcal{L}}_{\mathcal{Z}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}-\frac{\partial\mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial\mathbf{w}_{i,j}}\right|\leq O\left(\frac{|\mathbf{a}_i|B_{x\infty}}{n^{\frac{1}{3}}}\right). \tag{B.477}$$

□

Lemma B.4.36 (Existence of Good Networks under Empirical Risk. Modified version of Lemma B.3.13 Under Uniform Parity Setting)**.** *Suppose* $n>\Omega\left(\left(\frac{B_x}{\sqrt{B_{x2}}}+\log\frac{1}{p}+\frac{B_{x\infty}}{B_G|\ell'(0)|}+\frac{B_{x\infty}}{B_{G1}|\ell'(0)|}\right)^3+(\log($ *Let* $\lambda^{(1)}=\frac{1}{\eta^{(1)}}$*. For any* $B_\epsilon\in(0,B_b)$*, let* $\sigma_a=\Theta\left(\frac{\tilde{b}}{-|\ell'(0)|\eta^{(1)}B_GB_\epsilon}\right)$ *and* $\delta=2re^{-\sqrt{\frac{mp}{2}}}+\frac{1}{d^2}$*. Then, with probability at least* $1-\delta$ *over the initialization and training samples, there exists*

$\tilde{\mathbf{a}}_i$*'s such that* $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x}) = \sum_{i=1}^{4m} \tilde{\mathbf{a}}_i \sigma\left(\left\langle \mathbf{w}_i^{(1)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right)$ *satisfies*

$$\begin{aligned}
&\mathcal{L}_{\mathcal{D}}(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}) \qquad\qquad (B.478)\\
\leq& rB_{a1}\left(\frac{2B_{x1}B_{G1}B_b}{\sqrt{mp}B_G B_\epsilon} + \sqrt{2\log(d)d}\left(\gamma_\infty + O\left(\frac{B_{x\infty}}{B_G|\ell'(0)|n^{\frac{1}{3}}}\right)\right) + B_\epsilon\right) + \mathrm{OPT}_{d,r,B_F,S^\infty_{p,\gamma,B_G,B_{G1}}},
\end{aligned}$$

and $\|\tilde{\mathbf{a}}\|_0 = O\left(r(mp)^{\frac{1}{2}}\right)$, $\|\tilde{\mathbf{a}}\|_2 = O\left(\frac{B_{a2}B_b}{\tilde{b}(mp)^{\frac{1}{4}}}\right)$, $\|\tilde{\mathbf{a}}\|_\infty = O\left(\frac{B_{a1}B_b}{\tilde{b}(mp)^{\frac{1}{2}}}\right)$.

Proof of Lemma B.4.36. Denote $\rho = O\left(\exp\Theta\left(-n^{\frac{1}{3}}\right)\right)$ and $\beta = O\left(\frac{B_{x\infty}}{n^{\frac{1}{3}}}\right)$. Note that by symmetric initialization, we have $\ell'(yf_{\Xi^{(0)}}(\mathbf{x})) = |\ell'(0)|$ for any $\mathbf{x} \in \mathcal{X}$, so that, by Lemma B.4.35, we have $\left|\widetilde{G}(\mathbf{w}_i^{(0)}, \mathbf{b}_i)_j - G(\mathbf{w}_i^{(0)}, \mathbf{b}_i)_j\right| \leq \frac{\beta}{|\ell'(0)|}$ with probability at least $1-\rho$. Thus, by union bound, we can see that $S^\infty_{p,\gamma_\infty,B_G,B_{G1}} \subseteq \widetilde{S}^\infty_{p-\rho,\gamma_\infty+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|},B_{G1}+\frac{\beta}{|\ell'(0)|}}$. Consequently, we have $\mathrm{OPT}_{d,r,B_F,\widetilde{S}^\infty_{p-\rho,\gamma_\infty+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|},B_{G1}+\frac{\beta}{|\ell'(0)|}}} \leq \mathrm{OPT}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}}$. Exactly follow the proof in Lemma B.3.4 by replacing $S^\infty_{p,\gamma_\infty,B_G,B_{G1}}$ to $\widetilde{S}^\infty_{p-\rho,\gamma_\infty+\frac{\beta}{B_G|\ell'(0)|},B_G-\frac{\beta}{|\ell'(0)|},B_{G1}+\frac{\beta}{|\ell'(0)|}}$. Then, we finish the proof by $\rho \leq \frac{p}{2}$, $\frac{\beta}{|\ell'(0)|} \leq (1-1/\sqrt{2})B_G$, $\frac{\beta}{|\ell'(0)|} \leq (\sqrt{2}-1)B_{G1}$. □

Theorem B.4.37 (Online Convex Optimization under Empirical Risk. Modified version of Theorem B.3.17 Under Uniform Parity Setting)**.** *Consider training by Algorithm 1, and any* $\delta \in (0,1)$. *Assume* $d \geq \log m, \delta \leq O(\frac{1}{d^2})$. *Set*

$$\begin{aligned}
&\sigma_{\mathbf{w}} > 0, \quad \tilde{b} > 0, \quad \eta^{(t)} = \eta,\ \lambda^{(t)} = 0 \text{ for all } t \in \{2,3,\ldots,T\},\\
&\eta^{(1)} = \Theta\left(\frac{\min\{O(\eta), O(\eta\tilde{b})\}}{-\ell'(0)(B_{x1}\sigma_{\mathbf{w}}\sqrt{d}+\tilde{b})}\right),\ \lambda^{(1)} = \frac{1}{\eta^{(1)}}, \quad \sigma_a = \Theta\left(\frac{\tilde{b}(mp)^{\frac{1}{4}}}{-\ell'(0)\eta^{(1)}B_{x1}\sqrt{B_G B_b}}\right).
\end{aligned}$$

Let $0 < T\eta B_{x1} \leq o(1)$, $m = \Omega\left(\frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right)$ *and* $n > \Omega\left(\left(\frac{B_x}{\sqrt{B_{x2}}} + \log\frac{Tm}{p\delta} + (1+\frac{1}{B_G}+\frac{1}{B_{G1}})\frac{B_{x\infty}}{|\ell'(0)|}\right)^3\right)$. *With probability at least* $1-\delta$ *over the initialization and training samples, there exists* $t \in [T]$

such that

$$\mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.479}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}\left(\frac{2\sqrt{2}\sqrt{B_{x1}B_{G1}}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}} + \sqrt{2\log(d)d}\left(\gamma_\infty + O\left(\frac{B_{x\infty}}{B_G|\ell'(0)|n^{\frac{1}{3}}}\right)\right)\right)$$

$$+ \eta\left(\sqrt{r}B_{a2}B_bT\eta B_{x1}^2 + m\tilde{b}\right)O\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right) + O\left(\frac{B_{a2}^2B_b^2}{\eta T\tilde{b}^2(mp)^{\frac{1}{2}}}\right) \tag{B.480}$$

$$+ \frac{1}{n^{\frac{1}{3}}}O\Bigg(\left(\frac{rB_{a1}B_b}{\tilde{b}} + m\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + \frac{\tilde{b}}{B_{x1}}\right)\right) \tag{B.481}$$

$$\cdot\left(\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + T\eta^2B_{x1}\tilde{b}\right)B_x + \tilde{b}\right) + 2\Bigg) \tag{B.482}$$

$$+ \frac{1}{n^{\frac{1}{3}}}O\left(m\eta\left(\frac{\tilde{b}\sqrt{\log m}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + T\eta^2B_{x1}\tilde{b}\right)\sqrt{B_{x2}}\right). \tag{B.483}$$

Furthermore, for any $\epsilon \in (0,1)$*, set*

$$\tilde{b} = \Theta\left(\frac{B_G^{\frac{1}{4}}B_{a2}B_b^{\frac{3}{4}}}{\sqrt{rB_{a1}}}\right), \quad m = \Omega\left(\frac{1}{p\epsilon^4}\left(rB_{a1}\sqrt{B_{x1}B_{G1}}\sqrt{\frac{B_b}{B_G}}\right)^4 + \frac{1}{\sqrt{\delta}} + \frac{1}{p}\left(\log\left(\frac{r}{\delta}\right)\right)^2\right),$$

$$\eta = \Theta\left(\frac{\epsilon}{\left(\frac{\sqrt{r}B_{a2}B_bB_{x1}}{(mp)^{\frac{1}{4}}} + m\tilde{b}\right)\left(\frac{\sqrt{\log m}B_{x1}(mp)^{\frac{1}{4}}}{\sqrt{B_bB_G}} + 1\right)}\right), \quad T = \Theta\left(\frac{1}{\eta B_{x1}(mp)^{\frac{1}{4}}}\right),$$

$$n = \Omega\left(\left(\frac{mB_xB_{a2}^2\sqrt{B_b}(mp)^{\frac{1}{2}}\log m}{\epsilon rB_{a1}\sqrt{B_G}}\right)^3 + \left(\frac{B_x}{\sqrt{B_{x2}}} + \log\frac{Tm}{p\delta} + (1 + \frac{1}{B_G} + \frac{1}{B_{G1}})\frac{B_{x\infty}}{|\ell'(0)|}\right)^3\right),$$

we have there exists $t \in [T]$ *with*

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}})(\mathbf{x}) \neq y] \leq \mathcal{L}_{\mathcal{D}}\left(f_{\Xi^{(t)}}\right) \tag{B.484}$$

$$\leq \mathrm{OPT}_{d,r,B_F,S^\infty_{p,\gamma_\infty,B_G,B_{G1}}} + rB_{a1}\sqrt{2\log(d)d}\left(\gamma_\infty + O\left(\frac{B_{x\infty}}{B_G|\ell'(0)|n^{\frac{1}{3}}}\right)\right) + \epsilon. \tag{B.485}$$

Proof of Theorem B.4.37. Proof of the theorem and parameter choices remain the same as Theorem B.3.17 except for setting $B_\epsilon = \frac{\sqrt{B_{x1}B_{G1}}}{(mp)^{\frac{1}{4}}}\sqrt{\frac{B_b}{B_G}}$ and apply Lemma B.4.36. □

Feature Learning of Uniform Parity Functions

We denote

$$g_{i,j} = \mathbb{E}_{(\mathbf{x},y)} \left[y\sigma' \left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x} \right\rangle - \mathbf{b}_i \right] \mathbf{x}_j \right] \tag{B.486}$$

$$\xi_k = (-1)^{\frac{k-1}{2}} \frac{\binom{\frac{n-1}{2}}{\frac{k-1}{2}}}{\binom{n-1}{k-1}} \cdot 2^{-(n-1)} \binom{n-1}{\frac{n-1}{2}}. \tag{B.487}$$

Lemma B.4.38 (Uniform Parity Functions: Gradient Feature Learning. Corollary of Lemma 3 in [26])**.** *Assume that $n \geq 2(k+1)^2$. Then, the following holds:*

If $j \in A$, then

$$g_{i,j} = \xi_{k-1} \prod_{l \in A \setminus \{j\}} (\mathbf{w}_{i,l}^{(0)}). \tag{B.488}$$

If $i \notin A$, then

$$g_{i,j} = \xi_{k-1} \prod_{l \in A \cup \{j\}} (\mathbf{w}_{i,l}^{(0)}). \tag{B.489}$$

Lemma B.4.39 (Uniform Parity Functions: Existence of Good Networks (Alternative))**.** *Assume the same condition as in Lemma B.4.38. Define*

$$D = \frac{\sum_{l \in A} \mathbf{M}_l}{\| \sum_{l \in A} \mathbf{M}_l \|_2} \tag{B.490}$$

and

$$g^*(\mathbf{x}) = \sum_{i=0}^{k} (-1)^i \sqrt{k} \cdot \left[\sigma\left(\langle D, \mathbf{x}\rangle - \frac{2i-k-1}{\sqrt{k}}\right) - 2\sigma\left(\langle D, \mathbf{x}\rangle - \frac{2i-k}{\sqrt{k}}\right) + \sigma\left(\langle D, \mathbf{x}\rangle - \frac{2i-k+1}{\sqrt{k}}\right)\right]. \tag{B.491}$$

For $\mathcal{D}_{parity-uniform}$ *setting, we have* $g^* \in \mathcal{F}_{d,3(k+1),B_F,S^{\infty}_{p,\gamma_\infty,B_G,B_{G1}}}$ *where* $B_F = (B_{a1}, B_{a2}, B_b) = \left(2\sqrt{k}, 2\sqrt{(k(k+1))}, \frac{k+1}{\sqrt{k}}\right)$, $p = \Theta\left(\frac{1}{2^{k-1}}\right)$, $\gamma_\infty = O\left(\frac{\sqrt{k}}{d-k}\right)$, $B_G = \Theta(B_{G1}) = \Theta(d^{-k})$ *and* $B_{x1} = \sqrt{d}$, $B_{x_2} = d$. *We also have* $\mathrm{OPT}_{d,3(k+1),B_F,S^{\infty}_{p,\gamma_\infty,B_G,B_{G1}}} = 0$.

Proof of Lemma B.4.39. Fix index i, with probability $p_1 = \Theta(2^{-k})$, we will have $\mathbf{w}_{i,j}^{(0)} = \mathrm{sign}(\mathbf{a}_i^{(0)}) \cdot \mathrm{sign}(\xi_{k-1})$, for $\forall j$. For $\mathbf{w}_i^{(0)}$ that satisfy these conditions, we will have:

$$\mathrm{sign}(\mathbf{a}_i^{(0)}) g_{i,j} = |\xi_{k-1}|, \ \ \forall j \in A \tag{B.492}$$

$$\mathrm{sign}(\mathbf{a}_i^{(0)}) g_{i,j} = |\xi_{k+1}|, \ \ \forall j \notin A. \tag{B.493}$$

Then by Lemma 4 in [26], we have

$$\left\| \frac{\mathrm{sign}(\mathbf{a}_i^{(0)}) G(\mathbf{w}_i^{(0)}, \tilde{b})}{\|G(\mathbf{w}_i^{(0)}, \tilde{b})\|} - D \right\|_\infty \leq \max\left\{ \left| \frac{1}{k\sqrt{\frac{1}{k} + \frac{1}{d-k}}} - \frac{1}{\sqrt{k}} \right|, \left| \frac{1}{(d-k)\sqrt{\frac{1}{k} + \frac{1}{d-k}}} \right| \right\} \tag{B.494}$$

$$\leq \frac{\sqrt{k}}{d-k} \tag{B.495}$$

and

$$\|\mathrm{sign}(\mathbf{a}_i^{(0)}) G(\mathbf{w}_i^{(0)}, \tilde{b})\|_2 = \sqrt{k|\xi_{k-1}|^2 + (d-k)|\xi_{k+1}|^2} = \Theta(d^{\Theta(k)}). \tag{B.496}$$

From here, we can see that if we set $\gamma_\infty = \frac{\sqrt{k}}{d-k}$, $B_G = B_{G1} = \sqrt{k|\xi_{k-1}|^2 + (d-k)|\xi_{k+1}|^2}$, $p = p_1$, we will have $(D, +1), (D, -1) \in S^{\infty}_{p,\gamma_\infty,B_G,B_{G1}}$ by our symmetric initialization.

As a result, we have $f^* \in \mathcal{F}_{d,3(k+1),B_F,S^{\infty}_{p,\gamma_\infty},B_G,B_{G1}}$. Finally, it is easy to verify that $f^*(\mathbf{x}) = \mathrm{XOR}(\mathbf{x}_A)$, thus $\mathrm{OPT}_{d,3(k+1),B_F,S^{\infty}_{p,\gamma_\infty},B_G,B_{G1}} = 0$. □

Theorem B.4.40 (Uniform Parity Functions: Main Result (Alternative))**.** *For $\mathcal{D}_{parity-uniform}$ setting, for any $\delta \in (0,1)$ satisfying $\delta \leq O(\frac{1}{d^2})$ and for any $\epsilon \in (0,1)$ when*

$$m = \mathrm{poly}\left(\log\left(\frac{1}{\delta}\right), \frac{1}{\epsilon}, 2^{\Theta(k)}, d\right), T = \Theta\left(d^{\Theta(k)}\right), n = \Theta\left(d^{\Theta(k)}\right) \tag{B.497}$$

trained by Algorithm 1 with hinge loss, with probability at least $1-\delta$ over the initialization, with proper hyper-parameters, there exists $t \in [T]$ such that

$$\Pr[\mathrm{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq \frac{k^2\sqrt{d\log(d)}}{d-k} + \epsilon. \tag{B.498}$$

Proof of Theorem B.4.40. Plug the values of parameters into Theorem B.4.37 and directly get the result. □

B.4.7 Multiple Index Model with Low Degree Polynomial

Problem Setup

The multiple-index data problem has been used for studying network learning [32, 62]. We consider proving guarantees for the setting in [62], following our framework. We use the properties of the problem to prove the key lemma (i.e., the existence of good networks) in our framework and then derive the final guarantee from our theorem of the simple setting (Theorem 3.2.4).

Data Distributions. We draw input from the distribution $\mathcal{D}_{\mathcal{X}} = \mathcal{N}(0, I_{d\times d})$, and we assume the target function is $g^*(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$, where g^* is a degree τ polynomial normalized so that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}}[g^*(\mathbf{x})^2] = 1$.

Assumption B.4.41. There exists linearly independent vectors $u_1, \ldots, u_r$ such that $g^*(\mathbf{x}) = g(\langle \mathbf{x}, u_1\rangle, \ldots, \langle \mathbf{x}, u_r\rangle)$. $H := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}}[\nabla^2 g^*(\mathbf{x})]$ has rank r, where H is a Hessian matrix.

Definition B.4.42. Denote the normalized condition number of H by

$$\kappa := \frac{\|H^\dagger\|}{\sqrt{r}}. \tag{B.499}$$

Initialization and Loss. For $\forall i \in [m]$, we use the following initialization:

$$\mathbf{a}_i^{(0)} \sim \{-1, 1\}, \;\; \mathbf{w}_i^{(0)} \sim \mathcal{N}\left(0, \frac{1}{d} I_{d\times d}\right) \;\text{ and }\; \mathbf{b}_i = 0. \tag{B.500}$$

For this regression problem, we use mean square loss:

$$\mathcal{L}_{\mathcal{D}_\mathcal{X}}(f_\Xi) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\left[(f_\Xi(\mathbf{x}) - g^*(\mathbf{x}))^2\right]. \tag{B.501}$$

Training Process. We use the following one-step training algorithm for this specific data distribution.

Algorithm 7 Network Training via Gradient Descent [62]. Special case of Algorithm 4

Initialize $(\mathbf{a}^{(0)}, \mathbf{W}^{(0)}, \mathbf{b})$ as in Equation (3.8) and Equation (B.500); Sample $\mathcal{Z} \sim \mathcal{D}_\mathcal{X}^n$
$\rho = \frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}} g^*(\mathbf{x}), \;\; \beta = \frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}} g^*(\mathbf{x})\mathbf{x}$
$y = g^*(\mathbf{x}) - \rho - \beta\cdot\mathbf{x}$
$\mathbf{W}^{(1)} = \mathbf{W}^{(0)} - \eta^{(1)}(\nabla_\mathbf{W}\widetilde{\mathcal{L}}_\mathcal{Z}(f_{\Xi^{(0)}}) + \lambda^{(1)}\mathbf{W}^{(0)})$
Re-initialize $\mathbf{b}_i \sim \mathcal{N}(0,1)$
for $t = 2$ **to** T **do**
  $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} - \eta^{(t)}\nabla_\mathbf{a}\widetilde{\mathcal{L}}_\mathcal{Z}(f_{\Xi^{(t-1)}})$
end for

Lemma B.4.43 (Multiple Index Model with Low Degree Polynomial: Existence of Good Networks. Rephrase of Lemma 25 in [62])**.** *Assume $n \geq d^2 r\kappa^2(C_l\log(nmd))^{\tau+1}$, $d \geq C_d\kappa r^{3/2}$, and $m \geq r^\tau\kappa^{2\tau}(C_l\log(nmd))^{6\tau+1}$ for sufficiently large constants C_d, C_l, and let $\eta^{(1)} = \sqrt{\frac{d}{(C_l\log(nmd))^3}}$ and $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$. Then with probability $1 - \frac{1}{\mathrm{poly}(m,d)}$, there exists $\tilde{\mathbf{a}} \in \mathbb{R}^m$ such that $f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}$ satisfies*

$$\mathcal{L}_{\mathcal{D}_\mathcal{X}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}\right) \leq O\left(\frac{1}{n} + \frac{r^\tau\kappa^{2\tau}(C_l\log(nmd))^{6\tau+1})}{m}\right) \tag{B.502}$$

and

$$\|\tilde{\mathbf{a}}\|_2^2 \leq O\left(\frac{r^\tau \kappa^{2\tau}(C_l \log(nmd))^{6\tau}}{m}\right). \tag{B.503}$$

Multiple Index Model: Final Guarantee

Considering training by Algorithm 7, we have the following results.

Theorem B.4.44 (Multiple Index Model with Low Degree Polynomial: Main Result)**.** *Assume* $n \geq \Omega\left(d^2 r\kappa^2 (C_l \log(nmd))^{\tau+1} + m\right)$, $d \geq C_d \kappa r^{3/2}$, *and* $m \geq \Omega\left(\frac{1}{\epsilon} r^\tau \kappa^{2\tau} (C_l \log(nmd))^{6\tau+1}\right)$ *for sufficiently large constants* C_d, C_l. *Let* $\eta^{(1)} = \sqrt{\frac{d}{(C_l \log(nmd))^3}}$ *and* $\lambda^{(1)} = \frac{1}{\eta^{(1)}}$, *and* $\eta = \eta^{(t)} = \Theta(m^{-1})$, *for all* $t \in \{2, 3, \ldots, T\}$. *For any* $\epsilon \in (0, 1)$, *if* $T \geq \Omega\left(\frac{m^2}{\epsilon}\right)$, *then with properly set parameters and Algorithm 7, with high probability that there exists* $t \in [T]$ *such that*

$$\mathcal{L}_{\mathcal{D}_\mathcal{X}} f_{(\mathbf{a}^{(t)}, \mathbf{W}^{(1)}, \mathbf{b})} \leq \epsilon. \tag{B.504}$$

Proof of Theorem B.4.44. By Lemma B.4.43, we have for properly chosen hyper-parameters,

$$\mathrm{OPT}_{\mathbf{W}^{(1)}, \mathbf{b}, B_{a2}} \leq \mathcal{L}_{\mathcal{D}_\mathcal{X}}\left(f_{(\tilde{\mathbf{a}}, \mathbf{W}^{(1)}, \mathbf{b})}\right) \leq O\left(\frac{1}{n} + \frac{r^\tau \kappa^{2\tau}(C_l \log(nmd))^{6\tau+1})}{m}\right) \tag{B.505}$$

$$\leq \frac{\epsilon}{3}. \tag{B.506}$$

We compute the L-smooth constant of $\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a},\mathbf{W}^{(1)},\mathbf{b})}\right)$ to $\mathbf{a}$.

$$\left\|\nabla_{\mathbf{a}}\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}_1,\mathbf{W}^{(1)},\mathbf{b})}\right)-\nabla_{\mathbf{a}}\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}_2,\mathbf{W}^{(1)},\mathbf{b})}\right)\right\|_2 \tag{B.507}$$

$$=\left\|\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left[2\left(f_{(\mathbf{a}_1,\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x})-g^*-f_{(\mathbf{a}_2,\mathbf{W}^{(1)},\mathbf{b})}(\mathbf{x})+g^*\right)\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})\right]\right\|_2 \tag{B.508}$$

$$\leq\left\|\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left[2\left(\mathbf{a}_1^\top\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})-\mathbf{a}_2^\top\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})\right)\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})\right]\right\|_2 \tag{B.509}$$

$$\leq\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left[2\left\|\mathbf{a}_1-\mathbf{a}_2\right\|_2\left\|\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})\right\|_2^2\right]. \tag{B.510}$$

By the proof of Lemma 25 in [62], we have for $\forall i\in[4m]$, with probability at least $1-\frac{1}{\mathrm{poly}(m,d)}$, $|\langle\mathbf{w}_i,\mathbf{x}\rangle|\leq 1$, with some large polynomial $\mathrm{poly}(m,d)$. As a result, we have

$$\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left\|\mathbf{W}^{(1)\top}\mathbf{x}\right\|_2^2\leq m+\frac{1}{\mathrm{poly}(m,d)}\leq O(m). \tag{B.511}$$

Thus, we have,

$$L=O\left(\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left\|\sigma(\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b})\right\|_2^2\right) \tag{B.512}$$

$$\leq O\left(\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left\|\mathbf{W}^{(1)\top}\mathbf{x}-\mathbf{b}\right\|_2^2\right) \tag{B.513}$$

$$\leq O\left(\frac{1}{n}\sum_{\mathbf{x}\in\mathcal{Z}}\left\|\mathbf{W}^{(1)\top}\mathbf{x}\right\|_2^2+\|\mathbf{b}\|_2^2\right) \tag{B.514}$$

$$\leq O(m). \tag{B.515}$$

This means that we can let $\eta=\Theta\left(m^{-1}\right)$ and we will get our convergence result. We can bound $\|\mathbf{a}^{(1)}\|_2$ and $\|\tilde{\mathbf{a}}\|_2$ by $\|\mathbf{a}^{(1)}\|_2=O\left(\sqrt{m}\right)$ and $\|\tilde{\mathbf{a}}\|_2=O\left(\frac{r^\tau\kappa^{2\tau}(C_l\log(nmd))^{6\tau}}{m}\right)=O(\epsilon)$. So, if we choose $T\geq\Omega\left(\frac{m}{\epsilon\eta}\right)$, there exists $t\in[T]$ such that $\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\mathbf{a}^{(t)},\mathbf{W}^{(1)},\mathbf{b})}\right)-\widetilde{\mathcal{L}}_{\mathcal{Z}}\left(f_{(\tilde{\mathbf{a}},\mathbf{W}^{(1)},\mathbf{b})}\right)\leq O\left(\frac{L\|\mathbf{a}^{(1)}-\tilde{\mathbf{a}}\|_2^2}{T}\right)\leq\epsilon/3$.

We also have $\sqrt{\frac{\|\tilde{\mathbf{a}}\|_2^2(\|\mathbf{W}^{(1)}\|_F^2B_x^2+\|\mathbf{b}\|_2^2)}{n}}\leq\frac{\epsilon}{3}$. Then our theorem gets proved by Theorem 3.2.4. □

Discussion. We would like to unify [62], whcih are very closely related to our framework: their analysis for multiple index data follows the same principle and analysis approach as our general framework, although it does not completely fit into our Theorem 3.2.12 due to some technical differences. We can cover it with our Theorem 3.2.4.

Our work and [62] share the same principle and analysis approach. [62] shows that the first layer learns good features by one gradient step update, which can approximate the true labels by a low-degree polynomial function. Then, a classifier (the second layer) is trained on top of the learned first layer which leads to the final guarantees. This is consistent with our framework: we first show that the first layer learns good features by one gradient step update, which can approximate the true labels, and then show a good classifier can be learned on the first layer.

Our work and [62] have technical differences. First, in the second stage, [62] fix the first layer and only update the top layer which is a convex optimization. Our framework allows updates in the first layer and uses online convex learning techniques for the analysis. Second, they consider the square loss (this is used to calculate Hermite coefficients explicitly for gradients, which are useful in the low-degree polynomial function approximation). While in our online convex learning analysis, we need boundedness of the derivative of the loss to show that the first layer weights' changes are bounded in the second stage. Given the above two technicalities, we analyze their training algorithm (Algorithm 4) which fixes the first layer weights and fits into our Theorem 3.2.4.

B.5 Auxiliary Lemmas

In this section, we present some Lemmas used frequently.

Lemma B.5.1 (Lemmas on Gradients)**.**

$$\nabla_{\mathbf{W}} \mathcal{L}_{(\mathbf{x},y)}(f_\Xi) = \left[\frac{\partial \mathcal{L}_{(\mathbf{x},y)}(f_\Xi)}{\partial \mathbf{w}_1}, \ldots, \frac{\partial \mathcal{L}_{(\mathbf{x},y)}(f_\Xi)}{\partial \mathbf{w}_i}, \ldots, \frac{\partial \mathcal{L}_{(\mathbf{x},y)}(f_\Xi)}{\partial \mathbf{w}_{4m}}\right], \tag{B.516}$$

$$\frac{\partial \mathcal{L}_{(\mathbf{x},y)}(f_\Xi)}{\partial \mathbf{w}_i} = \mathbf{a}_i \ell'(y f_\Xi(\mathbf{x})) y \left[\sigma'\left(\langle \mathbf{w}_i, \mathbf{x}\rangle - \mathbf{b}_i\right)\right] \mathbf{x}, \tag{B.517}$$

$$\nabla_{\mathbf{W}} \mathcal{L}_{\mathcal{D}}(f_\Xi) = \left[\frac{\partial \mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial \mathbf{w}_1}, \ldots, \frac{\partial \mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial \mathbf{w}_i}, \ldots, \frac{\partial \mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial \mathbf{w}_{4m}}\right], \tag{B.518}$$

$$\frac{\partial \mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial \mathbf{w}_i} = \mathbf{a}_i \mathbb{E}_{(\mathbf{x},y)}\left[\ell'(y f_\Xi(\mathbf{x})) y \left[\sigma'\left(\langle \mathbf{w}_i, \mathbf{x}\rangle - \mathbf{b}_i\right)\right] \mathbf{x}\right], \tag{B.519}$$

$$\frac{\partial \mathcal{L}_{\mathcal{D}}(f_\Xi)}{\partial \mathbf{a}_i} = \mathbb{E}_{(\mathbf{x},y)}\left[\ell'(y f_\Xi(\mathbf{x})) y \left[\sigma\left(\langle \mathbf{w}_i, \mathbf{x}\rangle - \mathbf{b}_i\right)\right]\right]. \tag{B.520}$$

Proof. These can be verified by direct calculation. □

Lemma B.5.2 (Property of Symmetric Initialization)**.** *For any $\mathbf{x} \in \mathbb{R}^d$, we have $f_{\Xi^{(0)}}(\mathbf{x}) = 0$. For all $i \in [2m]$, we have $\mathbf{w}_i^{(1)} = -\mathbf{w}_{i+2m}^{(1)}$. When input data is symmetric, i.e, $\mathbb{E}_{(\mathbf{x},y)}[y\mathbf{x}] = \mathbf{0}$, for all $i \in [m]$, we have $\mathbf{w}_i^{(1)} = \mathbf{w}_{i+m}^{(1)}$.*

Proof of Lemma B.5.2. By symmetric initialization, we have $f_{\Xi^{(0)}}(\mathbf{x}) = 0$. For all $i \in [2m]$, we have

$$\begin{aligned}
\mathbf{w}_i^{(1)} =& -\eta^{(1)} \ell'(0) \mathbf{a}_i^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right] \mathbf{x}\right] && \text{(B.521)}\\
=& \eta^{(1)} \ell'(0) \mathbf{a}_{i+2m}^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_{i+2m}^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_{i+2m}\right] \mathbf{x}\right] && \text{(B.522)}\\
=& -\mathbf{w}_{i+2m}^{(1)}. && \text{(B.523)}
\end{aligned}$$

When $\mathbb{E}_{(\mathbf{x},y)}[y\mathbf{x}] = \mathbf{0}$, for all $i \in [m]$, we have

$$\begin{aligned}
\mathbf{w}_i^{(1)} =& -\eta^{(1)} \ell'(0) \mathbf{a}_i^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle \mathbf{w}_i^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_i\right] \mathbf{x}\right] && \text{(B.524)}\\
=& \eta^{(1)} \ell'(0) \mathbf{a}_{i+m}^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle -\mathbf{w}_{i+m}^{(0)}, \mathbf{x}\right\rangle + \mathbf{b}_{i+m}\right] \mathbf{x}\right] && \text{(B.525)}\\
=& \eta^{(1)} \ell'(0) \mathbf{a}_{i+m}^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[y\sigma'\left[\left\langle -\mathbf{w}_{i+m}^{(0)}, \mathbf{x}\right\rangle + \mathbf{b}_{i+m}\right] \mathbf{x} - y\mathbf{x}\right] && \text{(B.526)}\\
=& \eta^{(1)} \ell'(0) \mathbf{a}_{i+m}^{(0)} \mathbb{E}_{(\mathbf{x},y)}\left[-y\sigma'\left[\left\langle \mathbf{w}_{i+m}^{(0)}, \mathbf{x}\right\rangle - \mathbf{b}_{i+m}\right] \mathbf{x}\right] && \text{(B.527)}\\
=& \mathbf{w}_{i+m}^{(1)}. && \text{(B.528)}
\end{aligned}$$

□

Lemma B.5.3 (Property of Direction Neighborhood). *If* $\mathbf{w} \in \mathcal{C}_{D,\gamma}$*, we have* $\rho\mathbf{w} \in \mathcal{C}_{D,\gamma}$ *for any* $\rho \neq 0$*. We also have* $\mathbf{0} \notin \mathcal{C}_{D,\gamma}$*. Also, if* $(D, s) \in S_{p,\gamma,B_G}$*, we have* $(-D, s) \in S_{p,\gamma,B_G}$*.*

Proof. These can be verified by direct calculation. □

Lemma B.5.4 (Maximum Gaussian Tail Bound). M_n *is the maximum of* n *i.i.d. standard normal Gaussian. Then*

$$\Pr\left(M_n \geq \sqrt{2\log n} + \frac{z}{\sqrt{2\log n}}\right) \leq e^{-z}. \tag{B.529}$$

Proof. These can be verified by direct calculation. □

Lemma B.5.5 (Chi-squared Tail Bound). *If* X *is a* $\chi^2(k)$ *random variable. Then,* $\forall z \in \mathbb{R}$*, we have*

$$\Pr(X \geq k + 2\sqrt{kz} + 2z) \leq e^{-z}. \tag{B.530}$$

Proof. These can be verified by direct calculation. □

Lemma B.5.6 (Gaussian Tail Bound). *If* g *is standard Gaussian and* $z > 0$*, we have*

$$\frac{1}{\sqrt{2\pi}}\frac{z}{z^2+1}e^{-z^2/2} < \Pr_{g\sim\mathcal{N}(0,1)}[g > z] < \frac{1}{\sqrt{2\pi}}\frac{1}{z}e^{-z^2/2}. \tag{B.531}$$

Proof. These can be verified by direct calculation. □

Lemma B.5.7 (Gaussian Tail Expectation Bound). *If* g *is standard Gaussian and* $z \in \mathbb{R}$*, we have*

$$|\mathbb{E}_{g\sim\mathcal{N}(0,1)}[\mathbb{I}[g > z]g]| < 2\Pr_{g\sim\mathcal{N}(0,1)}[g > z]^{0.9}. \tag{B.532}$$

Proof of Lemma B.5.7. For any $p \in (0,1)$, we have

$$\left| \int_{-\infty}^{\sqrt{2}\mathrm{erf}^{-1}(2p-1)} \frac{e^{-\frac{x^2}{2}} x}{\sqrt{2\pi}} dx \right| < 2p^{0.9}, \tag{B.533}$$

where $\sqrt{2}\mathrm{erf}^{-1}(2p-1)$ is the quantile function of the standard Gaussian. We finish the proof by replacing p to be $\Pr_{g\sim\mathcal{N}(0,1)}[g > z]$. □

Lemma B.5.8. *If a function g satisfy $h(n+2) = 2h(n+1) - (1-\rho^2)h(n) + \beta$ for $n \in \mathbb{N}_+$ where $\rho, \beta > 0$, then $h(n) = -\frac{\beta}{\rho^2} + c_1(1-\rho)^n + c_2(1+\rho)^n$, where c_1, c_2 only depends on $h(1)$ and $h(2)$.*

Proof. These can be verified by direct calculation. □

Lemma B.5.9 (Rademacher Complexity Bounds. Rephrase of Lemma 48 in [62])**.** *For fixed $\mathbf{W}, \mathbf{b}$, let $\mathcal{F} = \{f_{(\mathbf{a},\mathbf{W},\mathbf{b})} : \|\mathbf{a}\| \le B_{a2}\}$. Then,*

$$\mathfrak{R}(\mathcal{F}) \le \sqrt{\frac{B_{a2}^2(\|\mathbf{W}\|_F^2 B_x^2 + \|\mathbf{b}\|_2^2)}{n}}. \tag{B.534}$$

Appendix C

Discussions, Complete Proofs and Additional Experiments in Chapter 4 Towards Few-shot Adaptation of Foundation Models via Multitask Finetuning

In this appendix, we first state our limitation in Appendix C.1. Then, we provide more related work in **??**. The proof of our theoretical results for the binary case is presented in Appendix C.2, where we formalize the theoretical settings and assumptions and elaborate on the results to contrastive pretraining in Appendix C.2.1 and supervised pretraining in Appendix C.2.2. We prove the main theory in Appendix C.2.4, which is a direct derivative of C.2.1 and C.2.2. We generalize the setting to multiclass and provide proof in Appendix C.3. We include the full proof of the general linear case study in Appendix C.4. We provide additional experimental results of vision tasks in Appendix C.5, language tasks in Appendix C.6, and vision-language tasks in Appendix C.7.

C.1 Limitation

We recognize an interesting phenomenon within multitask finetuning and dig into deeper exploration with theoretical analysis, while our experimental results may or may not beat state-of-the-art (SOTA) performance, as our focus is not on presenting multitask finetuning as a novel approach nor on achieving SOTA performance. On the other hand, the estimation of our diversity and consistency parameters accurately on real-world datasets is valuable but time-consuming. Whether there exists an efficient algorithm to estimate these parameters is unknown. We leave this challenging problem as our future work.

C.2 Deferred Proofs

In this section, we provide a formal setting and proof. We first formalize our setting in multiclass. Consider our task $\mathcal{T}$ contains r classes where $r \geq 2$.

Contrastive Learning. In contrastive learning, we sampled one example x from any latent class y, then apply the data augmentation module that randomly transforms such sample into another view of the original example denoted x^+. We also sample other $r-1$ examples $\{x_k^-\}_{k=1}^r$ from other latent classes $\{y_k^-\}_{k=1}^{r-1}$. We treat (x, x^+) as a positive pair and (x, x_k^-) as negative pairs. We define $\mathcal{D}_{\text{con}}(\eta)$ over sample $(x, x^+, x_1^-, \ldots, x_{r-1}^-)$ by following sampling procedure

$$(y, y_1^-, \ldots, y_{r-1}^-) \sim \eta^r \tag{C.1}$$

$$x \sim \mathcal{D}(y),\ x^+ \sim \mathcal{D}(y),\ x_k^- \sim \mathcal{D}(y_k^-), k = 1, \ldots, r-1. \tag{C.2}$$

We consider general contrastive loss $\ell_u\left(\left\{\phi(x)^\top\left(\phi(x^+) - \phi(x_k^-)\right)\right\}_{k=1}^{r-1}\right)$, where loss function ℓ_u is non-negative decreasing function. Minimizing the loss is equivalent to maximizing the similarity between positive pairs while minimizing it between negative pairs. In particular, logistic loss $\ell_u(\boldsymbol{v}) = \log\left(1 + \sum_i \exp\left(-\boldsymbol{v}_i\right)\right)$ for $\boldsymbol{v} \in \mathbb{R}^{r-1}$ recovers the one used in most empirical works: $-\log\left(\frac{\exp\left\{\phi(x)^\top\phi(x^+)\right\}}{\exp\left\{\phi(x)^\top\phi(x^+)\right\} + \sum_{i=1}^{r-1}\exp\left\{\phi(x)^\top\phi(x_i^-)\right\}}\right)$. The popula-

tion contrastive loss is defined as $\mathcal{L}_{con-pre}(\phi) := \mathbb{E}\left[\ell_u\left(\left\{\phi(x)^\top\left(\phi(x^+)-\phi(x_k^-)\right)\right\}_{k=1}^{r-1}\right)\right]$. Let $\mathcal{S}_{con-pre} := \left\{x_j, x_j^+, x_{j1}^-, \ldots, x_{j(r-1)}^-\right\}_{j=1}^N$ denote our contrastive training set with N samples, sampled from $\mathcal{D}_{\text{con}}(\eta)$, we have empirical contrastive loss $\widehat{\mathcal{L}}_{con-pre}(\phi) := \frac{1}{N}\sum_{i=1}^N\left[\ell_u\left(\left\{\phi(x)^\top\left(\phi(x^+)-\phi(x_k^-)\right)\right\}_{k=1}^{r-1}\right)\right]$.

Supervised Learning. In supervised learning we have a labeled dataset denoted as $\mathcal{S}_{con-pre} := \{x_j, y_j\}_{j=1}^N$ with N samples, by following sampling procedure:

$$y \sim \eta \tag{C.3}$$

$$x \sim \mathcal{D}(y). \tag{C.4}$$

There are in total K classes, denote $\mathcal{C}$ as the set consists of all classes. On top of the representation function ϕ, there is a linear function $f \in \mathcal{F} \subset \left\{\mathbb{R}^d \to \mathbb{R}^K\right\}$ predicting the labels, denoted as $g(x) = f \circ \phi(x)$. We consider general supervised loss on data point (x, y) is

$$\ell(g(x), y) := \ell_u\left((g(x))_y - (g(x))_{y'\neq y, y'\in\mathcal{C}}\right). \tag{C.5}$$

where loss function ℓ_u is non-negative decreasing function. In particular, logistic loss $\ell_u(\boldsymbol{v}) = \log\left(1+\sum_i \exp\left(-\boldsymbol{v}_i\right)\right)$ for $\boldsymbol{v} \in \mathbb{R}^{K-1}$ recovers the one used in most empirical works:

$$\ell(g(x), y) = \ell_u\left((g(x))_y - (g(x))_{y'\neq y, y'\in\mathcal{C}}\right) \tag{C.6}$$

$$= \log\left\{1 + \sum_{k\neq y}^{K} \exp\left(-\left[(g(x))_y - (g(x))_k\right]\right)\right\} \tag{C.7}$$

$$= -\log\left\{\frac{\exp(g(x))_y}{\sum_{k=1}^K \exp(g(x))_k}\right\}. \tag{C.8}$$

The population supervised loss is

$$\mathcal{L}_{sup-pre}(\phi) = \min_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{x,y} \left[\ell \left(f \circ \phi(x), y \right) \right]. \tag{C.9}$$

For training set $\mathcal{S}_{sup-pre} := \{x_i, y_i\}_{i=1}^N$ with N samples, the empirical supervised pretraining loss is $\widehat{\mathcal{L}}_{sup-pre}(\phi) := \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \left[\ell \left(f \circ \phi(x_i), y_i \right) \right]$.

Masked Language Modeling. Masked language modeling is a self-supervised learning method. It can be viewed as a specific form of supervised pretraining above. The pretraining data is a substantial dataset of sentences, often sourced from Wikipedia. In the pretraining phase, a random selection of words is masked within each sentence, and the training objective is to predict these masked words using the context provided by the remaining words in the sentence. This particular pretraining task can be viewed as a multi-class classification problem, where the number of classes (denoted as K) corresponds to the size of the vocabulary. Considering BERT and its variations, we have function ϕ as a text encoder. This encoder outputs a learned representation, often known as `[CLS]` token. The size of such learned representation is d, which is 768 for $\text{BERT}_{\text{BASE}}$ or 1024 for $\text{BERT}_{\text{LARGE}}$.

Supervised Tasks. Given a representation function ϕ, we apply a task-specific linear transformation W to the representation to obtain the final prediction. Consider r-way supervised task $\mathcal{T}$ consist a set of distinct classes $(y_1, \ldots, y_r) \subseteq \mathcal{C}$. We define $\mathcal{D}_{\mathcal{T}}(y)$ as the distribution of randomly drawing $y \in (y_1, \ldots, y_r)$, we denote this process as $y \sim \mathcal{T}$. Let $\mathcal{S}_{\mathcal{T}} := \{x_j, y_j\}_{j=1}^m$ denote our labeled training set with m samples, sampled i.i.d. from $y_j \sim \mathcal{T}$ and $x_j \sim \mathcal{D}(y_j)$. Define $g(\phi(\mathbf{x})) := W\phi(x) \in \mathbb{R}^r$ as prediction logits, where $W \in \mathbb{R}^{r \times d}$. The typical supervised logistic loss is $\ell(g \circ \phi(x), y) := \ell_u \left(\left\{ g(\phi(\mathbf{x}))_y - g(\phi(\mathbf{x}))_{y'} \right\}_{y' \neq y} \right)$. Similar to [19], define supervised loss w.r.t the task $\mathcal{T}$

$$\mathcal{L}_{sup}(\mathcal{T}, \phi) := \min_{W \in \mathbb{R}^{r \times d}} \mathop{\mathbb{E}}_{y \sim \mathcal{T}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}(y)} \left[\ell \left(W \cdot \phi(x), y \right) \right]. \tag{C.10}$$

Algorithm 8 Multitask Finetuning

Input: multitasks $\mathcal{T}_1, \ldots, \mathcal{T}_M$, pretrained model $\hat{\phi}$ with parameter θ, step size γ

Initialize ϕ with $\hat{\phi}$
repeat
  for all $\mathcal{T}_i$ **do**
    $\theta \leftarrow \theta - \gamma \nabla_\theta \widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi)$ { $\widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi)$ is defined in (4.2)}
  end for
until converge

Output: The final model, denoted as ϕ'

Define supervised loss with mean classifier as $\mathcal{L}^{\mu}_{sup}(\mathcal{T}, \phi) := \mathop{\mathbb{E}}_{y\sim\mathcal{T}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)} [\ell(W^{\mu} \cdot \phi(x), y)]$ where each row of W^{μ} is the mean of each class in $\mathcal{T}$, $W^{\mu}_{y_k} := \mu_{y_k} = \mathop{\mathbb{E}}_{x\sim y_k}(\phi(x)), k = 1, \ldots, r$. In the target task, suppose we have r distinct classes from $\mathcal{C}$ with equal weights. Consider $\mathcal{T}$ follows a general distribution ζ. Define expected supervised loss as $\mathcal{L}_{sup}(\phi) := \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta} [\mathcal{L}_{sup}(\mathcal{T}, \phi)]$.

Mutlitask Finetuning. Suppose we have M auxiliary tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$, each with m labeled samples $\mathcal{S}_i := \{(x^i_j, y^i_j) : j \in [m]\}$. The finetuning data are $\mathcal{S} := \cup_{i\in[M]}\mathcal{S}_i$. Given a pretrained model $\hat{\phi}$, we further finetune it using the objective:

$$\min_{\phi\in\Phi} \frac{1}{M} \sum_{i=1}^{M} \widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi), \quad \text{where } \widehat{\mathcal{L}}_{sup}(\mathcal{T}_i, \phi) := \min_{\boldsymbol{w}_i\in\mathbb{R}^d} \frac{1}{m} \sum_{j=1}^{m} \ell(\boldsymbol{w}_i^{\top} \phi(x^i_j), y^i_j). \tag{C.11}$$

This can be done via gradient descent from the initialization $\hat{\phi}$ (see Algorithm 8).

Algorithm 8 has similar pipeline as [223] where in the inner loop only a linear layer on top of the embeddings is learned. However, our algorithm is centered on multitask finetuning, where no inner loop is executed.

Finally, we formalize our assumption Assumption 4.2.1 below.

Assumption C.2.1 (Regularity Conditions)**.** The following regularity conditions hold:

(A1) Representation function ϕ satisfies $\|\phi\|_2 \le R$.

(A2) Linear operator W satisfies bounded spectral norm $\|W\|_2 \le B$.

(A3) The loss function ℓ_u are bounded by $[0, C]$ and $\ell(\cdot)$ is L-Lipschitz.

(A4) The supervised loss $\mathcal{L}_{sup}(\mathcal{T}, \phi)$ is $\tilde{L}$-Lipschitz with respect to ϕ for $\forall \mathcal{T}$.

C.2.1 Contrastive Pretraining

In this section, we will show how multitask finetuning improves the model from contrastive pretraining. We present pretraining error in binary classification and $\mathcal{D}_{\mathcal{T}}(y)$ as uniform. See the result for the general condition with multi-class in Appendix C.3.

Contrastive Pretraining and Direct Adaptation

In this section, we show the error bound of a foundation model on a target task, where the model is pretrained by contrastive loss followed directly by adaptation.

We first show how pretraining guarantees the expected supervised loss:

$$\mathcal{L}_{sup}(\phi) = \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta} \left[\mathcal{L}_{sup}(\mathcal{T}, \phi)\right]. \tag{C.12}$$

The error on the target task can be bounded by $\mathcal{L}_{sup}(\phi)$. We use ϵ^* denote $\mathcal{L}_{sup}(\phi^*_\zeta)$.

Lemma C.2.1 (Lemma 4.3 in [19]). For $\forall \phi \in \Phi$ pretrained in contrastive loss, we have $\mathcal{L}_{sup}(\phi) \leq \frac{1}{1-\tau}(\mathcal{L}_{con-pre}(\phi) - \tau)$.

We state the theorem below.

Theorem C.2.2. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{con-pre}(\hat{\phi}) \leq \epsilon_0$. Let $\tau := \Pr_{(y_1,y_2)\sim\eta^2}\{y_1 = y_2\}$. Consider pretraining set $\mathcal{S}_{con-pre} = \left\{x_j, x_j^+, x_j^-\right\}_{j=1}^N$. For any $\delta \geq 0$, if

$$N \geq \frac{1}{\epsilon_0}\left[8LR\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right].$$

Then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subset \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\frac{1}{1-\tau}(2\epsilon_0 - \tau) - \mathcal{L}_{sup}(\phi^*)\right] + \kappa. \tag{C.13}$$

The pretraining sample complexity is $O(\frac{\mathcal{R}_N(\Phi)}{\epsilon_0} + \frac{\log(1/\delta)}{\epsilon_0^2})$. The first term is the Rademacher complexity of the entire representation space Φ with sample size N. The second term relates to the generalization bound. Pretraining typically involves a vast and varied dataset, sample complexity is usually not a significant concern during this stage.

Proof of Theorem C.2.2. Recall in binary classes, $\mathcal{S}_{con-pre} = \left\{x_j, x_j^+, x_j^-\right\}_{j=1}^N$ denote our contrastive training set, sampled from $\mathcal{D}_{\text{con}}(\eta)$. Then by Lemma A.2 in [19], with **(A1)** and **(A3)**, we have for $\forall \phi \in \Phi$ with probability $1-\delta$,

$$\mathcal{L}_{con-pre}(\phi) - \widehat{\mathcal{L}}_{con-pre}(\phi) \leq \frac{4LR\mathcal{R}_N(\Phi)}{N} + C\sqrt{\frac{\log\frac{1}{\delta}}{N}}. \tag{C.14}$$

To have above $\leq \epsilon_0$, we have sample complexity

$$N \geq \frac{1}{\epsilon_0}\left[8LR\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right].$$

In pretraining, we have $\hat{\phi}$ such that

$$\hat{\mathcal{L}}_{con-pre}(\hat{\phi}) \leq \epsilon_0.$$

Then with the above sample complexity, we have pretraining $\hat{\phi}$

$$\mathcal{L}_{con-pre}(\hat{\phi}) \leq 2\epsilon_0.$$

Recall ν-diversity and κ-consistency, for target task $\mathcal{T}_0$, we have that for $\hat{\phi}$ and ϕ^*,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) = \mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \tag{C.15}$$

$$\leq d_{\mathcal{C}_0}(\hat{\phi}, \phi_\zeta^*) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \tag{C.16}$$

$$\leq \bar{d}_\zeta(\hat{\phi}, \phi_\zeta^*)/\nu + \kappa \tag{C.17}$$

$$\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\hat{\phi}) - \mathcal{L}_{sup}(\phi_\zeta^*)\right] + \kappa \tag{C.18}$$

$$= \frac{1}{\nu}\left[\frac{1}{1-\tau}(\mathcal{L}_{con-pre}(\hat{\phi}) - \tau) - \epsilon^*\right] + \kappa \tag{C.19}$$

$$\leq \frac{1}{\nu}\left[\frac{1}{1-\tau}(2\epsilon_0 - \tau) - \epsilon^*\right] + \kappa, \tag{C.20}$$

where the second to last inequality comes from Lemma C.2.1. □

Contrastive Pretraining and Multitask Finetuning

In this section, we show the error bound of a foundation model on a target task can be further reduced by multitask finetuning. We achieve this by showing that expected supervised loss $\mathcal{L}_{sup}(\phi)$ can be further reduced after multitask finetuning. The error on the target task can be bounded by $\mathcal{L}_{sup}(\phi)$. We use ϵ^* denote $\mathcal{L}_{sup}(\phi_\zeta^*)$.

Following the intuition in [99], we first re-state the definition of representation space.

Definition 3. The subset of representation space is

$$\Phi(\tilde{\epsilon}) = \left\{\phi \in \Phi : \hat{\mathcal{L}}_{pre}(\phi) \leq \tilde{\epsilon}\right\}.$$

Recall $\mathcal{S} = \{(x_j^i, y_j^i) : i \in [M], j \in [m]\}$ as finetuning dataset.

We define two function classes and associated Rademacher complexity.

Definition 4. Consider function class

$$\mathcal{G}_\ell(\tilde{\epsilon}) = \left\{g_{W,\phi}(x,y) : g_{W,\phi}(x,y) = \ell(W\phi(x_j^i), y_j^i), \phi \in \Phi(\tilde{\epsilon}), \|W\|_2 \leq B\right\}.$$

We define Rademacher complexity as

$$\mathcal{R}_n(\mathcal{G}_\ell(\tilde{\epsilon})) = \mathop{\mathbb{E}}_{\{\sigma_i\}_{j=1}^n, \{x_j, y_j\}_{j=1}^n} \left[\sup_{\ell \in \mathcal{G}_\ell(\tilde{\epsilon})} \sum_{j=1}^{n} \sigma_j \ell(W \cdot \phi(x_j), y_j) \right].$$

Definition 5. Consider function class

$$\mathcal{G}(\tilde{\epsilon}) = \{ g_\phi : g_\phi(\mathcal{T}) = \mathcal{L}_{sup}(T, \phi), \phi \in \Phi(\tilde{\epsilon}) \}.$$

We define Rademacher complexity as

$$\mathcal{R}_M(\mathcal{G}(\tilde{\epsilon})) = \mathop{\mathbb{E}}_{\{\sigma_i\}_{i=1}^M, \{\mathcal{T}_i\}_{i=1}^M} \left[\sup_{\phi \in \Phi(\tilde{\epsilon})} \sum_{i=1}^{M} \sigma_i \mathcal{L}_{sup}(\mathcal{T}_i, \phi) \right].$$

The key idea is multitask finetuning further reduce the expected supervised loss of a pretrained foundation model ϕ:

$$\mathcal{L}_{sup}(\phi) = \mathop{\mathbb{E}}_{\mathcal{T} \sim \zeta} [\mathcal{L}_{sup}(\mathcal{T}, \phi)]. \tag{C.21}$$

We first introduce some key lemmas. These lemmas apply to general r classes in a task $\mathcal{T}$.

Lemma C.2.3 (Bounded Rademacher complexity). By **(A2)** and **(A3)**, we have for $\forall n$

$$\mathcal{R}_n(\mathcal{G}_\ell(\tilde{\epsilon})) \leq 4\sqrt{r-1} LB \mathcal{R}_n(\Phi(\tilde{\epsilon})).$$

Proof of Lemma C.2.3. We first prove $\ell(g(\phi(x)), y)$ is $\sqrt{2(r-1)}LB$-Lipschitz with respect to ϕ for all $\forall y \in \mathcal{C}$. Consider

$$f_y(g(\phi(\mathbf{x}))) = \left\{ g(\phi(\mathbf{x}))_y - g(\phi(\mathbf{x}))_{y'} \right\}_{y' \neq y},$$

where $f_y : \mathbb{R}^r \to \mathbb{R}^{r-1}$. Note that

$$\begin{aligned}\ell(g \circ \phi(x), y) &= \ell\left(\left\{g(\phi(\mathbf{x}))_y - g(\phi(\mathbf{x}))_{y'}\right\}_{y' \neq y}\right) \\ &= \ell(f_y(g(\phi(\mathbf{x})))).\end{aligned}$$

By **(A3)**, we have ℓ is L-Lipschitz. We then prove f_y is $\sqrt{2(r-1)}$-Lipschitz. Without loss generality, consider $y = r$. We have $f_y(y) = [y_r - y_i]_{i=1}^{r-1}$. We have $\frac{\partial f_j}{y_i} = -\mathbb{1}\{j = i\}, i = 1, \ldots, r-1$, $\frac{\partial f_j}{y_r} = 1$. The Jacobian J satisfies $\|J\|_2 \leq \|J\|_F = \sqrt{2(r-1)}$.

Since g is B-Lipschitz by **(A2)**:$\|W\|_2 \leq B$. Then $\ell(g(\phi(x)), y)$ is $\sqrt{2(r-1)}LB$-Lipschitz with respect to ϕ for all $\forall y \in \mathcal{C}$. The conclusion follows Corollary 4 in [177]. □

Lemma C.2.4 (Bounded $\tilde{\epsilon}$). After finite steps in Multitask finetuning in Algorithm 8, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{1}{1-\tau}(2\epsilon_0 - \tau)$ and obtain ϕ'. Then there exists a bounded $\tilde{\epsilon}$ such that $\phi' \in \Phi(\tilde{\epsilon})$.

Proof of Lemma C.2.4. Given finite number of steps and finite step size γ in Algorithm 8, we have bounded $\|\phi' - \hat{\phi}\|$. Then with **(A2)** and **(A3)**, using Lemma C.2.3 we have $\ell(g(\phi(x)), y)$ is $\sqrt{2(r-1)}LB$-Lipschitz with respect to ϕ for all $\forall y$, using theorem A.2 in [19] we have l_u is LC-Lipschitz with respect to ϕ, we have $\widehat{\mathcal{L}}_{pre}(\phi)$ is M-Lipschitz with respect to ϕ with bounded M. We have $\exists\ \epsilon$ such that $\widehat{\mathcal{L}}_{pre}(\phi') - \widehat{\mathcal{L}}_{pre}(\hat{\phi}) \leq \epsilon\|\phi' - \hat{\phi}\|$. We have $\widehat{\mathcal{L}}_{pre}(\phi') \leq \epsilon_0 + \epsilon\|\phi' - \hat{\phi}\|$. Take $\tilde{\epsilon} = \epsilon_0 + \epsilon\|\phi' - \hat{\phi}\|$ yields the result. □

Lemma C.2.5. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ. Suppose for some small constant $\alpha \in (0, 1)$ and $\tilde{\epsilon}$, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{1}{1-\tau}(2\epsilon_0 - \tau)$ and obtain ϕ'. For any $\delta > 0$, if

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[8\sqrt{r-1}LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then expected supervised loss $\mathcal{L}_{sup}(\phi') \leq \alpha\frac{1}{1-\tau}(2\epsilon_0 - \tau)$, with probability $1 - \delta$.

Proof of Lemma C.2.5. Recall $\mathcal{S} := \{(x_j^i, y_j^i) : i \in [M], j \in [m]\}$ as finetuning dataset. Con-

sider in Equation (4.2) we have $\widehat{\mathbf{W}} := (\widehat{W}_1, \ldots, \widehat{W}_M)$ and ϕ' such that $\frac{1}{M}\sum_{i=1}^{M}\frac{1}{m}\sum_{j=1}^{m}\ell(\widehat{W}_i \cdot \phi'(x_j^i), y_j^i) \leq \epsilon_1 < \frac{\alpha}{3}\epsilon_0$.

We tried to bound

$$\mathcal{L}_{sup}(\phi') - \frac{1}{m}\sum_{j=1}^{m}\ell(\widehat{W}_i \cdot \phi'(x_j^i), y_j^i).$$

Recall that

$$\mathcal{L}_{sup}(\mathcal{T}_i, \phi) = \min_{W\in\mathbb{R}^{r\times d}} \mathop{\mathbb{E}}_{y\sim\mathcal{T}_i} \mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)} \left[\ell\left(W \cdot \phi(x), y\right)\right].$$

For $\forall\phi \in \Phi(\tilde{\epsilon})$

$$\mathcal{L}_{sup}(\phi) = \mathbb{E}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T}, \phi)\right] = \mathbb{E}_{\mathcal{T}\sim\zeta}\left[\min_{W\in\mathbb{R}^{r\times d}} \mathop{\mathbb{E}}_{y\sim\mathcal{T}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)} \left[\ell\left(W \cdot \phi(\mathbf{x}), y\right)\right]\right].$$

We have for $\forall\phi \in \Phi(\tilde{\epsilon})$, by uniform convergence (see [184] Theorem 3.3), we have with probability $1 - \delta/2$

$$\mathbb{E}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T}, \phi)\right] - \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{sup}(\mathcal{T}_i, \phi) \leq \frac{2\mathcal{R}_M(\mathcal{G}(\tilde{\epsilon}))}{M} + \sqrt{\frac{\log(2/\delta)}{M}} \tag{C.22}$$

$$\leq \frac{2\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon}))}{M} + \sqrt{\frac{\log(2/\delta)}{M}}, \tag{C.23}$$

where the last inequality comes from **(A4)** and Corollary 4 in [177]. To have above $\leq \epsilon_1/2$, we have sample complexity

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right].$$

Then we consider generalization bound for $\forall\phi$ and $\mathbf{W} := (W_1, \ldots, W_M)$

$$\mathcal{L}_{sup}(\phi, \mathbf{W}) = \frac{1}{M}\sum_{i=1}^{M} \mathop{\mathbb{E}}_{y^i\sim\mathcal{T}_i} \mathop{\mathbb{E}}_{x^i\sim\mathcal{D}(y^i)} \ell\left(W_i \cdot \phi(x^i), y^i\right) \tag{C.24}$$

$$\hat{\mathcal{L}}_{sup}(\phi, \mathbf{W}) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{m}\sum_{j=1}^{m}\ell(W_i \cdot \phi(x_j^i), y_j^i), \tag{C.25}$$

where $\mathbf{W} = (W_1, \ldots, W_M)$.

By uniform convergence (see [184] Theorem 3.3), we have with probability $1-\delta/2$,

$$\mathcal{L}_{sup}(\phi,\mathbf{W})-\hat{\mathcal{L}}_{sup}(\phi,\mathbf{W}) \leq \frac{2\mathcal{R}_{Mm}(\mathcal{G}_\ell)}{Mm}+\sqrt{\frac{\log(2/\delta)}{Mm}} \leq \frac{8\sqrt{r-1}LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon}))}{Mm}+C\sqrt{\frac{\log(2/\delta)}{Mm}},$$

where the last inequality comes from Lemma C.2.3. To have above $\leq \epsilon_1/2$, we have sample complexity

$$Mm \geq \frac{1}{\epsilon_1}\left[8\sqrt{r-1}LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon}))+\frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

satisfying $\forall \phi \in \Phi(\tilde{\epsilon})$

$$\begin{aligned}
\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{sup}(\mathcal{T}_i,\phi) &= \frac{1}{M}\sum_{i=1}^{M}\min_{W\in\mathbb{R}^{r\times d}}\mathop{\mathbb{E}}_{y\sim\mathcal{T}_i}\mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)}\left[\ell\left(W\cdot\phi(x),y\right)\right] \\
&\leq \frac{1}{M}\sum_{i=1}^{M}\mathop{\mathbb{E}}_{y\sim\mathcal{T}_i}\mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)}\left[\ell\left(\widehat{W}_i\cdot\phi(x),y\right)\right] \\
&= \mathcal{L}_{sup}(\phi,\widehat{\mathbf{W}}) \\
&\leq \hat{\mathcal{L}}_{sup}(\phi,\widehat{\mathbf{W}})+\epsilon_1/2.
\end{aligned}$$

Then combine above with Equation (C.22)

$$\begin{aligned}
\mathcal{L}_{sup}(\phi) &= \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T},\phi)\right] \\
&\leq \hat{\mathcal{L}}_{sup}(\phi,\widehat{\mathbf{W}})+\epsilon_1.
\end{aligned}$$

We have

$$\begin{aligned}
&\mathcal{L}_{sup}(\phi')-\frac{1}{m}\sum_{j=1}^{m}\ell(\widehat{W}_i\cdot\phi'(x_j^i),y_j^i) \leq \epsilon_1 \\
&\mathcal{L}_{sup}(\phi') \leq 2\epsilon_1 \leq \alpha\frac{1}{1-\tau}(2\epsilon_0-\tau).
\end{aligned}$$

The boundedness of $\tilde{\epsilon}$ follows Lemma C.2.4. □

We state the theorem below.

Theorem C.2.6. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency

with respect to ϕ^*, ϕ^*_ζ. Suppose for some small constant $\alpha \in (0,1)$, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{1}{1-\tau}(2\epsilon_0 - \tau)$ and obtain ϕ'. For any $\delta > 0$, if

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[8LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\alpha\frac{1}{1-\tau}(2\epsilon_0 - \tau) - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa. \tag{C.26}$$

Proof of Theorem C.2.6. Recall with ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ, for target task $\mathcal{T}_0$, we have that for ϕ' and ϕ^*,

$$\begin{aligned}
\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) &= \mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \\
&\leq \frac{1}{\nu}\bar{d}_\zeta(\phi', \phi^*_\zeta) + \kappa \\
&\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\phi') - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa \\
&\leq \frac{1}{\nu}\left[\alpha\frac{1}{1-\tau}(2\epsilon_0 - \tau) - \epsilon^*\right] + \kappa,
\end{aligned}$$

where the last inequality comes from Lemma C.2.5, where taking $r = 2$. □

C.2.2 Supervised Pretraining

In this section, we will show how multitask finetuning improves the model from supervised pretraining. We present pretraining error in binary classification and $\mathcal{D}_\mathcal{T}(y)$ as uniform. See the result for the general condition with multi-class in Appendix C.3.

Supervised Pretraining and Direct Adaptation

In this section, we show the error bound of a foundation model on a target task, where the model is pretrained by supervised loss followed directly by adaptation. For general $y \sim \eta$. Let $p_i := \Pr_{y\sim\eta}\{y = y_i\}$, where $\sum_{i=1}^K p_i = 1$.

Lemma C.2.7. Suppose $y \sim \eta$ and $l \leq \Pr_{y\sim\eta}\{y = y_i\} \leq u$. Consider a task $\mathcal{T}$ containing r classes, which is a subset of the total class set $\mathcal{C}$. We have $\forall \phi \in \Phi$,

$$\mathcal{L}_{sup}(\phi) \leq \left(\frac{u}{l}\right)^r \mathcal{L}_{sup-pre}(\phi),$$

where

$$\mathcal{L}_{sup-pre}(\phi) = \min_{f\in\mathcal{F}} \mathop{\mathbb{E}}_{x,y} \left[\ell\left(f \circ \phi(x), y\right)\right]. \tag{C.27}$$

Proof of Appendix C.2.2. We first prove $r = 3$, where $\mathcal{T} = \{y_1, y_2, y_3\}$. Then in supervised pretraining, we have:

$$\mathcal{L}_{sup-pre}(\phi) = \min_{f\in\mathcal{F}} \mathop{\mathbb{E}}_{y\sim\mathcal{T}} \quad \mathop{\mathbb{E}}_{x\sim y} \left[\ell\left(f \circ \phi(x), y\right)\right]. \tag{C.28}$$

Let $f = (f_1, f_2, f_3)^\top$ be the best linear classifier on top of ϕ, the prediction logits are $g(x) = f \circ \phi(x) = (g_1(x), g_2(x), g_3(x))^\top$. Then we have:

$$\mathop{\mathbb{E}}_{x\sim y_1} \left[\ell\left(g \circ \phi(x), y\right)\right] = -\log \frac{\exp(g_1(x))}{\sum_{k=1}^3 \exp(g_k(x))}.$$

We let $y_k(x) = \exp(g_k(x)), k = 1, 2, 3$. Then

$$\begin{aligned}
&\mathcal{L}_{sup-pre}(\phi) \\
=&-\left[p_1 \mathop{\mathbb{E}}_{x\sim y_1}\left(\log \frac{y_1(x)}{\sum_{k=1}^3 y_k(x)}\right) + p_2 \mathop{\mathbb{E}}_{x\sim y_2}\left(\log \frac{y_2(x)}{\sum_{k=1}^3 y_k(x)}\right) + p_3 \mathop{\mathbb{E}}_{x\sim y_3}\left(\log \frac{y_3(x)}{\sum_{k=1}^3 y_k(x)}\right)\right] \\
=&p_1 \mathop{\mathbb{E}}_{x\sim y_1}\left(\log \frac{\sum_{k=1}^3 y_k(x)}{y_1(x)}\right) + p_2 \mathop{\mathbb{E}}_{x\sim y_2}\left(\log \frac{\sum_{k=1}^3 y_k(x)}{y_2(x)}\right) + p_3 \mathop{\mathbb{E}}_{x\sim y_3}\left(\log \frac{\sum_{k=1}^3 y_k(x)}{y_3(x)}\right).
\end{aligned}$$

Recall

$$\mathcal{L}_{sup}(\mathcal{T}, \phi) := \min_{\boldsymbol{w}} \mathop{\mathbb{E}}_{y\sim\mathcal{T}} \quad \mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)} \left[\ell\left(\boldsymbol{w}^\top \phi(x), y\right)\right]. \tag{C.29}$$

Consider

$$\mathcal{L}^*_{sup}(\mathcal{T}, \phi) := \mathop{\mathbb{E}}_{y\sim\mathcal{T}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}(y)} \left[\ell\left(\boldsymbol{w}^\top \phi(x), y\right)\right], \tag{C.30}$$

where $\boldsymbol{w}$ is the corresponding sub-vector of f according to task (for e.g., $\boldsymbol{w} = (f_1, f_2)^\top$ if $\mathcal{T} = \{y_1, y_2\}$). Then we have

$$\begin{aligned}
\mathcal{L}^*_{sup}(\mathcal{T}, \phi) =& -\frac{p_1p_2}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_1}\left(\log\frac{y_1(x)}{y_1(x)+y_2(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_2}\left(\log\frac{y_2(x)}{y_1(x)+y_2(x)}\right)\right] \\
&-\frac{p_1p_3}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_1}\left(\log\frac{y_1(x)}{y_1(x)+y_3(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_3}\left(\log\frac{y_3(x)}{y_1(x)+y_3(x)}\right)\right] \\
&-\frac{p_2p_3}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_2}\left(\log\frac{y_2(x)}{y_2(x)+y_3(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_3}\left(\log\frac{y_3(x)}{y_2(x)+y_3(x)}\right)\right] \\
=&\frac{p_1p_2}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_1}\left(\log\frac{y_1(x)+y_2(x)}{y_1(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_2}\left(\log\frac{y_1(x)+y_2(x)}{y_2(x)}\right)\right] \\
&\frac{p_1p_3}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_1}\left(\log\frac{y_1(x)+y_3(x)}{y_1(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_3}\left(\log\frac{y_1(x)+y_3(x)}{y_3(x)}\right)\right] \\
&\frac{p_2p_3}{p_1p_2 + p_1p_3 + p_2p_3} \cdot \frac{1}{2}\left[\mathop{\mathbb{E}}_{x\sim y_2}\left(\log\frac{y_2(x)+y_3(x)}{y_2(x)}\right) + \mathop{\mathbb{E}}_{x\sim y_3}\left(\log\frac{y_2(x)+y_3(x)}{y_3(x)}\right)\right].
\end{aligned}$$

By observing the terms with $y_1(x)$ as denominator (similar as $y_2(x), y_3(x)$), we want to prove:

$$p_1\left(\frac{u}{l}\right)^2 \geq \frac{1}{2}\left(\frac{p_1p_2 + p_1p_3}{p_1p_2 + p_1p_3 + p_2p_3}\right).$$

This obtained by $\left(\frac{u}{l}\right)^2 \geq \frac{1}{3}\frac{u}{l^2}$.

We have

$$\mathcal{L}^*_{sup}(\mathcal{T}, \phi) \leq \left(\frac{u}{l}\right)^2 \mathcal{L}_{sup-pre}(\phi).$$

For the general K-class setting, we follow similar steps, we have

$$\mathcal{L}_{sup-pre}(\phi) = -\left[\sum_{i=1}^{r} p_i \mathop{\mathbb{E}}_{x\sim y_i}\left(\log\frac{y_i(x)}{\sum_{k=1}^{K} y_k(x)}\right)\right].$$

We denote J as all possible r product of $p_i \in \{p_1, \ldots, p_K\}$, $J = \{p_1 \cdots p_r, \ldots\}$. Similarly,

we have

$$\mathcal{L}^*_{sup}(\mathcal{T},\phi) = -\frac{1}{r}\left\{\sum_{\mathcal{T}\subsetneq\mathcal{C}}\left[\frac{\prod_{i\in\mathcal{T}} p_i}{J}\sum_{i\in\mathcal{T}}\mathop{\mathbb{E}}_{x\sim y_i}\left(\log\frac{y_i(x)}{\sum_{j\in\mathcal{T}} y_j(x)}\right)\right]\right\},$$

where $\mathcal{T}$ are all tasks with r classes. By observing, inside the summation there are in total $\binom{K-1}{r-1}$ terms with $y_1(x)$ as the numerator, where corresponding probability is

$$\frac{p_1\prod_{i\in\mathcal{T},i\neq 1} p_i}{J},$$

where each term can be upper bounded by $-\left(\frac{u}{l}\right)^r p_1 \mathop{\mathbb{E}}_{x\sim y_i}\left(\log\frac{y_i(x)}{\sum_{k=1}^K y_k(x)}\right)$ (similar as $y_j(x), j\in\mathcal{T}$). □

We state the theorem below.

Theorem C.2.8. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{sup-pre}(\hat{\phi}) \le \epsilon_0$. Let $p_i := \Pr_{y\sim\eta}\{y = y_i\}$, where $\sum_{i=1}^K p_i = 1$. Let $\rho := \frac{\max_i p_i}{\min_j p_j}$. Consider pretraining set $\mathcal{S}_{sup-pre} := \{x_i, y_i\}_{i=1}^N$, for any $\delta \ge 0$, if

$$N \ge \frac{1}{\epsilon_0}\left[8LR\sqrt{K}\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right].$$

Then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subset \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0,\hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0,\phi^*) \le \frac{1}{\nu}\left[2\rho^2\epsilon_0 - \epsilon^*\right] + \kappa. \tag{C.31}$$

Proof of Theorem C.2.8. The proof follows similar steps in Theorem C.2.2. For supervised pretraining, the sample complexity is similar to Theorem C.2.2, note that there is an extra $\sqrt{K}$ term. We show how we have this term below:

Consider function class

$$\mathcal{G}_\ell = \left\{g_{W,\phi}(x,y) : g_{W,\phi}(x,y) = \ell(W^\top\phi(x^i_j), y^i_j), \phi\in\Phi, \|W\|_2 \le B\right\}.$$

The Rademacher complexity is

$$\mathcal{R}_n(\mathcal{G}_\ell) = \mathop{\mathbb{E}}_{\{\sigma_i\}_{j=1}^n, \{x_j, y_j\}_{j=1}^n} \left[\sup_{\ell \in \mathcal{G}_\ell} \sum_{j=1}^n \sigma_j \ell(W \cdot \phi(x_j), y_j) \right].$$

Then from Lemma C.2.3, the pretraining is a large task with classification among K classes.

$$\mathcal{R}_n(\mathcal{G}_\ell) \leq 4\sqrt{K} LB \mathcal{R}_n(\Phi).$$

Then by Theorem 3.3 in [184], with **(A1)** and **(A3)**, we have for $\forall \phi \in \Phi$ with probability $1 - \delta$,

$$\mathcal{L}_{sup-pre}(\phi) - \widehat{\mathcal{L}}_{sup-pre}(\phi) \leq \frac{4LR\sqrt{K}\mathcal{R}_N(\Phi)}{N} + C\sqrt{\frac{\log \frac{1}{\delta}}{N}}. \tag{C.32}$$

To have above $\leq \epsilon_0$, we have sample complexity

$$N \geq \frac{1}{\epsilon_0} \left[8LR\sqrt{K}\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0} \log(\frac{2}{\delta}) \right].$$

With the above sample complexity of $\mathcal{S}_{sup-pre} = \{x_i, y_i\}_{i=1}^N$, we have pretraining $\hat{\phi}$

$$\mathcal{L}_{sup-pre}(\hat{\phi}) \leq 2\epsilon_0.$$

Recall ν-diversity and κ-consistency, with respect to ϕ^*, ϕ_ζ^*, for target task $\mathcal{T}_0$, we have that for $\hat{\phi}$ and ϕ^*,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq d_{\mathcal{C}_0}(\hat{\phi}, \phi_\zeta^*) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \tag{C.33}$$

$$\leq \bar{d}_\zeta(\hat{\phi}, \phi_\zeta^*)/\nu + \kappa \tag{C.34}$$

$$\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\hat{\phi}) - \mathcal{L}_{sup}(\phi_\zeta^*)\right] + \kappa \tag{C.35}$$

$$\leq \frac{1}{\nu}\left[\rho^2 \mathcal{L}_{sup-pre}(\hat{\phi}) - \epsilon^*\right] + \kappa \tag{C.36}$$

$$\leq \frac{1}{\nu}\left[2\rho^2\epsilon_0 - \epsilon^*\right] + \kappa \tag{C.37}$$

where the second to last inequality comes from Lemma C.2.7.

□

Supervised Pretraining and Multitask Finetuning

In this section, we show the error bound of a supervised pretrained foundation model on a target task can be further reduced by multitask finetuning. We follow similar steps in Appendix C.2.1. Recall Definition 3, similar to Lemma C.2.5, we introduce the following lemma under supervised pretraining loss.

Lemma C.2.9. Assume Assumption 4.2.1 and that Φ has (ν, ϵ)-diversity for ζ and $\mathcal{C}_0$. Suppose for some small constant $\alpha \in (0, 1)$, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3} 2\rho^2\epsilon_0$ and obtain ϕ'. For any $\delta > 0$, if

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[16LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then expected supervised loss $\mathcal{L}_{sup}(\phi') \leq 2\alpha\rho^2\epsilon_0$, with probability $1 - \delta$.

Proof of Lemma C.2.9. The steps follow similar steps in Lemma C.2.5. □

We state the main theorem below.

Theorem C.2.10. Assume Assumption 4.2.1 and that Φ has (ν, ϵ)-diversity for ζ and $\mathcal{C}_0$. Suppose for some small constant $\alpha \in (0, 1)$, we solve Equation (4.2) with empirical loss

lower than $\epsilon_1 = \frac{\alpha}{3} 2\rho^2 \epsilon_0$ and obtain ϕ'. For any $\delta > 0$, if

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[16LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left(2\alpha\rho^2\epsilon_0 - \mathcal{L}_{sup}(\phi^*)\right) + \epsilon. \tag{C.38}$$

Proof of Theorem C.2.10. Recall ν-diversity and κ-consistency, with respect to ϕ^*, ϕ^*_ζ, for target task $\mathcal{T}_0$, we have that for $\hat{\phi}$ and ϕ^*,

$$\begin{aligned}
\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) &\leq d_{\mathcal{C}_0}(\phi', \phi^*_\zeta) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) && \text{(C.39)}\\
&\leq \bar{d}_\zeta(\phi', \phi^*_\zeta)/\nu + \kappa && \text{(C.40)}\\
&\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\phi') - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa && \text{(C.41)}\\
&\leq \frac{1}{\nu}\left(2\alpha\rho^2\epsilon_0 - \epsilon^*\right) + \kappa, && \text{(C.42)}
\end{aligned}$$

where the last inequality comes from Lemma C.2.9.

□

C.2.3 Masked Language Pretraining

The theoretical guarantee in masked language pretraining follows the same error bound in supervised pretraining, with K representing the size of the vocabulary.

C.2.4 Unified Main Theory

We now prove the main theory below. We first re-state the theorem.

Theorem 4.2.1. (No Multitask Finetuning) Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^* and ϕ^*_ζ. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{pre}(\hat{\phi}) \leq \epsilon_0$.

Let $\tau := \Pr_{(y_1,y_2)\sim\eta^2}\{y_1 = y_2\}$. Then for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\frac{2\epsilon_0}{1-\tau} - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa. \tag{4.3}$$

Proof of Theorem 4.2.1. The result is a direct combination of Theorem C.2.2 and Theorem C.2.8. □

Theorem 4.2.2. (With Multitask Finetuning) Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^* and ϕ^*_ζ. Suppose for some constant $\alpha \in (0,1)$, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{2\epsilon_0}{1-\tau}$ and obtain ϕ'. For any $\delta > 0$, if for $\tilde{\epsilon} = \widehat{\mathcal{L}}_{pre}(\phi')$,

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[16LB\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\alpha\frac{2\epsilon_0}{1-\tau} - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa. \tag{4.4}$$

Proof of Theorem 4.2.2. Follow the similar steps in proof of Lemma C.2.5, we have

$$\mathcal{L}_{sup}(\phi') \leq 2\epsilon_1 \leq \alpha\frac{2\rho^2}{1-\tau}\epsilon_0.$$

Recall ν-diversity and κ-consistency, with respect to ϕ^*, ϕ^*_ζ, for target task $\mathcal{T}_0$, we have that for ϕ' and ϕ^*,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq d_{\mathcal{C}_0}(\phi', \phi_\zeta^*) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \tag{C.43}$$

$$\leq \bar{d}_\zeta(\phi', \phi_\zeta^*)/\nu + \kappa \tag{C.44}$$

$$\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\phi') - \mathcal{L}_{sup}(\phi_\zeta^*)\right] + \kappa \tag{C.45}$$

$$\leq \frac{1}{\nu}\left[\alpha\frac{2\rho^2}{1-\tau}\epsilon_0 - \epsilon^*\right] + \kappa. \tag{C.46}$$

□

The sample complexity of finetuning depends on $\tilde{\epsilon} = \widehat{\mathcal{L}}_{pre}(\phi')$. Below we show that $\tilde{\epsilon}$ can be upper bounded in finite step finetuning.

Lemma C.2.11 (Bounded $\tilde{\epsilon}$). After finite steps in Multitask finetuning in Algorithm 8, we solve Equation (4.2) with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3}\frac{1}{1-\tau}(2\epsilon_0 - \tau)$ and obtain ϕ'. Then there exists a bounded $\tilde{\epsilon}$ such that $\phi' \in \Phi(\tilde{\epsilon})$.

Proof of Lemma C.2.11. Given finite number of steps and finite step size γ in Algorithm 8, we have bounded $\|\phi' - \hat{\phi}\|$. Then with **(A2)** and **(A3)**, using Lemma C.2.3 and lemma A.3 in [19], we have $\widehat{\mathcal{L}}_{pre}(\phi)$ is M-Lipschitz with respect to ϕ with bounded M. We have $\exists\ \epsilon$ such that $\widehat{\mathcal{L}}_{pre}(\phi') - \widehat{\mathcal{L}}_{pre}(\hat{\phi}) \leq \epsilon\|\phi' - \hat{\phi}\|$. We have $\widehat{\mathcal{L}}_{pre}(\phi') \leq \epsilon_0 + \epsilon\|\phi' - \hat{\phi}\|$. Take $\tilde{\epsilon} = \epsilon_0 + \epsilon\|\phi' - \hat{\phi}\|$ yields the result. □

C.2.5 Bounded Task Loss by Task Diversity

By the previous lemma and claim, we have the below corollary.

Corollary C.2.12. Suppose we have ϕ in pretraining: for $\forall\phi \in \Phi$, $\mathcal{L}_{sup}(\phi) \leq \frac{1}{1-\tau}\left(\frac{u}{l}\right)^r \mathcal{L}_{pre}(\phi)$, where $\mathcal{L}_{pre}(\phi)$ is $\mathcal{L}_{con-pre}(\phi)$ if contrastive learning and $\mathcal{L}_{sup-pre}(\phi)$ if supervised learning.

Consider $\rho = \frac{u}{l}$ and Corollary C.2.12,

Recall ν-diversity and κ-consistency, with respect to ϕ^*, ϕ_ζ^*, for target task $\mathcal{T}_0$, we have that for $\hat{\phi}$ and ϕ^*,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq d_{\mathcal{C}_0}(\hat{\phi}, \phi_\zeta^*) + \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \tag{C.47}$$

$$\leq \bar{d}_\zeta(\hat{\phi}, \phi_\zeta^*)/\nu + \kappa \tag{C.48}$$

$$\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\hat{\phi}) - \mathcal{L}_{sup}(\phi_\zeta^*)\right] + \kappa \tag{C.49}$$

$$\leq \frac{1}{\nu}\left[\frac{\rho^r}{1-\tau}\mathcal{L}_{pre}(\hat{\phi}) - \mathcal{L}_{sup}(\phi^*)\right] + \kappa. \tag{C.50}$$

C.3 Multi-class Classification

In this section, we provide a general result for multi-classes.

C.3.1 Contrastive Pretraining

Lemma C.3.1 (Theorem 6.1 in [19]). For multi-classes, we have

$$\mathcal{L}_{sup}(\phi) \leq \mathcal{L}_{sup}^{\mu}(\phi) \leq \frac{1}{1-\tau_r}\mathcal{L}_{con-pre}(\phi), \tag{C.51}$$

where $\tau_r = \mathop{\mathbb{E}}_{(y, y_1^-, \ldots, y_{r-1}^-) \sim \eta^r} \mathbb{1}\{y \text{ does not appear in } (y_1^-, \ldots, y_{r-1}^-)\}$.

Proof of Lemma C.3.1. The proof of Lemma C.3.1 follows the first two steps in the proof of Theorem B.1 of [19]. we denote distribution of $y \sim \mathcal{T}$ as $\mathcal{D}_{\mathcal{T}}(y)$ and it's uniform distribution. □

We first provide contrastive pretraining error similar to Theorem C.2.2 in a multiclass setting.

Theorem C.3.2. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ_ζ^*. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{con-pre}(\hat{\phi}) \leq \epsilon_0$. Consider a pretraining set $\mathcal{S}_{un} = \left\{x_j, x_j^+, x_{j1}^-, \ldots, x_{j(r-1)}\right\}_{j=1}^N$. For target task $\mathcal{T}_0$, with sample complexity

$$N \geq \frac{1}{\epsilon_0}\left[8LR\sqrt{r-1}\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right],$$

it's sufficient to learn an $\hat{\phi}$ with classification error $\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\frac{2}{1-\tau_r}\epsilon_0 - \epsilon^*\right] + \epsilon$, with probability $1-\delta$.

Proof of Theorem C.3.2. Following similar step of proof of Theorem C.2.2, we have with

$$N \geq \frac{1}{\epsilon_0}\left[8LR\sqrt{r-1}\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right].$$

Then pretraining $\hat{\phi}$

$$\mathcal{L}_{con-pre}(\hat{\phi}) \leq 2\epsilon_0.$$

Recall ν-diversity and κ-consistency, for target task $\mathcal{T}_0$, we have that for $\hat{\phi}$ and ϕ^*,

$$\begin{aligned}
\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) &\leq \bar{d}_\zeta(\hat{\phi}, \phi^*_\zeta)/\nu + \kappa && \text{(C.52)}\\
&\leq \frac{1}{\nu}\left[\mathcal{L}_{sup}(\hat{\phi}) - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa && \text{(C.53)}\\
& && \text{(C.54)}
\end{aligned}$$

Consider Lemma C.3.1, we have:

$$\begin{aligned}
\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) &\leq \frac{1}{\nu}\left[\frac{1}{1-\tau_r}\mathcal{L}_{con-pre}(\hat{\phi}) - \epsilon^*\right] + \kappa && \text{(C.55)}\\
&= \frac{1}{\nu}\left(\frac{2\epsilon_0}{1-\tau_r} - \epsilon^*\right) + \kappa. && \text{(C.56)}
\end{aligned}$$

□

Below, we provide our main result similar to Theorem C.2.6 for multi-classes setting.

Theorem C.3.3. For target evaluation task $\mathcal{T}_0$, consider the error bound in pretraining is $\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[\frac{2\epsilon_0}{1-\tau_r} - \epsilon^*\right] + \kappa$. Consider α as any small constant, for any $\epsilon_1 < \frac{\alpha}{3}\frac{2\epsilon_0}{1-\tau_r}$, consider a multitask finetuning set $\mathcal{S} = \{(x_j^i, y_j^i) : i \in [M], j \in [m]\}$, with M

number of tasks, and m number of samples in each task. Then, with sample complexity

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right]$$
$$Mm \geq \frac{1}{\epsilon_1}\left[8LB\sqrt{r-1}\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right].$$

Solving Equation (4.2) with empirical risk lower than ϵ_1 is sufficient to learn an ϕ' with classification error $\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}(\alpha\frac{2\epsilon_0}{1-\tau_r} - \epsilon^*) + \kappa$, with probability $1-\delta$.

Proof of Theorem C.3.3. Recalling Lemma C.2.3 and Lemma C.2.5, the proof follows the same steps in the proof of Theorem C.2.6, with different r.

□

C.3.2 Supervised Pretraining

We first provide contrastive pretraining error similar to Theorem C.2.8 in the multiclass setting.

Theorem C.3.4. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ. Suppose $\hat{\phi}$ satisfies $\hat{\mathcal{L}}_{sup-pre}(\hat{\phi}) \leq \epsilon_0$. Let $p_i := \Pr_{y\sim\eta}\{y = y_i\}$, where $\sum_{i=1}^K p_i = 1$. Let $\rho := \frac{\max_i p_i}{\min_j p_j}$. Consider pretraining set $\mathcal{S}_{sup-pre} := \{x_i, y_i\}_{i=1}^N$, for any $\delta \geq 0$, if

$$N \geq \frac{1}{\epsilon_0}\left[8LR\sqrt{K}\mathcal{R}_N(\Phi) + \frac{8C^2}{\epsilon_0}\log(\frac{2}{\delta})\right].$$

Then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subset \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \hat{\phi}) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left[2\rho^r\epsilon_0 - \mathcal{L}_{sup}(\phi^*_\zeta)\right] + \kappa. \tag{C.57}$$

Proof of Theorem C.3.4. The proof follows similar steps of Theorem C.2.8. □

Below, we provide our main result similar to Theorem C.2.10 for multi-classes setting.

Theorem C.3.5. Assume Assumption 4.2.1 and that Φ has ν-diversity and κ-consistency with respect to ϕ^*, ϕ^*_ζ. Suppose for some small constant $\alpha \in (0,1)$, we solve Equation (4.2)

with empirical loss lower than $\epsilon_1 = \frac{\alpha}{3} 2\rho^r \epsilon_0$ and obtain ϕ'. For any $\delta > 0$, if

$$M \geq \frac{1}{\epsilon_1}\left[4\sqrt{2}\tilde{L}\mathcal{R}_M(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right], Mm \geq \frac{1}{\epsilon_1}\left[8LB\sqrt{r-1}\mathcal{R}_{Mm}(\Phi(\tilde{\epsilon})) + \frac{4C^2}{\epsilon_1}\log(\frac{2}{\delta})\right],$$

then with probability $1-\delta$, for any target task $\mathcal{T}_0 \subseteq \mathcal{C}_0$,

$$\mathcal{L}_{sup}(\mathcal{T}_0, \phi') - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*) \leq \frac{1}{\nu}\left(2\alpha\rho^r\epsilon_0 - \mathcal{L}_{sup}(\phi_\zeta^*)\right) + \kappa. \tag{C.58}$$

Proof of Theorem C.3.5. Recalling Lemma C.2.3 and Lemma C.2.5, the proof follows the same steps in the proof of Theorem C.2.10, with different r. □

C.4 Linear Case Study

In this section, we provide a full analysis of the linear case study to provide intuition about our consistency, diversity, and task selection algorithm. Intuitively, we have multiple classes, each centered around its mean vector. Target data has a subset of classes, while training data has another subset of classes. Consistency and diversity are related to how these two subsets overlap, i.e., the number of shared features and the number of disjoint features. Then, we can link it to the task selection algorithm.

In this section, z_i means the i-th dimension of vector z rather than the sample index.

C.4.1 Problem Setup

Linear Data and Tasks. We consider dictionary learning or sparse coding settings, which is a classic data model (e.g., [204, 270, 33, 241, 242]). Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space and we have input data $x \in \mathcal{X}$. Suppose $Q \in \mathbb{R}^{d \times D}$ is an unknown dictionary with D columns that can be regarded as patterns or features. For simplicity, assume $d = D$ and Q is orthonormal. We have $z \in \{0, -1, +1\}^d$ as a latent class, where z is a hidden vector that indicates the presence of each pattern. Each latent class z has a distribution $\mathcal{D}_z(x)$ over inputs x. We assume $\mathcal{D}_z(x)$ be a distribution with mean Qz, i.e., $x = Qz + \boldsymbol{e}_z$, where $\boldsymbol{e}_z \in \mathbb{R}^d$ is some noise vector drawing from a zero-mean distribution.

For simplicity, we consider each task to be a binary classification task, where $\mathcal{Y} = \{-1, +1\}$ is the label space. In each task (in multitask finetuning or target task), we have two latent classes z, z' (denote the task as $\mathcal{T}_{z,z'}$) and we randomly assign -1 and $+1$ to each latent class. W.l.o.g., we have in $\mathcal{T}_{z,z'}$:

$$x = \begin{cases} Qz + \boldsymbol{e}_z, & \text{if } y = -1 \\ Qz' + \boldsymbol{e}_{z'}, & \text{if } y = +1. \end{cases} \tag{C.59}$$

For simplicity, we consider a balanced class setting in all tasks, i.e., $p(y = -1) = p(y = +1) = \frac{1}{2}$.

Now, we define multitask finetuning tasks and target tasks. Suppose there is a set of latent classes $\mathcal{C} \subseteq \{0, -1, +1\}^d$ used for multitask finetuning tasks, which has an index set $J_{\mathcal{C}} \subseteq [d], k_{\mathcal{C}} := |J_{\mathcal{C}}|$ such that for any $z \in \mathcal{C}$, we have $z_{J_{\mathcal{C}}} \in \{-1, +1\}^{k_{\mathcal{C}}}$ and $z_{[d]\setminus J_{\mathcal{C}}} \in \{0\}^{d-k_{\mathcal{C}}}$. Similarly, suppose there is a set of latent classes $\mathcal{C}_0 \subseteq \{0, -1, +1\}^d$ used for target tasks whose index set is $J_0 \subseteq [d], k_0 := |J_0|$. Note that $J_{\mathcal{C}}$ may or may not overlap with J_0 and denote the set of features encoded both by $\mathcal{C}_0$ and $\mathcal{C}$ as $L_{\mathcal{C}} := J_0 \cap J_{\mathcal{C}}, l_{\mathcal{C}} := |L_{\mathcal{C}}|$. Intuitively, $L_{\mathcal{C}}$ represents the target features covered by training data. Let ζ over $\mathcal{C} \times \mathcal{C}$ be the distribution of multitask finetuning tasks. Then, in short, our data generation pipeline for multitask finetuning tasks is (1) sample two latent classes $(z, z') \sim \zeta$ as a task $\mathcal{T}_{z,z'}$; (2) assign label $-1, +1$ to two latent classes; (3) sample input data from $\mathcal{D}_z(x)$ and $\mathcal{D}_{z'}(x)$ with balanced probabilities.

For simplicity, we have a symmetric assumption and a non-degenerate assumption for ζ. Symmetric assumption means each dimension is equal important and non-degenerate assumption means any two dimensions are not determined by each other in all tasks.

Assumption C.4.1 (Symmetric)**.** We assume for any multitask finetuning tasks distribution ζ, for any $j, k \in J_{\mathcal{C}}$, switching two dimensions z_j and z_k does not change the distribution ζ.

Assumption C.4.2 (Non-degenerate)**.** We assume for any multitask finetuning tasks distribution ζ, for any $j, k \in J_{\mathcal{C}}$, over ζ we have $p(z_j = z'_j, z_k \neq z'_k) > 0$.

Remark C.4.3. There exists many ζ satisfying above assumptions, e.g., (1) $z_{J_\mathcal{C}}$ and $z'_{J_\mathcal{C}}$ uniformly sampling from $\{-1,+1\}^{k_\mathcal{C}}$; or (2) let $k_\mathcal{C}=2$, $z_{J_\mathcal{C}}$ and $z'_{J_\mathcal{C}}$ uniformly sampling from $\{(+1,+1),(-1,+1),(+1,-1)\}$ (note that uniformly sampling from $\{(+1,+1),(-1,-1)\}$ does not satisfy non-degenerate assumption). Also, we note that even when $\mathcal{C}=\mathcal{C}_0$, the target latent class may not exist in the multitask finetuning tasks.

Linear Model and Loss Function. Let Φ be the hypothesis class of models $\phi:\mathcal{X}\to\overline{\mathcal{Z}}$, where $\overline{\mathcal{Z}}\subseteq\mathbb{R}^d$ is the output space of the model. We consider a linear model class with regularity Assumption 4.2.1, i.e., $\Phi=\{\phi\in\mathbb{R}^{d\times d}:\|\phi\|_F\le R\}$ and linear head $w\in\mathbb{R}^d$ where $\|w\|_2\le B$. Thus, the final output of the model and linear head is $w^\top\phi x$. We use linear loss in [245], i.e., $\ell\left(w^\top\phi x,y\right)=-yw^\top\phi x$ and we have

$$\mathcal{L}_{sup}(\mathcal{T},\phi):=\min_{w\in\mathbb{R}^d,\|w\|_2\le B}\mathop{\mathbb{E}}_{z,y\sim\mathcal{T}}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_z(x)}\left[\ell\left(w^\top\phi x,y\right)\right] \tag{C.60}$$

$$\mathcal{L}_{sup}(\phi):=\mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T},\phi)\right] \tag{C.61}$$

$$\phi^*_\zeta:=\arg\min_{\phi\in\Phi}\mathcal{L}_{sup}(\phi), \tag{C.62}$$

where ϕ^*_ζ is the optimal representation for multitask finetuning.

C.4.2 Diversity and Consistency Analysis

Optimal Representation for Multitask Finetuning

Lemma C.4.1. Assume Assumption C.4.1 and Assumption C.4.2. We have $\phi^*_\zeta=U\Lambda^*Q^{-1}$, where U is any orthonormal matrix, $\Lambda^*=\mathrm{diag}(\lambda^*)$. For any $i\in J_\mathcal{C}$, $\lambda^*_i=\frac{R}{\sqrt{k_\mathcal{C}}}$ and $\lambda^*_i=0$ otherwise.

Proof of Lemma C.4.1. We have the singular value decomposition $\phi=U\Lambda V^\top$, where

$\Lambda = \text{diag}(\lambda)$, where $\lambda \in \mathbb{R}^d$. Then, we have

$$\mathcal{L}_{sup}(\phi) = \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T},\phi)\right] \tag{C.63}$$

$$= \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\min_{w\in\mathbb{R}^d,\|w\|_2\leq B}\mathop{\mathbb{E}}_{z,y\sim\mathcal{T}}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_z(x)}\left[\ell\left(w^\top\phi x, y\right)\right]\right] \tag{C.64}$$

$$= \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\min_{w\in\mathbb{R}^d,\|w\|_2\leq B}\frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}_z(x)}\left[\ell\left(w^\top\phi x,-1\right)\right]+\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{z'}(x)}\left[\ell\left(w^\top\phi x,+1\right)\right]\right)\right] \tag{C.65}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\min_{w\in\mathbb{R}^d,\|w\|_2\leq B}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_z(x)}\left[w^\top\phi x\right]+\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{z'}(x)}\left[-w^\top\phi x\right]\right] \tag{C.66}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\min_{w\in\mathbb{R}^d,\|w\|_2\leq B} w^\top\phi Q z - w^\top\phi Q z'\right] \tag{C.67}$$

$$= -\frac{B}{2}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\|\phi Q(z-z')\|_2\right] \tag{C.68}$$

$$= -\frac{B}{2}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\|\Lambda V^\top Q(z-z')\|_2\right]. \tag{C.69}$$

W.l.o.g., we can assume $V^\top = Q^{-1}$. As $\|\phi\|_F = \|\Lambda\|_F = \|\lambda\|_2$ thus we have

$$\begin{aligned}
\min_{\phi\in\Phi}\mathcal{L}_{sup}(\phi) &= -\frac{B}{2}\max_{\|\Lambda\|_F\leq R}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\|\Lambda(z-z')\|_2\right] \\
&= -\frac{B}{2}\max_{\|\lambda\|_2\leq R}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\sqrt{\sum_{i=1}^d\lambda_i^2(z_i-z_i')^2}\right] \\
&= -\frac{B}{2}\max_{\|\lambda\|_2= R}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\sqrt{\sum_{i=1}^d\lambda_i^2(z_i-z_i')^2}\right] \\
&= -B\max_{\|\lambda\|_2= R}\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'}\sim\zeta}\left[\sqrt{\sum_{i\in J_{\mathcal{C}}}\lambda_i^2\mathbb{1}[z_i\neq z_i']}\right],
\end{aligned} \tag{C.70}$$

where $\mathbb{1}[z_i \neq z_i']$ is a Boolean function, mapping True to 1 and False to 0.

Let $\phi_\zeta^* = U\Lambda^* Q^{-1}$ with corresponding λ^*. Now, we use contradiction to prove for any

$j, k \in J_{\mathcal{C}}$, we have $\lambda_j^* = \lambda_k^*$. W.l.o.g., suppose $\lambda_j^* < \lambda_k^*$,

$$
\begin{aligned}
&\mathcal{L}_{sup}(\phi_\zeta^*) \\
&= -B \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_j^{*2} \mathbb{1}[z_j \neq z_j'] + \lambda_k^{*2} \mathbb{1}[z_k \neq z_k'] + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \right] \\
&= -B \Bigg\{ p(z_j \neq z_j', z_k \neq z_k') \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_j^{*2} + \lambda_k^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j \neq z_j', z_k \neq z_k' \right] \\
&\quad + p(z_j = z_j', z_k \neq z_k') \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_k^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j = z_j', z_k \neq z_k' \right] \\
&\quad + p(z_j \neq z_j', z_k = z_k') \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_j^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j \neq z_j', z_k = z_k' \right] \Bigg\}.
\end{aligned}
$$

By symmetric Assumption C.4.1 and non-degenerate Assumption C.4.2, we have $p(z_j = z_j', z_k \neq z_k') = p(z_j \neq z_j', z_k = z_k') > 0$, and

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_k^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j = z_j', z_k \neq z_k' \right] \\
&\quad + \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_j^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j \neq z_j', z_k = z_k' \right] \\
=& \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_k^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j = z_j', z_k \neq z_k' \right] \\
&\quad + \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\lambda_j^{*2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_k \neq z_k', z_j = z_j' \right] \\
<& 2 \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\frac{\lambda_j^{*2} + \lambda_k^{*2}}{2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j = z_j', z_k \neq z_k' \right] \\
=& \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\frac{\lambda_j^{*2} + \lambda_k^{*2}}{2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_j = z_j', z_k \neq z_k' \right] \\
&\quad + \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\frac{\lambda_j^{*2} + \lambda_k^{*2}}{2} + \sum_{i \in J_{\mathcal{C}} \setminus \{j,k\}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \middle| z_k \neq z_k', z_j = z_j' \right].
\end{aligned}
$$

where two equality follows Assumption C.4.1 and the inequality follows Jensen's inequality. Let $\phi' = U\Lambda' Q^{-1}$ with corresponding λ', where $\lambda'_j = \lambda'_k = \sqrt{\frac{\lambda_j^{*2}+\lambda_k^{*2}}{2}}$ and for any $i \in J_{\mathcal{C}} \setminus \{j,k\}$, $\lambda'_i = \lambda^*_i$. We have $\|\phi'\|_F = \|\phi^*_\zeta\|_F$ and $\mathcal{L}_{sup}(\phi^*_\zeta) > \mathcal{L}_{sup}(\phi')$ which is a contradiction as we have ϕ^*_ζ is the optimal solution. Thus, for any $j,k \in J_{\mathcal{C}}$, we have $\lambda^*_j = \lambda^*_k$ and we finish the proof under simple calculation. □

Now, we are ready to analyze consistency and diversity under this linear case study.

Consistency

The intuition is that ζ not only covers $\mathcal{C}_0$ but contains too much unrelated information. Recall that the consistent term in Definition 2 is $\kappa = \sup_{\mathcal{T}_0 \subseteq \mathcal{C}_0} \left[\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_0)\right]$.

We first define some notation we will use later. Let ζ_0 be a multitask finetuning tasks distribution over $\mathcal{C}_0 \times \mathcal{C}_0$ and denote the corresponding optimal representation model as ϕ^*_0. Suppose for any target task $\mathcal{T}_0$ contains two latent classes z, z' from $\mathcal{C}_0$. W.l.o.g., denote z, z' differ in n_0 entries ($1 \le n_0 \le k_0$), whose $n_{\mathcal{C}}$ entries fall in $L_{\mathcal{C}}$, where $0 \le n_{\mathcal{C}} \le n_0$. Then, we get the lemma below:

Lemma C.4.2. Assume Assumption C.4.1 and Assumption C.4.2. We have

$$\kappa = \sup_{\mathcal{T}_0 \subseteq \mathcal{C}_0} \left[\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_0)\right] = BR\left(\sqrt{\frac{n_0}{k_0}} - \sqrt{\frac{n_{\mathcal{C}}}{k_{\mathcal{C}}}}\right). \tag{C.71}$$

Proof of Lemma C.4.2. Recall $1 \le n_0 \le k_0$ and $0 \le n_{\mathcal{C}} \le n_0$. By Lemma C.4.1, we have $\phi^*_\zeta = U\Lambda^* Q^{-1}$, where U is any orthonormal matrix, $\Lambda^* = \mathrm{diag}(\lambda^*)$. For any $i \in J_{\mathcal{C}}$, $\lambda^*_i = \frac{R}{\sqrt{k_{\mathcal{C}}}}$ and $\lambda^*_i = 0$ otherwise. We also have $\phi^*_0 = U_0 \Lambda^*_0 Q^{-1}$, where U_0 is any orthonormal matrix, $\Lambda^*_0 = \mathrm{diag}(\lambda^{0,*})$. For any $i \in J_0$, $\lambda^{0,*}_i = \frac{R}{\sqrt{k_0}}$ and $\lambda^{0,*}_i = 0$ otherwise. Thus, we have

$$\kappa = \sup_{\mathcal{T}_0 \subseteq \mathcal{C}_0} \left[\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_\zeta) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi^*_0)\right] \tag{C.72}$$

$$= BR\left(\sqrt{\frac{n_0}{k_0}} - \sqrt{\frac{n_{\mathcal{C}}}{k_{\mathcal{C}}}}\right). \tag{C.73}$$

□

Let $n'_{\mathcal{C}} = k_{\mathcal{C}} - n_{\mathcal{C}}$. Note this $k_{\mathcal{C}}$ is an increasing factor if $\mathcal{C}$ contains more features. Moreover, $n_{\mathcal{C}}$ is the number of features encoded by both target and training data, representing the information of target data covered by training data, $n_{\mathcal{C}}$ increases as more target information covered by training data, the loss will decrease. $n'_{\mathcal{C}}$ is the number of features encoded in training data but not encoded by target data, representing the un-useful information, $n'_{\mathcal{C}}$ increases as more un-related information is covered by training data, the loss will increase.

Diversity

We first review some definitions in Definition 1. The **averaged representation difference** for two model $\phi, \tilde{\phi}$ on a distribution ζ over tasks is

$$\bar{d}_{\zeta}(\phi, \tilde{\phi}) := \mathop{\mathbb{E}}_{\mathcal{T}\sim\zeta}\left[\mathcal{L}_{sup}(\mathcal{T}, \phi) - \mathcal{L}_{sup}(\mathcal{T}, \tilde{\phi})\right] = \mathcal{L}_{sup}(\phi) - \mathcal{L}_{sup}(\tilde{\phi}). \tag{C.74}$$

The **worst-case representation difference** between representations $\phi, \tilde{\phi}$ on the family of classes $\mathcal{C}_0$ is

$$d_{\mathcal{C}_0}(\phi, \tilde{\phi}) := \sup_{\mathcal{T}_0\subseteq\mathcal{C}_0}\left|\mathcal{L}_{sup}(\mathcal{T}_0, \phi) - \mathcal{L}_{sup}(\mathcal{T}_0, \tilde{\phi})\right|. \tag{C.75}$$

We say the model class Φ has ν-diversity for ζ and $\mathcal{C}_0$ if for any $\phi \in \Phi$ and ϕ^*_{ζ},

$$d_{\mathcal{C}_0}(\phi, \phi^*_{\zeta}) \le \bar{d}_{\zeta}(\phi, \phi^*_{\zeta})/\nu. \tag{C.76}$$

To find the minimum value of ν in Definition 1, we need further information about ζ. For simplicity, we have a fixed distance assumption, e.g., uniformly sampling from $\{(+1, +1, -1), (+1, -1, +1), (-1, +1, +1)\}$. Then, we consider two different cases below. We consider that all $\mathcal{T}_0 \subseteq \mathcal{C}_0$ such containing z, z' that differ in only 1 entry.

Assumption C.4.4 (Fixed Distance)**.** We assume for any multitask finetuning tasks distribution ζ, for any two latent classes $(z, z') \sim \zeta$, we have z, z' differ in n_k entries.

Case $L_{\mathcal{C}} \neq J_0$. In this case, $J_0 \setminus L_{\mathcal{C}} \neq \emptyset$, we have the features learned in multitask finetuning that do not cover all features used in the target task. Then, we have the following lemma, which means if $L_{\mathcal{C}} \neq J_0$ we can have infinitesimal ν to satisfy the diversity definition.

Lemma C.4.3. Assume Assumption C.4.1, Assumption C.4.2 and Assumption C.4.4. When $L_{\mathcal{C}} \neq J_0$, we have $\nu \to 0$.

Proof of Lemma C.4.3. As features in $\mathcal{C}_0$ not covered by $\mathcal{C}$, we can always find a $\mathcal{T}_0$ such containing z, z' that only differ in entries in $J_0 \setminus L_{\mathcal{C}}$, we say as entry $\tilde{i}$.

By Lemma C.4.1, we have $\phi_\zeta^* = U\Lambda^* Q^{-1}$, where U is any orthonormal matrix, $\Lambda^* = \text{diag}(\lambda^*)$. For any $i \in J_{\mathcal{C}}$, $\lambda_i^* = \frac{R}{\sqrt{k_{\mathcal{C}}}}$ and $\lambda_i^* = 0$ otherwise. We have $\mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) = 0$ and by Equation (C.70),

$$\mathcal{L}_{sup}(\phi_\zeta^*) = -B \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\sum_{i \in J_{\mathcal{C}}} \lambda_i^{*2} \mathbb{1}[z_i \neq z_i']} \right] \tag{C.77}$$

$$= -BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}. \tag{C.78}$$

On the other hand, for any $\phi \in \Phi$, we have $\mathcal{L}_{sup}(\mathcal{T}_0, \phi) = -B|\lambda_{\tilde{i}}|$. Thus, we have

$$\begin{aligned}
\nu &= \min_{\phi \in \Phi} \frac{\mathcal{L}_{sup}(\phi) - \mathcal{L}_{sup}(\phi_\zeta^*)}{\left| \mathcal{L}_{sup}(\mathcal{T}_0, \phi) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) \right|} \\
&= \min_{\phi \in \Phi} \frac{\mathcal{L}_{sup}(\phi) + BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{B|\lambda_{\tilde{i}}|} \\
&= \min_{\phi \in \Phi} \frac{-B \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\sum_{i \in J_{\mathcal{C}}} \lambda_i^2 \mathbb{1}[z_i \neq z_i']} \right] + BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{B|\lambda_{\tilde{i}}|} \\
&\leq \frac{- \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta} \left[\sqrt{\sum_{i \in J_{\mathcal{C}}} \frac{R^2 - \lambda_i^2}{k_{\mathcal{C}}} \mathbb{1}[z_i \neq z_i']} \right] + R\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|\lambda_{\tilde{i}}|} \\
&= \frac{-\sqrt{\frac{(R^2 - \lambda_{\tilde{i}}^2) n_k}{k_{\mathcal{C}}}} + R\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|\lambda_{\tilde{i}}|},
\end{aligned} \tag{C.79}$$

where the first inequality is by constructing a specific ϕ. Note that Equation (C.79) $\to 0$

when $|\lambda_{\tilde{i}}| \to 0$. ϕ is constructed as: for any $i \in J_{\mathcal{C}}$, $\lambda_i = \sqrt{\frac{R^2 - \lambda_{\tilde{i}}^2}{k_{\mathcal{C}}}}$ and $|\lambda_{\tilde{i}}| \to 0$. Thus, we finish the proof. □

Case $L_{\mathcal{C}} = J_0$. In this case $J_0 \setminus L_{\mathcal{C}} = \emptyset$, we have all features in $\mathcal{C}_0$ covered by $\mathcal{C}$.

Lemma C.4.4. Assume Assumption C.4.1, Assumption C.4.2 and Assumption C.4.4. When all $\mathcal{T}_0 \subseteq \mathcal{C}_0$ such containing z, z' that differ in only 1 entry and $L_{\mathcal{C}} = J_0$, we have ν is lower bounded by some constant $\tilde{c} = \sqrt{n_k}\left(1 - \sqrt{\frac{1}{k_{\mathcal{C}}(k_{\mathcal{C}}-1)}}\left(\sqrt{n_k(n_k-1)} + k_{\mathcal{C}} - n_k\right)\right)$.

Proof of Lemma C.4.4. We say the differ entry in $\mathcal{T}_0$ as entry $\tilde{i}$. By Lemma C.4.1, we have $\phi_\zeta^* = U\Lambda^* Q^{-1}$, where U is any orthonormal matrix, $\Lambda^* = \mathrm{diag}(\lambda^*)$. For any $i \in J_{\mathcal{C}}$, $\lambda_i^* = \frac{R}{\sqrt{k_{\mathcal{C}}}}$ and $\lambda_i^* = 0$ otherwise. By Equation (C.70), we have $\mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*) = -BR\sqrt{\frac{n_0}{k_{\mathcal{C}}}}$ and $\mathcal{L}_{sup}(\phi_\zeta^*) = -BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}$.

On the other hand, for any $\phi \in \Phi$, we have $\mathcal{L}_{sup}(\mathcal{T}_0, \phi) = -B|\lambda_{\tilde{i}}|$. Thus, by Assumption C.4.1, we have

$$
\begin{aligned}
\nu &= \min_{\mathcal{T}_0 \subseteq \mathcal{C}_0, \phi \in \Phi} \frac{\mathcal{L}_{sup}(\phi) - \mathcal{L}_{sup}(\phi_\zeta^*)}{\left|\mathcal{L}_{sup}(\mathcal{T}_0, \phi) - \mathcal{L}_{sup}(\mathcal{T}_0, \phi_\zeta^*)\right|} \\
&= \min_{\mathcal{T}_0 \subseteq \mathcal{C}_0, \phi \in \Phi} \frac{\mathcal{L}_{sup}(\phi) + BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|-B|\lambda_{\tilde{i}}| + BR\sqrt{\frac{1}{k_{\mathcal{C}}}}|} \\
&= \min_{\mathcal{T}_0 \subseteq \mathcal{C}_0, \phi \in \Phi} \frac{-B \mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta}\left[\sqrt{\sum_{i \in J_{\mathcal{C}}} \lambda_i^2 \mathbb{1}[z_i \neq z_i']}\right] + BR\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|-B|\lambda_{\tilde{i}}| + BR\sqrt{\frac{1}{k_{\mathcal{C}}}}|} \\
&= \min_{\mathcal{T}_0 \subseteq \mathcal{C}_0, \phi \in \Phi} \frac{-\mathop{\mathbb{E}}_{\mathcal{T}_{z,z'} \sim \zeta}\left[\sqrt{\lambda_{\tilde{i}}^2 \mathbb{1}[z_{\tilde{i}} \neq z_{\tilde{i}}'] + \sum_{i \in J_{\mathcal{C}} \setminus \{\tilde{i}\}} \frac{R^2 - \lambda_{\tilde{i}}^2}{k_{\mathcal{C}}-1} \mathbb{1}[z_i \neq z_i']}\right] + R\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|-|\lambda_{\tilde{i}}| + R\sqrt{\frac{1}{k_{\mathcal{C}}}}|} \\
&= \min_{\mathcal{T}_0 \subseteq \mathcal{C}_0, \phi \in \Phi} \frac{-\left[\frac{n_k}{k_{\mathcal{C}}}\sqrt{\lambda_{\tilde{i}}^2 + \frac{R^2 - \lambda_{\tilde{i}}^2}{k_{\mathcal{C}}-1}(n_k - 1)} + \frac{k_{\mathcal{C}} - n_k}{k_{\mathcal{C}}}\sqrt{\frac{n_k(R^2 - \lambda_{\tilde{i}}^2)}{k_{\mathcal{C}}-1}}\right] + R\sqrt{\frac{n_k}{k_{\mathcal{C}}}}}{|-|\lambda_{\tilde{i}}| + R\sqrt{\frac{1}{k_{\mathcal{C}}}}|} \\
&= \sqrt{n_k}\left(1 - \sqrt{\frac{1}{k_{\mathcal{C}}(k_{\mathcal{C}}-1)}}\left(\sqrt{n_k(n_k-1)} + k_{\mathcal{C}} - n_k\right)\right), \qquad \text{(C.80)}
\end{aligned}
$$

where the last equality take $\lambda_{\tilde{i}} = 0$. □

C.4.3 Proof of Main Results

Proof of Theorem 4.2.3. Note that $R = B = n_0 = k_0 = 1, n_k = 2$.

We see that ζ satisfies Assumption C.4.1, Assumption C.4.2 and Assumption C.4.4. We finish the proof by Lemma C.4.2, Lemma C.4.3 and Lemma C.4.4 with some simple calculations. □

Thus, we can link our diversity and consistency parameters to the number of features in z encoded by training tasks or target tasks. Based on this intuition, we propose a selection algorithm, where selection is based on x, we want to select data that encodes more relevant features of z, this can be achieved by comparing x from target data and training data either using cosine similarity or KDE.

C.5 Vision Experimental Results

We first provide a summary of dataset and protocal we use, we provide details in following sections.

Datasets and Models. We use four widely used few-shot learning benchmarks: mini-ImageNet [271], tieredImageNet [228], DomainNet [218] and Meta-dataset [265], following the protocol in [51, 259]. We use exemplary foundation models with different pretraining schemes (MoCo-v3 [49], DINO-v2 [208], and supervised learning with ImageNet [231]) and architectures (ResNet [121] and ViT [73]).

Experiment Protocol. We consider few-shot tasks consisting of N classes with K support samples and Q query samples per class (known as N-way K-shot). The goal is to classify the query samples into the N classes based on the support samples. Tasks used for finetuning are constructed by samples from the training split. Each task is formed randomly by sampling 15 classes, with every class drawing 1 or 5 support samples and 10

query samples. Target tasks are similarly constructed, yet from the test set. We follow [51] for multitask finetuning and target task adaptation. During multitask finetuning, we update all parameters in the model using a nearest centroid classifier, in which all samples are encoded, class centroids are computed, and cosine similarity between a query sample and those centroids are treated as the class logits. For adaptation to a target task, we only retain the model encoder and consider a similar nearest centroid classifier. This experiment protocol applies to all three major experiments (Sections 4.3.1 to 4.3.3).

C.5.1 Datasets

The miniImageNet dataset is a common benchmark for few-shot learning. It contains 100 classes sampled from ImageNet, then is randomly split into 64, 16, and 20 classes as training, validation, and testing set respectively.

The tieredImageNet dataset is another widely used benchmark for few-shot learning. It contains 608 classes from 34 super-categories sampled from ImageNet. These categories are then subdivided into 20 training categories with 351 classes, 6 validation categories with 97 classes, and 8 testing categories with 160 classes

DomainNet is the largest domain adaptation benchmark with about 0.6 million images. It consists of around 0.6 million images of 345 categories from 6 domains: clipart (clp), infograph (inf), quickdraw (qdr), real (rel) and sketch (skt). We split it into 185, 65, 100 classes as training, validation, and testing set respectively. We conduct experiments on Sketch (skt) subsets.

Meta-Dataset encompasses ten publicly available image datasets covering a wide array of domains: ImageNet-1k, Omniglot, FGVC-Aircraft, CUB-200-2011, Describable Textures, QuickDraw, FGVCx Fungi, VGG Flower, Traffic Signs, and MSCOCO. Each of these datasets is split into training, validation, and testing subsets. For additional information on the Meta-Dataset can be found in Appendix 3 of [265].

C.5.2 Experiment Protocols

Our evaluation and the finetuning process take the form of few-shot tasks, where a target task consists of N classes with K support samples and Q query samples in each class. The objective is to classify the query samples into the N classes based on the support samples. To accomplish this, we take the support samples in each class and feed them through an image encoder to obtain representations for each sample. We then calculate the average of these representations within each class to obtain the centroid of each class. For a given query sample x, we compute the probability that x belongs to class y based on the cosine similarity between the representation of x and the centroid of class y.

In our testing stage, we constructed 1500 target tasks, each consisting of 15 classes randomly sampled from the test split of the dataset. Within each class, we randomly selected 1 or 5 of the available images as shot images and 15 images as query images. These tasks are commonly referred to as 1-shot or 5-shot tasks. We evaluated the performance of our model on these tasks and reported the average accuracy along with a 95% confidence interval.

During multitask finetuning, the image encoder is directly optimized on few-shot classification tasks. To achieve this, we construct multitasks in the same format as the target tasks and optimize from the same evaluation protocol. Specifically, we create a total of 200 finetuning tasks, each task consists of 15 classes sampled from the train split of data, where each class contains 1 support image and 9 query images, resulting in 150 images per task. The classes in a finetuning task are sampled from the train split of the data.

To ensure a fair comparison with the finetuning baseline, we used the same training and testing data, as well as batch size, and applied standard finetuning. During standard finetuning, we added a linear layer after the encoder and trained the model. We also utilized the linear probing then finetuning (LP-FT) technique proposed by [148], which has been shown to outperform finetuning alone on both in-distribution and out-of-distribution data. In the testing stage, we removed the linear layer and applied the same few-shot testing pipeline to the finetuned encoders.

For task selection, we employ the CLIP ViT-B image encoder to obtain image embeddings. We assess consistency by measuring the cosine similarity of the mean embeddings and we evaluate diversity through a coverage score derived from the ellipsoid formula outlined in Section 4.2.2.

For optimization, we use the SGD optimizer with momentum 0.9, the learning rate is 1e-5 for CLIP and moco v3 pretrained models, and is 2e-6 for DINO v2 pretrained models. The models were finetuned over varying numbers of epochs in each scenario until they reached convergence.

C.5.3 Existence of Task Diversity

Task diversity is crucial for the foundation model to perform well on novel classes in target tasks.

In this section, we prove for task satisfying consistency, greater diversity in the related data can help reduce the error on the target task. Specifically, for the target task, where the target tasks data originates from the test split of a specific dataset, we utilized the train split of the same dataset as the finetuning tasks data. Then finetuning tasks satisfied consistency. In experiments, we varied the number of classes accessible to the model during the finetuning stage, while keeping the total sample number the same. This serves as a measure of the diversity of training tasks.

miniImageNet and Omniglot

We show the results of CLIP encoder on miniImageNet and Omniglot. We vary the number of classes model access to in finetuning stage. The number of classes varies from all classes, i.e., 64 classes, to 8 classes. Each task contains 5 classes. For finetuning tasks, each class contains 1 shot image and 10 query images. For target tasks, each class contains the 1-shot image and 15 query images.

Table C.1 shows the accuracy of ViT-B32 across different numbers of classes during the finetuning stage. The “Class 0” represents direct evaluation without any finetuning.

# **limited classes**	64	32	16	8	0
Accuracy	90.02 ± 0.15	88.54 ± 1.11	87.94 ± 0.22	87.07 ± 0.20	83.03 ± 0.24

Table C.1: Class diversity on ViT-B32 backbone on miniImageNet.

We observe that finetuning the model leads to an average accuracy improvement of 4%. Furthermore, as the diversity of classes increases, we observe a corresponding increase in performance. This indicates that incorporating a wider range of classes during the finetuning process enhances the model's overall accuracy.

For task diversity, we also use dataset Omniglot [149]. The Omniglot dataset is designed to develop more human-like learning algorithms. It contains 1623 different handwritten characters from 50 different alphabets. The 1623 classes are divided into 964, 301, and 358 classes as training, validation, and testing sets respectively. We sample multitask in finetuning stage from training data and the target task from testing data.

# **limited classes**	964	482	241	50	10	0
Accuracy	95.35 ± 0.14	95.08 ± 0.14	94.29 ± 0.15	88.48 ± 0.20	80.26 ± 0.24	74.69 ± 0.26

Table C.2: Class diversity on ViT-B32 backbone on Omniglot.

Table C.2 shows the accuracy of ViT-B32 on different numbers of classes in finetuning stage, where class 0 indicates direct evaluation without finetuning. Finetuning improves the average accuracy by 5.5%. As class diversity increases, performance increases.

tieredImageNet

We then show results on tieredImageNet across learning settings for the ViT-B backbone. We follow the same setting where we restrain each task that contains 15 classes.

We found that using more classes from related data sources during finetuning improves accuracy. This result indicates that upon maintaining consistency, a trend is observed where increased diversity leads to an enhancement in performance.

Pretrained	351	175	43	10
DINOv2	84.74	82.75	82.60	82.16
CLIP	68.57	67.70	67.06	63.52
Supervised	89.97	89.69	89.19	88.92

Table C.3: The performance of the ViT-B backbone using different pretraining methods on tieredImagenet, varying the number of classes accessible to the model during the finetuning stage. Each column represents the number of classes within the training data.

C.5.4 Ablation Study

In Section 4.3 and the result in Table 4.2, we utilize the train split from the same dataset to construct the finetuning data. It is expected that the finetuning data possess a diversity and consistency property, encompassing characteristics that align with the test data while also focusing on its specific aspects.

In the following ablation study, we explore the relationship between the diversity and consistency of data in finetuning tasks, sample complexity, and finetuning methods. We seek to answer the following questions: Does multitask finetuning benefit only from certain aspects? How do these elements interact with each other?

Violate both consistency and diversity: Altering Finetuning Task Data with Invariant Sample Complexity

In this portion, we examine the performance when the model is finetuned using data completely unrelated to the target task data. With the same finetuning sample complexity, the performance cannot be compared to the accuracy we have currently attained.

In this section, we present the performance of MoCo v3 with a ViT-B backbone on the DomainNet dataset. We finetuned the model using either ImageNet data or DomainNet train-split data and evaluated its performance on the test-split of DomainNet. We observed that finetuning the model with data selected from the DomainNet train-split resulted in improved performance on the target task. This finding aligns with our expectations and highlights the significance of proper finetuning data selection.

When considering the results presented in Table C.4, we also noticed that for MoCo v3 with a ResNet50 backbone and DINO v2 with a ViT-S backbone, multitask finetuning on ImageNet led to a decrease in model performance compared to direct adaptation. This suggests that inappropriate data selection can have a detrimental effect on the final model performance. This conclusion is also supported by the findings of [148].

pretrained	**backbone**	**FT data**	**Accuracy**
MoCo v3	ViT-B	ImageNet	24.88 (0.25)
		DomainNet	32.88 (0.29)
	ResNet50	ImageNet	27.22 (0.27)
		DomainNet	33.53 (0.30)
DINO v2	ViT-S	ImageNet	51.69 (0.39)
		DomainNet	61.57 (0.40)
	ViT-B	ImageNet	62.32 (0.40)
		DomainNet	68.22 (0.40)
Supervised	ViT-B	ImageNet	31.16 (0.31)
		DomainNet	48.02 (0.38)
	ResNet50	ImageNet	29.56 (0.28)
		DomainNet	39.09 (0.34)

Table C.4: Finetuning data selection on model performance. FT data: dataset we select for multitask finetuning. Report the accuracy on the test-split of DomainNet.

Violating consistency while retaining diversity: The Trade-Off between Task Consistency and Sample Complexity

Finetuning tasks with superior data are expected to excel under identical complexity, a natural question can be proposed: Does additional data enhance performance? Our results in this section negate this question. Testing the model on the DomainNet test-split, we employ two settings. In the first setting, we finetune the model on the DomainNet train-split. In the second, the model is finetuned with a combination of the same data from DomainNet as in the first setting, along with additional data from ImageNet.

Within our theoretical framework, mixing data satisfies diversity but fails consistency. The finetuning data, although containing related information, also encompasses excessive unrelated data. This influx of unrelated data results in a larger consistency parameter κ in our theoretical framework, adversely impacting model performance on the target task. We offer empirical evidence to affirm our theoretical conclusion.

Pretrained	**DomainNet**	**DomainNet + ImageNet**
DINOv2	68.22	66.93
CLIP	64.97	63.48
Supervised	48.02	43.76

Table C.5: Results evaluating on DomainNet test-split using ViT-B backbone. First column shows performance where model finetune on data from DomainNet train-split alone, second column shows the performance of the model finetuned using a blend of the same data from DomainNet, combined with additional data from ImageNet.

Table C.5 shows mixed data of domainNet and ImageNet will doesn't provide the same advantages as using only DomainNet data. In this case, an increasing in data does not necessarily mean better performance.

Diversity and Consistency of Task Data and Finetuning Methods

To provide a more comprehensive understanding of the impact of task data and finetuning methods on model performance, we conduct additional experiments, utilizing varying finetuning methods and data. The model is tested on the DomainNet test split. We employ either multitask finetuning or standard finetuning, where a linear layer is added after the pretrained model. This linear layer maps the representations learned by encoders to the logits. The data of finetuning tasks derive from either the DomainNet train-split or ImageNet.

In Table C.6, we detail how data quality and finetuning methods of tasks impact the ultimate performance. Standard finetuning (SFT) with unrelated data diminishes performance compared to direct adaptation (col-1 vs col-2). On the other hand, multitask finetuning using unrelated data (ImageNet), or SFT with related data (DomainNet), both

Pretrained	1 Adaptation	2 ImageNet (SFT)	3 ImageNet (Ours)	4 DomainNet (SFT)	5 DomainNet (Ours)
DINOv2	61.65	59.80	62.32	61.84	68.22
CLIP	46.39	46.50	58.94	47.72	64.97
Supervised	28.70	28.52	31.16	30.93	48.02

Table C.6: Results evaluating on DomainNet test-split using ViT-B backbone. Adaptation: Direction adaptation without finetuning; SFT: Standard finetuning; Ours: Our multitask finetuning. Col-1 shows performance without any finetuning, Col-2,3,4,5 shows performance with different finetuning methods and data.

outperform direct adaptation. However, multitask finetuning with unrelated data proves more beneficial than the latter (col-3 vs col-4). The peak performance is attained through multitask finetuning on related data (col-5).

Ablation Study on Task Selection Algorithm

We show a simplified diagram for task selection in Figure 4.2.

We first provide some details of Table 4.1. We first create an array of finetuning tasks, and then apply our task selection algorithm to these tasks. Specifically, we design 100 finetuning tasks by randomly selecting 15 classes, each providing 1 support sample and 10 query samples. The target tasks remain consistent with those discussed in Section 4.3. For a more comprehensive analysis of our algorithm, we performed ablation studies on the task selection algorithm, concentrating solely on either consistency or diversity, while violating the other. **Violate Diversity**: If the algorithm terminates early without fulfilling the stopping criteria, the data utilized in finetuning tasks fails to encompass all the attributes present in the target data. This leads to a breach of the diversity principle. **Violate Consistency**: Conversely, if the algorithm persists beyond the stopping criteria, the finetuning tasks become overly inclusive, incorporating an excessive amount of unrelated data, thus breaching the consistency.

This section details an ablation study on task selection for the dataset, we implement our task selection process on a meta-dataset, treating each dataset as a distinct task and choosing datasets to serve as data sources for the finetuning tasks. We show the result in Table C.7.

Pretrained	Selection	INet	Omglot	Acraft	CUB	QDraw	Fungi	Flower	Sign	COCO
CLIP	All	60.87	70.53	31.67	66.98	40.28	34.88	80.80	37.82	33.71
	Selected	60.87	**77.93**	**32.02**	**69.15**	**42.36**	**36.66**	**80.92**	**38.46**	**37.21**
DINOv2	All	83.04	72.92	36.52	94.01	49.65	52.72	98.54	34.59	47.05
	Selected	83.04	**80.29**	**36.91**	**94.12**	**52.21**	**53.31**	**98.65**	**36.62**	**50.09**
MoCo v3	All	59.62	60.85	18.72	40.49	40.96	32.65	59.60	33.94	33.42
	Selected	59.62	**63.08**	**19.03**	**40.74**	**41.16**	**32.89**	**59.64**	**35.25**	**33.51**

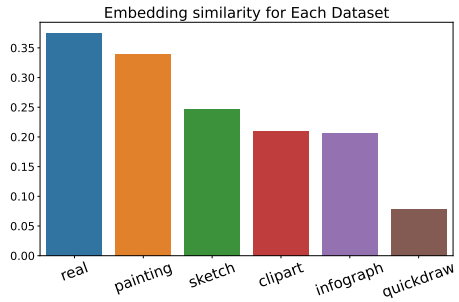
Table C.7: Results evaluating our task selection algorithm on Meta-dataset using ViT-B backbone.

Table C.7 indicates that maintaining both consistency and diversity in the task selection algorithm is essential for optimal performance. This is evident from the comparison between the Random selection and the our approach, where the latter often shows improved performance across multiple datasets. ImageNet as the target task is an exception where the two approaches give the best results. Due to its extensive diversity, all samples from all other datasets are beneficial for finetuning. Consequently, the task selection algorithm tends to select all the candidate tasks.

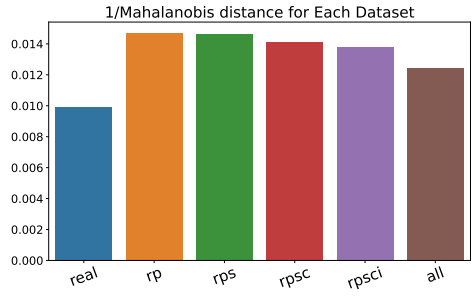
C.5.5 Task Selection Algorithm on DomainNet

We verify our task selection algorithm by applying it on DomainNet. Here, the mini-ImageNet test-split is regarded as the target task source, and diverse domains (such as clipart (clp), infograph (inf), quickdraw (qdr), real (rel), and sketch (skt)) are considered as sources for finetuning tasks. We view different domain datasets as distinct finetuning tasks. With 6 domains in focus, our objective is to select a subset that optimizes model performance. We systematically apply Algorithm 2. Initially, we calculate the cosine similarity of mean embeddings between each domain and target tasks, ordering them from most to least similar: real, painting, sketch, clipart, infograph, and quickdraw. Sequentially adding datasets in this order, the process continues until the diversity score (1 over Mahalanobis distance) stops exhibiting significant increase.

As we can see in Figure C.1, the diversity does not increase when we just select *real* and *painting* as our finetuning task data. For a comprehensive analysis, each combination is finetuned and the model performance accuracy on the target task is displayed.



(a) Mean embedding similarity for each data sort from most similar to least similar.



(b) Diversity score when adding task one by one, where *rp*: *real* and *painting*; *rps*: *real* and *painting* and *sketch* and so on.

Figure C.1: Dataset selection based on consistency and diversity on domainNet. Figure C.1a shows the consistency. Figure C.1b shows the diversity.

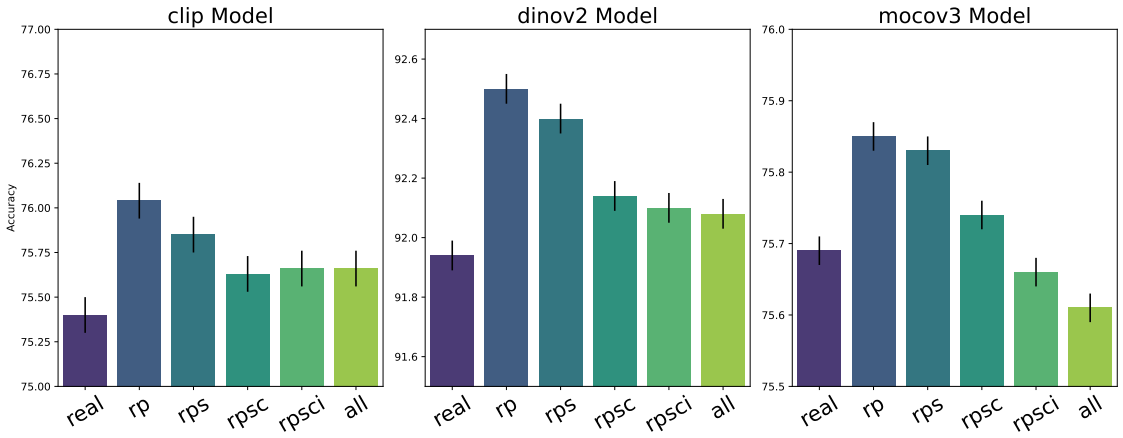


Figure C.2: Finetuning with different selection of domain datasets, where *rp*: *real* and *painting*; *rps*: *real* and *painting* and *sketch* and so on.

As we can see in Figure C.2,the accuracy aligns with the conclusions drawn based on consistency and diversity. Remarkably, only *real* and *painting* suffice for the model to excel on the target task.

C.5.6 More Results with CLIP Encoder

In this section, we show additional results on CLIP [221] model.

We can observe from Table C.8 standard finetuning improves performance compared to direct adaptation. However, our proposed multitask finetuning approach consistently

		miniImageNet		tieredImageNet		DomainNet	
backbone	**method**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
CLIP-ViTB32	Direct Adaptation	68.41 (0.54)	87.43 (0.15)	59.55 (0.21)	79.51 (0.27)	46.48 (0.37)	72.01 (0.29)
	Standard FT	69.39 (0.30)	88.39 (0.15)	61.20 (0.37)	80.65 (0.27)	47.72 (0.37)	72.82 (0.29)
	Multitask FT (Ours)	**78.62** (0.15)	**93.22** (0.11)	**68.57** (0.37)	**84.79** (0.22)	**64.97** (0.39)	**80.49** (0.25)
CLIP-ResNet50	Direct Adaptation	61.31 (0.31)	82.03 (0.18)	51.76 (0.36)	71.40 (0.30)	40.55 (0.36)	64.90 (0.31)
	Standard FT	63.15 (0.31)	83.45 (0.17)	55.77 (0.35)	75.28 (0.29)	43.77 (0.38)	67.30 (0.31)
	Multitask FT (Ours)	**67.03** (0.30)	**85.09** (0.17)	**57.56** (0.36)	**75.80** (0.28)	**52.67** (0.39)	**72.19** (0.30)

Table C.8: **Comparison on 15-way classification.** Average few-shot classification accuracies (%) with 95% confidence intervals clip encoder.

achieves even better results than the standard baseline.

Task (M) vs Sample (m). We vary the task size and sample size per task during finetuning. We verify the trend of different numbers of tasks and numbers of images per task. Each task contains 5 classes. For finetuning tasks, $m = 50$ indicates each class contains the 1-shot image and 9-query images. $m = 100$ indicates each class contains 2-shot and 18-query images. $m = 200$ indicates each class contains 4-shot and 36-query images. $M = m = 0$ indicates direct evaluation without finetuning. For target tasks, each class contains the 1-shot image and 15 query images.

Sample (m) / **Task (M)**	0	50	100	200
0	83.03 ± 0.24			
200		89.07 ± 0.20	89.95± 0.19	**90.09 ± 0.19**
400		89.31 ± 0.19	**90.11± 0.19**	90.70 ± 0.18
800		**89.71 ± 0.19**	90.27± 0.19	90.80 ± 0.18

Table C.9: Accuracy with a varying number of tasks and samples (ViT-B32 backbone).

Table C.9 shows the results on the pretrained CLIP model using the ViT backbone. For direct adaptation without finetuning, the model achieves 83.03% accuracy. Multitask finetuning improves the average accuracy at least by 6%. For a fixed number of tasks or samples per task, increasing samples or tasks improves accuracy. These results suggest that the total number of samples ($M \times m$) will determine the overall performance, supporting our main theorem.

Few-shot Effect. We perform experiments on the few-shot effects of finetuning tasks. We aim to evaluate whether increasing the number of few-shot images in the finetuning task leads to significant improvements. Each finetuning task consists of 5 classes, and we maintain a fixed number of 10 query images per class while gradually increasing the number of shot images, as illustrated in Table C.10. As for the target tasks, we ensure each class contains 1 shot image and 15 query images for evaluation.

# **shot images**	20	10	5	1	0
Accuracy	91.03 ± 0.18	90.93 ± 0.18	90.54 ± 0.18	90.02 ± 0.15	83.03 ± 0.24

Table C.10: Few-shot effect on ViT-B32 backbone on miniImageNet.

Table C.10 displays the accuracy results of ViT-B32 when varying the number of few-shot images in the finetuning tasks. We observe that increasing the number of few-shot images, thereby augmenting the sample size within each task, leads to improved performance. This finding is quite surprising, considering that the finetuning tasks and target tasks have different numbers of shot images. However, this aligns with our understanding of sample complexity, indicating that having access to more training examples can enhance the model's ability to generalize and perform better on unseen data.

C.5.7 Sample Complexity on Performance for tieredImageNet

We provide a table and visualization of the trend of the number of tasks and the number of samples per task for the MoCo v3 ViT model on tieredImageNet in Table C.11 and Figure C.3.

As demonstrated in the paper, we have observed that increasing the number of tasks generally leads to performance improvements, while keeping the number of samples per task constant. Conversely, when the number of samples per task is increased while maintaining the same number of tasks, performance also tends to improve. These findings emphasize the positive relationship between the number of tasks and performance, as well as the influence of sample size within each task.

Task (M) \ Sample (m)	150	300	450	600
200	68.32 (0.35)	71.42 (0.35)	73.84 (0.35)	75.58 (0.35)
400	71.41 (0.35)	75.60 (0.35)	77.57 (0.34)	78.66 (0.34)
600	73.85 (0.35)	77.59 (0.34)	79.04 (0.33)	79.76 (0.33)
800	75.56 (0.35)	78.68 (0.34)	79.78 (0.33)	80.26 (0.33)

Table C.11: Accuracy with a varying number of tasks and samples (ViT-B32 backbone).

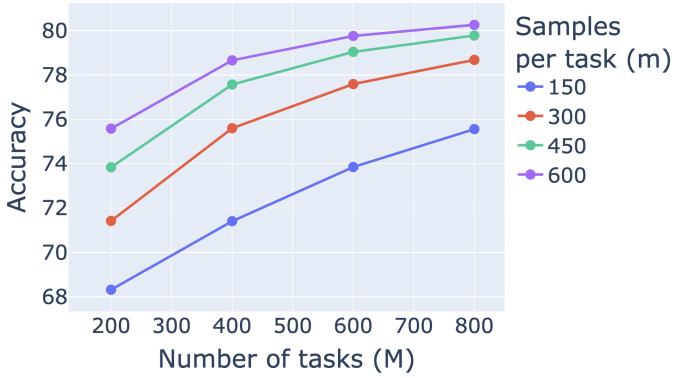


Figure C.3: Finetuning using tieredImageNet train-split, test on test-split.

C.5.8 Full results for Effectiveness of Multitask Finetuning

In this section, we provide another baseline in complement to the results in Section 4.3.3.

We incorporated the Model-Agnostic Meta-Learning (MAML) algorithm, as outlined by [82], as another baseline for our few-shot tasks. MAML operates in a two-step process: it initially updates parameters based on within-episode loss (the inner loop), then it evaluates and updates loss based on learned parameters (the outer loop). We follow the pipeline in [265] to implement MAML for few-shot tasks. We show results in Table C.12.

Table C.12 reveals that MAML exhibits variable performance across different settings. For instance, it outperforms both Adaptation and Standard FT methods in scenarios like MoCo v3 ViT-B on miniImageNet, DomainNet, and ResNet 50 on supervised training for tieredImageNet. However, its performance is less impressive in other contexts, such as DINOv2 ViT-B on miniImageNet and ViT-B on supervised training for miniImageNet. This variability in performance is attributed to the constraints of our few-shot tasks, where the limited number of support samples restricts the model's capacity to adapt to new tasks.

			miniImageNet		tieredImageNet		DomainNet	
pretrained	**backbone**	**method**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
MoCo v3	ViT-B	Adaptation	75.33 (0.30)	92.78 (0.10)	62.17 (0.36)	83.42 (0.23)	24.84 (0.25)	44.32 (0.29)
		Standard FT	75.38 (0.30)	92.80 (0.10)	62.28 (0.36)	83.49 (0.23)	25.10 (0.25)	44.76 (0.27)
		MAML	79.26 (0.28)	93.02 (0.08)	67.96 (0.32)	84.66 (0.19)	28.91 (0.39)	51.12 (0.28)
		Ours	**80.62** (0.26)	**93.89** (0.09)	**68.32** (0.35)	**85.49** (0.22)	**32.88** (0.29)	**54.17** (0.30)
	ResNet50	Adaptation	68.80 (0.30)	88.23 (0.13)	55.15 (0.34)	76.00 (0.26)	27.34 (0.27)	47.50 (0.28)
		Standard FT	68.85 (0.30)	88.23 (0.13)	55.23 (0.34)	76.07 (0.26)	27.43 (0.27)	47.65 (0.28)
		MAML	69.28 (0.26)	88.78 (0.12)	55.31 (0.32)	75.51 (0.19)	27.53 (0.39)	47.73 (0.28)
		Ours	**71.16** (0.29)	**89.31** (0.12)	**58.51** (0.35)	**78.41** (0.25)	**33.53** (0.30)	**55.82** (0.29)
DINO v2	ViT-S	Adaptation	85.90 (0.22)	95.58 (0.08)	74.54 (0.32)	89.20 (0.19)	52.28 (0.39)	72.98 (0.28)
		Standard FT	86.75 (0.22)	95.76 (0.08)	74.84 (0.32)	89.30 (0.19)	54.48 (0.39)	74.50 (0.28)
		MAML	86.67 (0.24)	95.54 (0.08)	74.63 (0.34)	89.60 (0.19)	52.72 (0.34)	73.35 (0.28)
		Ours	**88.70** (0.22)	**96.08** (0.08)	**77.78** (0.32)	**90.23** (0.18)	**61.57** (0.40)	**77.97** (0.27)
	ViT-B	Adaptation	90.61 (0.19)	97.20 (0.06)	82.33 (0.30)	92.90 (0.16)	61.65 (0.41)	79.34 (0.25)
		Standard FT	91.07 (0.19)	97.32 (0.06)	82.40 (0.30)	93.07 (0.16)	61.84 (0.39)	79.63 (0.25)
		MAML	90.77 (0.18)	97.20 (0.08)	82.54 (0.32)	92.88 (0.19)	62.30 (0.39)	79.01 (0.28)
		Ours	**92.77** (0.18)	**97.68** (0.06)	**84.74** (0.30)	**93.65** (0.16)	**68.22** (0.40)	**82.62** (0.24)
Supervised pretraining on ImageNet	ViT-B	Adaptation	94.06 (0.15)	97.88 (0.05)	83.82 (0.29)	93.65 (0.13)	28.70 (0.29)	49.70 (0.28)
		Standard FT	95.28 (0.13)	98.33 (0.04)	86.44 (0.27)	94.91 (0.12)	30.93 (0.31)	52.14 (0.29)
		MAML	95.35 (0.12)	98.50 (0.08)	86.79 (0.32)	94.72 (0.19)	30.53 (0.39)	52.21 (0.28)
		Ours	**96.91** (0.11)	**98.76** (0.04)	**89.97** (0.25)	**95.84** (0.11)	**48.02** (0.38)	**67.25** (0.29)
	ResNet50	Adaptation	81.74 (0.24)	94.08 (0.09)	65.98 (0.34)	84.14 (0.21)	27.32 (0.27)	46.67 (0.28)
		Standard FT	84.10 (0.22)	94.81 (0.09)	74.48 (0.33)	88.35 (0.19)	34.10 (0.31)	55.08 (0.29)
		MAML	82.07 (0.28)	94.12 (0.08)	75.69 (0.32)	89.30 (0.19)	35.10 (0.39)	56.51 (0.28)
		Ours	**87.61** (0.20)	**95.92** (0.07)	**77.74** (0.32)	**89.77** (0.17)	**39.09** (0.34)	**60.60** (0.29)

Table C.12: **Results of few-shot image classification.** We report average classification accuracy (%) with 95% confidence intervals on test splits. Adaptation: Direction adaptation without finetuning; Standard FT: Standard finetuning; MAML: MAML algorithm in [82]; Ours: Our multitask finetuning; 1-/5-shot: number of labeled images per class in the target task.

Despite these fluctuations, our multitask finetuning approach consistently surpasses the mentioned baselines, often by a significant margin, across all evaluated scenarios.

C.6 NLP Experimental Results

We first provide a summary of the experimental setting and results in the below subsection. Then we provide details in the following subsections.

C.6.1 Summary

To further validate our approach, we conducted prompt-based finetuning experiments on masked language models, following the procedure outlined in [95].

Datasets and Models. We consider a collection of 14 NLP datasets, covering 8 single-sentence and 6 sentence-pair English tasks. This collection includes tasks from the GLUE benchmark [276], as well as 7 other popular sentence classification tasks. The objective is to predict the label based on a single sentence or a sentence-pair. Specifically, the goal is to predict sentiments for single sentences or to estimate the relationship between sentence pairs. Each of the datasets is split into training and test set. See details in Appendix C.6.2. We experiment with a pretrained model RoBERTa [166].

Experiment Protocols. We consider prompt-based finetuning for language models [95]. This approach turns a prediction task into a masked language modeling problem, where the model generates a text response to a given task-specific prompt as the label. Our experiment protocol follows [95]. The experiments are divided into 14 parallel experiments, each corresponding to a dataset. For the few-shot experiment, we use test split data as the target task data and sample 16 examples per class from the train split as finetuning data. The evaluation metric is measured by prompt-based prediction accuracy.

During the testing stage, we conduct experiments in zero-shot and few-shot settings for a given dataset. In the zero-shot setting, we directly evaluate the model's prompt-based prediction accuracy. In the few-shot setting, we finetune the model using support samples from the same dataset and assess its accuracy on the test split. For multitask finetuning, we select support samples from other datasets and construct tasks for prompt-based finetuning.

	SST-2 (acc)	**SST-5** (acc)	**MR** (acc)	**CR** (acc)	**MPQA** (acc)	**Subj** (acc)	**TREC** (acc)	**CoLA** (Matt.)
Prompt-based zero-shot	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
Multitask FT zero-shot	**92.9**	37.2	86.5	88.8	73.9	55.3	36.8	-0.065
Prompt-based FT†	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	**91.2** (1.1)	84.8 (5.1)	**9.3** (7.3)
Multitask Prompt-based FT	92.0 (1.2)	**48.5** (1.2)	86.9 (2.2)	90.5 (1.3)	**86.0** (1.6)	89.9 (2.9)	83.6 (4.4)	5.1 (3.8)
+ task selection	92.6 (0.5)	47.1 (2.3)	**87.2** (1.6)	**91.6** (0.9)	85.2 (1.0)	90.7 (1.6)	**87.6** (3.5)	3.8 (3.2)

	MNLI (acc)	**MNLI-mm** (acc)	**SNLI** (acc)	**QNLI** (acc)	**RTE** (acc)	**MRPC** (F1)	**QQP** (F1)
Prompt-based zero-shot	50.8	51.7	49.5	50.8	51.3	61.9	49.7
Multitask FT zero-shot	63.2	65.7	61.8	65.8	74.0	81.6	63.4
Prompt-based FT†	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)
Multitask Prompt-based FT	70.9 (1.5)	73.4 (1.4)	**78.7** (2.0)	71.7 (2.2)	**74.0** (2.5)	**79.5** (4.8)	67.9 (1.6)
+ task selection	**73.5** (1.6)	**75.8** (1.5)	77.4 (1.6)	**72.0** (1.6)	70.0 (1.6)	76.0 (6.8)	**69.8** (1.7)

Table C.13: **Results of few-shot learning with NLP benchmarks.** All results are obtained using RoBERTa-large. We report mean (and standard deviation) of metrics over 5 different splits. †: Result in [95]; FT: finetuning; task selection: select multitask data from customized datasets.

We then evaluate the performance of the finetuned model on the target task. More details can be found in Appendix C.6.3.

Task Selection. We select datasets by using task selection algorithm of feature vectors, which are obtained by computing the representations of each dataset and analyzing their relationship. We first obtain text features for each data point in the dataset. We select few-shot samples for generating text features. For each example, we replace the masked word with the true label in its manual template, then we forward them through the BERT backbone. Then, we compute the first principal component to obtain a feature vector for each dataset. Dataset selection provides certain improvements on some datasets, as elaborated below. Further details can be found in Appendix C.6.4.

Results. Our results are presented in Table C.13. Again, our method outperforms direct adaptation on target tasks across most datasets. For zero-shot prediction, our method provides improvements on all datasets except CoLA. Our multitask finetuning approach results in performance improvements on 12 out of 15 target tasks for few-shot prediction, with the exceptions being SST-2, Subj, and CoLA. CoLA is also reported by [95] as an exception that contains non-grammatical sentences that are outside of the distribution of the pretrained language model. SST-2 already achieves high accuracy in zero-shot prediction,

and our model performs best in such setting. Subj is unique in that its task is to predict whether a given sentence is subjective or objective, therefore multitasking with few-shot samples from other datasets may not provide significant improvement for this task.

C.6.2 Datasets and Models

The text dataset consisted of 8 single-sentence and 6 sentence-pair English tasks, including tasks from the GLUE benchmark [276], as well as 7 other popular sentence classification tasks (SNLI [37], SST-5 [250], MR [215], CR [127], MPQA [293], Subj [214], TREC [273]). The objective was to predict the label based on a single sentence or a sentence-pair. Specifically, for single sentences, we aimed to predict their semantics as either positive or negative, while for sentence-pairs, we aimed to predict the relationship between them. We experiment with the pretrained model `RoBERTa`. We have 14 datasets in total. We split each dataset into train and test split, see details below. We experiment with the pretrained model `RoBERTa`.

We follow [95] in their train test split. We use the original development sets of SNLI and datasets from GLUE for testing. For datasets such as MR, CR, MPQA, and Subj that require a cross-validation evaluation, we randomly select 2,000 examples for testing and exclude them from training. For SST5 and TREC, we utilize their official test sets.

To construct multitask examples from support samples, we gather support samples from all datasets except the testing dataset. For each task, we randomly select ten support samples and prompt-based finetuning the model.

C.6.3 Experiment Protocols

[95] proposed a prompt-based finetuning pipeline for moderately sized language models such as `BERT, RoBERTa`. Prompt-based prediction converts the downstream prediction task as a (masked) language modeling problem, where the model directly generates a textual response also known as a label word, to a given prompt defined by a task-specific template. As an illustration, consider the SST-2 dataset, which comprises sentences expressing positive

Task	Template	Label words
SST-2	<S_1> It was [MASK] .	positive: great, negative: terrible
SST-5	<S_1> It was [MASK] .	v.positive: great, positive: good, neutral: okay, negative: bad, v.negative: terrible
MR	<S_1> It was [MASK] .	positive: great, negative: terrible
CR	<S_1> It was [MASK] .	positive: great, negative: terrible
Subj	<S_1> This is [MASK] .	subjective: subjective, objective: objective
TREC	[MASK] : <S_1>	abbreviation: Expression, entity: Entity, description: Description human: Human, location: Location, numeric: Number
COLA	<S_1> This is [MASK] .	grammatical: correct, not_grammatical: incorrect
MNLI	<S_1> ? [MASK] , <S_2>	entailment: Yes, netural: Maybe, contradiction: No
SNLI	<S_1> ? [MASK] , <S_2>	entailment: Yes, netural: Maybe, contradiction: No
QNLI	<S_1> ? [MASK] , <S_2>	entailment: Yes, not_entailment: No
RTE	<S_1> ? [MASK] , <S_2>	entailment: Yes, not_entailment: No
MRPC	<S_1> [MASK] , <S_2>	equivalent: Yes, not_equivalent: No
QQP	<S_1> [MASK] , <S_2>	equivalent: Yes, not_equivalent: No

Table C.14: Manual templates and label words that we used in our experiments, following [95].

or negative sentiment. The binary classification task can be transformed into a masked prediction problem using the template `<S>, it was <MASK>.`, where `<S>` represents the input sentence and `<MASK>` is the label word (e.g., "great" or "terrible") that the model is supposed to predict, see full templates in Table C.14. Prompt-based finetuning updates the model with prompt-based prediction loss for a given example, such as a sentence or sentence-pair.

To conduct the few-shot experiment, we use all data from the test split as the target task data for each dataset, and sample 16 examples per class from the train split as the support samples. The experiments are divided into 14 parallel experiments, with each corresponding to one dataset. The evaluation accuracy is measured as the prompt-based prediction accuracy. We subsampled 5 different sets of few-shot examples to run replicates experiments and report average performance.

During the testing stage, for a given dataset (e.g. QNLI), we consider the entire test split as the target task and divide the experiment into zero-shot and few-shot settings. In the zero-shot setting, we directly evaluate the model by measuring the accuracy of prompt-based predictions. In the few-shot setting, we first prompt-based finetune the model with support samples from the same dataset (QNLI) and then evaluate the accuracy on the test split. This experimental protocol follows the same pipeline as described in [95].

To perform multitask finetuning for a target task on a particular dataset (e.g. QNLI), we select support samples from other datasets (e.g. SST-2, Subj, QQP, etc.) as finetuning examples. We construct tasks using these examples and apply the same prompt-based finetuning protocol to multitask finetune the model on these tasks. Finally, we evaluate the performance of the finetuned model on the target task.

C.6.4 Task Selection

The importance of the relationship between the data used in the training tasks and the target task cannot be overstated in multitask finetuning. Our theory measures this relationship through diversity and consistency statements, which require that our finetuning data are diverse enough to capture the characteristics of the test data, while still focusing on the specific regions where the test data aligns. We visualize this diversity and relationship through the feature maps of the datasets.

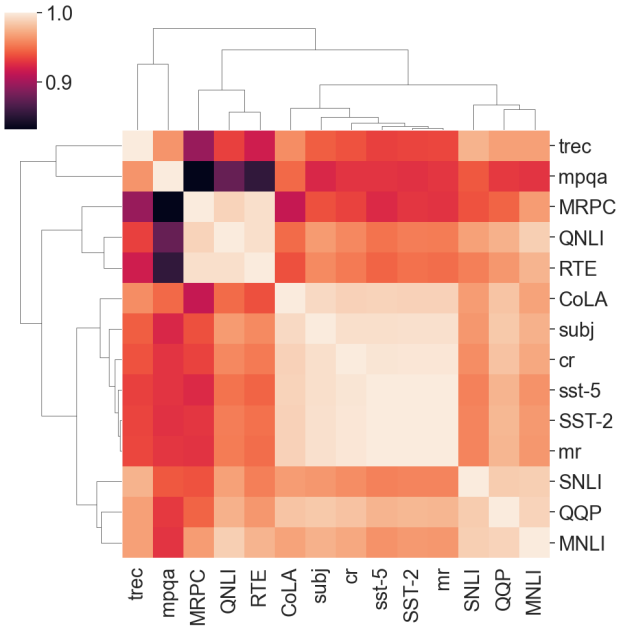


Figure C.4: Linear similarity among features vectors among 14 language datasets.

To visualize the relationship between feature vectors of different datasets, we first obtain text features for each data point in the dataset. We select few-shot samples for generating

cola: mr, cr,sst-2,sst-5,subj
sst-2: cola,mr, cr,sst-5,subj,
mrpc: qnli, rte
qqp: snli, mnli
mnli: snli, qqp
snli: qqp, mnli
qnli: mrpc, rte
rte: mrpc, qnli
mr: cola, cr,sst-2,sst-5,subj
sst-5: cola,mr, cr,sst-2,subj
subj: cola,mr, cr,sst-2,sst-5
trec: mpqa
cr: cola,mr,sst-2,sst-5,subj
mpqa: trec

Table C.15: Dataset selection.

text features. For each example, we replace the masked word with the true label in its manual template, then we forward them through the `BERT` backbone. The reason for using `BERT` over `RoBERTa` is that the latter only has masked token prediction in pretraining, the `[CLS]` in pretrained `RoBERTa` model might not contain as much sentence information as `BERT`. Then, we compute the first principal component to obtain a feature vector for each dataset. We illustrate the relationship between these feature vectors in Figure C.4.

We further perform training data selection based on the task selection algorithm among the feature vectors, the selected dataset is shown in table Table C.15.

By performing task selection, we observed further improvements in multitask prompt-based finetuning on MR, CR, TREC, MNLI, QNLI, and QQP datasets. However, it's worth noting that the CoLA dataset is an exception, as it involves predicting the grammaticality of sentences, and its inputs may include non-grammatical sentences that are outside the distribution of masked language models, as noted in [95]. Overall, our approach shows promising results for multitask learning in language tasks.

	SST-2 (acc)	**SST-5** (acc)	**MR** (acc)	**CR** (acc)	**MPQA** (acc)	**Subj** (acc)	**TREC** (acc)	**CoLA** (Matt.)
Prompt-based zero-shot	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
Multitask FT zero-shot	**92.9**	37.2	86.5	88.8	73.9	55.3	36.8	-0.065
+ task selection	92.5	34.2	87.1	88.7	71.8	72.0	36.8	0.001
Prompt-based FT†	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	**91.2** (1.1)	84.8 (5.1)	**9.3** (7.3)
Multitask Prompt-based FT	92.0 (1.2)	**48.5** (1.2)	86.9 (2.2)	90.5 (1.3)	**86.0** (1.6)	89.9 (2.9)	83.6 (4.4)	5.1 (3.8)
+ task selection	92.6 (0.5)	47.1 (2.3)	**87.2** (1.6)	**91.6** (0.9)	85.2 (1.0)	90.7 (1.6)	**87.6** (3.5)	3.8 (3.2)

	MNLI (acc)	**MNLI-mm** (acc)	**SNLI** (acc)	**QNLI** (acc)	**RTE** (acc)	**MRPC** (F1)	**QQP** (F1)
Prompt-based zero-shot	50.8	51.7	49.5	50.8	51.3	61.9	49.7
Multitask FT zero-shot	63.2	65.7	61.8	65.8	74.0	81.6	63.4
+ task selection	62.4	64.5	65.5	61.6	64.3	75.4	57.6
Prompt-based FT†	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)
Multitask Prompt-based FT	70.9 (1.5)	73.4 (1.4)	**78.7** (2.0)	71.7 (2.2)	**74.0** (2.5)	**79.5** (4.8)	67.9 (1.6)
+ task selection	**73.5** (1.6)	**75.8** (1.5)	77.4 (1.6)	**72.0** (1.6)	70.0 (1.6)	76.0 (6.8)	**69.8** (1.7)

Table C.16: **Results of few-shot learning with NLP benchmarks.** All results are obtained using RoBERTa-large. We report the mean (and standard deviation) of metrics over 5 different splits. †: Result in [95] in our paper; FT: finetuning; task selection: select multitask data from customized datasets.

Full Results with Task Selection

To complement task selection in Table C.13, we provide full results here and explain each method thoroughly.

We first elaborate on what each method did in each stage. During the testing stage, we conducted experiments in zero-shot and few-shot settings for a given dataset following [95], who applied prompt-based methods on moderately sized language models such as RoBERTa. Prompt-based finetuning method updates the model with prompt-based prediction loss for a given example. The given example can either be from a testing dataset or other datasets.

Table C.16 shows our multitask finetuning and task selection provide helps on target tasks, as detailed in Appendix C.6.1. We will elaborate on what each method did in the "Multitask fintuning phase" and "Downstream phase".

In the "Multitask fintuning phase": For prompt-based zero-shot (col-1) and prompt-based FT (col-4) we do not finetune any model. For Multitask Prompt-based finetuning (col-2,3,5,6), we conduct prompt-based finetuning methods using finetuning(auxiliary) tasks. The data of tasks are from datasets other than testing datasets. For instance, consider a model designated to adapt to a dataset (say SST-2), we choose data from other datasets (mr,

cr, etc.) and combine these data together and form multiple auxiliary tasks, these tasks updated the model using prompt-based finetuning methods. In the "downstream phase" where we adapt the model: In the zero-shot setting (col-1,2,3), we directly evaluate the model's prompt-based prediction accuracy. In the few-shot setting (col-4,5,6), we finetune the model using shot samples from the same dataset (sst-2) and assess its accuracy on the test split.

Additional Results on simCSE

	SST-2 (acc)	**SST-5** (acc)	**MR** (acc)	**CR** (acc)	**MPQA** (acc)	**Subj** (acc)	**TREC** (acc)	**CoLA** (Matt.)
Prompt-based zero-shot	50.9	19.3	50	50	50	50.4	**27.2**	0
Multitask FT zero-shot	51.3	13.8	50	50	50	50.6	18.8	0
Prompt-based FT†	**51.8**(2.6)	20.5 (6.1)	**50.6** (0.8)	50.8 (1.1)	52.3 (1.9)	**55.4** (3.7)	19.8 (7.3)	0.8 (0.9)
Multitask Prompt-based FT	50.6 (0.7)	**22.1** (6.2)	50.5 (1.0)	51.5 (1.7)	**53.4** (2.7)	51.0 (1.4)	26.4 (8.5)	**0.9** (1.3)
+ task selection	51.7 (1.7)	19.7 (5.6)	**50.6** (0.8)	**51.6** (1.6)	52.3 (2.7)	54.7 (2.5)	23.2 (9.9)	0.5 (0.7)
	MNLI (acc)	**MNLI-mm** (acc)	**SNLI** (acc)	**QNLI** (acc)	**RTE** (acc)	**MRPC** (F1)	**QQP** (F1)	
Prompt-based zero-shot	35.4	35.2	33.8	50.5	47.3	1.4	1.5	
Multitask FT zero-shot	35.4	35.2	33.6	49.5	47.3	53.8	53.8	
Prompt-based FT†	32.9 (0.8)	33.0 (0.7)	33.7 (0.6)	**50.6** (1.4)	48.7 (3.7)	**79.2** (4.1)	53.5 (2.7)	
Multitask Prompt-based FT	32.5 (0.6)	32.5 (0.7)	33.5 (0.4)	**50.6** (2.4)	50.0 (2.0)	76.3 (6.5)	**54.2** (0.8)	
+ task selection	**33.2** (1.2)	**33.2** (1.1)	**35.0** (0.8)	50.3 (0.4)	**51.8** (2.0)	72.2 (10.8)	52.9 (3.0)	

Table C.17: Our main results using simCSE [97]. We report mean (and standard deviation) performance over 5 splits of few-shot examples. FT: fine-tuning; task selection: select multitask data from customized dataset.

We present our results using the same approach as described in our paper. However, we used a different pretrained loss, namely simCSE, as proposed by [97]. However, the results are not promising, The reason is simCSE is trained with a contrastive loss instead of masked language prediction, making it less suitable for prompt-based finetuning.

C.7 Vision Language Tasks

Pretrained vision-language as another type of foundation model has achieved tremendous success across various downstream tasks. These models, such as CLIP [221] and ALIGN [139], align images and text in a shared space, enabling zero-shot classification in target tasks.

Finetuning such models has resulted in state-of-the-art accuracy in several benchmarks.

Vision-language model enables the classification of images through prompting, where classification weights are calculated by a text encoder. The text encoder inputs text prompts containing class information, and outputs features aligned with features from the vision encoder in the same space.

However, standard finetuning can be affected by minor variations underperforming direct adaptation [148, 295]. Additionally, standard finetuning can be computationally expensive, as it requires training the model on a large amount of target task data.

We perform our multitask finetuning pipeline on the vision-language model and observe certain improvements. It's worth mentioning although the vision-language model is pretrained using contrastive learning, the model does not align with our framework. Vision-language model computes contrastive loss between image and text encoder, whereas our pretraining pipeline formulates the contrastive loss between the same representation function ϕ for positive and negative sample pairs. Despite the discrepancy, we provide some results below.

C.7.1 Improving Zero-shot Performance

We investigate the performance of CLIP models in a zero-shot setting, following the established protocol for our vision tasks. Each task includes 50 classes, with one query image per class. We employ text features combined with class information as the centroid to categorize query images within the 50 classes. During adaptation, we classify among randomly selected classes in the test split, which consists of 50 classes.

We experimented with our methods on tieredImageNet and DomainNet. The text template utilized in *tieredImageNet* was adapted from the CLIP documentation. In adaptation, we classify among all classes in the test split (160 classes in tieredImageNet and 100 classes in DomainNet). For text features on tieredImageNet, we use 8 templates adapted from CLIP `a photo of a {}, itap of a {}, a bad photo of the {}, a origami {}, a photo of the large {}, a {} in a video game, art of the {}, a`

`photo of the small {}`. For templates on *DomainNet*, we simply use `a photo of a {}`. In the DomainNet The text template used for this experiment is `"a photo of {}"`. We perform Locked-Text Tuning, where we fixed the text encoder and update the vision encoder alone.

Backbone	**Method**	**tieredImageNet**	**DomainNet**
ViT-B	Adaptation	84.43 (0.25)	70.93 (0.32)
	Ours	84.50 (0.25)	73.31 (0.30)
ResNet50	Adaptation	81.01 (0.28)	63.61 (0.34)
	Ours	81.02 (0.27)	65.55 (0.34)

Table C.18: Multitask finetune on zero-shot performance with CLIP model.

Table C.18 demonstrates that CLIP already exhibits a high level of zero-shot performance. This is due to the model classifying images based on text information rather than relying on another image from the same class, which enables the model to utilize more accurate information to classify among query images. We show the effectiveness of zero-shot accuracy in tieredImageNet and DomainNet. It is worth highlighting that our multitask finetuning approach enhances the model's zero-shot performance, particularly on the more realistic DomainNet dataset. We have observed that our multitask finetuning pipeline yields greater improvements for tasks on which the model has not been extensively trained.

C.7.2 Updating Text Encoder and Vision Encoder

We also investigated whether updating the text encoder will provide better performance. On the tieredImageNet dataset, We finetune the text encoder and vision encoder simultaneously using the contrastive loss, following the protocol in [109].

Method	Zero-shot	Multitask finetune
Accuracy	84.43 (0.25)	85.01 (0.76)

Table C.19: Multitask finetune on zero-shot performance with ViT-B32 backbone on tieredImageNet.

In Table C.19, we observed slightly better improvements compared to updating the

vision encoder alone. We anticipate similar performance trends across various datasets and backbone architectures. We plan to incorporate these findings into our future work.

C.7.3 CoCoOp

We also multitask finetune the vision language model following the protocol outlined in [324]. This approach involved prepending an image-specific token before the prompt to enhance prediction accuracy. To generate this token, we trained a small model on the input image. We evaluate the performance of our model on all classes in the test split, which corresponds to a 160-way classification task. This allows us to comprehensively assess the model's ability to classify among a large number of categories.

Method	Zero-shot	Multitask finetune
ViT-B32	69.9	71.4

Table C.20: Multitask finetune on zero-shot performance with ViT-B32 backbone on tieredImageNet.

Table C.20 showed the result of the performance of the CoCoOp method. We observed an improvement of 1.5% in accuracy on direct adaptation.

Appendix D

Discussions, Complete Proofs and Additional Experiments in Chapter 5: Why Larger Language Models do In-context Learning Differently

D.1 Limitations

We study and understand an interesting phenomenon of in-context learning: smaller models are more robust to noise, while larger ones are more easily distracted, leading to different ICL behaviors. Although we study two stylized settings and give the closed-form solution, our analysis cannot extend to real Transformers easily due to the high non-convex function and complicated design of multiple-layer Transformers. Also, our work does not study optimization trajectory, which we leave as future work. On the other hand, we use simple binary classification real-world datasets to verify our analysis, which still has a gap for the practical user using the LLM scenario.

D.2 Deferred Proof for Linear Regression

D.2.1 Proof of Theorem 5.2.1

Here, we provide the proof of Theorem 5.2.1.

Theorem 5.2.1 (Optimal rank-r solution for regression). Recall the loss function $\tilde{\ell}$ in Lemma 5.2.1. Let

$$\mathbf{U}^*, u^* = \operatorname*{arg\,min}_{\mathbf{U}\in\mathbb{R}^{d\times d},\operatorname{rank}(\mathbf{U})\le r, u\in\mathbb{R}} \tilde{\ell}(\mathbf{U}, u).$$

Then $\mathbf{U}^* = c\mathbf{Q}\mathbf{V}^*\mathbf{Q}^\top, u = \frac{1}{c}$, where c is any nonzero constant, and $\mathbf{V}^* = \operatorname{diag}([v_1^*, \ldots, v_d^*])$ satisfies for any $i \le r, v_i^* = \frac{N}{(N+1)\lambda_i + \operatorname{tr}(\mathbf{D})}$ and for any $i > r, v_i^* = 0$.

Proof of Theorem 5.2.1. Note that,

$$\begin{aligned}\operatorname*{arg\,min}_{\mathbf{U}\in\mathbb{R}^{d\times d},\operatorname{rank}(\mathbf{U})\le r, u\in\mathbb{R}} \tilde{\ell}(\mathbf{U}, u) &= \operatorname*{arg\,min}_{\mathbf{U}\in\mathbb{R}^{d\times d},\operatorname{rank}(\mathbf{U})\le r, u\in\mathbb{R}} \tilde{\ell}(\mathbf{U}, u) - \min_{\mathbf{U}\in\mathbb{R}^{d\times d}, u\in\mathbb{R}} \tilde{\ell}(\mathbf{U}, u) \\ &= \operatorname*{arg\,min}_{\mathbf{U}\in\mathbb{R}^{d\times d},\operatorname{rank}(\mathbf{U})\le r, u\in\mathbb{R}} \left(\tilde{\ell}(\mathbf{U}, u) - \min_{\mathbf{U}\in\mathbb{R}^{d\times d}, u\in\mathbb{R}} \tilde{\ell}(\mathbf{U}, u)\right).\end{aligned}$$

Thus, we may consider Equation (D.4) in Lemma D.2.1 only. On the other hand, we have

$$\begin{aligned}\Gamma &= \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_{d\times d} \\ &= \left(1 + \frac{1}{N}\right)\mathbf{Q}\mathbf{D}\mathbf{Q}^\top + \frac{1}{N}\operatorname{tr}(\mathbf{D})\mathbf{Q} I_{d\times d}\mathbf{Q}^\top \\ &= \mathbf{Q}\left(\left(1 + \frac{1}{N}\right)\mathbf{D} + \frac{1}{N}\operatorname{tr}(\mathbf{D}) I_{d\times d}\right)\mathbf{Q}^\top.\end{aligned}$$

We denote $\mathbf{D}' = \left(1 + \frac{1}{N}\right)\mathbf{D} + \frac{1}{N}\operatorname{tr}(\mathbf{D}) I_{d\times d}$. We can see $\Lambda^{\frac{1}{2}} = \mathbf{Q}\mathbf{D}^{\frac{1}{2}}\mathbf{Q}^\top$, $\Gamma^{\frac{1}{2}} = \mathbf{Q}\mathbf{D}'^{\frac{1}{2}}\mathbf{Q}^\top$, and $\Gamma^{-1} = \mathbf{Q}\mathbf{D}'^{-1}\mathbf{Q}^\top$. We denote $\mathbf{V} = u\mathbf{Q}^\top\mathbf{U}\mathbf{Q}$. Since Γ and Λ are commutable and the Frobenius norm (F-norm) of a matrix does not change after multiplying it by an

orthonormal matrix, we have Equation (D.4) as

$$\begin{aligned}\tilde{\ell}(\mathbf{U},u)-\min_{\mathbf{U}\in\mathbb{R}^{d\times d},u\in\mathbb{R}}\tilde{\ell}(\mathbf{U},u)=&\frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\left(u\Lambda^{\frac{1}{2}}\mathbf{U}\Lambda^{\frac{1}{2}}-\Lambda\Gamma^{-1}\right)\right\|_F^2\\=&\frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\left(u\mathbf{U}-\Gamma^{-1}\right)\Lambda^{\frac{1}{2}}\right\|_F^2\\=&\frac{1}{2}\left\|\mathbf{D}'^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\left(\mathbf{V}-\mathbf{D}'^{-1}\right)\mathbf{D}^{\frac{1}{2}}\right\|_F^2.\end{aligned}$$

As $\mathbf{W}^{KQ}$ is a matrix whose rank is at most r, we have $\mathbf{V}$ is also at most rank r. Then, we denote $\mathbf{V}^*=\arg\min_{\mathbf{V}\in\mathbb{R}^{d\times d},\mathrm{rank}(\mathbf{V})\leq r}\left\|\mathbf{D}'^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\left(\mathbf{V}-\mathbf{D}'^{-1}\right)\mathbf{D}^{\frac{1}{2}}\right\|_F^2$. We can see that $\mathbf{V}^*$ is a diagonal matrix. Denote $\mathbf{D}'=\mathrm{diag}([\lambda'_1,\ldots,\lambda'_d])$ and $\mathbf{V}^*=\mathrm{diag}([v_1^*,\ldots,v_d^*])$. Then, we have

$$\left\|\mathbf{D}'^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\left(\mathbf{V}-\mathbf{D}'^{-1}\right)\mathbf{D}^{\frac{1}{2}}\right\|_F^2 \tag{D.1}$$

$$=\sum_{i=1}^{d}\left(\lambda_i'^{\frac{1}{2}}\lambda_i\left(v_i^*-\frac{1}{\lambda'_i}\right)\right)^2 \tag{D.2}$$

$$=\sum_{i=1}^{d}\left(\left(1+\frac{1}{N}\right)\lambda_i+\frac{\mathrm{tr}(\mathbf{D})}{N}\right)\lambda_i^2\left(v_i^*-\frac{1}{\left(1+\frac{1}{N}\right)\lambda_i+\frac{\mathrm{tr}(\mathbf{D})}{N}}\right)^2. \tag{D.3}$$

As $\mathbf{V}^*$ is the minimum rank r solution, we have that $v_i^*\geq 0$ for any $i\in[d]$ and if $v_i^*>0$, we have $v_i^*=\frac{1}{\left(1+\frac{1}{N}\right)\lambda_i+\frac{\mathrm{tr}(\mathbf{D})}{N}}$. Denote $g(x)=\left(\left(1+\frac{1}{N}\right)x+\frac{\mathrm{tr}(\mathbf{D})}{N}\right)x^2\left(\frac{1}{\left(1+\frac{1}{N}\right)x+\frac{\mathrm{tr}(\mathbf{D})}{N}}\right)^2=x^2\left(\frac{1}{\left(1+\frac{1}{N}\right)x+\frac{\mathrm{tr}(\mathbf{D})}{N}}\right)$. It is easy to see that $g(x)$ is an increasing function on $[0,\infty)$. Now, we use contradiction to show that $\mathbf{V}^*$ only has non-zero entries in the first r diagonal entries. Suppose $i>r$, such that $v_i^*>0$, then we must have $j\leq r$ such that $v_j^*=0$ as $\mathbf{V}^*$ is a rank r solution. We find that if we set $v_i^*=0, v_j^*=\frac{1}{\left(1+\frac{1}{N}\right)\lambda_j+\frac{\mathrm{tr}(\mathbf{D})}{N}}$ and all other values remain the same, Equation (D.3) will strictly decrease as $g(x)$ is an increasing function on $[0,\infty)$. Thus, here is a contradiction. We finish the proof by $\mathbf{V}^*=u\mathbf{Q}^\top\mathbf{U}^*\mathbf{Q}$. □

D.2.2 Behavior Difference

Theorem 5.2.2 (Behavior difference for regression). Let $\mathbf{w}=\mathbf{Q}(\mathbf{s}+\xi)\in\mathbb{R}^d$ where $\mathbf{s},\xi\in\mathbb{R}^d$ are truncated and residual vectors defined above. The optimal rank-r solution $f_{\mathrm{LSA},\theta}$ in

Theorem 5.2.1 satisfies:

$$
\begin{aligned}
&\mathcal{L}(f_{\mathrm{LSA},\theta}; \widehat{\mathbf{E}}) \\
:=&\mathbb{E}_{\mathbf{x}_1,\epsilon_1,\ldots,\mathbf{x}_M,\epsilon_M,\mathbf{x}_q} \left(f_{\mathrm{LSA},\theta}(\widehat{\mathbf{E}}) - \langle \mathbf{w}, \mathbf{x}_q \rangle \right)^2 \\
=&\frac{1}{M} \|\mathbf{s}\|^2_{(\mathbf{V}^*)^2\mathbf{D}^3} + \frac{1}{M} \left(\|\mathbf{s} + \xi\|^2_{\mathbf{D}} + \sigma^2 \right) \mathrm{tr}\left((\mathbf{V}^*)^2 \mathbf{D}^2 \right) \\
&+ \|\xi\|^2_{\mathbf{D}} + \sum_{i \in [r]} \mathbf{s}_i^2 \lambda_i \left(\lambda_i v_i^* - 1 \right)^2 .
\end{aligned}
$$

Proof of Theorem 5.2.2. By Theorem 5.2.1, w.l.o.g, letting $c = 1$, the optimal rank-r solution $f_{\mathrm{LSA},\theta}$ satisfies $\theta = (\mathbf{W}^{PV}, \mathbf{W}^{KQ})$, and

$$
\mathbf{W}^{*PV} = \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \mathbf{W}^{*KQ} = \begin{pmatrix} \mathbf{U}^* & 0_d \\ 0_d^\top & 0 \end{pmatrix},
$$

where $\mathbf{U}^* = \mathbf{Q}\mathbf{V}^*\mathbf{Q}^\top$.

We can see that $\mathbf{U}^*$ and Λ commute. Denote $\widehat{\Lambda} := \frac{1}{M}\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top$. Note that we have

$$
\begin{aligned}
\widehat{y}_q =& f_{\mathrm{LSA},\theta}(\widehat{\mathbf{E}}) \\
=& \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix} \left(\frac{\widehat{\mathbf{E}}\widehat{\mathbf{E}}^\top}{M} \right) \begin{pmatrix} \mathbf{U}^* & 0_d \\ 0_d^\top & 0 \end{pmatrix} \mathbf{x}_q \\
=& \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\left(\mathbf{x}_q\mathbf{x}_q^\top + \sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top \right) & \frac{1}{M}\left(\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top \mathbf{w} + \sum_{i=1}^M \epsilon_i \mathbf{x}_i \right) \\ \frac{1}{M}\left(\sum_{i=1}^M \mathbf{w}^\top \mathbf{x}_i\mathbf{x}_i^\top + \sum_{i=1}^M \epsilon_i \mathbf{x}_i^\top \right) & \frac{1}{M}\sum_{i=1}^M (\mathbf{w}^\top \mathbf{x}_i + \epsilon_i)^2 \end{pmatrix} \\
& \cdot \begin{pmatrix} \mathbf{U}^* & 0_d \\ 0_d^\top & 0 \end{pmatrix} \mathbf{x}_q \\
=& \left(\mathbf{w}^\top \widehat{\Lambda} + \frac{1}{M}\sum_{i=1}^M \epsilon_i \mathbf{x}_i^\top \right) \mathbf{U}^* \mathbf{x}_q .
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x}_1,\epsilon_1,\ldots,\mathbf{x}_M,\epsilon_M,\mathbf{x}_q}\left(\widehat{y}_q - \langle \mathbf{w}, \mathbf{x}_q\rangle\right)^2 \\
=&\mathbb{E}_{\mathbf{x}_1,\epsilon_1,\ldots,\mathbf{x}_M,\epsilon_M,\mathbf{x}_q}\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q + \frac{1}{M}\sum_{i=1}^{M}\epsilon_i\mathbf{x}_i^\top\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\mathbf{x}_q\right)^2 \\
=&\underbrace{\mathbb{E}\left[\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\mathbf{x}_q\right)^2\right]}_{\text{(I)}} + \underbrace{\mathbb{E}\left[\left(\frac{1}{M}\sum_{i=1}^{M}\epsilon_i\mathbf{x}_i^\top\mathbf{U}^*\mathbf{x}_q\right)^2\right]}_{\text{(II)}},
\end{aligned}
$$

where the last equality is due to i.i.d. of ϵ_i. We see that the label noise can only have an effect in the second term. For the term (I) we have,

$$
\begin{aligned}
\text{(I)} =&\mathbb{E}\left[\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q + \mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\mathbf{x}_q\right)^2\right] \\
=&\underbrace{\mathbb{E}\left[\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q\right)^2\right]}_{\text{(III)}} + \underbrace{\mathbb{E}\left[\left(\mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q - \mathbf{w}^\top\mathbf{x}_q\right)^2\right]}_{\text{(IV)}},
\end{aligned}
$$

where the last equality is due to $\mathbb{E}[\widehat{\Lambda}] = \Lambda$ and $\widehat{\Lambda}$ is independent with $\mathbf{x}_q$. Note the fact that $\mathbf{U}^*$ and Λ commute. For the (III) term, we have

$$
\begin{aligned}
\text{(III)} =&\mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q\right)^2 + \left(\mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q\right)^2 - 2\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q\right)\left(\mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q\right)\right]\middle|\mathbf{x}_q\right] \\
=&\mathbb{E}\left[\left(\mathbf{w}^\top\widehat{\Lambda}\mathbf{U}^*\mathbf{x}_q\right)^2 - \left(\mathbf{w}^\top\Lambda\mathbf{U}^*\mathbf{x}_q\right)^2\right].
\end{aligned}
$$

By the property of trace, we have,

$$
\begin{aligned}
\text{(III)} =& \mathbb{E}\left[\operatorname{tr}\left(\widehat{\Lambda}\mathbf{w}\mathbf{w}^\top\widehat{\Lambda}(\mathbf{U}^*)^2\Lambda\right)\right] - \|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} \\
=& \mathbb{E}\left[\frac{1}{M^2}\operatorname{tr}\left(\left(\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{w}\mathbf{w}^\top\left(\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top\right)(\mathbf{U}^*)^2\Lambda\right)\right] - \|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} \\
=& \mathbb{E}\left[\frac{M-1}{M}\operatorname{tr}\left(\Lambda\mathbf{w}\mathbf{w}^\top\Lambda(\mathbf{U}^*)^2\Lambda\right) + \frac{1}{M}\operatorname{tr}\left(\mathbf{x}_1\mathbf{x}_1^\top\mathbf{w}\mathbf{w}^\top\mathbf{x}_1\mathbf{x}_1^\top(\mathbf{U}^*)^2\Lambda\right)\right] - \|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} \\
=& -\frac{1}{M}\|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} + \frac{1}{M}\mathbb{E}\left[\operatorname{tr}\left(\mathbf{x}_1\mathbf{x}_1^\top\mathbf{w}\mathbf{w}^\top\mathbf{x}_1\mathbf{x}_1^\top(\mathbf{U}^*)^2\Lambda\right)\right] \\
=& -\frac{1}{M}\|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} + \frac{1}{M}\mathbb{E}\left[\operatorname{tr}\left(\left(\|\mathbf{w}\|^2_\Lambda\Lambda + 2\Lambda\mathbf{w}^\top\mathbf{w}\Lambda\right)(\mathbf{U}^*)^2\Lambda\right)\right] \\
=& \frac{1}{M}\|\mathbf{w}\|^2_{(\mathbf{U}^*)^2\Lambda^3} + \frac{1}{M}\|\mathbf{w}\|^2_\Lambda \operatorname{tr}\left((\mathbf{U}^*)^2\Lambda^2\right),
\end{aligned}
$$

where the third last equality is by Lemma D.2.2. Furthermore, injecting $\mathbf{w} = \mathbf{Q}(\mathbf{s}+\xi)$, as $\xi^\top\mathbf{V}^*$ is a zero vector, we have

$$
\begin{aligned}
\text{(III)} &= \frac{1}{M}\|\mathbf{s}+\xi\|^2_{(\mathbf{V}^*)^2\mathbf{D}^3} + \frac{1}{M}\|\mathbf{s}+\xi\|^2_{\mathbf{D}}\operatorname{tr}\left((\mathbf{V}^*)^2\mathbf{D}^2\right) \\
&= \frac{1}{M}\|\mathbf{s}\|^2_{(\mathbf{V}^*)^2\mathbf{D}^3} + \frac{1}{M}\|\mathbf{s}+\xi\|^2_{\mathbf{D}}\operatorname{tr}\left((\mathbf{V}^*)^2\mathbf{D}^2\right).
\end{aligned}
$$

Similarly, for the term (IV), we have

$$
\begin{aligned}
\text{(IV)} =& \mathbb{E}\left[\left((\mathbf{s}+\xi)^\top\mathbf{Q}^\top\Lambda\mathbf{U}^*\mathbf{x}_q - (\mathbf{s}+\xi)^\top\mathbf{Q}^\top\mathbf{x}_q\right)^2\right] \\
=& \mathbb{E}\left[\left(\mathbf{s}^\top\mathbf{D}\mathbf{V}^*\mathbf{Q}^\top\mathbf{x}_q - \mathbf{s}^\top\mathbf{Q}^\top\mathbf{x}_q - \xi^\top\mathbf{Q}^\top\mathbf{x}_q\right)^2\right] \\
=& \mathbf{s}^\top(\mathbf{V}^*)^2\mathbf{D}^3\mathbf{s} + \mathbf{s}^\top\mathbf{D}\mathbf{s} + \xi^\top\mathbf{D}\xi - 2\mathbf{s}^\top\mathbf{V}^*\mathbf{D}^2\mathbf{s} \\
=& \xi^\top\mathbf{D}\xi + \sum_{i\in[r]}\mathbf{s}_i^2\lambda_i\left(\lambda_i^2(v_i^*)^2 - 2\lambda_i v_i^* + 1\right) \\
=& \|\xi\|^2_{\mathbf{D}} + \sum_{i\in[r]}\mathbf{s}_i^2\lambda_i\left(\lambda_i v_i^* - 1\right)^2,
\end{aligned}
$$

where the third equality is due to $\mathbf{s}^\top\mathbf{A}\xi = 0$ for any diagonal matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$.

Now, we analyze the label noise term. By $\mathbf{U}^*$ and Λ being commutable, for the term

(II), we have

$$\begin{aligned}
\text{(II)} =& \frac{\sigma^2}{M^2}\mathbb{E}\left[\left(\sum_{i=1}^{M}\mathbf{x}_i^\top\mathbf{U}^*\mathbf{x}_q\right)^2\right] \\
=& \frac{\sigma^2}{M^2}\mathbb{E}\left[\operatorname{tr}\left(\left(\sum_{i=1}^{M}\mathbf{x}_i\right)^\top\mathbf{U}^*\Lambda\mathbf{U}^*\left(\sum_{i=1}^{M}\mathbf{x}_i\right)\right)\right] \\
=& \frac{\sigma^2}{M}\mathbb{E}\left[\operatorname{tr}\left(\mathbf{x}_1^\top\mathbf{U}^*\Lambda\mathbf{U}^*\mathbf{x}_1\right)\right] \\
=& \frac{\sigma^2}{M}\operatorname{tr}\left((\mathbf{V}^*)^2\mathbf{D}^2\right),
\end{aligned}$$

where all cross terms vanish in the second equality. We conclude by combining four terms. □

Theorem 5.2.3 (Behavior difference for regression, special case). Let $0 \le r \le r' \le d$ and $\mathbf{w} = \mathbf{Q}\mathbf{s}$ where $\mathbf{s}$ is r-dim truncated vector. Denote the optimal rank-r solution as f_1 and the optimal rank-r' solution as f_2. Then,

$$\begin{aligned}
&\mathcal{L}(f_2;\widehat{\mathbf{E}}) - \mathcal{L}(f_1;\widehat{\mathbf{E}}) \\
=& \frac{1}{M}\left(\|\mathbf{s}\|_{\mathbf{D}}^2 + \sigma^2\right)\left(\sum_{i=r+1}^{r'}\left(\frac{N\lambda_i}{(N+1)\lambda_i + \operatorname{tr}(\mathbf{D})}\right)^2\right).
\end{aligned}$$

Proof of Theorem 5.2.3. Let $\mathbf{V}^* = \operatorname{diag}([v_1^*, \ldots, v_d^*])$ satisfying for any $i \le r, v_i^* = \frac{N}{(N+1)\lambda_i + \operatorname{tr}(\mathbf{D})}$ and for any $i > r, v_i^* = 0$. Let $\mathbf{V'}^* = \operatorname{diag}([v'^*_1, \ldots, v'^*_d])$ be satisfied for any $i \le r', v'^*_i = \frac{N}{(N+1)\lambda_i + \operatorname{tr}(\mathbf{D})}$ and for any $i > r', v'^*_i = 0$. Note that $\mathbf{V}^*$ is a truncated diagonal matrix of

$\mathbf{V}'^*$. By Theorem 5.2.1 and Theorem 5.2.2, we have

$$\begin{aligned}
\mathcal{L}(f_2;\widehat{\mathbf{E}})-\mathcal{L}(f_1;\widehat{\mathbf{E}}) =& \left(\frac{1}{M}\|\mathbf{s}\|^2_{(\mathbf{V}'^*)^2\mathbf{D}^3}+\frac{1}{M}\left(\|\mathbf{s}\|^2_{\mathbf{D}}+\sigma^2\right)\operatorname{tr}\left((\mathbf{V}'^*)^2\mathbf{D}^2\right)+\sum_{i\in[r']}\mathbf{s}_i^2\lambda_i\left(\lambda_i v_i'^*-1\right)^2\right)\\
&-\left(\frac{1}{M}\|\mathbf{s}\|^2_{(\mathbf{V}^*)^2\mathbf{D}^3}+\frac{1}{M}\left(\|\mathbf{s}\|^2_{\mathbf{D}}+\sigma^2\right)\operatorname{tr}\left((\mathbf{V}^*)^2\mathbf{D}^2\right)+\sum_{i\in[r]}\mathbf{s}_i^2\lambda_i\left(\lambda_i v_i^*-1\right)^2\right)\\
=&\frac{1}{M}\left(\|\mathbf{s}\|^2_{\mathbf{D}}+\sigma^2\right)\left(\operatorname{tr}\left((\mathbf{V}'^*)^2\mathbf{D}^2\right)-\operatorname{tr}\left((\mathbf{V}^*)^2\mathbf{D}^2\right)\right)\\
=&\frac{1}{M}\left(\|\mathbf{s}\|^2_{\mathbf{D}}+\sigma^2\right)\left(\sum_{i=r+1}^{r'}\left(\frac{N\lambda_i}{(N+1)\lambda_i+\operatorname{tr}(\mathbf{D})}\right)^2\right).
\end{aligned}$$

□

D.2.3 Auxiliary Lemma

Lemma D.2.1 provides the structure of the quadratic form of our MSE loss.

Lemma D.2.1 (Corollary A.2 in [315])**.** *The loss function $\tilde{\ell}$ in Lemma 5.2.1 satisfies*

$$\min_{\mathbf{U}\in\mathbb{R}^{d\times d},u\in\mathbb{R}}\tilde{\ell}(\mathbf{U},u)=-\frac{1}{2}\operatorname{tr}[\Lambda^2\Gamma^{-1}],$$

where $\mathbf{U}=c\Gamma^{-1},u=\frac{1}{c}$ for any non-zero constant c are minimum solution. We also have

$$\tilde{\ell}(\mathbf{U},u)-\min_{\mathbf{U}\in\mathbb{R}^{d\times d},u\in\mathbb{R}}\tilde{\ell}(\mathbf{U},u)=\frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\left(u\Lambda^{\frac{1}{2}}\mathbf{U}\Lambda^{\frac{1}{2}}-\Lambda\Gamma^{-1}\right)\right\|_F^2. \tag{D.4}$$

Lemma D.2.2. *Let $\mathbf{x}\sim\mathcal{N}(0,\Lambda),\epsilon\sim\mathcal{N}(0,\sigma^2)$ and $y=\langle\mathbf{w},\mathbf{x}\rangle+\epsilon$, where $\mathbf{w}\in\mathbb{R}^d$ is a fixed vector. Then we have*

$$\begin{aligned}
\mathbb{E}\left[y^2\mathbf{x}\mathbf{x}^\top\right]=&\sigma^2\Lambda+\|\mathbf{w}\|^2_\Lambda\Lambda+2\Lambda\mathbf{w}^\top\mathbf{w}\Lambda,\\
\mathbb{E}(y\mathbf{x})\mathbb{E}(y\mathbf{x})^\top=&\Lambda^\top\mathbf{w}\mathbf{w}^\top\Lambda,\\
\mathbb{E}\left[(y\mathbf{x}-\mathbb{E}(y\mathbf{x}))(y\mathbf{x}-\mathbb{E}(y\mathbf{x}))^\top\right]=&\sigma^2\Lambda+\|\mathbf{w}\|^2_\Lambda\Lambda+\Lambda\mathbf{w}^\top\mathbf{w}\Lambda.
\end{aligned}$$

Proof of Lemma D.2.2. As y is a zero mean Gaussian, by Isserlis' theorem [292, 180], for

any $i, j \in [d]$ we have

$$\begin{aligned}\mathbb{E}[y^2\mathbf{x}_i\mathbf{x}_j] =& \mathbb{E}[y^2]\mathbb{E}[\mathbf{x}_i\mathbf{x}_j] + 2\mathbb{E}[y\mathbf{x}_i]\mathbb{E}[y\mathbf{x}_j] \\ =& \left(\sigma^2 + \mathbf{w}^\top\Lambda\mathbf{w}\right)\Lambda_{i,j} + 2\Lambda_i^\top\mathbf{w}\mathbf{w}^\top\Lambda_j.\end{aligned}$$

Thus, we have $\mathbb{E}\left[y^2\mathbf{x}\mathbf{x}^\top\right] = \left(\sigma^2 + \mathbf{w}^\top\Lambda\mathbf{w}\right)\Lambda + 2\Lambda\mathbf{w}^\top\mathbf{w}\Lambda$. Similarly, we also have $\mathbb{E}(y\mathbf{x})\mathbb{E}(y\mathbf{x})^\top = \Lambda^\top\mathbf{w}\mathbf{w}^\top\Lambda$. Thus, we have

$$\begin{aligned}&\mathbb{E}\left[(y\mathbf{x} - \mathbb{E}(y\mathbf{x}))(y\mathbf{x} - \mathbb{E}(y\mathbf{x}))^\top\right] \\ =&\mathbb{E}\left[y^2\mathbf{x}\mathbf{x}^\top - y\mathbf{x}\mathbb{E}(y\mathbf{x})^\top - \mathbb{E}(y\mathbf{x})y\mathbf{x}^\top + \mathbb{E}(y\mathbf{x})\mathbb{E}(y\mathbf{x})^\top\right] \\ =&\mathbb{E}\left[y^2\mathbf{x}\mathbf{x}^\top\right] - \mathbb{E}(y\mathbf{x})\mathbb{E}(y\mathbf{x})^\top \\ =&\left(\sigma^2 + \mathbf{w}^\top\Lambda\mathbf{w}\right)\Lambda + \Lambda\mathbf{w}^\top\mathbf{w}\Lambda.\end{aligned}$$

□

D.3 Deferred Proof for Parity Classification

D.3.1 Proof of Theorem 5.3.1

Here, we provide the proof of Theorem 5.3.1.

Proof of Theorem 5.3.1. Recall $\mathbf{t}_\tau = (i_\tau, j_\tau)$. Let $\mathbf{z}_\tau \in \mathbb{R}^d$ satisfy $\mathbf{z}_{\tau,i_\tau} = \mathbf{z}_{\tau,j_\tau} = 2\gamma$ and all other entries are zero. Denote $\mathbf{V}^{(i)} = \mathbf{G}^\top\mathbf{W}^{(i)}\mathbf{G}$. Notice that $\|\mathbf{W}^{(i)}\|_F^2 = \|\mathbf{V}^{(i)}\|_F^2$. Thus,

we denote $\mathbf{V}^{*,(i)} = \mathbf{G}^\top \mathbf{W}^{*,(i)} \mathbf{G}$. Then, we have

$$
\begin{aligned}
&\mathbb{E}_\tau \left[\ell \left(y_{\tau,q} \cdot g(\mathbf{X}_\tau, \mathbf{y}_\tau, \mathbf{x}_{\tau,q}) \right) \right] \\
=&\mathbb{E}_\tau \left[\ell \left(y_{\tau,q} \left(\sum_{i\in[m]} \mathbf{a}_i \sigma \left[\frac{\mathbf{y}_\tau^\top \mathbf{X}_\tau}{N} \mathbf{W}^{(i)} \mathbf{x}_{\tau,q} \right] \right) \right) \right] \\
=&\mathbb{E}_\tau \left[\ell \left(y_{\tau,q} \left(\sum_{i\in[m]} \mathbf{a}_i \sigma \left[\mathbf{z}_\tau^\top \mathbf{V}^{(i)} \phi_{\tau,q} \right] \right) \right) \right] \\
=&\mathbb{E}_\tau \left[\ell \left(y_{\tau,q} \left(\sum_{i\in[m]} \mathbf{a}_i \sigma \left[2\gamma (\mathbf{V}^{(i)}_{i_\tau,:} + \mathbf{V}^{(i)}_{j_\tau,:}) \phi_{\tau,q} \right] \right) \right) \right].
\end{aligned}
$$

We can see that for any $i \in [m]$, $|\mathbf{a}_i^*| = 1$ and $\mathbf{V}^{*,(i)}_{j,l} = 0$ when $j \neq l$. As ReLU is a homogeneous function, we have

$$
\begin{aligned}
&\mathbb{E}_\tau \left[\ell \left(y_{\tau,q} \cdot g^*(\mathbf{X}_\tau, \mathbf{y}_\tau, \mathbf{x}_{\tau,q}) \right) \right] \\
=&\underbrace{(1-p_\mathcal{T}) \mathbb{E} \left[\ell \left(2\gamma \phi_{\tau,q,i_\tau} \phi_{\tau,q,j_\tau} \left(\sum_{i\in[m]} \mathbf{a}_i^* \sigma \left[\mathbf{V}^{*,(i)}_{i_\tau,i_\tau} \phi_{\tau,q,i_\tau} + \mathbf{V}^{*,(i)}_{j_\tau,j_\tau} \phi_{\tau,q,j_\tau} \right] \right) \right) \middle| \mathbf{t}_\tau \in S_1 \right]}_{\text{(I)}} \\
&+ \underbrace{p_\mathcal{T} \mathbb{E} \left[\ell \left(2\gamma \phi_{\tau,q,i_\tau} \phi_{\tau,q,j_\tau} \left(\sum_{i\in[m]} \mathbf{a}_i^* \sigma \left[\mathbf{V}^{*,(i)}_{i_\tau,i_\tau} \phi_{\tau,q,i_\tau} + \mathbf{V}^{*,(i)}_{j_\tau,j_\tau} \phi_{\tau,q,j_\tau} \right] \right) \right) \middle| \mathbf{t}_\tau \in S_2 \right]}_{\text{(II)}}.
\end{aligned}
$$

We have

$$\begin{aligned}(\mathrm{I}) =&(1-p_{\mathcal{T}})\cdot\Bigg\{(\frac{1}{4}+\gamma)\mathbb{E}\left[\ell\left(2\gamma\left(\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[\mathbf{V}_{i_\tau,i_\tau}^{*,(i)}+\mathbf{V}_{j_\tau,j_\tau}^{*,(i)}\right]\right)\right)\middle|\mathbf{t}_\tau\in S_1\right]\\&+\frac{1}{4}\mathbb{E}\left[\ell\left(-2\gamma\left(\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[\mathbf{V}_{i_\tau,i_\tau}^{*,(i)}-\mathbf{V}_{j_\tau,j_\tau}^{*,(i)}\right]\right)\right)\middle|\mathbf{t}_\tau\in S_1\right]\\&+(\frac{1}{4}-\gamma)\mathbb{E}\left[\ell\left(2\gamma\left(\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[-\mathbf{V}_{i_\tau,i_\tau}^{*,(i)}-\mathbf{V}_{j_\tau,j_\tau}^{*,(i)}\right]\right)\right)\middle|\mathbf{t}_\tau\in S_1\right]\\&+\frac{1}{4}\mathbb{E}\left[\ell\left(-2\gamma\left(\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[-\mathbf{V}_{i_\tau,i_\tau}^{*,(i)}+\mathbf{V}_{j_\tau,j_\tau}^{*,(i)}\right]\right)\right)\middle|\mathbf{t}_\tau\in S_1\right]\Bigg\}.\end{aligned}$$

We can get a similar equation for (II).

We make some definitions to be used. We define a pattern as $(z_1,\{(i_\tau,z_2),(j_\tau,z_3)\})$, where $z_1,z_2,z_3\in\{\pm1\}$. We define a pattern is covered by a neuron means there exists $i\in[m]$, such that $\mathbf{a}_i^*=z_1$ and $\mathrm{sign}(\mathbf{V}_{i_\tau,i_\tau}^{*,(i)})=z_2$ and $\mathrm{sign}(\mathbf{V}_{j_\tau,j_\tau}^{*,(i)})=z_3$. We define a neuron as being positive when its $\mathbf{a}_i^*=+1$ and being negative when its $\mathbf{a}_i^*=-1$. We define a pattern as being positive if $z_1=+1$ and being negative if $z_1=-1$.

Then all terms in (I) and (II) can be written as:

$$\alpha\mathbb{E}\left[\ell\left(2\gamma z_1\left(\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[z_2\mathbf{V}_{i_\tau,i_\tau}^{*,(i)}+z_3\mathbf{V}_{j_\tau,j_\tau}^{*,(i)}\right]\right)\right)\right],$$

where α is the scalar term. Note that there are total $\frac{k(k-1)}{2}\times4$ patterns in (I) and $\left(\frac{d(d-1)}{2}-\frac{k(k-1)}{2}\right)\times4$ patterns in (II). The loss depends on the weighted sum of non-covered patterns. To have zero loss, we need all patterns to be covered by m neurons, i.e., $(\mathbf{a}^*,\mathbf{V}^{*,(1)},\ldots,\mathbf{V}^{*,(m)})$.

Note that one neuron at most cover $\frac{d(d-1)}{2}$ patterns. Also, by $0<p_{\mathcal{T}}<\frac{\frac{1}{4}-\gamma}{\frac{d(d-1)}{2}(\frac{1}{4}+\gamma)+\frac{1}{4}-\gamma}$, we have

$$\frac{d(d-1)}{2}p_{\mathcal{T}}(\frac{1}{4}+\gamma)<(1-p_{\mathcal{T}})(\frac{1}{4}-\gamma),$$

which means the model will only cover all patterns in (I) before covering a pattern in (II) in purpose.

Now, we show that the minimum number of neurons to cover all patterns in (I) and (II) is $2(\nu_2+1)$.

First, we show that $2(\nu_2+1)$ neurons are enough to cover all patterns in (I) and (II). For $i \in [\nu_2]$ and $i_\tau \in [d]$, $\mathbf{V}^{(i)}_{i_\tau,i_\tau} = (2\,\text{digit}(\text{bin}(i_\tau-1),i)-1)/(4\gamma)$ and all non-diagonal entries in $\mathbf{V}^{(i)}$ being zero and $\mathbf{a}_i = -1$. For $i = \nu_2+1$ and $i_\tau \in [d]$, $\mathbf{V}^{(i)}_{i_\tau,i_\tau} = -\nu_2/(4\gamma)$ and all non-diagonal entries in $\mathbf{V}^{(i)}$ being zero and $\mathbf{a}_i = +1$. For $i \in [2(\nu_2+1)] \setminus [\nu_2+1]$, let $\mathbf{V}^{(i)} = -\mathbf{V}^{(i-\nu_2-1)}$ and $\mathbf{a}_i = \mathbf{a}_{i-\nu_2-1}$.

We can check that this construction can cover all patterns in (I) and (II) and only needs $2(\nu_2+1)$ neurons. $\mathbf{V}^{(\nu_2+1)}$ and $\mathbf{V}^{(2(\nu_2+1))}$ cover all positive patterns. All other neurons cover all negative patterns. This is because $\text{bin}(i_\tau)$ and $\text{bin}(j_\tau)$ have at least one digit difference. If $\text{bin}(i_\tau)$ and $\text{bin}(j_\tau)$ are different in the i-th digit, then $(-1,\{(i_\tau,-1),(j_\tau,+1)\})$ and $(-1,\{(i_\tau,+1),(j_\tau,-1)\})$ are covered by the i-th and $i+\nu_2+1$-th neuron.

We can also check that the scalar $\frac{1}{4\gamma}$ and $\frac{\nu_2}{4\gamma}$ is the optimal value. Note that

(1) For any negative patterns, the positive neurons will not have a cancellation effect on the negative neurons, i.e., when $y_q = -1$, the positive neurons will never activate.

(2) For each negative neuron, there exist some patterns that are uniquely covered by it.

(3) For any positive patterns, there are at most $\nu_2 - 1$ negative neurons that will have a cancellation effect on the positive neurons, i.e., when $y_q = +1$, these negative neurons will activate simultaneously. Also, we can check that there is a positive pattern such that there are $\nu_2 - 1$ negative neurons that will have a cancellation effect.

(4) For two positive neurons, there exist some patterns that are uniquely covered by one of them.

Due to hinge loss, we can see that $\frac{1}{4\gamma}$ is tight for negative neurons as (1) and (2). Similarly, we can also see that $\frac{\nu_2}{4\gamma}$ is tight for positive neurons as (3) and (4).

Second, we prove that we need at least $2(\nu_2+1)$ neurons to cover all patterns in (I) and (II). We can see that we need at least 2 positive neurons to cover all positive patterns. Then, we only need to show that $2\nu_2-1$ neurons are not enough to cover all negative patterns. We can prove that all negative patterns are covered equivalent to all numbers from $\{0,1,\ldots,2^{\nu_2}-1\}$ are encoded by $\left\{\left(\mathbf{V}_{i,i}^{(1)},\ldots,\mathbf{V}_{i,i}^{(\nu_2)}\right) \;\middle|\; i\in[k]\right\}$. Then $2\nu_2-1$ is not enough to do so.

Therefore, the minimum number of neurons to cover all patterns in (I) and (II) is $2(\nu_2+1)$.

Thus, when $m=2(\nu_1+1)$, the optimal solution will cover all patterns in (I) but not all in (II). When $m\geq 2(\nu_2+1)$, the optimal solution will cover all patterns in (I) and (II). We see that g_1^* neurons as the subset of g_2^* neurons, while the only difference is that the scalar of positive neurons is $\frac{\nu_1}{4\gamma}$ for g_1^* and $\frac{\nu_2}{4\gamma}$ for g_2^*. Thus, we finished the proof. □

D.3.2 Proof of Theorem 5.3.2

Here, we provide the proof of Theorem 5.3.2.

Proof of Theorem 5.3.2. Let $\Phi^\tau=[\phi_{\tau,1},\ldots,\phi_{\tau,M}]^\top\in\mathbb{R}^{M\times d}$. Recall $\mathbf{t}_\tau=(i_\tau,j_\tau)$. Let $\mathbf{z}_\tau\in\mathbb{R}^d$ satisfy $\mathbf{z}_{\tau,i_\tau}=\mathbf{z}_{\tau,j_\tau}=2\gamma$ and all other entries are zero. We see $\mathbf{t}_\tau$ as an index set and let $\mathbf{r}_\tau=[d]\setminus\mathbf{t}_\tau$. Then, we have

$$
\begin{aligned}
&g_2^*(\mathbf{X}_\tau,\mathbf{y}_\tau,\mathbf{x}_{\tau,q})\\
=&\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[\frac{\mathbf{y}_\tau^\top\mathbf{X}_\tau}{M}\mathbf{W}^{*,(i)}\mathbf{x}_{\tau,q}\right]\\
=&\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[\frac{\mathbf{y}_\tau^\top\Phi^\tau}{M}\mathbf{V}^{*,(i)}\phi_{\tau,q}\right]\\
=&\sum_{i\in[m]}\mathbf{a}_i^*\sigma\left[\frac{\mathbf{y}_\tau^\top\Phi^\tau_{:,\mathbf{t}_\tau}}{M}\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,:}\phi_{\tau,q,\mathbf{t}_\tau}+\frac{\mathbf{y}_\tau^\top\Phi^\tau_{:,\mathbf{r}_\tau}}{M}\mathbf{V}^{*,(i)}_{\mathbf{r}_\tau,:}\phi_{\tau,q,\mathbf{r}_\tau}\right].
\end{aligned}
$$

Note that we can absorb the randomness of $\mathbf{y}_\tau,\Phi^\tau_{:,\mathbf{r}_\tau},\phi_{\tau,q,\mathbf{r}_\tau}$ together.

Let z_i for $i\in[n]$ uniformly draw from $\{-1,+1\}$. By Chernoff bound for binomial

distribution (Lemma D.3.1), for any $0 < \epsilon < 1$, we have

$$\Pr\left(\left|\frac{\sum_{i\in[n]} z_i}{n}\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2 n}{6}\right).$$

Thus, for any $0 < \delta < 1$, with probability at least $1-\delta$ over the randomness of evaluation data, such that

$$\left|\Xi_{\mathbf{t}_\tau}^\top \operatorname{diag}(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau})\right| \leq O\left(\sqrt{\frac{1}{M}\log\frac{1}{\delta}}\right).$$

Then, for any $0 < \delta < 1$, with probability at least $1-\delta$ over the randomness of evaluation data, we have

$$\begin{aligned}
&g_2^*(\mathbf{X}_\tau, \mathbf{y}_\tau, \mathbf{x}_{\tau,q}) \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[\frac{\mathbf{y}_\tau^\top \Phi^\tau_{:,\mathbf{t}_\tau}}{M} \mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,:} \phi_{\tau,q,\mathbf{t}_\tau} + \Xi^\top \operatorname{diag}(\mathbf{V}^{*,(i)}) - \Xi_{\mathbf{t}_\tau}^\top \operatorname{diag}(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau})\right] \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[\mathbf{z}_\tau^\top \mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,:} \phi_{\tau,q,\mathbf{t}_\tau} + \Xi^\top \operatorname{diag}(\mathbf{V}^{*,(i)}) - \Xi_{\mathbf{t}_\tau}^\top \operatorname{diag}(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau})\right] \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[2\gamma \operatorname{diag}\left(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau}\right)^\top \phi_{\tau,q,\mathbf{t}_\tau} + \Xi^\top \operatorname{diag}(\mathbf{V}^{*,(i)}) - \Xi_{\mathbf{t}_\tau}^\top \operatorname{diag}(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau})\right] \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[\operatorname{diag}\left(\mathbf{V}^{*,(i)}\right)^\top \left(2\gamma\hat{\phi}_{\tau,q} + \Xi\right) - \Xi_{\mathbf{t}_\tau}^\top \operatorname{diag}(\mathbf{V}^{*,(i)}_{\mathbf{t}_\tau,\mathbf{t}_\tau})\right] \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[\operatorname{diag}\left(\mathbf{V}^{*,(i)}\right)^\top \left(2\gamma\hat{\phi}_{\tau,q} + \Xi\right) + O\left(\sqrt{\frac{1}{M}\log\frac{1}{\delta}}\right)\right] \\
&= \sum_{i\in[m]} \mathbf{a}_i^* \sigma\left[\operatorname{diag}\left(\mathbf{V}^{*,(i)}\right)^\top \left(2\gamma\hat{\phi}_{\tau,q} + P_{\mathbf{D}_2}(\Xi)\right) + O\left(\sqrt{\frac{1}{M}\log\frac{1}{\delta}}\right)\right] \\
&= h(\theta_2, 2\gamma\hat{\phi}_{\tau,q} + P_{\mathbf{D}_2}(\Xi)) + O\left(\sqrt{\frac{\nu_2}{M}\log\frac{1}{\delta}}\right).
\end{aligned}$$

Similarly, we have $g_1^*(\mathbf{X}_\tau, \mathbf{y}_\tau, \mathbf{x}_{\tau,q}) = h(\theta_1, 2\gamma\hat{\phi}_{\tau,q} + P_{\mathbf{D}_1}(\Xi)) + O\left(\sqrt{\frac{\nu_1}{M}\log\frac{1}{\delta}}\right)$.

As $\mathbf{t}_\tau \in S_1$ and the number of $(\phi_{i_\tau}, \phi_{j_\tau})$ being balanced as training, by careful checking, we can see that $\ell(y_q \cdot h(\theta_1, 2\gamma\hat{\phi}_{\tau,q})) = \ell(y_q \cdot h(\theta_2, 2\gamma\hat{\phi}_{\tau,q})) = 0$ and we have $2\gamma\hat{\phi}_{\tau,q}$ is the

signal part.

On the other hand, we know that all the first half columns in $\mathbf{D}_2$ are orthogonal with each other, and the second half columns in $\mathbf{D}_2$ are opposite to the first half columns. We have the same fact to $\mathbf{D}_1$. As Ξ is a symmetric noise distribution, we have $\frac{\mathbb{E}[\|P_{\mathbf{D}_1}(\Xi))\|_2^2]}{\mathbb{E}[\|P_{\mathbf{D}_2}(\Xi))\|_2^2]} = \frac{\nu_1+1}{\nu_2+1}$ and we have $P_{\mathbf{D}_1}(\Xi))$ and $P_{\mathbf{D}_2}(\Xi))$ is the noise part. □

D.3.3 Auxiliary Lemma

Lemma D.3.1 (Chernoff bound for binomial distribution)**.** *Let* $Z \sim \mathrm{Bin}(n, p)$ *and let* $\mu = \mathbb{E}[Z]$*. For any* $0 < \epsilon < 1$*, we have*

$$\Pr(|Z - \mu| \geq \epsilon\mu) \leq 2\exp\left(-\frac{\epsilon^2\mu}{3}\right).$$

Bibliography

[1] Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. "The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks". In: *Conference on Learning Theory*. PMLR. 2022.

[2] Emmanuel Abbe et al. "Learning to Reason with Neural Networks: Generalization, Unseen Data and Boolean Measures". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022.

[3] Sweta Agrawal et al. "In-context examples selection for machine translation". In: *arXiv preprint arXiv:2212.02437* (2022).

[4] Kwangjun Ahn et al. "Linear attention is (maybe) all you need (to understand Transformer optimization)". In: *The Twelfth International Conference on Learning Representations*. 2024.

[5] Kwangjun Ahn et al. "Transformers learn to implement preconditioned gradient descent for in-context learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[6] Shunta Akiyama and Taiji Suzuki. "Excess Risk of Two-Layer ReLU Neural Networks in Teacher-Student Settings and its Superiority to Kernel Methods". In: *The Eleventh International Conference on Learning Representations*. 2023.

[7] Shunta Akiyama and Taiji Suzuki. "On learnability via gradient method for two-layer relu neural networks in teacher-student setting". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 152–162.

[8] Ekin Akyurek et al. "What learning algorithm is in-context learning? Investigations with linear models". In: *The Eleventh International Conference on Learning Representations*. 2023.

[9] Zeyuan Allen-Zhu and Yuanzhi Li. "Backward feature correction: How deep learning performs deep learning". In: *arXiv preprint arXiv:2001.04413* (2020).

[10] Zeyuan Allen-Zhu and Yuanzhi Li. "Feature purification: How adversarial training performs robust deep learning". In: *arXiv preprint arXiv:2005.10190* (2020).

[11] Zeyuan Allen-Zhu and Yuanzhi Li. "Physics of Language Models: Part 1, Context-Free Grammar". In: *arXiv preprint arXiv:2305.13673* (2023).

[12] Zeyuan Allen-Zhu and Yuanzhi Li. "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning". In: *arXiv preprint arXiv:2012.09816* (2020).

[13] Zeyuan Allen-Zhu and Yuanzhi Li. "What Can ResNet Learn Efficiently, Going Beyond Kernels?" In: *Advances in Neural Information Processing Systems*. 2019.

[14] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. "Learning and generalization in overparameterized neural networks, going beyond two layers". In: *Advances in neural information processing systems*. 2019.

[15] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. "A Convergence Theory for Deep Learning via Over-Parameterization". In: *International Conference on Machine Learning*. 2019.

[16] Shengnan An et al. "How Do In-Context Examples Affect Compositional Generalization?" In: *arXiv preprint arXiv:2305.04835* (2023).

[17] Sanjeev Arora and Anirudh Goyal. "A theory for emergence of complex skills in language models". In: *arXiv preprint arXiv:2307.15936* (2023).

[18] Sanjeev Arora et al. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks". In: *International Conference on Learning Representations*. 2018.

[19] Sanjeev Arora et al. "A theoretical analysis of contrastive unsupervised representation learning". In: *36th International Conference on Machine Learning, ICML 2019*. International Machine Learning Society (IMLS). 2019.

[20] Sanjeev Arora et al. "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 322–332.

[21] Sanjeev Arora et al. "On exact computation with an infinitely wide neural net". In: *arXiv preprint arXiv:1904.11955* (2019).

[22] Jimmy Ba et al. "High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation". In: *arXiv preprint arXiv:2205.01445* (2022).

[23] Yu Bai and Jason D Lee. "Beyond Linearization: On Quadratic and Higher-Order Approximation of Wide Neural Networks". In: *International Conference on Learning Representations*. 2019.

[24] Yu Bai et al. "Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[25] Boaz Barak et al. "Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit". In: *Advances in Neural Information Processing Systems*. 2022.

[26] Boaz Barak et al. "Hidden progress in deep learning: Sgd learns parities near the computational limit". In: *arXiv preprint arXiv:2207.08799* (2022).

[27] Boaz Barak et al. "Hidden progress in deep learning: Sgd learns parities near the computational limit". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21750–21764.

[28] Peter L Bartlett et al. "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* (2020).

[29] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.

[30] Srinadh Bhojanapalli et al. “Low-rank bottleneck in multi-head attention models”. In: *International conference on machine learning*. PMLR. 2020.

[31] Alberto Bietti et al. “Birth of a Transformer: A Memory Viewpoint”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[32] Alberto Bietti et al. “Learning single-index models with shallow neural networks”. In: *Advances in Neural Information Processing Systems* (2022).

[33] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[34] Avrim Blum and Ronald L Rivest. “Training a 3-node neural network is NP-complete”. In: *Advances in neural information processing systems*. 1989, pp. 494–501.

[35] Avrim Blum et al. “Weakly learning DNF and characterizing statistical query learning using Fourier analysis”. In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. 1994, pp. 253–262.

[36] Rishi Bommasani et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[37] Samuel R. Bowman et al. “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015.

[38] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* (2020).

[39] Sébastien Bubeck et al. “Sparks of artificial general intelligence: Early experiments with gpt-4”. In: *arXiv preprint arXiv:2303.12712* (2023).

[40] Yuan Cao and Quanquan Gu. “Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks”. In: *arXiv preprint arXiv:1905.13210* (2019).

[41] Yuan Cao et al. “Benign Overfitting in Two-layer Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022.

[42] Yuan Cao et al. *Towards Understanding the Spectral Bias of Deep Learning*. 2020. arXiv: `1912.01198 [cs.LG]`.

[43] Niladri S Chatterji, Philip M Long, and Peter L Bartlett. “When Does Gradient Descent with Logistic Loss Find Interpolating Two-Layer Networks?” In: *Journal of Machine Learning Research* (2021), pp. 1–48.

[44] Beidi Chen et al. “Scatterbrain: Unifying sparse and low-rank attention”. In: *Advances in Neural Information Processing Systems* (2021).

[45] Minshuo Chen et al. “Efficient approximation of deep relu networks for functions on low dimensional manifolds”. In: *Advances in neural information processing systems* 32 (2019), pp. 8174–8184.

[46] Minshuo Chen et al. “Nonparametric regression on low-dimensional manifolds using deep ReLU networks: Function approximation and statistical recovery”. In: *arXiv preprint arXiv:1908.01842* (2019).

[47] Minshuo Chen et al. “Towards Understanding Hierarchical Learning: Benefits of Neural Representations”. In: *arXiv preprint arXiv:2006.13436* (2020).

[48] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International Conference on Machine Learning*. 2020.

[49] Xinlei Chen, Saining Xie, and Kaiming He. “An empirical study of training self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[50] Yanda Chen et al. “Meta-learning via Language Model In-context Tuning”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022.

[51] Yinbo Chen et al. “Meta-baseline: Exploring simple meta-learning for few-shot learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[52] Zhengdao Chen, Eric Vanden-Eijnden, and Joan Bruna. “On Feature Learning in Neural Networks with Global Convergence Guarantees”. In: *International Conference on Learning Representations*. 2022.

[53] Xiang Cheng, Yuxin Chen, and Suvrit Sra. “Transformers Implement Functional Gradient Descent to Learn Non-Linear Functions In Context”. In: *arXiv preprint arXiv:2312.06528* (2023).

[54] Lenaic Chizat and Francis Bach. “A note on lazy training in supervised differentiable programming”. In: *arXiv preprint arXiv:1812.07956* (2018).

[55] Lenaic Chizat and Francis Bach. “Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss”. In: *Conference on Learning Theory*. PMLR. 2020.

[56] Lenaic Chizat and Francis Bach. “On the global convergence of gradient descent for over-parameterized models using optimal transport”. In: *Advances in neural information processing systems* 31 (2018).

[57] Lenaic Chizat, Edouard Oyallon, and Francis Bach. “On Lazy Training in Differentiable Programming”. In: *Advances in Neural Information Processing Systems*. 2019.

[58] Aakanksha Chowdhery et al. “Palm: Scaling language modeling with pathways”. In: *arXiv preprint arXiv:2204.02311* (2022).

[59] Hyung Won Chung et al. “Scaling instruction-finetuned language models”. In: *arXiv preprint arXiv:2210.11416* (2022).

[60] Alexis Conneau and Douwe Kiela. “SentEval: An Evaluation Toolkit for Universal Sentence Representations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), 2018.

[61] Damai Dai et al. “Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers”. In: *arXiv preprint arXiv:2212.10559* (2022).

[62] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. “Neural networks can learn representations with gradient descent”. In: *Conference on Learning Theory*. PMLR. 2022.

[63] Amit Daniely and Eran Malach. “Learning parities with neural networks”. In: *Advances in Neural Information Processing Systems* (2020).

[64] Amit Daniely and Eran Malach. “Learning parities with neural networks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20356–20365.

[65] Amit Daniely, Nathan Srebro, and Gal Vardi. “Efficiently Learning Neural Networks: What Assumptions May Suffice?” In: *arXiv preprint arXiv:2302.07426* (2023).

[66] Amit Daniely and Gal Vardi. “Hardness of learning neural networks with natural weights”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 930–940.

[67] Jyotikrishna Dass et al. “Vitality: Unifying low-rank and sparse approximation for vision transformer acceleration with a linear taylor attention”. In: *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE. 2023.

[68] Li Deng. “The mnist database of handwritten digit images for machine learning research”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[69] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[70] Ilias Diakonikolas et al. “Approximation schemes for relu regression”. In: *Conference on Learning Theory*. 2020.

[71] Zhiyan Ding et al. “Overparameterization of deep ResNet: zero loss and mean-field analysis”. In: *The Journal of Machine Learning Research* (2022).

[72] Qingxiu Dong et al. “A survey for in-context learning”. In: *arXiv preprint arXiv:2301.00234* (2022).

[73] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.

[74] Xialiang Dou and Tengyuan Liang. “Training neural networks as learning data-adaptive kernels: Provable representation and approximation benefits”. In: *Journal of the American Statistical Association* (2020).

[75] Simon Du et al. “Gradient Descent Finds Global Minima of Deep Neural Networks”. In: *International Conference on Machine Learning*. 2019.

[76] Simon S Du et al. “Gradient Descent Provably Optimizes Over-parameterized Neural Networks”. In: *International Conference on Learning Representations*. 2018.

[77] Simon Shaolei Du et al. “Few-Shot Learning via Learning the Representation, Provably”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[78] Xinyan Fan et al. “Lighter and better: low-rank decomposed self-attention networks for next-item recommendation”. In: *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2021.

[79] Cong Fang, Hanze Dong, and Tong Zhang. “Mathematical models of overparameterized neural networks”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 683–703.

[80] Cong Fang, Hanze Dong, and Tong Zhang. *Over Parameterized Two-level Neural Networks Can Learn Near Optimal Feature Representations*. 2019. arXiv: `1910.11508 [cs.LG]`.

[81] Yu Feng and Yuhai Tu. *Phases of learning dynamics in artificial neural networks: with or without mislabeled data*. 2021. arXiv: `2101.06509 [cs.LG]`.

[82] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[83] Jonathan Frankle and Michael Carbin. “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks”. In: *International Conference on Learning Representations*. 2018.

[84] Jonathan Frankle et al. “Stabilizing the lottery ticket hypothesis”. In: *arXiv preprint arXiv:1903.01611* (2019).

[85] Spencer Frei, Yuan Cao, and Quanquan Gu. “Agnostic learning of a single neuron with gradient descent”. In: *Advances in Neural Information Processing Systems*. 2020.

[86] Spencer Frei, Yuan Cao, and Quanquan Gu. “Provable Generalization of SGD-trained Neural Networks of Any Width in the Presence of Adversarial Label Noise”. In: *arXiv preprint arXiv:2101.01152* (2021).

[87] Spencer Frei, Niladri S Chatterji, and Peter L Bartlett. “Random feature amplification: Feature learning and generalization in neural networks”. In: *arXiv preprint arXiv:2202.07626* (2022).

[88] Spencer Frei and Quanquan Gu. “Proxy convexity: A unified framework for the analysis of neural networks trained by gradient descent”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[89] Spencer Frei et al. “Benign Overfitting in Linear Classifiers and Leaky ReLU Networks from KKT Conditions for Margin Maximization”. In: *arXiv preprint arXiv:2303.01462* (2023).

[90] Spencer Frei et al. “Implicit Bias in Leaky ReLU Networks Trained on High-Dimensional Data”. In: *arXiv preprint arXiv:2210.07082* (2022).

[91] Spencer Frei et al. “The Double-Edged Sword of Implicit Bias: Generalization vs. Robustness in ReLU Networks”. In: *arXiv preprint arXiv:2303.01456* (2023).

[92] Yao Fu et al. “Improving language model negotiation with self-play and in-context learning from ai feedback”. In: *arXiv preprint arXiv:2305.10142* (2023).

[93] Tomer Galanti, András György, and Marcus Hutter. “Generalization bounds for transfer learning with pretrained classifiers”. In: *arXiv preprint arXiv:2212.12532* (2022).

[94] Peng Gao et al. “Llama-adapter v2: Parameter-efficient visual instruction model”. In: *arXiv preprint arXiv:2304.15010* (2023).

[95] Tianyu Gao, Adam Fisch, and Danqi Chen. “Making Pre-trained Language Models Better Few-shot Learners”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. 2021.

[96] Tianyu Gao, Adam Fisch, and Danqi Chen. “Making Pre-trained Language Models Better Few-shot Learners”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. 2021.

[97] Tianyu Gao, Xingcheng Yao, and Danqi Chen. “SimCSE: Simple Contrastive Learning of Sentence Embeddings”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2021.

[98] Shivam Garg et al. “What can transformers learn in-context? a case study of simple function classes”. In: *Advances in Neural Information Processing Systems* (2022).

[99] Siddhant Garg and Yingyu Liang. “Functional Regularization for Representation Learning: A Unified Theoretical Perspective”. In: *arXiv preprint arXiv:2008.02447* (2020).

[100] Weifeng Ge and Yizhou Yu. “Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[101] Mario Geiger, Leonardo Petrini, and Matthieu Wyart. “Landscape and training regimes in deep learning”. In: *Physics Reports* 924 (2021), pp. 1–18.

[102] Mario Geiger et al. “Disentangling feature and lazy training in deep neural networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.11 (2020), p. 113301.

[103] Mor Geva et al. “Transformer Feed-Forward Layers Are Key-Value Memories”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 5484–5495.

[104] Behrooz Ghorbani et al. “Limitations of Lazy Training of Two-layers Neural Networks”. In: *arXiv preprint arXiv:1906.08899* (2019).

[105] Behrooz Ghorbani et al. “When Do Neural Networks Outperform Kernel Methods?” In: *Advances in Neural Information Processing Systems*. 2020.

[106] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. “Implicit regularization of discrete gradient dynamics in linear neural networks”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[107] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Computer Vision and Pattern Recognition*. 2014.

[108] Sebastian Goldt et al. “Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup”. In: *Advances in neural information processing systems* 32 (2019).

[109] Sachin Goyal et al. “Finetune like you pretrain: Improved finetuning of zero-shot vision models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[110] Jean-Bastien Grill et al. “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in neural information processing systems* (2020).

[111] Jiuxiang Gu et al. “Conv-Basis: A New Paradigm for Efficient Attention Inference and Gradient Computation in Transformers”. In: *arXiv preprint arXiv:2405.05219* (2024).

[112] Jiuxiang Gu et al. “Exploring the Frontiers of Softmax: Provable Optimization, Applications in Diffusion Model, and Beyond”. In: *arXiv preprint arXiv:2405.03251* (2024).

[113] Jiuxiang Gu et al. “Fourier Circuits in Neural Networks: Unlocking the Potential of Large Language Models in Mathematical Reasoning and Modular Arithmetic”. In: *arXiv preprint arXiv:2402.09469* (2024).

[114] Jiuxiang Gu et al. “Tensor Attention Training: Provably Efficient Learning of Higher-order Transformers”. In: *arXiv preprint arXiv:2405.16411* (2024).

[115] Jiuxiang Gu et al. “Unraveling the Smoothness Properties of Diffusion Models: A Gaussian Mixture Perspective”. In: *arXiv preprint arXiv:2405.16418* (2024).

[116] Suriya Gunasekar et al. “Characterizing implicit bias in terms of optimization geometry”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1832–1841.

[117] Tianyu Guo et al. “How do transformers learn in-context beyond simple functions? a case study on learning with representations”. In: *The Twelfth International Conference on Learning Representations*. 2024.

[118] Boris Hanin and Mihai Nica. “Finite Depth and Width Corrections to the Neural Tangent Kernel”. In: *International Conference on Learning Representations*. 2019.

[119] Jeff Z HaoChen et al. “Provable guarantees for self-supervised deep learning with spectral contrastive loss”. In: *Advances in Neural Information Processing Systems* (2021).

[120] Kaiming He et al. “Deep residual learning for image recognition”. In: *Computer Vision and Pattern Recognition*. 2016.

[121] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[122] Kaiming He et al. “Masked autoencoders are scalable vision learners”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[123] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Computer Vision and Pattern Recognition*. 2020.

[124] Timothy Hospedales et al. “Meta-learning in neural networks: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[125] Edward J Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models”. In: *International Conference on Learning Representations*. 2022.

[126] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2022.

[127] Minqing Hu and Bing Liu. "Mining and summarizing customer reviews". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004.

[128] Shell Xu Hu et al. "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[129] Jiaoyang Huang and Horng-Tzer Yau. "Dynamics of deep neural networks and neural tangent hierarchy". In: *International conference on machine learning*. PMLR. 2020, pp. 4542–4551.

[130] Weiran Huang et al. "Towards the Generalization of Contrastive Self-Supervised Learning". In: *The Eleventh International Conference on Learning Representations*. 2023.

[131] Yu Huang, Yuan Cheng, and Yingbin Liang. "In-context convergence of transformers". In: *arXiv preprint arXiv:2310.05249* (2023).

[132] Srinivasan Iyer et al. "Opt-iml: Scaling language model instruction meta learning through the lens of generalization". In: *arXiv preprint arXiv:2212.12017* (2022).

[133] Arthur Jacot. "Implicit Bias of Large Depth Networks: a Notion of Rank for Nonlinear Functions". In: *The Eleventh International Conference on Learning Representations*. 2023.

[134] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems*. 2018.

[135] Samy Jelassi, Michael Sander, and Yuanzhi Li. "Vision transformers provably learn spatial structure". In: *Advances in Neural Information Processing Systems* (2022).

[136] Ziwei Ji and Matus Telgarsky. "Directional convergence and alignment in deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17176–17186.

[137] Ziwei Ji and Matus Telgarsky. "Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks". In: *International Conference on Learning Representations*. 2019.

[138] Ziwei Ji and Matus Telgarsky. "The implicit bias of gradient descent on nonseparable data". In: *Conference on Learning Theory*. PMLR. 2019, pp. 1772–1798.

[139] Chao Jia et al. "Scaling up visual and vision-language representation learning with noisy text supervision". In: *International Conference on Machine Learning*. PMLR. 2021.

[140] Pritish Kamath, Omar Montasser, and Nathan Srebro. "Approximate is good enough: Probabilistic variants of dimensional and margin complexity". In: *Conference on Learning Theory*. 2020.

[141] Hamed Karimi, Julie Nutini, and Mark Schmidt. "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2016, pp. 795–811.

[142] Michael Kearns. "Efficient noise-tolerant learning from statistical queries". In: *Journal of the ACM* (1998).

[143] Omar Khattab et al. "Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP". In: *arXiv preprint arXiv:2212.14024* (2022).

[144] Frederic Koehler and Andrej Risteski. "The comparative power of relu networks and polynomial kernels in the presence of sparse latent structure". In: *International Conference on Learning Representations*. 2018.

[145] Jonas Moritz Kohler and Aurelien Lucchi. "Sub-sampled cubic regularization for non-convex optimization". In: *International Conference on Machine Learning*. PMLR. 2017.

[146] Guy Kornowski, Gilad Yehudai, and Ohad Shamir. "From Tempered to Benign Overfitting in ReLU Neural Networks". In: *arXiv preprint arXiv:2305.15141* (2023).

[147] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *University of Toronto* (2012).

[148] Ananya Kumar et al. "Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[149] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* (2015).

[150] Ya Le and Xuan Yang. "Tiny imagenet visual recognition challenge". In: *CS 231N* (2015).

[151] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[152] Jaehoon Lee et al. "Deep Neural Networks as Gaussian Processes". In: *International Conference on Learning Representations*. 2018.

[153] Jaehoon Lee et al. "Finite versus infinite neural networks: an empirical study". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15156–15172.

[154] Jaehoon Lee et al. "Wide neural networks of any depth evolve as linear models under gradient descent". In: *Advances in neural information processing systems* (2019).

[155] Brian Lester, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

[156] Hongkang Li et al. "A Theoretical Understanding of Shallow Vision Transformers: Learning, Generalization, and Sample Complexity". In: *The Eleventh International Conference on Learning Representations*. 2023.

[157] Hongkang Li et al. "Transformers as Multi-Task Feature Selectors: Generalization Analysis of In-Context Learning". In: *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*. 2023.

[158] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2021.

[159] Yingcong Li et al. "Dissecting Chain-of-Thought: Compositionality through In-Context Filtering and Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[160] Yingcong Li et al. "Transformers as Algorithms: Generalization and Stability in In-context Learning". In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2023.

[161] Yuanzhi Li and Yingyu Liang. "Learning overparameterized neural networks via stochastic gradient descent on structured data". In: *Advances in Neural Information Processing Systems*. 2018.

[162] Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. "Learning Over-Parametrized Two-Layer Neural Networks beyond NTK". In: *Conference on Learning Theory*. 2020.

[163] Yuanzhi Li, Colin Wei, and Tengyu Ma. "Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks". In: *Advances in Neural Information Processing Systems* (2019).

[164] Yuchen Li, Yuanzhi Li, and Andrej Risteski. "How Do Transformers Learn Topic Structure: Towards a Mechanistic Understanding". In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2023.

[165] Chen Liu et al. "Learning a few-shot embedding model with contrastive learning". In: *Proceedings of the AAAI conference on artificial intelligence*. 2021.

[166] Yinhan Liu et al. "RoBERTa: A robustly optimized BERT pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[167] Lajanugen Logeswaran and Honglak Lee. "An efficient framework for learning sentence representations". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[168] Tao Luo et al. "Phase diagram for two-layer ReLU neural networks at infinite-width limit". In: *Journal of Machine Learning Research* (2021).

[169] Zeping Luo et al. "Understanding The Robustness of Self-supervised Learning Through Topic Modeling". In: *The Eleventh International Conference on Learning Representations*. 2023.

[170] Kaifeng Lyu and Jian Li. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks". In: *International Conference on Learning Representations*. 2019.

[171] Kaifeng Lyu et al. “Gradient descent on two-layer nets: Margin maximization and simplicity bias”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12978–12991.

[172] Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. “One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention”. In: *arXiv preprint arXiv:2307.03576* (2023).

[173] Eran Malach et al. “Quantifying the benefit of using differentiable learning over tangent kernels”. In: *arXiv preprint arXiv:2103.01210* (2021).

[174] Sadhika Malladi et al. “Fine-Tuning Language Models with Just Forward Passes”. In: *Advances in Neural Information Processing Systems* (2023).

[175] Christopher D Manning et al. “Emergent linguistic structure in artificial neural networks trained by self-supervision”. In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30046–30054.

[176] Alexander G de G Matthews et al. “Gaussian process behaviour in wide deep neural networks”. In: *International Conference on Learning Representations*. 2018.

[177] Andreas Maurer. “A vector-contraction inequality for rademacher complexities”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2016.

[178] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. “Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit”. In: *Conference on Learning Theory*. PMLR. 2019, pp. 2388–2464.

[179] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. “A mean field view of the landscape of two-layer neural networks”. In: *Proceedings of the National Academy of Sciences* (2018).

[180] JV Michalowicz et al. “An Isserlis’ theorem for mixed Gaussian variables: Application to the auto-bispectral density”. In: *Journal of Statistical Physics* (2009).

[181] Sewon Min et al. “MetaICL: Learning to Learn In Context”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022.

[182] Sewon Min et al. “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022.

[183] Swaroop Mishra et al. “Cross-Task Generalization via Natural Language Crowdsourcing Instructions”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022.

[184] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[185] Behrad Moniri et al. “A Theory of Non-Linear Feature Learning with One Gradient Step in Two-Layer Neural Networks”. In: *arXiv preprint arXiv:2310.07891* (2023).

[186] Andrea Montanari and Yiqiao Zhong. “The interpolation phase transition in neural networks: Memorization and generalization under lazy training”. In: *The Annals of Statistics* (2022).

[187] Edward Moroshko et al. “Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[188] Alireza Mousavi-Hosseini et al. “Neural Networks Efficiently Learn Low-Dimensional Representations with SGD”. In: *arXiv preprint arXiv:2209.14863* (2022).

[189] Niklas Muennighoff et al. “Crosslingual Generalization through Multitask Finetuning”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*. 2023.

[190] Fionn Murtagh. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2.5 (1991), pp. 183–197. ISSN: 0925-2312. DOI: https://doi.org/10.1016/0925-2312(91)90023-5. URL: https://www.sciencedirect.com/science/article/pii/092523129190023

[191] Shikhar Murty, Tatsunori B Hashimoto, and Christopher D Manning. “Dreca: A general task augmentation strategy for few-shot natural language inference”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021.

[192] Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. “Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 3051–3059.

[193] Mor Shpigel Nacson et al. “Convergence of gradient descent on separable data”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019.

[194] Mor Shpigel Nacson et al. “Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4683–4692.

[195] Vaishnavh Nagarajan and J Zico Kolter. “Uniform convergence may be unable to explain generalization in deep learning”. In: *Advances in Neural Information Processing Systems* (2019).

[196] Preetum Nakkiran et al. “Deep Double Descent: Where Bigger Models and More Data Hurt”. In: *International Conference on Learning Representations*. 2020.

[197] Preetum Nakkiran et al. “SGD on Neural Networks Learns Functions of Increasing Complexity”. In: *arXiv preprint arXiv:1905.11604* (2019).

[198] Yuval Netzer et al. “Reading digits in natural images with unsupervised feature learning”. In: (2011).

[199] Behnam Neyshabur. “Implicit regularization in deep learning”. In: *arXiv preprint arXiv:1709.01953* (2017).

[200] Jianmo Ni et al. “Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022.

[201] Roman Novak et al. “Bayesian convolutional neural networks with many channels are gaussian processes”. In: *International Conference on Learning Representations*. 2019.

[202] Maxwell Nye et al. "Show your work: Scratchpads for intermediate computation with language models". In: *arXiv preprint arXiv:2112.00114* (2021).

[203] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[204] B. Olshausen and D. Field. "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision Research* 37 (1997), pp. 3311–3325.

[205] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[206] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint arxiv:2303.08774* (2023).

[207] OpenAI. *Introducing ChatGPT*. https://openai.com/blog/chatgpt. Accessed: 2023-09-10. 2022.

[208] Maxime Oquab et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *arXiv:2304.07193* (2023).

[209] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* (2022).

[210] Samet Oymak and Mahdi Soltanolkotabi. "Overparameterized nonlinear learning: Gradient descent takes the shortest path?" In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4951–4960.

[211] Samet Oymak and Mahdi Soltanolkotabi. "Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks". In: *IEEE Journal on Selected Areas in Information Theory* (2020), pp. 84–105.

[212] Samet Oymak et al. "Generalization Guarantees for Neural Networks via Harnessing the Low-rank Structure of the Jacobian". In: *arXiv preprint arXiv:1906.05392* (2019).

[213] Jane Pan et al. "What In-Context Learning 'Learns' In-Context: Disentangling Task Recognition and Task Learning". In: *Findings of Association for Computational Linguistics (ACL)*. 2023.

[214] Bo Pang and Lillian Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. 2004.

[215] Bo Pang and Lillian Lee. "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 2005.

[216] Abhishek Panigrahi et al. "Trainable transformer in transformer". In: *arXiv preprint arXiv:2307.01189* (2023).

[217] Vardan Papyan, XY Han, and David L Donoho. "Prevalence of neural collapse during the terminal phase of deep learning training". In: *Proceedings of the National Academy of Sciences* (2020), pp. 24652–24663.

[218] Xingchao Peng et al. "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

[219] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. "True few-shot learning with language models". In: *Advances in neural information processing systems* (2021).

[220] Mohammadreza Pourreza and Davood Rafiei. "Din-sql: Decomposed in-context learning of text-to-sql with self-correction". In: *arXiv preprint arXiv:2304.11015* (2023).

[221] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021.

[222] Adityanarayanan Radhakrishnan et al. *Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features*. 2023. arXiv: 2212.13881 [cs.LG].

[223] Aniruddh Raghu et al. "Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML". In: *International Conference on Learning Representations*. 2020.

[224] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in Neural Information Processing Systems*. 2008.

[225] Allan Raventos et al. "Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[226] Gautam Reddy. "The mechanistic basis of data dependence and abrupt learning in an in-context classification task". In: *The Twelfth International Conference on Learning Representations*. 2024.

[227] Maria Refinetti et al. *Classifying high-dimensional Gaussian mixtures: Where kernel methods fail and neural networks succeed*. 2021. arXiv: 2102.11742 [cs.LG].

[228] Mengye Ren et al. "Meta-Learning for Semi-Supervised Few-Shot Classification". In: *International Conference on Learning Representations*. 2018.

[229] Yunwei Ren, Mo Zhou, and Rong Ge. "Depth Separation with Multilayer Mean-Field Networks". In: *The Eleventh International Conference on Learning Representations*. 2023.

[230] Nicholas Roberts et al. "Geometry-Aware Adaptation for Pretrained Models". In: *arXiv preprint arXiv:2307.12226* (2023).

[231] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* (2015).

[232] Itay Safran, Ronen Eldan, and Ohad Shamir. "Depth separations in neural networks: what is actually being separated?" In: *Conference on Learning Theory*. PMLR. 2019, pp. 2664–2666.

[233] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. 2022.

[234] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. "Linear transformers are secretly fast weight programmers". In: *International Conference on Machine Learning*. PMLR. 2021.

[235] Harshay Shah et al. "The Pitfalls of Simplicity Bias in Neural Networks". In: *NeurIPS*. 2020.

[236] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. "Failures of gradient-based deep learning". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3067–3075.

[237] Freda Shi et al. "Large language models can be easily distracted by irrelevant context". In: *International Conference on Machine Learning*. PMLR. 2023.

[238] Zhenmei Shi, Junyi Wei, and Yingyu Liang. "A THEORETICAL ANALYSIS ON FEATURE LEARNING IN NEURAL NETWORKS: EMERGENCE FROM INPUTS AND ADVANTAGE OVER FIXED FEATURES". In: *ICLR 2022-10th International Conference on Learning Representations*. 2022.

[239] Zhenmei Shi, Junyi Wei, and Yingyu Liang. "A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features". In: *International Conference on Learning Representations*. 2022.

[240] Zhenmei Shi, Junyi Wei, and Yingyu Liang. "A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features". In: *International Conference on Learning Representations*. 2022.

[241] Zhenmei Shi, Junyi Wei, and Yingyu Liang. "A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features". In: *International Conference on Learning Representations*. 2022.

[242] Zhenmei Shi, Junyi Wei, and Yingyu Liang. "Provable Guarantees for Neural Networks via Gradient Feature Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[243] Zhenmei Shi et al. "Domain Generalization via Nuclear Norm Regularization". In: *Conference on Parsimony and Learning (Proceedings Track)*. 2023.

[244] Zhenmei Shi et al. "Domain Generalization with Nuclear Norm Regularization". In: *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*. 2022.

[245] Zhenmei Shi et al. "The Trade-off between Universality and Label Efficiency of Representations from Contrastive Learning". In: *International Conference on Learning Representations*. 2023.

[246] Zhenmei Shi et al. "Why Larger Language Models Do In-context Learning Differently?" In: *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*. 2023.

[247] Zhenmei Shi et al. "Why Larger Language Models Do In-context Learning Differently?" In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=WOa96EG26M.

[248] Justin Sirignano and Konstantinos Spiliopoulos. "Mean field analysis of neural networks: A central limit theorem". In: *Stochastic Processes and their Applications* (2020), pp. 1820–1852.

[249] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* (2017).

[250] Richard Socher et al. “Recursive deep models for semantic compositionality over a sentiment treebank”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013.

[251] Haoyu Song et al. “CLIP Models are Few-Shot Learners: Empirical Studies on VQA and Visual Entailment”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022.

[252] Daniel Soudry et al. “The implicit bias of gradient descent on separable data”. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2822–2878.

[253] Dominik Stöger and Mahdi Soltanolkotabi. “Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23831–23843.

[254] Tianxiang Sun et al. “Black-box tuning for language-model-as-a-service”. In: *International Conference on Machine Learning*. PMLR. 2022.

[255] Yiyou Sun, Zhenmei Shi, and Yixuan Li. “A Graph-Theoretic Framework for Understanding Open-World Semi-Supervised Learning”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[256] Yiyou Sun et al. “When and How Does Known Class Help Discover Unknown Ones? Provable Understanding Through Spectral Analysis”. In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2023.

[257] Matus Telgarsky. “Feature selection with gradient descent on two-layer networks in low-rotation regimes”. In: *arXiv preprint arXiv:2208.02789* (2022).

[258] Yonglong Tian, Dilip Krishnan, and Phillipf Isola. “Contrastive multiview coding”. In: *European Conference on Computer Vision*. Springer. 2020.

[259] Yonglong Tian et al. “Rethinking few-shot image classification: a good embedding is all you need?” In: *European Conference on Computer Vision*. Springer. 2020.

[260] Yuandong Tian et al. *JoMA: Demystifying Multilayer Transformers via JOint Dynamics of MLP and Attention*. 2023. arXiv: 2310.00535.

[261] Yuandong Tian et al. “Scan and Snap: Understanding Training Dynamics and Token Composition in 1-layer Transformer”. In: *Advances in Neural Information Processing Systems* (2023).

[262] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. “Contrastive learning, multi-view redundancy, and linear models”. In: *Algorithmic Learning Theory*. PMLR. 2021.

[263] Hugo Touvron et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[264] Hugo Touvron et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[265] Eleni Triantafillou et al. “Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples”. In: *International Conference on Learning Representations*. 2020.

[266] Nilesh Tripuraneni, Chi Jin, and Michael Jordan. "Provable meta-learning of linear representations". In: *International Conference on Machine Learning*. PMLR. 2021.

[267] Nilesh Tripuraneni, Michael Jordan, and Chi Jin. "On the theory of transfer learning: The importance of task diversity". In: *Advances in Neural Information Processing Systems* (2020).

[268] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* (2017).

[269] Rodrigo Veiga et al. "Phase diagram of Stochastic Gradient Descent in high-dimensional two-layer neural networks". In: *arXiv preprint arXiv:2202.00293* (2022).

[270] William E Vinje and Jack L Gallant. "Sparse coding and decorrelation in primary visual cortex during natural vision". In: *Science* 287.5456 (2000), pp. 1273–1276.

[271] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* (2016).

[272] Johannes Von Oswald et al. "Transformers learn in-context by gradient descent". In: *International Conference on Machine Learning*. PMLR. 2023.

[273] Ellen M Voorhees and Dawn M Tice. "Building a question answering test collection". In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 2000.

[274] Tu Vu et al. "STraTA: Self-Training with Task Augmentation for Better Few-shot Learning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021.

[275] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2018.

[276] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2018.

[277] Tongzhou Wang and Phillip Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere". In: *International Conference on Machine Learning*. PMLR. 2020.

[278] Yaqing Wang et al. "Generalizing from a few examples: A survey on few-shot learning". In: *ACM computing surveys* (2020).

[279] Yifei Wang, Jonathan Lacotte, and Mert Pilanci. "The Hidden Convex Optimization Landscape of Two-Layer ReLU Neural Networks: an Exact Characterization of the Optimal Solutions". In: *arXiv e-prints* (2020), arXiv–2006.

[280] Yifei Wang et al. "Chaos is a Ladder: A New Theoretical Understanding of Contrastive Learning via Augmentation Overlap". In: *International Conference on Learning Representations*. 2022.

[281] Yizhong Wang et al. "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 5085–5109.

[282] Zhen Wang et al. "Multitask Prompt Tuning Enables Parameter-Efficient Transfer Learning". In: *The Eleventh International Conference on Learning Representations*. 2023.

[283] Colin Wei et al. "Regularization matters: Generalization and optimization of neural nets vs their induced kernel". In: *Advances in Neural Information Processing Systems* 32 (2019).

[284] Colin Wei et al. "Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data". In: *International Conference on Learning Representations*. 2021.

[285] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in Neural Information Processing Systems* (2022).

[286] Jason Wei et al. "Emergent abilities of large language models". In: *arXiv preprint arXiv:2206.07682* (2022).

[287] Jason Wei et al. "Finetuned Language Models are Zero-Shot Learners". In: *International Conference on Learning Representations*. 2022.

[288] Jerry Wei et al. "Larger language models do in-context learning differently". In: *arXiv preprint arXiv:2303.03846* (2023).

[289] Jerry Wei et al. "Symbol tuning improves in-context learning in language models". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

[290] Zixin Wen and Yuanzhi Li. "Toward understanding the feature learning process of self-supervised contrastive learning". In: *International Conference on Machine Learning*. PMLR. 2021.

[291] Kevin Christian Wibisono and Yixin Wang. "On the Role of Unstructured Training Data in Transformers' In-Context Learning Capabilities". In: *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*. 2023.

[292] Gian-Carlo Wick. "The evaluation of the collision matrix". In: *Physical review* (1950).

[293] Janyce Wiebe, Theresa Wilson, and Claire Cardie. "Annotating expressions of opinions and emotions in language". In: *Language resources and evaluation* (2005).

[294] Blake Woodworth et al. "Kernel and rich regimes in overparametrized models". In: *Conference on Learning Theory*. 2020.

[295] Mitchell Wortsman et al. "Robust fine-tuning of zero-shot models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[296] Jingfeng Wu et al. "How Many Pretraining Tasks Are Needed for In-Context Learning of Linear Regression?" In: *The Twelfth International Conference on Learning Representations*. 2024.

[297] Sang Michael Xie et al. "An Explanation of In-context Learning as Implicit Bayesian Inference". In: *International Conference on Learning Representations*. 2022.

[298] Sang Michael Xie et al. "Data selection for language models via importance resampling". In: *arXiv preprint arXiv:2302.03169* (2023).

[299] Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. "Do Large Language Models Have Compositional Ability? An Investigation into Limitations and Scalability". In: *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2024. URL: https://openreview.net/forum?id=4XPeF0SbJs.

[300] Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. "Do large language models have compositional ability? an investigation into limitations and scalability". In: *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2024.

[301] Zhuoyan Xu et al. "Improving Foundation Models for Few-Shot Learning via Multitask Finetuning". In: *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2023.

[302] Zhuoyan Xu et al. "Towards Few-Shot Adaptation of Foundation Models via Multitask Finetuning". In: *The Twelfth International Conference on Learning Representations*. 2024.

[303] Zhuoyan Xu et al. " Towards Few-Shot Adaptation of Foundation Models via Multitask Finetuning". In: *The Twelfth International Conference on Learning Representations*. 2024.

[304] Greg Yang. "Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation". In: *arXiv preprint arXiv:1902.04760* (2019).

[305] Greg Yang and Edward J Hu. "Feature learning in infinite-width neural networks". In: *arXiv preprint arXiv:2011.14522* (2020).

[306] Zhanyuan Yang, Jinghua Wang, and Yingying Zhu. "Few-shot classification with contrastive learning". In: *European Conference on Computer Vision*. Springer. 2022.

[307] Shunyu Yao et al. "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[308] Gilad Yehudai and Shamir Ohad. "Learning a single neuron with gradient methods". In: *Conference on Learning Theory*. 2020.

[309] Gilad Yehudai and Ohad Shamir. "On the power and limitations of random features for understanding neural networks". In: *Advances in Neural Information Processing Systems* (2019).

[310] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision*. 2014.

[311] Chiyuan Zhang, Samy Bengio, and Yoram Singer. "Are all layers created equal?" In: *arXiv preprint arXiv:1902.01996* (2019).

[312] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *International Conference on Learning Representations*. 2017.

[313] Hanlin Zhang et al. "A Study on the Calibration of In-context Learning". In: *arXiv preprint arXiv:2312.04021* (2023).

[314] Renrui Zhang et al. "Llama-adapter: Efficient fine-tuning of language models with zero-init attention". In: *arXiv preprint arXiv:2303.16199* (2023).

[315] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. "Trained Transformers Learn Linear Models In-Context". In: *arXiv preprint arXiv:2306.09927* (2023).

[316] Tianyi Zhang et al. "Revisiting few-sample BERT fine-tuning". In: *International Conference on Learning Representations*. 2020.

[317] Yulai Zhao, Jianshu Chen, and Simon Du. "Blessing of Class Diversity in Pre-training". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023.

[318] Zihao Zhao et al. "Calibrate before use: Improving few-shot performance of language models". In: *International Conference on Machine Learning*. PMLR. 2021.

[319] Huaixiu Steven Zheng et al. "Step-Back Prompting Enables Reasoning Via Abstraction in Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024.

[320] Ruiqi Zhong et al. "Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021.

[321] Chunting Zhou et al. "LIMA: Less Is More for Alignment". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[322] Hattie Zhou et al. "Teaching algorithmic reasoning via in-context learning". In: *arXiv preprint arXiv:2211.09066* (2022).

[323] Hattie Zhou et al. "What Algorithms can Transformers Learn? A Study in Length Generalization". In: *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*. 2023.

[324] Kaiyang Zhou et al. "Conditional prompt learning for vision-language models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[325] Kaiyang Zhou et al. "Learning to prompt for vision-language models". In: *International Journal of Computer Vision* (2022).

[326] Mo Zhou, Rong Ge, and Chi Jin. "A Local Convergence Theory for Mildly Over-Parameterized Two-Layer Neural Network". In: *Conference on Learning Theory*. 2021.

[327] Roland S Zimmermann et al. "Contrastive learning inverts the data generating process". In: *International Conference on Machine Learning*. PMLR. 2021.

[328] Difan Zou and Quanquan Gu. "An improved analysis of training over-parameterized deep neural networks". In: *Advances in neural information processing systems* 32 (2019).

[329] Difan Zou et al. "Gradient descent optimizes over-parameterized deep ReLU networks". In: *Machine Learning* 109.3 (2020), pp. 467–492. ISSN: 1573-0565.

[330] Difan Zou et al. "Stochastic gradient descent optimizes over-parameterized deep relu networks". In: *arXiv preprint arXiv:1811.08888* (2018).

[331] Difan Zou et al. “The Benefits of Mixup for Feature Learning”. In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2023, pp. 43423–43479.