Towards Reliable Foundation Models in the Open World

by

Yifei Ming

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2024

Date of final oral examination: 05/16/2024

The dissertation is approved by the members of the Final Oral Committee:

Yong Jae Lee, Associate Professor, Computer Sciences
Frederic Sala, Assistant Professor, Computer Sciences
Kangwook Lee, Assistant Professor, Electrical and Computer Engineering
Yixuan Li (Advisor), Assistant Professor, Computer Sciences

Acknowledgments

I am deeply grateful to everyone who contributed to the completion of my doctoral journey. This journey has been filled with both challenges and highlights, and I could not have navigated it without the support and guidance from numerous individuals and communities.

Foremost, I extend my heartfelt thanks to my advisor, Prof. Sharon Li. Her expertise, integrity, unwavering support, and rigorous academic standards have profoundly influenced not only my research but also my personal growth. Prof. Li guided me into the field of reliable machine learning as I began my PhD journey. Entering this new research area was exciting yet challenging. I am immensely thankful for her guidance during these critical years.

I also wish to express my sincere appreciation to my thesis committee members, Prof. Kangwook Lee, Prof. Yong Jae Lee, and Prof. Frederic Sala. Their insightful feedback and constructive comments have been essential in refining this thesis and broadening my perspectives.

My gratitude extends to my external collaborators and industrial mentors. The opportunity to engage with real-world industrial problems provided a fresh perspective and deepened my understanding of applications and scalable solutions. Their commitment and collaboration also created a supportive and enriching environment that were crucial for my career development.

Additionally, I am deeply thankful for the enduring support from my

parents and my partner, who have been my foundation throughout this PhD journey. Their emotional support during challenging times and their unwavering belief in my goals have been beacons of light, guiding and uplifting me through the difficult times.

Lastly, I wish to express my gratitude to the University of Wisconsin-Madison. The campus has been a safe haven, offering beautiful lake views and a diverse range of dining options that have made daily life both convenient and enjoyable.

Contents

List of Tables

List of Figures

Abstract

In recent years, the field of machine learning has undergone a paradigm shift with the rise of large models pre-trained on web-scale data. These models, collectively known as foundation models, have demonstrated remarkable performance on a wide range of downstream tasks. Despite the promising performance, inherent safety risks of these large pre-trained models can potentially manifest in downstream tasks, causing profound social impacts in safety-critical applications such as financial services, autonomous driving, and medical diagnosis. Understanding and mitigating these risks is critical for the deployment of these models in the open world, where models can encounter novel inputs with various distributional shifts. In this thesis, we investigate two representative tasks for reliable machine learning in the open world: out-of-distribution (OOD) detection and out-of-distribution generalization.

This doctoral thesis makes significant contributions to the field of reliable machine learning in the open world, with a focus on two central aspects underlying the success of modern foundation models: contrastive representation learning with hyperspherical embeddings, and learning with multiple modalities. The thesis is divided into three parts: Part I presents algorithmic foundations of reliable machine learning on the hypersphere. In this part, we present CIDER for OOD detection (Chapter 3) and HYPO for OOD generalization (Chapter 4). In Part II, we dive into the multi-modal paradigm and develop new algorithms and theories of

reliable multi-modal models with hyperspherical embeddings. In particular, we investigate the zero-shot scheme in Chapter 5 and the parameter-efficient fine-tuning scheme in Chapter 6. We conclude this part with the retrieval-augmented adaptation scheme in Chapter 7. In Part III, we move beyond the conventional vision-language models and discuss reliable multi-modal models for document understanding (Chapter 8). We also propose a novel OOD detection benchmark that tackles the effect of spurious correlation (Appendix A). We conclude this part with supplementary materials for the preceding chapters.

This thesis presents both novel practical algorithms and theoretical insights that enhance the reliability of foundation models in the open world. We hope this work will serve as a springboard, facilitating a deeper understanding and design of algorithms that adapt foundation models to a broad array of real-world applications.

Chapter 1

Introduction

Traditional machine learning approaches often operate under the closed-world assumption, where the test distribution is assumed to be the same as the training distribution (He et al., 2016; Huang et al., 2017). In contrast, when deploying machine learning models in the open world, it is important to ensure that the model remains reliable in the presence of out-of-distribution (OOD) inputs—samples that deviate from the training distribution.

A variety of distributional shifts can occur, where covariate and semantic shifts have received significant research attention in recent years, giving rise to the tasks of out-of-distribution (OOD) generalization and OOD detection, respectively.

As a motivating example, machine learning models are often utilized in identifying and filtering offensive language for content moderation on social media platforms. Imagine a scenario where a model has been trained on a dataset composed of common offensive expressions. However, as internet culture evolves, new slang emerges that conveys harmful meanings similar to those the model was trained on. This evolution represents a covariate shift: the fundamental task of identifying offensive content remains unchanged, but the specific language expressions have altered.

This gives rise to the task of OOD generalization, where the model is expected to adapt and recognize these forms as harmful.

Additionally, the model may face samples with semantic shifts, such as memes using a novel combination of images and text to convey offensive messages that the model has not been trained to recognize. We expect reliable models can detect these new meme formats as unknown, which gives rise to the task of OOD detection. Addressing both covariate and semantic shifts is crucial for machine learning models developed in the open world. The models are expected not only to adapt to new variations within previously understood categories of offensive content (covariate shift) but also to recognize and respond to entirely new categories of potentially harmful content (semantic shift).

This thesis introduces novel theoretical analysis and effective algorithms that significantly contribute to the field of OOD detection and OOD generalization. Section 1.1 provides an overview of advancements in OOD detection and Section 1.2 offers an overview of our contributions to OOD generalization. This thesis encompasses a broad spectrum of contemporary topics, covering general machine learning models in Part I (Chapters 2 to 4), and delving into recent foundation models across multiple modalities (Bommasani et al., 2021; Zhou et al., 2023) in Part II and Part III (Chapters 5 to 8).

1.1 Out-of-Distribution Detection

The task of out-of-distribution (OOD) detection is centered on addressing semantic shifts. At inference time, we desire models that are not only accurate when an input is drawn from the training distribution, but also raise alarms when encountering inputs from outside the training distribution. OOD detection can be framed as a binary classification problem, where the goal is to determine whether an input belongs to in-distribution

(ID) or not (OOD).

This thesis presents recent advances on OOD detection in the context of modern neural networks. Different from the classic anomalous detection problem in statistics, OOD detection is non-trivial for deep neural networks, which often exhibit overconfident predictions on OOD inputs (Nguyen et al., 2015).

The majority of works for OOD detection focuses on the supervised setting, where the goal is to derive a binary ID-OOD classifier along with a multi-class classification model for ID classification. An illustration of a typical workflow of OOD detection is shown in Figure 1.1. At inference time, the model will encounter a mixture of ID and OOD samples. Ideally, an open-world classifier $f(\cdot)$ is expected to achieve good performance on dual tasks: (1) ID classification which is often a multi-class classification problem; (2) OOD detection which is a binary classification problem. The OOD detector $G(\cdot)$ usually depends on a scoring function $S(\cdot)$ that indicates the likelihood of an input being ID.

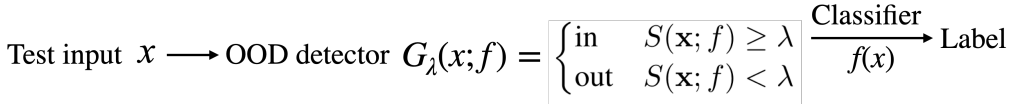


Figure 1.1: A typical workflow of OOD detection in the supervised setting. At inference time, the model will encounter a mixture of ID and OOD inputs. $f(\cdot)$ is the open-world classifier. $G(\cdot)$ denotes an OOD detector that depends on a scoring function $S(\cdot)$. λ is the threshold for classifying an input as ID.

A plethora of OOD detection algorithms have been developed in recent years, among which distance-based methods demonstrate promising performance (Lee et al., 2018; Xing et al., 2020; Tack et al., 2020; Sehwag et al., 2021; Sun et al., 2022). Distance-based methods leverage feature embeddings extracted from a pre-trained model, and operate under the assumption that the OOD samples are relatively far away from the clus-

ters of ID data. These approaches circumvent the shortcoming of alternative OOD scores that utilize the model's outputs or logits, which can be overconfident for OOD inputs.

Arguably, the efficacy of distance-based approaches can depend largely on the quality of feature embeddings. In this thesis, we will dive into one popular and successful representation learning paradigm that learns embeddings in the hyperspherical space. For example, contrastive learning (van den Oord et al., 2019; Khosla et al., 2020; Chen et al., 2020a) aims to learn a discriminative embedding where positive samples are aligned while negative ones are dispersed. The embeddings are ℓ_2-normalized which indicates that they reside on a unit hypersphere.

Recent works (Tack et al., 2020; Sehwag et al., 2021; Sun et al., 2022) directly employ off-the-shelf contrastive losses for OOD detection and demonstrate promising performance on a wide range of tasks. However, existing training objectives produce embeddings that suffice for classifying ID samples, but may remain sub-optimal in the open world. In Part I of this thesis, we will address a fundamental research question: how to exploit hyperspherical embeddings for OOD detection? We will introduce CIDER (Ming et al., 2023), a novel representation learning framework designed for open-world classification with hyperspherical embeddings.

Despite increasing attention, the vast majority of OOD detection methods are driven by single-modal learning (Hendrycks et al., 2020; Hsu et al., 2020; Jin et al., 2022; Shen et al., 2021; Xu et al., 2021a; Zhan et al., 2021; Zheng et al., 2020; Zhou et al., 2021b). For example, labels are typically encoded as one-hot vectors in image classification, leaving the semantic information encapsulated in texts largely unexploited. OOD detection relying on pure visual information can inherit the limitations, *e.g.*, when an OOD input is visually similar to in-distribution (ID) data yet semantically different from any ID class.

In Part II of this thesis, we will delve into a new landscape for OOD detection, departing from the classic single-modal toward a *multi-modal* regime. While the motivation is appealing, a core challenge remains: how to effectively utilize joint vision-language features for OOD detection? Successful distance-based OOD detection approaches in the visual domain do not directly translate into the multi-modal regime. Recent vision-language pre-training schemes such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) have emerged as promising paradigms that surpass single-modal representation learning. The main idea is to align an image with its corresponding textual description in the hyperspherical feature space. While the resulting representations are powerful, OOD detection based on such aligned multi-modal features is still in its infancy.

We will introduce a series of pioneering works (Ming et al., 2022a; Ming and Li, 2023) that explore distance-based OOD detection with multi-modal hyperspherical embeddings. We start from the zero-shot setting, where we propose a new OOD detection method that leverages the compatibility between visual features and textual features of pre-trained CLIP. By defining the textual features as the "*concept prototypes*" for each ID class, we characterize OOD uncertainty by the distance from the visual feature to the closest ID prototype. By a proper scaling of the distance, our proposed Maximum Concept Matching (MCM) score achieves strong ID-OOD separability and achieves significant performance gain compared to OOD scores that only use (single-modal) visual features on challenging benchmarks.

Subsequently, we depart from the zero-shot setting and delve into the fine-tuning scheme of vision-language models. In particular, we focus on recent developments in parameter-efficient fine-tuning methods tailored for large-scale models such as prompt learning (Zhou et al., 2022b; Khattak et al., 2023) and adaptor tuning (Zhang et al., 2022b; Udandarao et al., 2023). We will present a comprehensive study to understand how fine-

tuning impact OOD detection for few-shot downstream tasks. By framing OOD detection as multi-modal concept matching, we establish a connection between fine-tuning methods and various OOD scores. Our results suggest that a proper choice of OOD scores is essential for CLIP-based fine-tuning. In particular, the maximum concept matching (MCM) score provides a promising solution consistently. We also show that prompt learning demonstrates the state-of-the-art OOD detection performance over the zero-shot counterpart.

While the concept of OOD inputs is intuitive, the precise definition of OOD is often left in vagueness in the literature and falls short of the desired notion of OOD in reality. In Part III of this thesis, we provide a finer-grained definition of OOD that incorporates both the invariant and environmental (spurious) features (Ming et al., 2022c). We reveal insights on detection methods that are more effective in reducing the impact of spurious correlation, and provide theoretical analysis on why reliance on environmental features leads to high OOD detection error. Based on our analysis, we introduce a novel OOD detection benchmark named `Spurious OOD detection`.

Through rigorous theoretical analysis and algorithmic development, we hope the collections of works presented in this thesis will foster a deeper understanding of distance-based OOD detection and serve as a cornerstone for future advancements in open-world machine learning, particularly with multi-modal foundation models.

1.2 Out-of-Distribution Generalization

In contrast to the task of OOD detection which primarily focuses on semantic shifts, OOD generalization focuses on covariate shifts. As a concrete example, in medical diagnosis with machine learning models, the task is to analyze X-ray images to identify various conditions such as

pneumonia, fractures, or tumors. These models are often trained on in-distribution (ID) data consisting of X-ray images from a specific demographic or from certain types of equipment. However, when deployed in different geographic regions or with X-ray machines, the model may encounter inputs with covariate shifts due to differences in image quality, contrast levels, or patient demographics, which were not present in the training data. Under such circumstances, the model is expected to generalize to OOD data, accurately diagnosing conditions despite variations in the image characteristics.

A plethora of OOD generalization algorithms has been developed in recent years (see Zhou et al. (2022a) for a comprehensive survey), where a central theme is to learn domain-invariant representations—features that are consistent and meaningful across different environments[1] and can generalize to the unseen test environment. Theoretically, Ye et al. (2021) showed that the OOD generalization error can be bounded in terms of intra-class *variation* and inter-class *separation*. Intra-class variation measures the stability of representations across different environments, while inter-class separation assesses the dispersion of features among different classes. Ideally, features should display low variation and high separation, in order to generalize well to OOD data. Despite the theoretical analysis, a research question remains open in the field: How to design a practical learning algorithm that directly achieves these two properties, and what theoretical guarantees can the algorithm offer?

In Part I of the thesis, we address this question by introducing a new learning framework HYPO (**HYP**erspherical **O**OD generalization), which provably learns domain-invariant representations in the hyperspherical space with unit norm. Our key idea is to promote low variation (aligning representation across domains for every class) and high separation (separating prototypes across different classes) in the feature space. Ge-

[1] In the literature, the term environment and domain are used interchangeably.

ometrically, we show that our loss function can be understood through the lens of maximum likelihood estimation under the classic von Mises-Fisher distribution. Empirically, HYPO advances the state-of-the-art on a wide range of OOD and generalization benchmarks.

While Part I discusses algorithms and theories under the general representation learning setting, Part II focuses on the multi-modal scheme with foundation models. In particular, we consider contrastive vision-language models, which often struggle on fine-trained datasets with categories not adequately represented during pre-training. Recent works have shown promising results by utilizing samples from web-scale databases for retrieval-augmented adaptation (Udandarao et al., 2023). Despite the empirical success, understanding how retrieval impacts the adaptation of vision-language models remains an open research question. In this thesis, we adopt a reflective perspective and will present a comprehensive and systematic empirical study to understand the roles of key components in retrieval-augmented adaptation. In addition, we further present a novel theoretical framework that directly support our empirical observations. We aim to provide a better understanding on the impact of semantic shifts during multi-modal retrieval for CLIP-based adaptation.

1.3 Thesis Outline

This doctoral thesis focuses on two key research topics: algorithmic foundations of reliable machine learning on the hypersphere (Part I) and reliable multi-modal models with hyperspherical embeddings (Part II). The primary goal of this thesis is to facilitate the understanding and design of algorithms that help adapting foundation models to diverse downstream tasks in the real world. An outline of this thesis is illustrated in Figure 1.2.

Part I consists of Chapter 2, Chapter 3, and Chapter 4, which provide

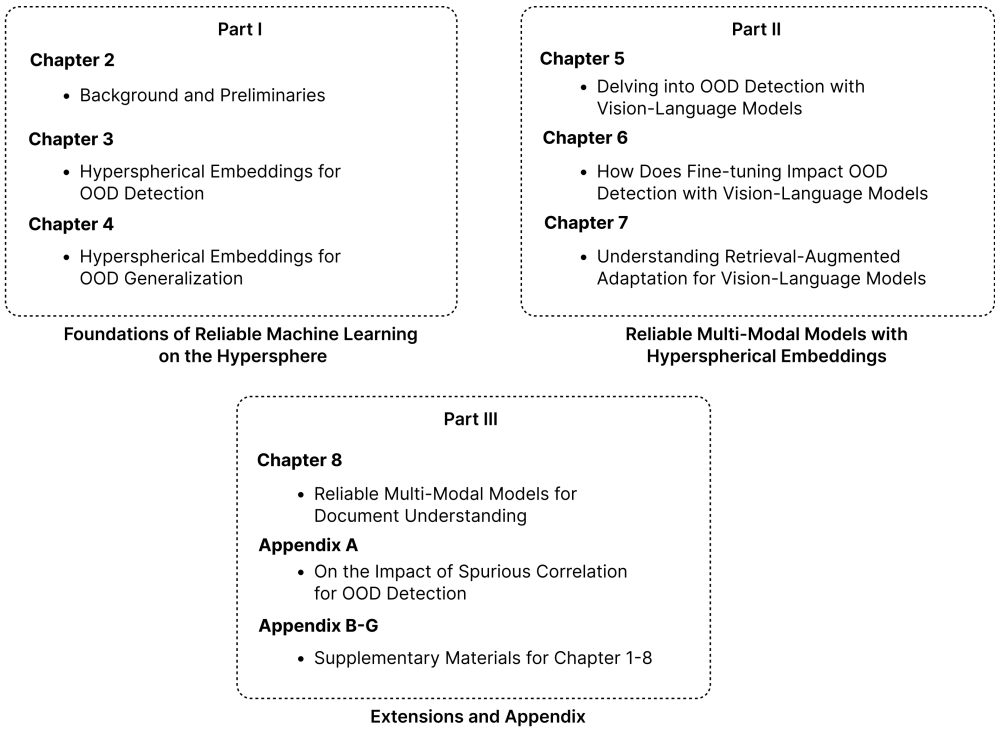


Figure 1.2: Thesis outline for Part I, II, and III.

the algorithmic foundations of reliable machine learning with a focus on hyperspherial embeddings.

Chapter 2 offers a holistic introduction on the background and preliminaries. We start with a detailed description on contrastive learning with hyperspherical embeddings, one of the most successful paradigms in modern multi-modal foundation models. Next, we formally introduce the problem setup on OOD detection and OOD generalization.

Chapter 3 introduces CIDER, a learning algorithm that exploits hyperspherical embeddings for OOD detection. In this chapter, we analyze and establish the unexplored relationship between OOD detection and the embedding properties in the hyperspherical space. Theoretically,

CIDER formalizes the latent representations as von Mises-Fisher distributions, thereby providing a theoretical interpretation of hyperspherical embeddings. Empirically, CIDER establishes superior performance on common benchmarks, outperforming the latest rival by a significant margin. The content of this chapter is primarily based on Ming et al. (2023).

Chapter 4 introduces HYPO, a learning algorithm that provably learns domain-invariant representations in the hyperspherical space. In this chapter, we introduce two guiding principles for shaping hyperspherical embeddings: intra-class variation and inter-class separation. In the ideal case, features from the same class (across different training domains) are closely aligned with their class prototypes, while different class prototypes are maximally separated. This chapter also provides theoretical justifications on how the prototypical learning objective improves the OOD generalization bound. The content of this chapter is primarily based on Ming et al. (2024).

Inspired by the recent success of vision-language pre-training, **Part II** extends the landscape of reliable machine learning from single-modal to multi-modal regimes. In this part, we explore the principles of reliable multi-modal models with hyperspherical embeddings. This part consists of Chapter 5, Chapter 6, and Chapter 7.

Chapter 5 introduces Maximum Concept Matching (MCM), a simple yet effective zero-shot OOD detection method based on aligning visual features with textual concepts. We contribute in-depth analysis and theoretical insights to understand the effectiveness of MCM. Extensive experiments demonstrate that MCM achieves superior performance on a wide variety of real-world tasks. The content of this chapter is primarily based on Ming et al. (2022a).

Chapter 6 extends Chapter 5 and aims to address an underexplored research question "How does fine-tuning impact out-of-distribution detection for Vision-Language Models?" In this chapter, we present a comprehensive study to understand how fine-tuning impact OOD detection for few-shot downstream tasks. By framing OOD detection as multi-modal concept matching, we establish a connection between fine-tuning methods and various OOD scores. Our results suggest that a proper choice of OOD scores is essential for CLIP-based fine-tuning. In particular, the maximum concept matching (MCM) score provides a promising solution consistently. The content of this chapter is primarily based on Ming and Li (2023).

Chapter 7 complements previous chapters and explores another popular paradigm where one has access to an external knowledge base for retrieval. In this chapter, we investigate an open research question: "How does retrieval impact the adaptation of vision-language models?" We adopt a reflective perspective by presenting a systematic study to understand the roles of key components in retrieval-augmented adaptation. We unveil new insights on uni-modal and cross-modal retrieval and highlight the critical role of logit ensemble for effective adaptation. We also present theoretical underpinnings that support our empirical observations.

Part III includes Appendices and supplementary materials for previous chapters. In addition, we provide extensions in Chapter 8 where we move beyond vision-language models with aligned feature space.

Chapter 8 presents a pioneering work on OOD detection for document understanding. Documents are multi-modal in nature yet no analogues of paired image-text pairs exist for documents. This chapter explores

if and how multi-modal information in documents can be exploited for OOD detection. We provide a systematic and in-depth analysis on OOD detection for document understanding models. We study the effects of model modality, pre-training, and fine-tuning across various types of OOD inputs. In particular, we find that spatial information is critical for document OOD detection. To better exploit spatial information, we propose a spatial-aware adapter, which serves as a parameter-efficient add-on module to adapt transformer-based language models to the document domain. The content of this chapter is primarily based on Gu et al. (2023).

Chapter 2

Background

In this chapter, we formally introduce the concept of representation learning, with a focus on learning hyperspherical representations. We also introduce the problem setup of *out-of-distribution detection* and *out-of-distribution generalization* as representative tasks for reliable machine learning in the open world. This chapter lays the foundation for subsequent chapters where we dive deeper into OOD detection and generalization problems in various contexts, especially with pre-trained foundation models.

2.1 Preliminaries

2.1.1 Representation Learning

Learning generalizable representations has been one of the central themes in the field of deep learning. In this work, we mainly focus on representation learning paradigms with hyperspherical embeddings.

Unit hypersphere. A hypersphere is a topological space that is homeomorphic to a standard n-sphere, which is the set of points in $(n+1)$-dimensional Euclidean space that are located at a constant distance from

the center. When the sphere has a unit radius, it is called the unit hypersphere. Formally, an n-dimensional unit-hypersphere

$$S^n := \{\mathbf{z} \in \mathbb{R}^{n+1} | \|\mathbf{z}\|_2 = 1\}$$

Geometrically, hyperspherical embeddings lie on the surface of a hypersphere.

Hyperspherical learning with a single modality. We denote the input space of the given modality (*e.g.*, image or text) as $\mathcal{X}$. In general, the model architecture consists of two components: (1) a deep neural network encoder $f : \mathcal{X} \mapsto \mathbb{R}^e$ that maps the augmented input $\mathbf{x}$ to a high dimensional feature embedding $f(\mathbf{x})$ (often referred to as the penultimate layer features); (2) a classification head that maps the high dimensional embedding $f(\mathbf{x})$ to logits for classification. The loss is typically applied to the normalized feature embedding $\mathbf{z} := f(\mathbf{x})/\|f(\mathbf{x})\|_2$. The normalized embeddings are also referred to as *hyperspherical embeddings*, since they are on a unit hypersphere.

Hyperspherical learning with multiple modalities. We introduce the pioneering framework CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021). CLIP adopts a simple dual-stream architecture with one text encoder $\mathcal{T} : t \to \mathbb{R}^d$ (*e.g.*, Transformer (Vaswani et al., 2017)) and one image encoder $\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$ (*e.g.*, ViT (Dosovitskiy et al., 2021)). CLIP is pre-trained on million text-image pairs with a simple contrastive loss that aligns the embeddings from different modalities in the latent space. The loss is applied to the normalized feature embedding for both modalities. To simplify notations, we also denote the encoder $f := \{\mathcal{I}, \mathcal{T}\}$.

2.1.2 Out-of-Distribution Detection

When deploying a machine model in the real world, a reliable classifier should not only accurately classify known in-distribution (ID) samples, but also identify as "unknown" any OOD input. This can be achieved by having an OOD detector, where OOD detection can be viewed as a binary classification problem. In the following, we first introduce the classical view of OOD detection, where models are trained from scratch. Next, we introduce OOD detection with models pre-trained on large-scale data, which are commonly employed in modern foundation models.

OOD detection with models trained from scratch. We consider multi-class classification, where $\mathcal{X}$ denotes the input space and $\mathcal{Y}^{\text{in}} = \{1, 2, ..., C\}$ denotes the ID labels. The training set $\mathcal{D}_{\text{tr}}^{\text{in}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is drawn *i.i.d.* from $P_{\mathcal{X}\mathcal{Y}^{\text{in}}}$. Let $P_{\mathcal{X}}$ denote the marginal distribution over $\mathcal{X}$, which is called the in-distribution (ID).

At test time, the goal of OOD detection is to decide whether a sample $\mathbf{x} \in \mathcal{X}$ is from $P_{\mathcal{X}}$ (ID) or not (OOD). In practice, OOD is often defined by a distribution that simulates unknowns encountered during deployment, such as samples from an irrelevant distribution whose label set has no intersection with $\mathcal{Y}^{\text{in}}$ and therefore should not be predicted by the model. Mathematically, let $\mathcal{D}_{\text{test}}^{\text{ood}}$ denote an OOD test set where the label space $\mathcal{Y}^{\text{ood}} \cap \mathcal{Y}^{\text{in}} = \emptyset$. The decision can be made via a level set estimation:

$$G_\lambda(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) = \mathbb{1}\{S(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) \geq \lambda\},$$

where samples with a higher OOD score $S(\mathbf{x})$ are classified as ID and vice versa. The threshold λ is typically chosen so that a high fraction of ID data (*e.g.* 95%) is correctly classified.

Remarks: In this formulation, $G_\lambda(\cdot)$ is the OOD detector, and $S(\cdot)$ is also known as *the scoring function*. By convention, 1 represents the positive

class (ID) and 0 indicates OOD. Therefore, OOD detection can also be written as:

$$G_\lambda(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) = \begin{cases} 1 & S(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) \geq \lambda \\ 0 & S(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) < \lambda \end{cases},$$

Alternatively, we can rewrite the above expression as follows:

$$G_\lambda(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) = \begin{cases} \text{ID} & S(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) \geq \lambda \\ \text{OOD} & S(\mathbf{x}; \mathcal{Y}_{\text{in}}, f) < \lambda \end{cases},$$

Both formulations are common in the literature. We will discuss OOD detection with hyperspherical embeddings in Chapter 3.

OOD detection with pre-trained models. Given a pre-trained model f, a (downstream) classification task is typically defined by a set of class labels/names $\mathcal{Y}_{\text{in}}$, which we refer to as the known (ID) classes. Here ID classes are defined *w.r.t.* the classification task of interest, instead of the classes used in pre-training. Accordingly, OOD is defined *w.r.t.* the ID classes, not the data distribution during pre-training. The goal of OOD detection is to (1) detect samples that do not belong to any of the known classes; (2) otherwise, assign test samples to one of the known classes. Therefore, the OOD detector can be viewed as a "safeguard" for the classification model. In particular, for pre-trained CLIP models, OOD detection can be performed based on only the names of the given classes in $\mathcal{Y}_{\text{in}}$. Different from standard supervised learning, there is no training on the ID samples involved. We denote such setting as **zero-shot OOD detection**. Further discussions are provided in Chapter 5 and Chapter 6.

2.1.3 Out-of-Distribution Generalization

Next, we introduce another important task for reliable machine learning in the open world: out-of-distribution generalization. As before, we consider multi-class classification where the input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ and the corresponding label $y \in \mathcal{Y} := \{1, 2, \cdots, C\}$. The joint distribution of X and Y is unknown and represented by $\mathbb{P}_{XY}$. The goal is to learn a predictor function, $f : \mathcal{X} \to \mathbb{R}^C$, that can accurately predict the label y for an input $\mathbf{x}$, where $(\mathbf{x}, y) \sim \mathbb{P}_{XY}$.

Unlike in standard supervised learning tasks, the out-of-distribution (OOD) generalization problem is challenged by the fact that one cannot sample directly from $\mathbb{P}_{XY}$. Instead, we can only sample (X, Y) under limited environmental conditions, each of which corrupts or varies the data differently. For example, in autonomous driving, these environmental conditions may represent different weathering conditions such as snow, rain, etc. We formalize this notion of environmental variations with a set of *environments* or domains $\mathcal{E}_{\text{all}}$. Sample pairs (X^e, Y^e) are randomly drawn from environment e. In practice, we may only have samples from a finite subset of *available environments* $\mathcal{E}_{\text{avail}} \subset \mathcal{E}_{\text{all}}$. Given $\mathcal{E}_{\text{avail}}$, the goal is to learn a predictor f that can generalize across all possible environments.

The problem can be formally described as follows. Let $\mathcal{E}_{\text{avail}} \subset \mathcal{E}_{\text{all}}$ be a set of training environments, and assume that for each environment $e \in \mathcal{E}_{\text{avail}}$, we have a dataset $\mathcal{D}^e = \{(\mathbf{x}_j^e, y_j^e)\}_{j=1}^{n_e}$, sampled i.i.d. from an unknown distribution $\mathbb{P}_{XY}^e$. The goal of OOD generalization is to find a classifier f^*, using the data from the datasets $\mathcal{D}^e$, that minimizes the worst-case risk over the entire family of environments $\mathcal{E}_{\text{all}}$:

$$\min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}_{\text{all}}} \mathbb{E}_{\mathbb{P}_{XY}^e} \ell(f(X^e), Y^e), \tag{2.1}$$

where $\mathcal{F}$ is hypothesis space and $l(\cdot, \cdot)$ is the loss function.

The problem is challenging since we do not have access to data from

domains outside $\mathcal{E}_{\text{avail}}$. In particular, the task is commonly referred to as **multi-source domain generalization** when $|\mathcal{E}_{\text{avail}}| > 1$. We will discuss OOD generalization with hyperspherical embeddings in Chapter 4.

2.2 Common Notations

In this section, we present the common notations used in the thesis. In each chapter, we will introduce additional notations when necessary.

Symbols	Descriptions
$[n]$	the set $\{1, ..., n\}$
$\mathbb{1}\{\cdot\}$	Indicator function
$\|\cdot\|_1$	l_1 norm of a matrix or a vector
$\|\cdot\|_2$	l_2 norm of a matrix or a vector
$\|\cdot\|_F$	The Frobenius norm of a matrix
$\mathbf{1}_n$	n-dimensional vector with all 1
$\mathbf{0}_n$	n-dimensional vector with all 0
I_n	Identity matrix with shape $n \times n$
$A_{(i,j)}/A_{ij}$	The value at i-th row and j-th column of a matrix A
$A_{k,(i,j)}$	The value at i-th row and j-th column of a matrix A_k
$A^\dagger$	Moore-Penrose inverse of matrix A
$\langle \mathbf{u}, \mathbf{v} \rangle$ / $\mathbf{u}^\top \mathbf{v}$	The inner product between $\mathbf{u}$ and $\mathbf{v}$
$\mathcal{N}(\mu, \sigma^2)$	Univariate Normal (Gaussian) distribution with mean μ and variance σ^2
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ
$\nabla f(\mathbf{x})$	Gradient of f at $\mathbf{x}$
$\frac{\partial f}{\partial x}$	Partial derivative of f with respect to x
S^n	n-dimensional unit-hypersphere $S^n := \{\mathbf{z} \in \mathbb{R}^{n+1} \mid \|\mathbf{z}\|_2 = 1\}$

Table 2.1: List of common notations.

Part I

Foundations of Reliable Machine Learning on the Hypersphere

Chapter 3

How to Exploit Hyperspherical Embeddings for Out-of-Distribution Detection?

Publication Statement. This chapter is a joint work with Yiyou Sun, Ousmane Dia, Yixuan Li. The paper version of this chapter appeared in ICLR 2023 (Ming et al., 2023).

Out-of-distribution (OOD) detection is a critical task for reliable machine learning. Recent advances in representation learning give rise to distance-based OOD detection, where testing samples are detected as OOD if they are relatively far away from the centroids or prototypes of in-distribution (ID) classes. However, prior methods directly take off-the-shelf contrastive losses that suffice for classifying ID samples, but are not optimally designed when test inputs contain OOD samples. In this chapter, we introduce CIDER, a novel representation learning framework that exploits hyperspherical embeddings for OOD detection. CIDER jointly optimizes two losses to promote strong ID-OOD separability: a dispersion loss that promotes large angular distances among different class prototypes, and

a compactness loss that encourages samples to be close to their class prototypes. We analyze and establish the unexplored relationship between OOD detection performance and the embedding properties in the hyperspherical space, and demonstrate the importance of dispersion and compactness. CIDER establishes superior performance, outperforming the latest rival by 13.33% in FPR95. Code is available at https://github.com/deeplearning-wisc/cider.

3.1 Introduction

When deploying machine learning models in the open world, it is important to ensure the reliability of the model in the presence of out-of-distribution (OOD) inputs—samples from an unknown distribution that the network has not been exposed to during training, and therefore should not be predicted with high confidence at test time. We desire models that are not only accurate when the input is drawn from the known distribution, but are also aware of the unknowns outside the training categories. This gives rise to the task of OOD detection, where the goal is to determine whether an input is in-distribution (ID) or not.

A plethora of OOD detection algorithms have been developed recently, among which distance-based methods demonstrated promise (Lee et al., 2018; Xing et al., 2020). These approaches circumvent the shortcoming of using the model's confidence score for OOD detection, which can be abnormally high on OOD samples (Nguyen et al., 2015) and hence not distinguishable from ID data. Distance-based methods leverage feature embeddings extracted from a model, and operate under the assumption that the test OOD samples are relatively far away from the clusters of ID data.

Arguably, the efficacy of distance-based approaches can depend largely on the quality of feature embeddings. Recent works including SSD+ (Se-

hwag et al., 2021) and KNN+ (Sun et al., 2022) directly employ off-the-shelf contrastive losses for OOD detection. In particular, these works use the supervised contrastive loss (SupCon) (Khosla et al., 2020) for learning the embeddings, which are then used for OOD detection with either parametric Mahalanobis distance (Lee et al., 2018; Sehwag et al., 2021) or non-parametric KNN distance (Sun et al., 2022). However, existing training objectives produce embeddings that suffice for classifying ID samples, but remain sub-optimal for OOD detection. For example, when trained on CIFAR-10 using SupCon loss, the average angular distance between ID and OOD data is only 29.86 degrees in the embedding space, which is too small for effective ID-OOD separation. This raises the important question:

How to exploit representation learning methods that maximally benefit OOD detection?

In this work, we propose CIDER, a **C**ompactness and **D**isp**E**rsion **R**egularized learning framework designed for OOD detection. Our method is motivated by the desirable properties of hyperspherical embeddings, which can be naturally modeled by the von Mises-Fisher (vMF) distribution. vMF is a classical and important distribution in directional statistics (Mardia et al., 2000), is analogous to spherical Gaussian distributions for features with unit norms. Our key idea is to design an end-to-end trainable loss function that enables optimizing hyperspherical embeddings into a mixture of vMF distributions, which satisfy two properties simultaneously: (**1**) each sample has a higher probability assigned to the correct class in comparison to incorrect classes, and (**2**) different classes are far apart from each other. To formalize our idea, CIDER introduces two losses: a *dispersion loss* that promotes large angular distances among different class prototypes, along with a *compactness loss* that encourages samples to be close to their class prototypes. These two terms are complementary to shape hyperspherical embeddings for both OOD detection and ID classification purposes. Unlike previous contrastive loss, CIDER explicitly

formalizes the latent representations as vMF distributions, thereby providing a direct theoretical interpretation of hyperspherical embeddings.

In particular, we show that promoting large inter-class dispersion is key to strong OOD detection performance, which has not been explored in previous literature. Previous methods including SSD+ directly use off-the-shelf SupCon loss, which produces embeddings that lack sufficient inter-class dispersion needed for OOD detection. CIDER mitigates the issue by explicitly optimizing for large inter-class margins and leads to more desirable hyperspherical embeddings. Noticeably, when trained on CIFAR-10, CIDER displays a relative 42.36% improvement of ID-OOD separability compared to SupCon. We further show that CIDER's strong representation can benefit different distance-based OOD scores, outperforming recent competitive methods SSD+ (Sehwag et al., 2021) and KNN+ (Sun et al., 2022) by a significant margin. Our key results and contributions are:

1. We propose CIDER, a novel representation learning framework designed for OOD detection. Compared to the latest rival (Sun et al., 2022), CIDER produces superior embeddings that lead to **13.33**% error reduction (in FPR95) on the challenging CIFAR-100 benchmark.

2. We are the first to establish the unexplored relationship between OOD detection performance and the embedding quality in the hyperspherical space, and provide measurements based on the notion of compactness and dispersion. This allows future research to quantify the embedding in the hyperspherical space for effective OOD detection.

3. We offer new insights on the design of representation learning for OOD detection. We also conduct extensive ablations to understand the efficacy and behavior of CIDER, which remains effective and competitive under various settings, including the ImageNet dataset.

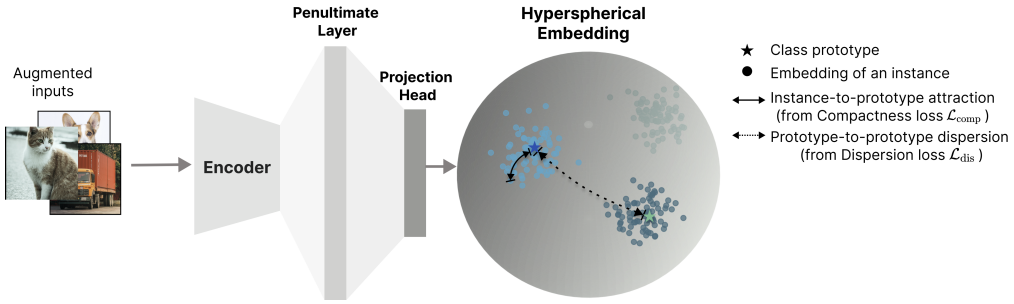


Figure 3.1: Overview of our compactness and dispersion regularized (**CIDER**) learning framework for OOD detection. We jointly optimize two complementary terms to encourage desirable properties of the embedding space: (1) a *dispersion loss* to encourage larger angular distances among different class prototypes, and (2) a *compactness loss* to encourage samples to be close to their class prototypes.

3.2 Preliminaries

We consider multi-class classification, where $\mathcal{X}$ denotes the input space and $\mathcal{Y}^{\text{in}} = \{1, 2, ..., C\}$ denotes the ID labels. The training set $\mathcal{D}_{\text{tr}}^{\text{in}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is drawn *i.i.d.* from $P_{\mathcal{X}\mathcal{Y}^{\text{in}}}$. Let $P_{\mathcal{X}}$ denote the marginal distribution over $\mathcal{X}$, which is called the in-distribution (ID).

Out-of-distribution detection. OOD detection can be viewed as a binary classification problem. At test time, the goal of OOD detection is to decide whether a sample $\mathbf{x} \in \mathcal{X}$ is from $P_{\mathcal{X}}$ (ID) or not (OOD). In practice, OOD is often defined by a distribution that simulates unknowns encountered during deployment, such as samples from an irrelevant distribution whose label set has no intersection with $\mathcal{Y}^{\text{in}}$ and therefore should not be predicted by the model. Mathematically, let $\mathcal{D}_{\text{test}}^{\text{ood}}$ denote an OOD test set where the label space $\mathcal{Y}^{\text{ood}} \cap \mathcal{Y}^{\text{in}} = \emptyset$. The decision can be made via a level set estimation: $G_\lambda(\mathbf{x}) = \mathbb{1}\{S(\mathbf{x}) \geq \lambda\}$, where samples with higher scores $S(\mathbf{x})$ are classified as ID and vice versa. The threshold λ is typically chosen so that a high fraction of ID data (*e.g.* 95%) is correctly classified.

Hyperspherical embeddings. A hypersphere is a topological space that is homeomorphic to a standard n-sphere, which is the set of points in $(n+1)$-dimensional Euclidean space that are located at a constant distance from the center. When the sphere has a unit radius, it is called the unit hypersphere. Formally, an n-dimensional unit-hypersphere

$$S^n := \{\mathbf{z} \in \mathbb{R}^{n+1} | \|\mathbf{z}\|_2 = 1\}$$

Geometrically, hyperspherical embeddings lie on the surface of a hypersphere.

3.3 Method

Overview. Our framework `CIDER` is illustrated in Figure 3.1. The general architecture consists of two components: (1) a deep neural network encoder $f : \mathcal{X} \mapsto \mathbb{R}^e$ that maps the augmented input $\tilde{\mathbf{x}}$ to a high dimensional feature embedding $f(\tilde{\mathbf{x}})$ (often referred to as the penultimate layer features); (2) a projection head $h : \mathbb{R}^e \mapsto \mathbb{R}^d$ that maps the high dimensional embedding $f(\tilde{\mathbf{x}})$ to a lower dimensional feature representation $\tilde{\mathbf{z}} := h(f(\tilde{\mathbf{x}}))$. The loss is applied to the normalized feature embedding $\mathbf{z} := \tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|_2$. The normalized embeddings are also referred to as *hyperspherical embeddings*, since they are on a unit hypersphere. Our goal is to shape the hyperspherical embedding space so that the learned embeddings can be mostly effective for distinguishing ID vs. OOD data.

3.3.1 Model Hyperspherical Embeddings

The hyperspherical embeddings can be naturally modeled by the von Mises-Fisher (vMF) distribution, a classical and important distribution in directional statistics (Mardia et al., 2000). In particular, vMF is analogous to spherical Gaussian distributions for features $\mathbf{z}$ with unit norms

($\|\mathbf{z}\|^2 = 1$). The probability density function for a unit vector $\mathbf{z} \in \mathbb{R}^d$ in class c is defined as:

$$p_d(\mathbf{z}; \boldsymbol{\mu}_c, \kappa) = Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top \mathbf{z}\right), \tag{3.1}$$

where $\boldsymbol{\mu}_c$ is the class prototype with unit norm, $\kappa \geq 0$ indicates the tightness of the distribution around the mean direction $\boldsymbol{\mu}_c$, and $Z_d(\kappa)$ is the normalization factor. The larger value of κ, the stronger the distribution is concentrated in the mean direction. In the extreme case of $\kappa = 0$, the sample points are distributed uniformly on the hypersphere.

Under this probability model, an embedding vector $\mathbf{z}$ is assigned to class c with the following normalized probability:

$$\mathbb{P}(y = c|\mathbf{z}; \{\kappa, \boldsymbol{\mu}_j\}_{j=1}^C) = \frac{Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top \mathbf{z}\right)}{\sum_{j=1}^C Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_j^\top \mathbf{z}\right)} \tag{3.2}$$

$$= \frac{\exp\left(\boldsymbol{\mu}_c^\top \mathbf{z}/\tau\right)}{\sum_{j=1}^C \exp\left(\boldsymbol{\mu}_j^\top \mathbf{z}/\tau\right)}, \tag{3.3}$$

where $\kappa = \frac{1}{\tau}$. Next, we outline our proposed training method that promotes class-conditional vMF distributions for OOD detection.

3.3.2 How to Optimize Hyperspherical Embeddings?

Training objective. Our key idea is to design a trainable loss function that enables optimizing hyperspherical embeddings into a mixture of vMF distributions, which satisfy two properties simultaneously: **(1)** each sample has a higher probability assigned to the correct class in comparison to incorrect classes, and **(2)** different classes are far apart from each other.

To achieve ❶, we can perform maximum likelihood estimation (MLE)

on the training data:

$$\text{argmax}_\theta \prod_{i=1}^{N} p(y_i|\mathbf{z}_i; \{\kappa_j, \boldsymbol{\mu}_j\}_{j=1}^{C}), \tag{3.4}$$

where i is the index of the embedding and N is the size of the training set. By taking the negative log-likelihood, the objective function is equivalent to minimizing the following loss:

$$\mathcal{L}_{\text{comp}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)}/\tau\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_j/\tau\right)}, \tag{3.5}$$

where $c(i)$ denotes the class index of a sample $\mathbf{x}_i$, and τ is the temperature parameter. We term this *compactness loss*, since it encourages samples to be closely aligned with its class prototype.

To promote property ❷, we propose the *dispersion loss*, optimizing large angular distances among different class prototypes:

$$\mathcal{L}_{\text{dis}} = \frac{1}{C}\sum_{i=1}^{C} \log \frac{1}{C-1}\sum_{j=1}^{C} \mathbb{1}\{j \neq i\} e^{\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau}. \tag{3.6}$$

While prototypes with larger pairwise angular distances may not impact ID classification accuracy, they are crucial for OOD detection, as we will show later in Section 3.4.2. Since the embeddings of OOD samples lie in-between ID clusters, optimizing large inter-class margin benefits OOD detection. The importance of inter-class dispersion can also be explained in Figure 3.2, where samples in the `fox` class (OOD) are semantically close to `cat` (ID) and `dog` (ID). A larger angular distance (*i.e.* smaller cosine similarity) between ID classes `cat` and `dog` in the hyperspherical space improves the separability from `fox`, and allows for more effective detection. We investigate this phenomenon quantitatively in Section 3.4.

Formally, our training objective **CIDER** (compactness and dispersion

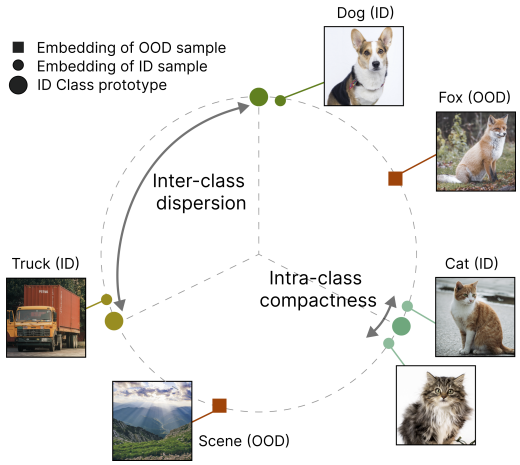


Figure 3.2: Illustration of desirable hyperspherical embeddings for OOD detection. As OOD samples lie *between* ID clusters, optimizing a large angular distance among ID clusters benefits OOD detection.

regularized learning) is:

$$\mathcal{L}_{\text{CIDER}} = \mathcal{L}_{\text{dis}} + \lambda_c \mathcal{L}_{\text{comp}}, \tag{3.7}$$

where λ_c is the co-efficient modulating the relative importance of two losses. These two terms are complementary to shape hyperspherical embeddings for both ID classification and OOD detection.

Prototype estimation and update. During training, an important step is to estimate the class prototype $\boldsymbol{\mu}_c$ for each class $c \in \{1, 2, ..., C\}$. One canonical way to estimate the prototypes is to use the mean vector of all training samples for each class and update it frequently during training. Despite its simplicity, this method incurs a heavy computational toll and causes undesirable training latency. Instead, the class-conditional prototypes can be effectively updated in an exponential-moving-average man-

ner (EMA) (Li et al., 2020b):

$$\boldsymbol{\mu}_c := \text{Normalize}(\alpha\boldsymbol{\mu}_c + (1-\alpha)\mathbf{z}),\ \forall c \in \{1, 2, \ldots, C\} \tag{3.8}$$

where the prototype $\boldsymbol{\mu}_c$ for class c is updated during training as the moving average of all embeddings with label c, and $\mathbf{z}$ denotes the normalized embedding of samples of class c. We ablate the effect of prototype update factor α in Section B.3. The pseudo-code for our method is in Appendix B.1.

Remark 1: Differences *w.r.t.* SSD+. We highlight three fundamental differences *w.r.t.* SSD+, in terms of training objective, test-time OOD detection, and theoretical interpretation. **(1)** At training time, SSD+ directly uses off-the-shelf SupCon loss (Khosla et al., 2020), which produces embeddings that *lack sufficient inter-class dispersion needed for OOD detection*. For example, when trained on CIFAR-10 using SupCon loss, the average angular distance between ID and OOD data is only 29.86 degrees in the embedding space. In contrast, CIDER enforces the inter-class dispersion by *explicitly* maximizing the angular distances among different ID class prototypes. As we will show in Section 3.4.2, CIDER displays a relative 42.36% improvement of ID-OOD separability compared to SSD+, due to the explicit inter-class dispersion. **(2)** At test time, SSD+ uses the Mahalanobis distance (Lee et al., 2018), which imposes a strong Gaussian distribution assumption on hyperspherical embeddings. In contrast, CIDER alleviates this assumption with a non-parametric distance score. **(3)** Lastly, CIDER explicitly models the latent representations as vMF distributions, providing a direct and clear geometrical interpretation of hyperspherical embeddings.

Remark 2: Differences *w.r.t.* Proxy-based methods. Our work also bears significant differences *w.r.t.* proxy-based metric learning methods.

(1) Our primary task is OOD detection, whereas deep metric learning is commonly used for face verification and image retrieval tasks; (2) Prior methods such as `ProxyAnchor` (Kim et al., 2020) lack explicit prototype-to-prototype dispersion, which we show is crucial for OOD detection. Moreover, `ProxyAnchor` initializes the proxies randomly and updates through gradients, while we estimate prototypes directly from sample embeddings using EMA. We provide experimental comparisons next.

3.4 Experiments

3.4.1 Common Setup

Datasets and training details. Following the common benchmarks in the literature, we consider `CIFAR-10` and `CIFAR-100` (Krizhevsky et al., 2009) as in-distribution datasets. For OOD test datasets, we use a suite of natural image datasets including `SVHN` (Netzer et al., 2011), `Places365` (Zhou et al., 2017), `Textures` (Cimpoi et al., 2014), `LSUN` (Yu et al., 2015), and `iSUN` (Xu et al., 2015). In our main experiments, we use ResNet-18 as the backbone for CIFAR-10 and ResNet-34 for CIFAR-100. We train the model using stochastic gradient descent with momentum 0.9, and weight decay 10^{-4}. To demonstrate the simplicity and effectiveness of `CIDER`, we adopt the **same** hyperparameters as in `SSD+` (Sehwag et al., 2021) with the `SupCon` loss: the initial learning rate is 0.5 with cosine scheduling, the batch size is 512, and the training time is 500 epochs. We choose the default weight $\lambda_c = 2$, so that the value of different loss terms are similar upon model initialization. Following the literature (Khosla et al., 2020), we use the embedding dimension of 128 for the projection head. The temperature τ is 0.1. We adjust the prototype update factor α, batch size, temperature, loss weight, prototype update factor, and model architecture in our ablation study (Appendix B.3). We report the ID classification results for `SSD+` and `CIDER` following the common linear evaluation

protocol (Khosla et al., 2020), where a linear classifier is trained on top of the normalized penultimate layer features. More experimental details are provided in Appendix B.2. Code and data is released publicly for reproducible research.

OOD detection scores. During test time, we employ a distance-based method for OOD detection. An input $\mathbf{x}$ is considered OOD if it is relatively far from the ID data in the embedding space. By default, we adopt a simple non-parametric KNN distance (Sun et al., 2022), which does not impose any distributional assumption on the feature space. Here the distance is the cosine similarity with respect to the k-th nearest neighbor, which is equivalent to the (negated) Euclidean distance as all features have unit norms. In the ablation study, we also consider the commonly used Mahalanobis score (Lee et al., 2018) for a fair comparison with `SSD+` (Sehwag et al., 2021).

Evaluation metrics. We report the following metrics: (1) the false positive rate (FPR95) of OOD samples when the true positive rate of ID samples is at 95%, (2) the area under the receiver operating characteristic curve (AUROC), and (3) ID classification accuracy (ID ACC).

3.4.2 Main Results and Analysis

CIDER outperforms competitive approaches. Table 3.1 contains a wide range of competitive methods for OOD detection. All methods are trained on ResNet-34 using CIFAR-100, without assuming access to auxiliary outlier datasets. For clarity, we divide the methods into two categories: trained with and without contrastive losses. For pre-trained model-based scores such as `MSP` (Hendrycks and Gimpel, 2017), `ODIN` (Liang et al., 2018), `Mahalanobis` (Lee et al., 2018), and `Energy` (Liu et al., 2020), the model is trained with the softmax cross-entropy (`CE`) loss. `GODIN` (Hsu et al.,

Table 3.1: OOD detection performance for CIFAR-100 (ID) with ResNet-34. Training with CIDER significantly improves OOD detection performance.

Method	OOD Dataset										Average	
	SVHN		Places365		LSUN		iSUN		Texture			
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
Without Contrastive Learning												
MSP	78.89	79.80	84.38	74.21	83.47	75.28	84.61	74.51	86.51	72.53	83.12	75.27
ODIN	70.16	84.88	82.16	75.19	76.36	80.10	79.54	79.16	85.28	75.23	78.70	79.11
Mahalanobis	87.09	80.62	84.63	73.89	84.15	79.43	83.18	78.83	61.72	84.87	80.15	79.53
Energy	66.91	85.25	81.41	76.37	59.77	86.69	66.52	84.49	79.01	79.96	70.72	82.55
GODIN	74.64	84.03	89.13	68.96	93.33	67.22	94.25	65.26	86.52	69.39	87.57	70.97
LogitNorm	59.60	90.74	80.25	78.58	81.07	82.99	84.19	80.77	86.64	75.60	78.35	81.74
With Contrastive Learning												
ProxyAnchor	87.21	82.43	70.10	79.84	37.19	91.68	70.01	84.96	65.64	84.99	66.03	84.78
CE + SimCLR	24.82	94.45	86.63	71.48	56.40	89.00	66.52	83.82	63.74	82.01	59.62	84.15
CSI	44.53	92.65	79.08	76.27	75.58	83.78	76.62	84.98	61.61	86.47	67.48	84.83
SSD+	31.19	94.19	77.74	79.90	79.39	85.18	80.85	84.08	66.63	86.18	67.16	85.90
KNN+	39.23	92.78	80.74	77.58	48.99	89.30	74.99	82.69	57.15	88.35	60.22	86.14
CIDER	23.09	95.16	79.63	73.43	16.16	96.33	71.68	82.98	43.87	90.42	**46.89**	**87.67**

2020) is trained using the DeConf-C loss, while LogitNorm (Wei et al., 2022) modifies CE loss via logit normalization. For methods involving contrastive losses, we consider ProxyAnchor (Kim et al., 2020), SimCLR (Winkens et al., 2020), CSI (Tack et al., 2020), SSD+ (Sehwag et al., 2021), and KNN+ (Sun et al., 2022). Both SSD+ and KNN+ use the SupCon loss in training. We use the same network structure and embedding dimension, while varying the training objective.

As shown in Table 3.1, OOD detection performance is significantly improved with CIDER. Three trends can be observed: (1) Compared to SSD+ and KNN+, CIDER explicitly optimizes for inter-class dispersion and leads more desirable embeddings. Moreover, CIDER alleviates the class-conditional Gaussian assumptions for OOD detection. Instead, a simple non-parametric distance-based score suffices. Specifically, CIDER outperforms the competitive methods SSD+ by **20.3**% and KNN+ by **13.3**% in FPR95; (2) While CSI (Tack et al., 2020) relies on sophisticated data augmentations and ensembles in testing, CIDER only uses the default data augmentations and thus is simpler in practice. Performance wise, CIDER reduces the average FPR95 by 20.6% compared to CSI; (3) Lastly, as a re-

sult of the improved embedding quality, CIDER improves the ID accuracy by 0.76% compared to training with the CE loss (Table B.5). We provide results on the less challenging task (CIFAR-10 as ID) in Appendix B.5, where CIDER's strong performance remains.

CIDER benefits different distance-based scores. We show that CIDER's strong representation can benefit different distance-based OOD scores. We consider the Mahalanobis score (denoted as Maha) due to its commonality, and to ensure a fair comparison with SSD+ under the same OOD score. The results are shown in Table 3.2. Under *both* KNN (non-parametric) and Maha (parametric) scores, CIDER consistently improves the OOD detection performance compared to training with SupCon. For example, CIDER with KNN significant reduces FPR95 by **13.33**$\%$ compared to SupCon+KNN. Moreover, compared to SSD+ (SupCon+Maha), CIDER+Maha reduces FPR95 by **22.77**$\%$. This further highlights the improved representation quality and generality of CIDER.

Table 3.2: Ablation on OOD detection score. Results are FPR95 on CIFAR-100 (ID) with ResNet-34. We evaluate both Mahalanobis and KNN score ($K = 300$).

Method	**OOD Dataset**					**AVG FPR95**
	SVHN	Places365	LSUN	iSUN	Texture	
SupCon+Maha (SSD+)	31.19	77.74	79.39	80.85	66.63	67.16
CIDER+Maha	**16.68**	**80.34**	**11.07**	**73.82**	**40.06**	**44.39**
SupCon+KNN (KNN+)	39.23	80.74	48.99	74.99	57.15	60.22
CIDER+KNN	**23.09**	**79.63**	**16.16**	**71.68**	**43.87**	**46.89**

Inter-class dispersion is key to strong OOD detection. Here we examine the effects of loss components on OOD detection. As shown in Table 3.3, we have the following observations: (1) For ID classification, training with $\mathcal{L}_{\text{comp}}$ alone leads to an accuracy of 75.19%, similar to the ID accuracy of SSD+ (75.11%). This suggests that promoting intra-class

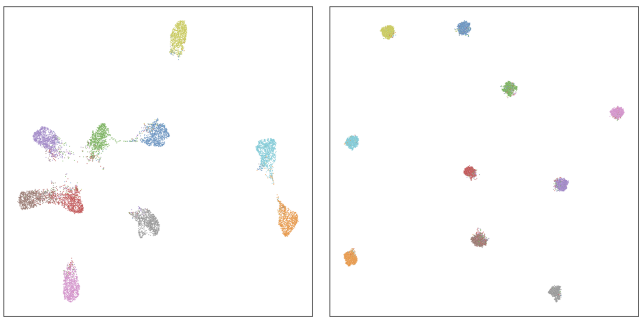
compactness and a moderate level of inter-class dispersion (as a result of sample-to-prototype negative pairs in $\mathcal{L}_{\text{comp}}$) are sufficient to discriminate different ID classes; (2) For OOD detection, further inter-class dispersion is critical, which is explicitly encouraged through the dispersion loss $\mathcal{L}_{\text{dis}}$. As a result, adding $\mathcal{L}_{\text{dis}}$ improved the average AUROC by 2%. However, promoting inter-class dispersion via $\mathcal{L}_{\text{dis}}$ alone without $\mathcal{L}_{\text{comp}}$ is not sufficient for neither ID classification nor OOD detection. Our ablation suggests that $\mathcal{L}_{\text{dis}}$ and $\mathcal{L}_{\text{comp}}$ work synergistically to improve the hyperspherical embeddings that are desirable for both ID classification and OOD detection.

Table 3.3: Ablation study on loss component. Results (in AUROC) are based on CIFAR-100 trained with ResNet-34. Training with only $\mathcal{L}_{\text{comp}}$ suffices for ID classification. Inter-class dispersion induced by $\mathcal{L}_{\text{dis}}$ is key to OOD detection.

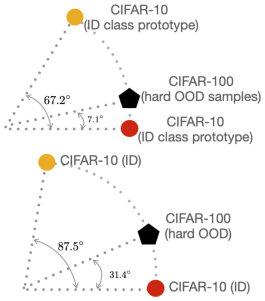
Loss Components		**AUROC↑**						**ID ACC↑**
$\mathcal{L}_{\text{comp}}$	$\mathcal{L}_{\text{dis}}$	Places365	LSUN	iSUN	Texture	SVHN	**AVG**	
✓		79.63	85.75	84.45	87.21	91.33	85.67	75.19
	✓	54.76	69.81	54.99	44.26	46.48	54.06	2.03
✓	✓	**73.43**	**96.33**	**82.98**	**90.42**	**95.16**	**87.67**	**75.35**

3.4.3 Characterizing and Understanding Embedding Quality

CIDER learns distinguishable representations. We visualize the learned feature embeddings in Figure 3.3 using UMAP (McInnes et al., 2018), where the colors encode different class labels. A salient observation is that embeddings obtained with `CIDER` enjoy much better compactness compared to embeddings trained with the `CE` loss (3.3a). Moreover, the classes are distributed more uniformly in the space, highlighting the efficacy of the dispersion loss.



(a) ID embeddings of CE (left) and CIDER (right)



(b) ID & OOD of CE (top) and CIDER (down)

Figure 3.3: (a): UMAP (McInnes et al., 2018) visualization of the features when the model is trained with CE vs. CIDER for CIFAR-10 (ID). (b): CIDER makes OOD samples more separable from ID compared to CE (*c.f.* Table 3.4).

CIDER improves inter-class dispersion and intra-class compactness. Beyond visualization, we also quantitatively measure the embedding quality. We propose two measurements: inter-class dispersion and intra-class compactness:

$$\text{Dispersion}(\boldsymbol{\mu}) = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{C-1}\sum_{j=1}^{C}\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_j\mathbb{1}\{j\neq i\}.$$

$$\text{Compactness}(\mathcal{D}_{\text{tr}}^{\text{in}},\boldsymbol{\mu}) = \frac{1}{C}\sum_{j=1}^{C}\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^\top\boldsymbol{\mu}_j\mathbb{1}\{y_i=j\},$$

where $\mathcal{D}_{\text{tr}}^{\text{in}} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, and $\mathbf{z}_i$ is the normalized embedding of $\mathbf{x}_i$ for all $1 \leq i \leq N$. Dispersion is measured by the average cosine similarity among pair-wise class prototypes. The compactness can be interpreted as the average cosine similarity between each feature embedding and its corresponding class prototype.

To make the measurements more interpretable, we convert cosine similarities to *angular degrees*. Hence, a higher inter-class dispersion (in de-

grees) indicates more separability among class prototypes, which is desirable. Similarly, lower intra-class compactness (in degrees) is better. The results are shown in Table 3.4 based on the CIFAR-10 test set. Compared to `SSD+` (with `SupCon` loss), `CIDER` significantly improves the inter-class dispersion by **12.03** degrees. Different from `SupCon`, `CIDER` explicitly optimizes the inter-class dispersion, which especially benefits OOD detection.

Table 3.4: Compactness and dispersion of CIFAR-10 feature embedding, along with the separability *w.r.t.* each OOD test set. We convert cosine similarity to angular degrees for better readability.

Training Loss	**Dispersion (ID)**↑	**Compactness (ID)**↓	**ID-OOD Separability**↑ (in degree)					
	(in degree)	(in degree)	CIFAR-100	LSUN	iSUN	Texture	SVHN	**AVG**
Cross-Entropy	67.17	24.53	7.11	14.57	13.70	13.76	11.08	12.04
SSD+ (SupCon loss)	75.50	22.08	23.90	28.55	25.70	33.45	37.70	29.86
CIDER (ours)	**87.53**	**21.35**	**31.41**	**48.37**	**41.54**	**39.60**	**51.65**	**42.51**

CIDER improves ID-OOD separability. Next, we quantitatively measure how the feature embedding quality affects the ID-OOD separability. We introduce a *separability score*, which measures on average how close the embedding of a sample from the OOD test set is to the closest ID class prototype, compared to that of an ID sample. The traditional notion of "OOD being far away from ID classes" is now translated to "OOD being somewhere between ID clusters on the hypersphere". A higher separability score indicates that the OOD test set is easier to be detected. Formally, we define the separability measurement as:

$$\uparrow \text{Separability} = \frac{1}{|\mathcal{D}_{\text{test}}^{\text{ood}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{test}}^{\text{ood}}} \max_{j \in [C]} \mathbf{z}_{\mathbf{x}}^{\top} \boldsymbol{\mu}_j - \frac{1}{|\mathcal{D}_{\text{test}}^{\text{in}}|} \sum_{\mathbf{x}' \in \mathcal{D}_{\text{test}}^{\text{in}}} \max_{j \in [C]} \mathbf{z}_{\mathbf{x}'}^{\top} \boldsymbol{\mu}_j, \quad (3.9)$$

where $\mathcal{D}_{\text{test}}^{\text{ood}}$ is the OOD test dataset and $\mathbf{z}_{\mathbf{x}}$ denotes the normalized embedding of sample $\mathbf{x}$. Table 3.4 shows that CIDER leads to higher separability and consequently superior OOD detection performance (*c.f.* Ta-

ble 3.1). Averaging across 5 OOD test datasets, our method displays a relative **42.36**% improvement of ID-OOD separability compared to `SupCon`. This further verifies the effectiveness of `CIDER` for improving OOD detection.

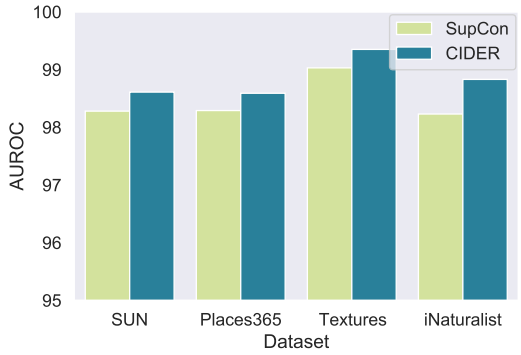


Figure 3.4: Fine-tuning pre-trained ResNet-34 on ImageNet-100 (ID).

3.4.4 Additional Ablations and Analysis

CIDER is competitive on large-scale datasets. To further examine the performance of CIDER on real-world tasks, we evaluate the performance of CIDER on the more challenging large-scale benchmarks. Specifically, we use ImageNet-100 as ID, a subset of ImageNet (Deng et al., 2009) consisting of 100 randomly sampled classes. For OOD test datasets, we use the same ones in (Huang and Li, 2021), including subsets of iNaturalist (Van Horn et al., 2018), Sun (Xiao et al., 2010), Places (Zhou et al., 2017), and Texture (Cimpoi et al., 2014). For each OOD dataset, the categories do not overlap with the ID dataset. For computational efficiency, we fine-tune pre-trained ResNet-34 with CIDER and SupCon losses for 10 epochs with an initial learning rate of 0.01. For each loss, we update the weights of the last residual block and the nonlinear projection head, while freezing the parameters in the first three residual blocks. At test time, we use the same detection score (KNN) to evaluate representation quality.

The performance (in AUROC) is shown in Figure 3.4 (more results are in Appendix B.4). We can see that CIDER remains very competitive on all the OOD test sets where CIDER consistently outperforms SupCon. This further verifies the benefits of explicitly promoting inter-class dispersion and intra-class compactness.

Ablation studies on weight scale, prototype update factor, learning rate, temperature, batch size, and architecture. We provide comprehensive ablation studies to understand the impact of various factors in Appendix B.3. For example, as shown in Figure 3.5, CIDER consistently outperforms SupCon across different batch sizes.

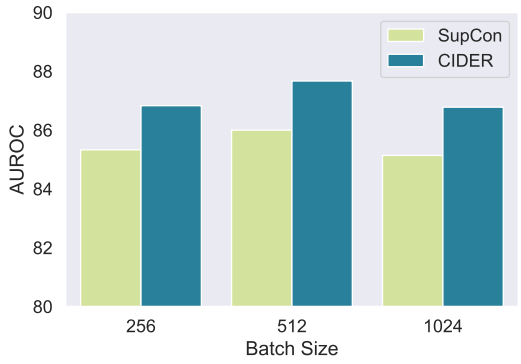


Figure 3.5: OOD Detection performance across different batch sizes..

3.5 Related Works

Out-of-distribution detection. The majority of works in the OOD detection literature focus on the supervised setting, where the goal is to derive a binary ID-OOD classifier along with a classification model for the in-distribution data. Compared to generative model-based methods (Kirichenko et al., 2020; Nalisnick et al., 2019; Ren et al., 2019; Serrà et al., 2020; Xiao et al., 2020), OOD detection based on supervised discriminative models

typically yield more competitive performance. For deep neural networks-based methods, most OOD detection methods derive confidence scores either based on the output (Bendale and Boult, 2016; Hendrycks and Gimpel, 2017; Hsu et al., 2020; Huang and Li, 2021; Liang et al., 2018; Liu et al., 2020; Sun et al., 2021; Ming et al., 2022b), gradient information (Huang et al., 2021a), or the feature embeddings (Lee et al., 2018; Sastry and Oore, 2020; Tack et al., 2020; Zhou et al., 2021c; Sehwag et al., 2021; Sun et al., 2022; Ming et al., 2022a; Du et al., 2022a). Our method can be categorized as a distance-based OOD detection method by exploiting the hyperspherical embedding space.

Contrastive representation learning. Contrastive representation learning (van den Oord et al., 2019) aims to learn a discriminative embedding where positive samples are aligned while negative ones are dispersed. It has demonstrated remarkable success for visual representation learning in unsupervised (Chen et al., 2020a,b; He et al., 2020; Robinson et al., 2021), semi-supervised (Assran et al., 2021), and supervised settings (Khosla et al., 2020). Recently, Li et al. (2021b) propose a prototype-based contrastive loss for unsupervised learning where prototypes are generated via a clustering algorithm, while our method is supervised where prototypes are updated based on labels. Li et al. (2021a) incorporate a prototype-based loss to tackle data noise. Wang and Isola (2020) analyze the asymptotic behavior of contrastive losses theoretically, while Wang and Liu (2021) empirically investigate the properties of contrastive losses for classification. Recently, Bucci et al. (2022) investigates contrastive learning for open-set domain adaptation. However, none of the works focus on OOD detection. We aim to fill the gap and facilitate the design and understanding of contrastive losses for OOD detection.

Representation learning for OOD detection. Self-supervised learning has been shown to improve OOD detection. Prior works (Sehwag et al.,

2021; Winkens et al., 2020) verify the effectiveness of directly applying the off-the-shelf multi-view contrastive losses such as `SupCon` and `SimCLR` for OOD detection. `CSI` (Tack et al., 2020) investigates the type of data augmentations that are particularly beneficial for OOD detection. Different from prior works, we focus on hyperspherical embeddings and propose to explicitly encourage the desirable properties for OOD detection, and thus alleviate the dependence on specific data augmentations or self-supervision. Moreover, CIDER explicitly models the latent representations as vMF distributions, providing a direct and clear geometrical interpretation of hyperspherical embeddings. Closest to our work, Du et al. (2022a) recently explores shaping representations into vMF distributions for object-level OOD detection. However, they do not consider the inter-class dispersion loss, which we show is crucial to achieving strong OOD detection performance.

Deep metric learning. Learning a desirable embedding is a fundamental goal in the deep metric learning community. Various losses have been proposed for face verification (Deng et al., 2019; Liu et al., 2017; Wang et al., 2018), person re-identification (Chen et al., 2017; Xiao et al., 2017), and image retrieval (Kim et al., 2019; Movshovitz-Attias et al., 2017; Oh Song et al., 2016; Teh et al., 2020). However, none of the works focus on desirable embeddings for OOD detection. The difference between CIDER and proxy-based methods has been discussed in Remark 2.

3.6 Conclusion and Outlook

In this work, we propose `CIDER`, a novel representation learning framework that exploits hyperspherical embeddings for OOD detection. `CIDER` jointly optimizes the *dispersion* and *compactness* losses to promote strong ID-OOD separability. We show that `CIDER` achieves superior performance

on common OOD benchmarks, including large-scale OOD detection tasks. Moreover, we introduce new measurements to quantify the hyperspherical embedding, and establish the relationship with OOD detection performance. We conduct extensive ablations to understand the efficacy and behavior of CIDER under various settings and hyperparameters. We hope our work can inspire future methods of exploiting hyperspherical representations for OOD detection.

Chapter 4

Hyperspherical Out-of-Distribution Generalization

Publication Statement. This chapter is a joint work with Haoyue Bai, Julian Katz-Samuels, and Yixuan Li. The paper version of this chapter appeared in ICLR 2024 (Ming et al., 2024).

Out-of-distribution (OOD) generalization is critical for machine learning models deployed in the real world. However, achieving this can be fundamentally challenging, as it requires the ability to learn invariant features across different domains or environments. In this chapter, we introduce a novel framework HYPO (**HYP**erspherical **O**OD generalization) that provably learns domain-invariant representations in a hyperspherical space. In particular, our hyperspherical learning algorithm is guided by intra-class variation and inter-class separation principles—ensuring that features from the same class (across different training domains) are closely aligned with their class prototypes, while different class prototypes are maximally separated. We further provide theoretical justifica-

tions on how our prototypical learning objective improves the OOD generalization bound. Through extensive experiments on challenging OOD benchmarks, we demonstrate that our approach outperforms competitive baselines and achieves superior performance. Code is available at https://github.com/deeplearning-wisc/hypo.

4.1 Introduction

Deploying machine learning models in real-world settings presents a critical challenge of generalizing under distributional shifts. These shifts are common due to mismatches between the training and test data distributions. For instance, in autonomous driving, a model trained on in-distribution (ID) data collected under sunny weather conditions is expected to perform well in out-of-distribution (OOD) scenarios, such as rain or snow. This underscores the importance of the OOD generalization problem, which involves learning a predictor that can generalize across all possible environments, despite being trained on a finite subset of training environments.

A plethora of OOD generalization algorithms has been developed in recent years (Zhou et al., 2022a), where a central theme is to learn domain-invariant representations—features that are consistent and meaningful across different environments (domains) and can generalize to the unseen test environment. Recently, Ye et al. (2021) theoretically showed that the OOD generalization error can be bounded in terms of intra-class *variation* and inter-class *separation*. Intra-class variation measures the stability of representations across different environments, while inter-class separation assesses the dispersion of features among different classes. Ideally, features should display low variation and high separation, in order to generalize well to OOD data (formally described in Section 4.3). Despite the theoretical analysis, a research question remains open in the field:

RQ: How to design a practical learning algorithm that directly achieves these two properties, and what theoretical guarantees can the algorithm offer?

To address the question, this chapter presents a learning framework HYPO (**HYP**erspherical **O**OD generalization), which provably learns domain-invariant representations in the hyperspherical space with unit norm (Section 4.4). Our key idea is to promote low variation (aligning representation across domains for every class) and high separation (separating prototypes across different classes). In particular, the learning objective shapes the embeddings such that samples from the same class (across all training environments) gravitate towards their corresponding class prototype, while different class prototypes are maximally separated. The two losses in our objective function can be viewed as optimizing the key properties of intra-class variation and inter-class separation, respectively. Since samples are encouraged to have a small distance with respect to their class prototypes, the resulting embedding geometry can have a small distribution discrepancy across domains and benefits OOD generalization. Geometrically, we show that our loss function can be understood through the lens of maximum likelihood estimation under the classic von Mises-Fisher distribution.

Empirical contribution. Empirically, we demonstrate strong OOD generalization performance by extensively evaluating HYPO on common benchmarks (Section 4.5). On the CIFAR-10 (ID) vs. CIFAR-10-Corruption (OOD) task, HYPO substantially improves the OOD generalization accuracy on challenging cases such as Gaussian noise, from 78.09% to 85.21%. Furthermore, we establish superior performance on popular domain generalization benchmarks, including PACS, Office-Home, VLCS, etc. For example, we achieve 88.0% accuracy on PACS which outperforms the best

loss-based method by 1.1%. This improvement is non-trivial using standard stochastic gradient descent optimization. When coupling our loss with specialized optimization SWAD (Cha et al., 2021), the accuracy is further increased to 89%. We provide visualization and quantitative analysis to verify that features learned by HYPO indeed achieve low intra-class variation and high inter-class separation.

Theoretical insight. We provide theoretical justification for how HYPO can guarantee improved OOD generalization, supporting our empirical findings. Our theory complements Ye et al. (2021), which does not provide a loss for optimizing the intra-class variation or inter-class separation. Thus, *a key contribution of this chapter is to provide a crucial link between provable understanding and a practical algorithm for OOD generalization in the hypersphere.* In particular, our Theorem 4.6 shows that when the model is trained with our loss function, we can upper bound intra-class variation, a key quantity to bound OOD generalization error. For a learnable OOD generalization task, the upper bound on generalization error is determined by the variation estimate on the training environments, which is effectively reduced by our loss function under sufficient sample size and expressiveness of the neural network.

4.2 Problem Setup

We consider a multi-class classification task that involves a pair of random variables (X, Y) over instances $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ and corresponding labels $y \in \mathcal{Y} := \{1, 2, \cdots, C\}$. The joint distribution of X and Y is unknown and represented by $\mathbb{P}_{XY}$. The goal is to learn a predictor function, $f : \mathcal{X} \to \mathbb{R}^C$, that can accurately predict the label y for an input $\mathbf{x}$, where $(\mathbf{x}, y) \sim \mathbb{P}_{XY}$.

Unlike in standard supervised learning tasks, the out-of-distribution (OOD) generalization problem is challenged by the fact that one cannot sample directly from $\mathbb{P}_{XY}$. Instead, we can only sample (X, Y) under lim-

ited environmental conditions, each of which corrupts or varies the data differently. For example, in autonomous driving, these environmental conditions may represent different weathering conditions such as snow, rain, etc. We formalize this notion of environmental variations with a set of *environments* or domains $\mathcal{E}_{\text{all}}$. Sample pairs (X^e, Y^e) are randomly drawn from environment e. In practice, we may only have samples from a finite subset of *available environments* $\mathcal{E}_{\text{avail}} \subset \mathcal{E}_{\text{all}}$. Given $\mathcal{E}_{\text{avail}}$, the goal is to learn a predictor f that can generalize across all possible environments. The problem is stated formally below.

Definition 4.1 (OOD Generalization)**.** *Let $\mathcal{E}_{\text{avail}} \subset \mathcal{E}_{\text{all}}$ be a set of training environments, and assume that for each environment $e \in \mathcal{E}_{\text{avail}}$, we have a dataset $\mathcal{D}^e = \{(\mathbf{x}_j^e, y_j^e)\}_{j=1}^{n_e}$, sampled i.i.d. from an unknown distribution $\mathbb{P}_{XY}^e$. The goal of OOD generalization is to find a classifier f^*, using the data from the datasets $\mathcal{D}^e$, that minimizes the worst-case risk over the entire family of environments $\mathcal{E}_{\text{all}}$:*

$$\min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}_{\text{all}}} \mathbb{E}_{\mathbb{P}_{XY}^e} \ell(f(X^e), Y^e), \tag{4.1}$$

where $\mathcal{F}$ is hypothesis space and $l(\cdot, \cdot)$ is the loss function.

The problem is challenging since we do not have access to data from domains outside $\mathcal{E}_{\text{avail}}$. In particular, the task is commonly referred to as multi-source domain generalization when $|\mathcal{E}_{\text{avail}}| > 1$.

4.3 Motivation of Algorithm Design

Our work is motivated by the theoretical findings in Ye et al. (2021), which shows that the OOD generalization performance can be bounded in terms of intra-class *variation* and inter-class *separation* with respect to various environments. The formal definitions are given as follows.

Definition 4.2 (Intra-class variation)**.** *The variation of feature ϕ across a domain set $\mathcal{E}$ is*

$$\mathcal{V}(\phi, \mathcal{E}) = \max_{y \in \mathcal{Y}} \sup_{e, e' \in \mathcal{E}} \rho\Big(\mathbb{P}(\phi^e|y), \mathbb{P}(\phi^{e'}|y)\Big), \tag{4.2}$$

where $\rho(\mathbb{P}, \mathbb{Q})$ is a symmetric distance (e.g., Wasserstein distance, total variation, Hellinger distance) between two distributions, and $\mathbb{P}(\phi^e|y)$ denotes the class-conditional distribution for features of samples in environment e.

Definition 4.3 (Inter-class separation[1])**.** *The separation of feature ϕ across domain set $\mathcal{E}$ is*

$$\mathcal{I}_\rho(\phi, \mathcal{E}) = \frac{1}{C(C-1)} \sum_{\substack{y \neq y' \\ y, y' \in \mathcal{Y}}} \min_{e \in \mathcal{E}} \rho\Big(\mathbb{P}(\phi^e|y), \mathbb{P}(\phi^e|y')\Big). \tag{4.3}$$

The intra-class variation $\mathcal{V}(\phi, \mathcal{E})$ measures the stability of feature ϕ over the domains in $\mathcal{E}$ and the inter-class separation $\mathcal{I}(\phi, \mathcal{E})$ captures the ability of ϕ to distinguish different labels. Ideally, features should display high separation and low variation.

Definition 4.4. *The OOD generalization error of classifier f is defined as follows:*

$$err(f) = \max_{e \in \mathcal{E}_{all}} \mathbb{E}_{\mathbb{P}^e_{XY}} \ell(f(X^e), Y^e) - \max_{e \in \mathcal{E}_{avail}} \mathbb{E}_{\mathbb{P}^e_{XY}} \ell(f(X^e), Y^e)$$

which is bounded by the variation estimate on $\mathcal{E}_{avail}$ with the following theorem.

Theorem 4.5 (OOD error upper bound, informal (Ye et al., 2021))**.** *Suppose the loss function $\ell(\cdot, \cdot)$ is bounded by $[0, B]$. For a learnable OOD generalization problem with sufficient inter-class separation, the OOD generalization*

[1] Referred to as "Informativeness" in Ye et al. (2021).

error $err(f)$ can be upper bounded by

$$\mathrm{err}(f) \leq O\Big(\Big(\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}})\Big)^{\frac{\alpha^2}{(\alpha+d)^2}}\Big), \tag{4.4}$$

for some $\alpha > 0$, and $\mathcal{V}^{\text{sup}}\left(h, \mathcal{E}_{\text{avail}}\right) \triangleq \sup_{\beta \in \mathcal{S}^{d-1}} \mathcal{V}\left(\beta^\top h, \mathcal{E}_{\text{avail}}\right)$ is the inter-class variation, $h(\cdot) \in \mathbb{R}^d$ is the feature vector, and β is a vector in unit hypersphere $\mathcal{S}^{d-1} = \left\{\beta \in \mathbb{R}^d : \|\beta\|_2 = 1\right\}$, and f is a classifier based on normalized feature h.

Remarks. The Theorem above suggests that both low intra-class variation and high inter-class separation are desirable properties for theoretically grounded OOD generalization. Note that in the **full formal** Theorem (see Appendix C.3), maintaining the inter-class separation is necessary for the learnability of the OOD generalization problem (Def. C.9). In other words, when the learned embeddings exhibit high inter-class separation, the problem becomes learnable. In this context, bounding intra-class variation becomes crucial for reducing the OOD generalization error.

Despite the theoretical underpinnings, it remains unknown to the field how to design a practical learning algorithm that directly achieves these two properties, and what theoretical guarantees can the algorithm offer. This motivates our work.

To reduce the OOD generalization error, our key motivation is to design a hyperspherical learning algorithm that directly promotes low variation (aligning representation across domains for every class) and high separation (separating prototypes across different classes).

4.4 Method

Following the motivation in Section 4.3, we now introduce the details of the learning algorithm HYPO (**HYP**erspherical **O**OD generalization), which is designed to promote domain invariant representations in the hyperspherical space. The key idea is to shape the hyperspherical embedding space so that samples from the same class (across all training environments $\mathcal{E}_{\text{avail}}$) are closely aligned with the corresponding class prototype. Since all points are encouraged to have a small distance with respect to the class prototypes, the resulting embedding geometry can have a small distribution discrepancy across domains and hence benefits OOD generalization. In what follows, we first introduce the learning objective (Section 4.4.1), and then we discuss the geometrical interpretation of the loss and embedding (Section 4.4.2). We will provide theoretical justification for HYPO in Section 4.6, which leads to a provably smaller intra-class variation, a key quantity to bound OOD generalization error.

4.4.1 Hyperspherical Learning for OOD Generalization

Loss function. The learning algorithm is motivated to directly optimize the two criteria: intra-class variation and inter-class separation. At a high level, HYPO aims to learn embeddings for each sample in the training environments by maintaining a class prototype vector $\boldsymbol{\mu}_c \in \mathbb{R}^d$ for each class $c \in \{1, 2, ..., C\}$. To optimize for low variation, the loss encourages the feature embedding of a sample to be close to its class prototype. To optimize for high separation, the loss encourages different class prototypes to be far apart from each other.

Specifically, we consider a deep neural network $h : \mathcal{X} \mapsto \mathbb{R}^d$ that maps an input $\tilde{\mathbf{x}} \in \mathcal{X}$ to a feature embedding $\tilde{\mathbf{z}} := h(\tilde{\mathbf{x}})$. The loss operates on the normalized feature embedding $\mathbf{z} := \tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|_2$. The normalized embeddings are also referred to as *hyperspherical embeddings*, since they are on

a unit hypersphere, denoted as $S^{d-1} := \{\mathbf{z} \in \mathbb{R}^d \mid \|\mathbf{z}\|_2 = 1\}$. The loss is formalized as follows:

$$\mathcal{L} = \underbrace{-\frac{1}{N}\sum_{e\in\mathcal{E}_{\text{avail}}}\sum_{i=1}^{|\mathcal{D}^e|}\log\frac{\exp\left(\mathbf{z}_i^{e\top}\boldsymbol{\mu}_{c(i)}/\tau\right)}{\sum_{j=1}^{C}\exp\left(\mathbf{z}_i^{e\top}\boldsymbol{\mu}_j/\tau\right)}}_{\mathcal{L}_{\text{var}}:\ \downarrow\ \textbf{variation}} + \underbrace{\frac{1}{C}\sum_{i=1}^{C}\log\frac{1}{C-1}\sum_{j\neq i, j\in\mathcal{Y}}\exp\left(\boldsymbol{\mu}_i^{\top}\boldsymbol{\mu}_j/\tau\right)}_{\uparrow\ \textbf{separation}},$$

where N is the number of samples, τ is the temperature, $\mathbf{z}$ is the normalized feature embedding, and $\boldsymbol{\mu}_c$ is the prototype embedding for class c. While hyperspherical learning algorithms have been studied in other context (Mettes et al., 2019; Khosla et al., 2020; Ming et al., 2023), *none of the prior works explored its provable connection to domain generalization, which is our distinct contribution. We will theoretically show in Section 4.6 that minimizing our loss function effectively reduces intra-class variation, a key quantity to bound OOD generalization error.*

The training objective in Equation 4.5 can be efficiently optimized end-to-end. During training, an important step is to estimate the class prototype $\boldsymbol{\mu}_c$ for each class $c \in \{1, 2, ..., C\}$. The class-conditional prototypes can be updated in an exponential-moving-average manner (EMA) (Li et al., 2020b):

$$\boldsymbol{\mu}_c := \text{Normalize}(\alpha\boldsymbol{\mu}_c + (1-\alpha)\mathbf{z}),\ \forall c \in \{1, 2, \ldots, C\} \tag{4.5}$$

where the prototype $\boldsymbol{\mu}_c$ for class c is updated during training as the moving average of all embeddings with label c, and $\mathbf{z}$ denotes the normalized embedding of samples of class c. An end-to-end pseudo algorithm is summarized in Appendix C.1.

Class prediction. In testing, classification is conducted by identifying the closest class prototype: $\hat{y} = \arg\max_{c\in[C]} f_c(\mathbf{x})$, where $f_c(\mathbf{x}) = \mathbf{z}^{\top}\boldsymbol{\mu}_c$ and $\mathbf{z} = \frac{h(\mathbf{x})}{\|h(\mathbf{x})\|_2}$ is the normalized feature embedding.

4.4.2 Geometrical Interpretation of Loss and Embedding

Geometrically, the loss function above can be interpreted as learning embeddings located on the surface of a unit hypersphere. The hyperspherical embeddings can be modeled by the von Mises-Fisher (vMF) distribution, a well-known distribution in directional statistics (Jupp and Mardia, 2009). For a unit vector $\mathbf{z} \in \mathbb{R}^d$ in class c, the probability density function is defined as

$$p(\mathbf{z} \mid y = c) = Z_d(\kappa) \exp(\kappa \boldsymbol{\mu}_c^\top \mathbf{z}), \tag{4.6}$$

where $\boldsymbol{\mu}_c \in \mathbb{R}^d$ denotes the mean direction of the class c, $\kappa \geq 0$ denotes the concentration of the distribution around $\boldsymbol{\mu}_c$, and $Z_d(\kappa)$ denotes the normalization factor. A larger κ indicates a higher concentration around the class center. In the extreme case of $\kappa = 0$, the samples are distributed uniformly on the hypersphere.

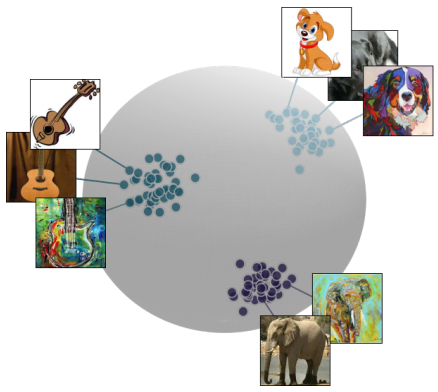


Figure 4.1: Illustration of hyperspherical embeddings. Images are from PACS (Li et al., 2017a).

Under this probabilistic model, an embedding $\mathbf{z}$ is assigned to the class

c with the following probability

$$p(y=c \mid \mathbf{z}; \{\kappa, \boldsymbol{\mu}_j\}_{j=1}^{C}) = \frac{Z_d(\kappa)\exp(\kappa \boldsymbol{\mu}_c^\top \mathbf{z})}{\sum_{j=1}^{C} Z_d(\kappa)\exp(\kappa \boldsymbol{\mu}_j^\top \mathbf{z})} = \frac{\exp(\boldsymbol{\mu}_c^\top \mathbf{z}/\tau)}{\sum_{j=1}^{C}\exp(\boldsymbol{\mu}_j^\top \mathbf{z}/\tau)}, \tag{4.7}$$

where $\tau = 1/\kappa$ denotes a temperature parameter.

Maximum likelihood view. Notably, minimizing the first term in our loss (*cf.* Eq. 4.5) is equivalent to performing maximum likelihood estimation under the vMF distribution:

$$\text{argmax}_\theta \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i; \{\kappa, \boldsymbol{\mu}_j\}_{j=1}^{C}), \text{where } (\mathbf{x}_i, y_i) \in \bigcup_{e \in \mathcal{E}_{\text{train}}} \mathcal{D}^e$$

where i is the index of sample, j is the index of the class, and N is the size of the training set. In effect, this loss encourages each ID sample to have a high probability assigned to the correct class in the mixtures of the vMF distributions.

4.5 Experiments

In this section, we show that HYPO achieves strong OOD generalization performance in practice, establishing competitive performance on several benchmarks. In what follows, we describe the experimental setup in Section 4.5.1, followed by main results and analysis in Section 4.5.2.

4.5.1 Experimental Setup

Datasets. Following the common benchmarks in literature, we use CIFAR-10 (Krizhevsky et al., 2009) as the in-distribution data. We use CIFAR-10-C (Hendrycks and Dietterich, 2019) as OOD data, with 19 different common corruption applied to CIFAR-10. In addition to CIFAR-10, we

conduct experiments on popular benchmarks including PACS (Li et al., 2017a), Office-Home (Gulrajani and Lopez-Paz, 2021), and VLCS (Gulrajani and Lopez-Paz, 2021) to validate the generalization performance. PACS contains 4 domains/environments (`photo`, `art painting`, `cartoon`, `sketch`) with 7 classes (`dog`, `elephant`, `giraffe`, `guitar`, `horse`, `house`, `person`). Office-Home comprises four different domains: art, clipart, product, and real. Results on additional OOD datasets Terra Incognita (Gulrajani and Lopez-Paz, 2021), and ImageNet can be found in Appendix C.6 and Appendix C.7.

Evaluation metrics. We report the following two metrics: **(1)** ID classification accuracy (ID Acc.) for ID generalization, and **(2)** OOD classification accuracy (OOD Acc.) for OOD generalization.

Experimental details. In our main experiments, we use ResNet-18 for CIFAR-10 and ResNet-50 for PACS, Office-Home, and VLCS. For these datasets, we use stochastic gradient descent with momentum 0.9, and weight decay 10^{-4}. For CIFAR-10, we train the model from scratch for 500 epochs using an initial learning rate of 0.5 and cosine scheduling, with a batch size of 512. Following common practice for contrastive losses (Chen et al., 2020a; Khosla et al., 2020; Yao et al., 2022), we use an MLP projection head with one hidden layer to obtain features. The embedding (output) dimension is 128 for the projection head. We set the default temperature τ as 0.1 and the prototype update factor α as 0.95. For PACS, Office-Home, and VLCS, we follow the common practice and initialize the network using ImageNet pre-trained weights. We fine-tune the network for 50 epochs. The embedding dimension is 512 for the projection head. We adopt the leave-one-domain-out evaluation protocol and use the training domain validation set for model selection (Gulrajani and Lopez-Paz, 2021), where the validation set is pooled from all training domains. Details on other hyperparameters are in Appendix C.4.

Algorithm	PACS	Office-Home	VLCS	Average Acc. (%)
ERM (Vapnik, 1999)	85.5	67.6	77.5	76.7
CORAL (Sun and Saenko, 2016)	86.2	68.7	78.8	77.9
DANN (Ganin et al., 2016)	83.7	65.9	78.6	76.1
MLDG (Li et al., 2018a)	84.9	66.8	77.2	76.3
CDANN (Li et al., 2018c)	82.6	65.7	77.5	75.3
MMD (Li et al., 2018b)	84.7	66.4	77.5	76.2
IRM (Arjovsky et al., 2019)	83.5	64.3	78.6	75.5
GroupDRO (Sagawa et al., 2019)	84.4	66.0	76.7	75.7
I-Mixup (Wang et al., 2020)	84.6	68.1	77.4	76.7
RSC (Huang et al., 2020)	85.2	65.5	77.1	75.9
ARM (Zhang et al., 2021a)	85.1	64.8	77.6	75.8
MTL (Blanchard et al., 2021)	84.6	66.4	77.2	76.1
VREx (Krueger et al., 2021)	84.9	66.4	78.3	76.5
Mixstyle (Zhou et al., 2021a)	85.2	60.4	77.9	74.5
SelfReg (Kim et al., 2021a)	85.6	67.9	77.8	77.1
SagNet (Nam et al., 2021)	86.3	68.1	77.8	77.4
GVRT (Min et al., 2022)	85.1	70.1	79.0	78.1
VNE (Kim et al., 2023a)	86.9	65.9	78.1	77.0
HYPO (Ours)	$88.0_{\pm 0.4}$	$71.7_{\pm 0.3}$	$78.2_{\pm 0.4}$	**79.3**

Table 4.1: Comparison with domain generalization methods on the PACS, Office-Home, and VLCS. All methods are trained on ResNet-50. The model selection is based on a training domain validation set. To isolate the effect of loss functions, all methods are optimized using standard SGD. We report the average and std of our method. $\pm x$ denotes the rounded standard error.

4.5.2 Main Results and Analysis

HYPO excels on common corruption benchmarks. As shown in Figure 4.2, HYPO achieves consistent improvement over the ERM baseline (trained with cross-entropy loss), on a variety of common corruptions. Our evaluation includes different corruptions including Gaussian noise, Snow, JPEG compression, Shot noise, Zoom blur, etc. The model is trained on CIFAR-10, without seeing any type of corruption data. In particular, our method brings significant improvement for challenging cases such as Gaussian noise, enhancing OOD accuracy from 78.09% to 85.21% ($+$**7.12**$\%$). Complete results on all 19 different corruption types are in Appendix C.5.

HYPO establishes competitive performance on popular benchmarks.

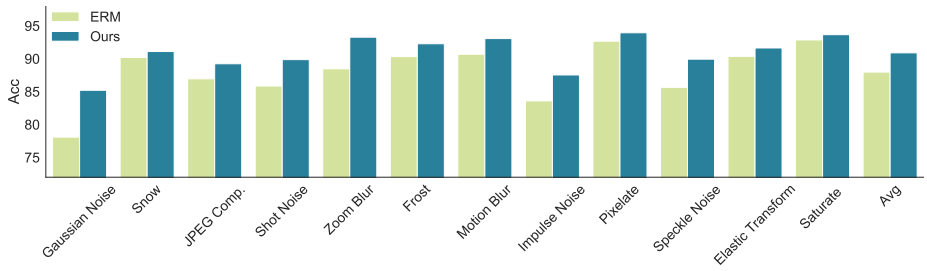


Figure 4.2: Our method HYPO significantly improves the OOD generalization performance compared to ERM on various OOD datasets w.r.t. CIFAR-10 (ID). Full results can be seen in Appendix C.5.

Our method delivers superior results in the popular domain generalization tasks, as shown in Table 4.1. HYPO outperforms an extensive collection of common OOD generalization baselines on popular domain generalization datasets, including PACS, Office-Home, VLCS. For instance, on PACS, HYPO improves the best loss-based method by **1.1**%. Notably, this enhancement is non-trivial since we are not relying on specialized optimization algorithms such as SWAD (Cha et al., 2021). Later in our ablation, we show that coupling HYPO with SWAD can further boost the OOD generalization performance, establishing superior performance on this challenging task.

With multiple training domains, we observe that it is desirable to emphasize hard negative pairs when optimizing the inter-class separation. As depicted in Figure 4.3, the embeddings of negative pairs from the same domain but different classes (such as dog and elephant in art painting) can be quite close on the hypersphere. Therefore, it is more informative to separate such hard negative pairs. This can be enforced by a simple modification to the denominator of our variation loss (Eq. C.4 in Appendix C.4), which we adopt for multi-source domain generalization tasks.

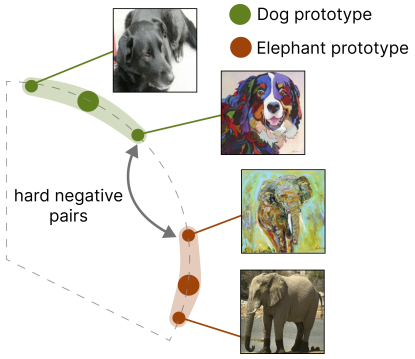


Figure 4.3: Illustration of hard negative pairs that share the same domain (art painting) but have different class labels.

Relations to PCL. PCL (Yao et al., 2022) adapts a proxy-based contrastive learning framework for domain generalization. We highlight several notable distinctions from ours: **(1)** While PCL offers no theoretical insights, HYPO is guided by theory. We provide a formal theoretical justification that our method reduces intra-class variation which is essential to bounding OOD generalization error (see Section 4.6); **(2)** Our loss function formulation is different and can be rigorously interpreted as shaping vMF distributions of hyperspherical embeddings (see Section 4.4.2), whereas PCL can not; **(3)** Unlike PCL (86.3% w/o SWAD), HYPO is able to achieve competitive performance (88.0%) without heavy reliance on special optimization SWAD (Cha et al., 2021), a dense and overfit-aware stochastic weight sampling (Izmailov et al., 2018) strategy for OOD generalization. As shown in Table 4.2, we also conduct experiments in conjunction with SWAD. Compared to PCL, HYPO achieves superior performance with **89**% accuracy, which further demonstrates its advantage.

Visualization of embedding. Figure 4.4 shows the UMAP (McInnes et al., 2018) visualization of feature embeddings for ERM (left) vs. HYPO (right). The embeddings are extracted from models trained on PACS. The red, orange, and green points are from the in-distribution, corresponding

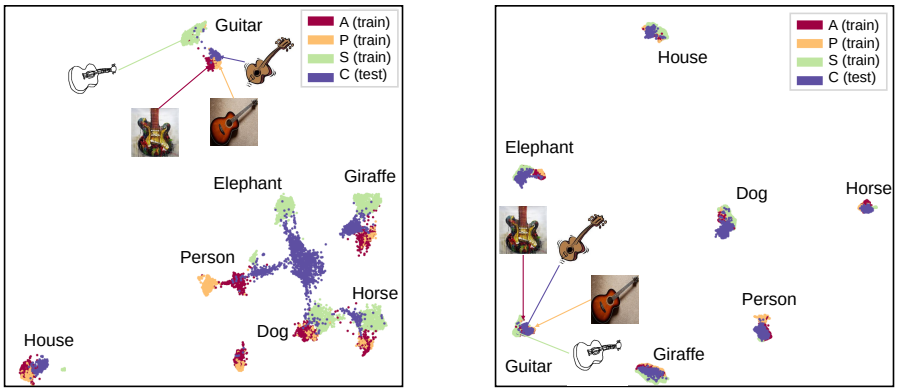
Algorithm	Art painting	Cartoon	Photo	Sketch	Average Acc. (%)
PCL w/ SGD (Yao et al., 2022)	88.0	78.8	98.1	80.3	86.3
HYPO w/ SGD (Ours)	87.2	82.3	98.0	84.5	**88.0**
PCL w/ SWAD (Yao et al., 2022)	90.2	83.9	98.1	82.6	88.7
HYPO w/ SWAD (Ours)	90.5	84.6	97.7	83.2	**89.0**

Table 4.2: Results for comparing PCL and HYPO with SGD-based and SWAD-based optimizations on the PACS benchmark. (*The performance reported in the original PCL paper Table 3 is implicitly based on SWAD).

to art painting (A), photo (P), and sketch (S) domains. The violet points are from the unseen OOD domain cartoon (C). There are two salient observations: **(1)** for any given class, the embeddings across domains $\mathcal{E}_{\text{all}}$ become significantly more aligned (and invariant) using our method compared to the ERM baseline. This directly verifies the low variation (*cf.* Equation 4.2) of our learned embedding. **(2)** The embeddings are well separated across different classes, and distributed more uniformly in the space than ERM, which verifies the high inter-class separation (*cf.* Equation 4.3) of our method. Overall, our observations well support the efficacy of HYPO.

Quantitative verification of intra-class variation. We provide empirical verification on intra-class variation in Figure 4.5, where the model is trained on PACS. We measure the intra-class *variation* with Sinkhorn divergence (entropy regularized Wasserstein distance). The horizontal axis (0)-(6) denotes different classes, and the vertical axis denotes different pairs of training domains ('P', 'A', 'S'). Darker color indicates lower Sinkhorn divergence. We can see that our method results in significantly lower intra-class variation compared to ERM, which aligns with our theoretical insights in Section 4.6.

Additional ablation studies. We provide additional experiments and ablations in the Appendix, including **(1)** results on other tasks from Do-



(a) ERM (**high variation**) (b) HYPO (**low variation**)

Figure 4.4: UMAP (McInnes et al., 2018) visualization of the features when the model is trained with `CE` vs. `HYPO` for PACS. The red, orange, and green points are from the in-distribution, which denote art painting (A), photo (P), and sketch (S). The violet points are from the unseen OOD domain cartoon (C).

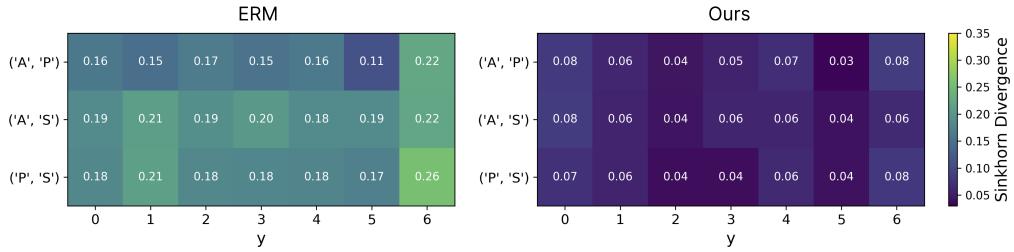


Figure 4.5: Intra-class variation for ERM (left) vs. HYPO (right) on PACS. For each class y, we measure the Sinkhorn Divergence between the embeddings of each pair of domains. Our method results in significantly lower intra-class variation across different pairs of training domains compared to ERM.

mainBed (Appendix C.6); **(2)** results on large-scale benchmarks such as ImageNet-100 (Appendix C.7); **(3)** ablation of different loss terms (Appendix C.8); **(4)** an analysis on the effect of τ and α (Appendix C.9).

4.6 Why HYPO Improves Out-of-Distribution Generalization?

In this section, we provide a formal justification of the loss function. Our main Theorem 4.6 gives a provable understanding of how the learning objective effectively reduces the variation estimate $\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}})$, thus directly reducing the OOD generalization error according to Theorem 4.5. For simplicity, we assume $\tau = 1$ and denote the prototype vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C \in \mathcal{S}^{d-1}$. Let $\mathcal{H} \subset \{h : \mathcal{X} \mapsto \mathcal{S}^{d-1}\}$ denote the function class induced by the neural network.

Theorem 4.6 (Variation upper bound using HYPO)**.** *When samples are aligned with class prototypes such that* $\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\mu}_{c(j)}^{\top}\mathbf{z}_j \geq 1-\epsilon$ *for some* $\epsilon \in (0,1)$*, then* $\exists \delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}}) \leq O(\epsilon^{1/3} + (\frac{\ln(2/\delta)}{N})^{1/6} + (\mathbb{E}_{\mathcal{D}}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N} \sup_{h\in\mathcal{H}} \sum_{i=1}^{N} \sigma_i \mathbf{z}_i^{\top} \boldsymbol{\mu}_{c(i)}])^{1/3}),$$

where $\mathbf{z}_j = \frac{h(\mathbf{x}_j)}{\|h(\mathbf{x}_j)\|_2}$*,* $\sigma_1, \ldots, \sigma_N$ *are Rademacher random variables and* $O(\cdot)$ *suppresses dependence on constants and* $|\mathcal{E}_{\text{avail}}|$*.*

Implications. In Theorem 4.6, we can see that the upper bound consists of three factors: the optimization error, the Rademacher complexity of the given neural network, and the estimation error which becomes close to 0 as the number of samples N increases. Importantly, the term ϵ reflects how sample embeddings are aligned with their class prototypes on the hyperspherical space (as we have $\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\mu}_{c(j)}^{\top}\mathbf{z}_j \geq 1-\epsilon$), *which is directly minimized by our proposed loss in Equation 4.5*. The above Theorem implies that when we train the model with the HYPO loss, we can effectively upper bound the intra-class variation, a key term for bounding OOD generation performance by Theorem 4.5. In Section C.8, we provide empirical verification of our bound by estimating $\hat{\epsilon}$, which is indeed

close to 0 for models trained with HYPO loss. We defer proof details to Appendix C.3.

Necessity of inter-class separation loss. We further present a theoretical analysis in Appendix C.10 explaining how our loss promotes inter-class separation, which is necessary to ensure the learnability of the OOD generalization problem. We provide a brief summary in Appendix C.3 and discuss the notion of OOD learnability, and would like to refer readers to Ye et al. (2021) for an in-depth and formal treatment. Empirically, to verify the impact of inter-class separation, we conducted an ablation study in Appendix C.8, where we compare the OOD performance of our method (with separation loss) vs. our method (without separation loss). We observe that incorporating separation loss indeed achieves stronger OOD generalization performance, echoing the theory.

4.7 Related Works

Out-of-distribution generalization. OOD generalization is an important problem when the training and test data are sampled from different distributions. Compared to domain adaptation (Daume III and Marcu, 2006; Ben-David et al., 2010; Tzeng et al., 2017; Kang et al., 2019; Wang et al., 2022e), OOD generalization is more challenging (Blanchard et al., 2011; Muandet et al., 2013; Gulrajani and Lopez-Paz, 2021; Bai et al., 2021b; Zhou et al., 2021a; Koh et al., 2021; Bai et al., 2021a; Wang et al., 2022d; Ye et al., 2022; Cha et al., 2022; Kim et al., 2023a; Guo et al., 2023; Dai et al., 2023; Tong et al., 2023), which aims to generalize to unseen distributions without any sample from the target domain. In particular, A popular direction is to extract domain-invariant feature representation. Prior works show that the invariant features from training domains can help discover invariance on target domains for linear models (Peters et al., 2016; Rojas-Carulla et al., 2018). IRM (Arjovsky et al., 2019) and its variants (Ahuja

et al., 2020; Krueger et al., 2021) aim to find invariant representation from different training domains via an invariant risk regularizer. Mahajan et al. (2021) propose a causal matching-based algorithm for domain generalization. Other lines of works have explored the problem from various perspectives such as causal discovery (Chang et al., 2020), distributional robustness (Sagawa et al., 2019; Zhou et al., 2020), model ensembles (Chen et al., 2023c; Rame et al., 2023), and test-time adaptation (Park et al., 2023; Chen et al., 2023b). In this chapter, we focus on improving OOD generalization via hyperspherical learning, and provide a new theoretical analysis of the generalization error.

Theory for OOD generalization. Although the problem has attracted great interest, theoretical understanding of desirable conditions for OOD generalization is under-explored. Generalization to arbitrary OOD is impossible since the test distribution is unknown (Blanchard et al., 2011; Muandet et al., 2013). Numerous general distance measures exist for defining a set of test domains around the training domain, such as KL divergence (Joyce, 2011), MMD (Gretton et al., 2006), and EMD (Rubner et al., 1998). Based on these measures, some prior works focus on analyzing the OOD generalization error bound. For instance, Albuquerque et al. (2019) obtain a risk bound for linear combinations of training domains. Ye et al. (2021) provide OOD generalization error bounds based on the notation of variation. In this work, we provide a hyperspherical learning algorithm that provably reduces the variation, thereby improving OOD generalization both theoretically and empirically.

Contrastive learning for domain generalization Contrastive learning methods have been widely explored in different learning tasks. For example, Wang and Isola (2020) analyze the relation between the alignment and uniformity properties on the hypersphere for unsupervised learning, while we focus on supervised learning with domain shift. Tapaswi

et al. (2019) investigates a contrastive metric learning approach for hyperspherical embeddings in video face clustering, which differs from our objective of OOD generalization. Von Kügelgen et al. (2021) provide theoretical justification for self-supervised learning with data augmentations. Recently, contrastive losses have been adopted for OOD generalization. For example, CIGA (Chen et al., 2022) captures the invariance of graphs to enable OOD generalization for graph data. CNC (Zhang et al., 2022a) is specifically designed for learning representations robust to spurious correlation by inferring pseudo-group labels and performing supervised contrastive learning. SelfReg (Kim et al., 2021a) proposes a self-supervised contrastive regularization for domain generalization with non-hyperspherical embeddings, while we focus on hyperspherical features with theoretically grounded loss formulations.

4.8 Conclusion

In this chapter, we present a theoretically justified algorithm for OOD generalization via hyperspherical learning. HYPO facilitates learning domain-invariant representations in the hyperspherical space. Specifically, we encourage low variation via aligning features across domains for each class and promote high separation by separating prototypes across different classes. Theoretically, we provide a provable understanding of how our loss function reduces the OOD generalization error. Minimizing our learning objective can reduce the variation estimates, which determine the general upper bound on the generalization error of a learnable OOD generalization task. Empirically, HYPO achieves superior performance compared to competitive OOD generalization baselines. We hope our work can inspire future research on OOD generalization and provable understanding.

Part II

Reliable Multi-Modal Models with Hyperspherical Embeddings

Chapter 5

Delving into Out-of-Distribution Detection with Vision-Language Representations

Publication Statement. This chapter is a joint work with Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. The paper version of this chapter appeared in NeurIPS 2022 (Ming et al., 2022a).

Recognizing out-of-distribution (OOD) samples is critical for machine learning systems deployed in the open world. The vast majority of OOD detection methods are driven by a single modality (*e.g.*, either vision or language), leaving the rich information in multi-modal representations untapped. Inspired by the recent success of vision-language pre-training, this work enriches the landscape of OOD detection from a single-modal to a multi-modal regime. Particularly, we propose Maximum Concept Matching (MCM), a simple yet effective zero-shot OOD detection method based on aligning visual features with textual concepts. We contribute in-depth analysis and theoretical insights to understand the effectiveness of MCM. Extensive experiments demonstrate that MCM achieves superior

performance on a wide variety of real-world tasks. MCM with vision-language features outperforms a common baseline with pure visual features on a hard OOD task with semantically similar classes by 56.60% (FPR95). Code is available at https://github.com/deeplearning-wisc/MCM.

5.1 Introduction

Out-of-distribution (OOD) detection is critical for deploying machine learning models in the wild, where samples from novel classes can naturally emerge and should be flagged for caution. Despite increasing attention, the vast majority of OOD detection methods are driven by single-modal learning (Hendrycks et al., 2020; Hsu et al., 2020; Jin et al., 2022; Shen et al., 2021; Xu et al., 2021a; Zhan et al., 2021; Zheng et al., 2020; Zhou et al., 2021b). For example, labels are typically encoded as one-hot vectors in image classification, leaving the semantic information encapsulated in texts largely unexploited. OOD detection relying on pure visual information can inherit the limitations, *e.g.*, when an OOD input is visually similar to in-distribution (ID) data yet semantically different from any ID class.

In this chapter, we delve into a new landscape for OOD detection, departing from the classic single-modal toward a *multi-modal* regime. While the motivation is appealing, a core challenge remains: *how to effectively utilize joint vision-language features for OOD detection?* In the visual domain, existing methods typically require good feature representations (Sehwag et al., 2021; Tack et al., 2020), and a distance metric under which OOD data points are relatively far away from the in-distribution (ID) data (Lee et al., 2018; Sun et al., 2022). These approaches, however, do not directly translate into the multi-modal regime. On the representation learning side, recent vision-language pre-training schemes such as CLIP (Radford

et al., 2021) and ALIGN (Jia et al., 2021) have emerged as promising alternatives for visual representation learning. The main idea is to align an image with its corresponding textual description in the feature space. While the resulting representations are powerful, OOD detection based on such aligned multi-modal features is still in its infancy.

We bridge the gap by exploring a distance-based OOD detection approach, leveraging the joint vision-language representations. Our method capitalizes on the compatibility between visual features and textual features. By defining the textual features as the "*concept prototypes*" for each ID class, we characterize OOD uncertainty by the distance from the visual feature to the closest ID prototype. That is, images closer to one of the textual embeddings of ID classes are more likely to be ID and vice versa. By a proper scaling of the distance, our proposed Maximum Concept Matching (**MCM**) score achieves strong ID-OOD separability (see Figure 5.1). MCM stands in contrast with the previous distance-based approaches, such as Mahalanobis (Lee et al., 2018), which defines class prototypes based on pure visual embeddings. Indeed, we show later in Section 5.5 that MCM (with multi-modal vision-language features) is far more competitive than Mahalanobis (with single-modal visual features). Moreover, while prior works of CLIP-based OOD detection (Esmaeilpour et al., 2022; Fort et al., 2021) rely on a set of candidate OOD labels, MCM is OOD-agnostic and alleviates the need for any prior information about test inputs.

Our work also advances the field by showcasing the promise of zero-shot OOD detection, which offers strong performance and generality without training on the ID samples. In particular, classic OOD detection methods often require training from scratch (Chen et al., 2021; Hendrycks et al., 2018) or fine-tuning (Fort et al., 2021; Huang and Li, 2021) on a given ID dataset. In this setting, a classifier and its companion OOD detector are good at only one task. Every new task (ID dataset) requires addi-

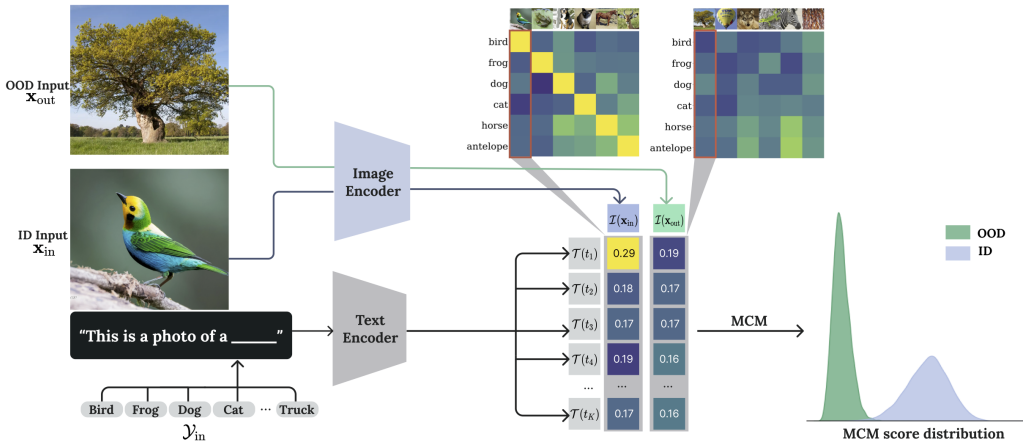


Figure 5.1: Overview of the proposed zero-shot OOD detection framework. The ID classification task is defined by a set of class labels $\mathcal{Y}_{\text{in}}$. The goal of OOD detection is to detect samples that do not belong to $\mathcal{Y}_{\text{in}}$. We view the textual embeddings of ID classes (wrapped by text templates) as concept prototypes. The OOD uncertainty of an input image can be characterized by the distance from visual features to the closest ID prototype. By properly scaling the distance, the MCM score achieves strong ID-OOD separability. See Section 5.3 for details.

tional training and brings additional computation and storage costs. In contrast, we show for the first time that: **(1)** MCM achieves superior performance across a wide variety of real-world tasks—with just *one single pre-trained model*. This is encouraging given that there is no training or any OOD information involved. **(2)** On the challenging ImageNet-1k benchmark, MCM's zero-shot OOD detection performance favorably matches and even outperforms strong task-specific baselines fine-tuned on BiT (Huang and Li, 2021) and ViT models (Fort et al., 2021). **(3)** MCM remains robust against hard OOD inputs, including both semantically hard OODs (Winkens et al., 2020) and spurious OODs (Ming et al., 2022c).

We summarize our main contributions as follows:

1. We propose MCM, a simple yet effective OOD detection method

based on aligned vision-language features. MCM offers several compelling advantages over other OOD detection methods: generalizable (one model supports many tasks), OOD-agnostic (no information required from OOD data), training-free (no downstream fine-tuning required), and scalable to large real-world tasks.

2. We conduct extensive experiments and show that MCM achieves superior performance on a wide range of real-world tasks. On ImageNet-1k, MCM achieves an average AUROC of 91.49%, outperforming methods that require training. Moreover, MCM remains competitive under challenging hard OOD evaluation tasks.

3. We provide in-depth empirical and theoretical analysis, providing insights to understand the effectiveness of MCM. We hope that this work will serve as a springboard for future works on OOD detection with multi-modal features.

5.2 Preliminaries

Contrastive vision-language pre-training. Compared to visual representation learning models such as ViT (Dosovitskiy et al., 2021), vision-language representation learning demonstrates superior performance on image classification tasks. For instance, CLIP (Radford et al., 2021) adopts a self-supervised contrastive objective (*i.e.*, InfoNCE loss (Van den Oord et al., 2018)) to align an image with its corresponding textual description in the feature space. Specifically, CLIP adopts a simple dual-stream architecture with one text encoder $\mathcal{T} : t \to \mathbb{R}^d$ (*e.g.*, Transformer (Vaswani et al., 2017)) and one image encoder $\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$ (*e.g.*, ViT (Dosovitskiy et al., 2021)). After pre-training on a dataset of 400 million text-image pairs, the joint vision-language embeddings of CLIP well associate objects in different modalities. Despite the promise, existing CLIP-like models

perform zero-shot classification in a *closed-world* setting. That is, it will match an input into a fixed set of categories, even if it is irrelevant (*e.g.*, a tree being predicted as a bird in Figure 5.1). This motivates our work to leverage the multi-modal representation for OOD detection, which is largely unexplored.

Zero-shot OOD detection. Given a pre-trained model, a classification task of interest is defined by a set of class labels/names $\mathcal{Y}_{\text{in}}$, which we refer to as the known (ID) classes. Here ID classes are defined *w.r.t.* the classification task of interest, instead of the classes used in pre-training. Accordingly, OOD is defined *w.r.t.* the ID classes, not the data distribution during pre-training. The goal of OOD detection is to (1) detect samples that do not belong to any of the known classes; (2) otherwise, assign test samples to one of the known classes. Therefore, the OOD detector can be viewed as a "safeguard" for the classification model. Formally, we denote the OOD detector as a binary function:

$$G(\mathbf{x}; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) : \mathcal{X} \rightarrow \{\text{in}, \text{out}\}$$

, where $\mathbf{x} \in \mathcal{X}$ denotes a test image. Our method is based on only the names of the given classes in $\mathcal{Y}_{\text{in}}$, and a pre-trained model. Different from standard supervised learning, there is no training on the ID samples involved, hence zero-shot.

5.3 OOD Detection via Concept Matching

We illustrate our approach in Figure 5.1, which derives the OOD detector $G(\cdot)$ based on *concept matching*. For a given task with label set $\mathcal{Y}_{\text{in}} = \{y_1, y_2, ..., y_K\}$, we can construct a collection of concept vectors $\mathcal{T}(t_i), i \in \{1, 2, ..., K\}$, where t_i is the text prompt "`this is a photo of a` $\langle y_i \rangle$" for

a label y_i. The concept vectors are represented by the embeddings of the text prompts.

For any test input image $\mathbf{x}'$, we can calculate the label-wise matching score based on the cosine similarity between the image feature $\mathcal{I}(\mathbf{x}')$ and the concept vector $\mathcal{T}(t_i)$: $s_i(\mathbf{x}') = \frac{\mathcal{I}(\mathbf{x}')\cdot\mathcal{T}(t_i)}{\|\mathcal{I}(\mathbf{x}')\|\cdot\|\mathcal{T}(t_i)\|}$. Formally, we define the maximum concept matching (**MCM**) score as:

$$S_{\text{MCM}}(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) = \max_i \frac{e^{s_i(\mathbf{x}')/\tau}}{\sum_{j=1}^{K} e^{s_j(\mathbf{x}')/\tau}}, \tag{5.1}$$

where τ is the temperature. For ID data, it will be matched to one of the concept vectors (textual prototypes) with a high score; and vice versa. Formally, our OOD detection function can be formulated as:

$$G(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) = \begin{cases} 1 & S_{\text{MCM}}(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) \geq \lambda \\ 0 & S_{\text{MCM}}(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) < \lambda \end{cases},$$

where by convention 1 represents the positive class (ID) and 0 indicates OOD. λ is chosen so that a high fraction of ID data (*e.g.*, 95%) is above the threshold. For samples that are classified as ID, one can obtain the class prediction based on the closest concept: $\hat{y} = \arg\max_{i\in[K]} s_i$.

Remark: (1) Our work differs from (and is complementary to) CLIP by focusing on OOD detection rather than (closed-world) zero-shot classification. We show new theoretical insights that softmax scaling plays a unique role in zero-shot OOD detection—improving the separability between ID and OOD data. This role has not been studied rigorously for zero-shot OOD detection. Readers familiar with CLIP may notice that MCM can be used for zero-shot classification in the closed world. This also makes MCM practically convenient for dual goals: detect OOD samples and assign ID data to one of the known classes. (2) Our method in principle is not limited to CLIP; it can be generally applicable for con-

trastive vision-language pre-training models that promote multi-modal feature alignment.

New insights on softmax scaling for zero-shot OOD detection. We provide theoretical justifications that softmax scaling improves the separability between ID and OOD data for CLIP-based OOD detection, which is *contrary* to models trained with cross-entropy (CE) loss. In particular, CLIP-like models are trained with a multi-modal contrastive loss, which maximizes the cosine similarity between an image and its textual description in the feature space. The resulting cosine similarity scores display strong *uniformity*[1] across labels, as evidenced in Figure 5.2 (right). Compared to OOD inputs, the gap between the maximum cosine similarity and the average is larger for ID inputs. However, the gap can be small when the number of ID classes increases where ID samples occur with lower highest cosine similarity. As a result, the highest cosine similarity for ID samples and OOD samples can be highly close (*c.f.* Figure 5.2 (left)).

Motivated by these observations, MCM employs softmax as a post hoc mechanism to **magnify** the difference. This is *fundamentally different from the softmax score derived from a model trained with cross-entropy loss*, which inherently maximizes the posterior $p(y|\mathbf{x})$ for the ground-truth label, and minimizes the probability for other labels. Unlike CLIP-like models, logit scores displaying uniformity would be heavily penalized by the CE loss. As a result, the logit score corresponding to the ground-truth label can already be significantly higher than other labels. Applying softmax on

[1]This can be explained both theoretically (Wang and Isola, 2020) and empirically (Wang and Liu, 2021). It has been shown that self-supervised contrastive learning with a smaller temperature (*e.g.*, initialized as 0.07 for CLIP) promotes uniform distribution for L_2-normalized features. Moreover, as CLIP features lie on a high-dimensional space (512 for CLIP-B/16 and 768 for CLIP-L/14), uniformly distributed points in a high-dimensional sphere tend to be equidistant to each other (Vershynin, 2018). Therefore, for OOD inputs, we observe approximately uniform cosine similarity with concept vectors.

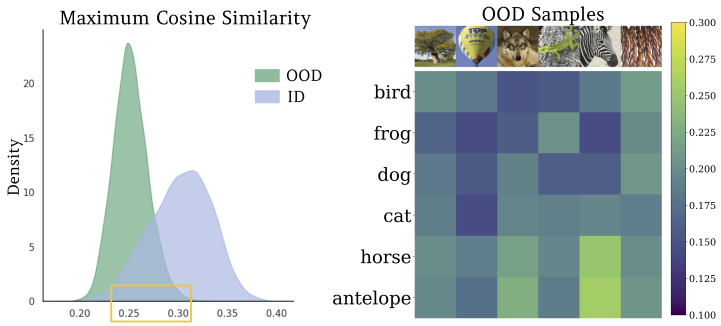


Figure 5.2: Left: Maximum cosine similarity for ID and OOD inputs. There exists overlapping regions (shown in yellow); Right: Cosine similarities between OOD inputs and ID concept vectors. For OOD inputs, the cosine similarities display uniformity.

the logit scores can exacerbate overconfident predictions, and reduce the separability between ID and OOD data (Liang et al., 2018). Indeed, for a model trained with cross-entropy loss, a logit-based score such as Energy (Liu et al., 2020) is shown to be much more effective than the softmax score.

Interestingly, for CLIP-like models, the trend is the opposite—applying softmax helps sharpen the uniform-like inner product scores, and increases the separability between ID and OOD data. To help readers better understand the insights, we first formalize our observations that OOD inputs trigger *similar cosine similarities* across ID concepts (Figure 5.2, right) as the following assumption:

Assumption 5.1. *Let* $z := \mathbb{1}\{y \in \mathcal{Y}_{in}\}$. $Q_{\mathbf{x}}$ *denotes the out-of-distribution* $\mathbb{P}_{\mathbf{x}|z=0}$ *(marginal distribution of* $\mathbf{x}$ *conditioned on* $z = 0$*). Assume* $\exists\, \delta > 0$ *such that*

$$Q_{\mathbf{x}}\left(\frac{1}{K-1}\sum_{i\neq\hat{y}}[s_{\hat{y}_2}(\mathbf{x}) - s_i(\mathbf{x})] < \delta\right) = 1,$$

where $\hat{y} := argmax_{i\in[K]} s_i(\mathbf{x})$ *and* $\hat{y}_2 := argmax_{i\neq\hat{y}, i\in[K]} s_i(\mathbf{x})$ *denote the indices of the largest and second largest cosine similarities for an OOD input* $\mathbf{x}$*.*

Now we provide formal guarantees that using softmax can provably reduce the false positive rate (FPR) compared to that without softmax.

Theorem 5.2. *Given a task with ID label set $\mathcal{Y}_{in} = \{y_1, y_2, ..., y_K\}$ and a pre-trained CLIP-like model $(\mathcal{T}, \mathcal{I})$. If $Q_{\mathbf{x}}$ satisfies Assumption 5.1, then there exists a constant $T = \frac{\lambda(K-1)\left(\lambda^{wo}+\delta-s_{\hat{y}_2}\right)}{K\lambda-1}$ such that for any temperature $\tau > T$, we have*

$$\mathrm{FPR}(\tau, \lambda) \leq \mathrm{FPR}^{wo}(\lambda^{wo}),$$

where $\mathrm{FPR}(\tau, \lambda)$ is the false positive rate based on softmax scaling with *temperature τ and detection threshold λ; $\mathrm{FPR}^{wo}(\lambda^{wo})$ is the false positive rate* without *softmax scaling based on threshold λ^{wo}. This suggests that applying softmax scaling with a moderate temperature results in superior OOD detection performance compared to that without softmax scaling. The proof is in Appendix D.1. Later in Section 5.5, we empirically verify on a real-world ImageNet dataset that our bound can indeed be satisfied in CLIP where the thresholds are chosen at 95% true positive rate.*

What MCM offers: Beyond theoretical insights, we would like to highlight several compelling advantages of our zero-shot OOD detection approach, owing to the strong pre-trained CLIP model:

- **Generalizable to many tasks**: Traditional OOD detection methods are based on a task-specific model. As a result, the OOD detector is not suitable for a realistic online scenario where the task changes from one to another. In contrast, we will show in Section 5.4 that MCM can perform a wide variety of OOD detection tasks, with just one single model. For a new task, only the names of the task's visual concepts $\mathcal{Y}_{\text{in}}$ are required.
- **OOD-agnostic**: Our method does not rely on any OOD information, and thus suits many real-world scenarios where one cannot

anticipate what the unknowns would be ahead of time. This also mitigates the shortcoming of a recent approach (Fort et al., 2021), which assumes that a set of unseen labels are given as some weak information about OOD data.

- **Training-free**: MCM enables OOD detection in a zero-shot fashion. This stands in contrast to the vast majority of OOD detection literature, which often requires training from scratch or fine-tuning to achieve competitive performance.

- **Scalable**: The contrastive vision-language pre-training paradigm makes MCM scalable to a large number of class labels and realistic high-resolution images.

We now proceed to the experimental results, demonstrating these advantages on real-world tasks.

5.4 A Comprehensive Analysis of MCM

5.4.1 Datasets and Implementation Details

Datasets. Most previous works on OOD detection only focus on small-scale datasets with blurry images such as CIFAR (Krizhevsky et al., 2009) and TinyImageNet (Le and Yang, 2015). With pre-trained models such as CLIP, OOD detection can be extended to more realistic and complex datasets. In this work, we scale up evaluations in terms of (1) image resolution, (2) dataset variety, and (3) number of classes. We consider the following ID datasets: Cub-200 (Wah et al., 2011), Stanford-Cars (Krause et al., 2013), Food-101 (Bossard et al., 2014), Oxford-Pet (Parkhi et al., 2012) and variants of ImageNet (Deng et al., 2009). For OOD test datasets, we use the same ones in (Huang and Li, 2021), including subsets of iNaturalist (Van Horn et al., 2018), Sun (Xiao et al., 2010), Places (Zhou et al.,

2017), and TEXTURE (Cimpoi et al., 2014). For each OOD dataset, the categories are not overlapping with the ID dataset. We also use subsets of ImageNet-1k for fine-grained analysis. For example, we construct ImageNet-10 that mimics the class distribution of CIFAR-10 but with high-resolution images. For hard OOD evaluation, we curate ImageNet-20, which consists of 20 classes semantically similar to ImageNet-10 (*e.g.*, dog (ID) vs. wolf (OOD)).

Model. In our experiments, we adopt CLIP (Radford et al., 2021) as the target pre-trained model, which is one of the most popular and publicly available vision-language models. Note that our method is not limited to CLIP; it can generally be applicable for contrastive vision-language pre-training models that promote multi-modal feature alignment. Specifically, we mainly use CLIP-B/16, which consists of a ViT-B/16 Transformer as the image encoder and a masked self-attention Transformer (Vaswani et al., 2017) as the text encoder. To indicate the input patch size in ViT models, we append "/x" to model names. We prepend -B, -L to indicate `Base` and `Large` versions of the corresponding architecture. For instance, ViT-B/16 implies the `Base` variant with an input patch resolution of 16×16. We also use CLIP-L/14 which is based on ViT-L/14 as a representative of large models. Unless specified otherwise, the temperature τ is 1 for all experiments. Details of the datasets, experimental setup, and hyperparameters are provided in Appendix D.2.

Metrics. For evaluation, we use the following metrics: (1) the false positive rate (FPR95) of OOD samples when the true positive rate of in-distribution samples is at 95%, (2) the area under the receiver operating characteristic curve (AUROC), and (3) ID classification accuracy (ID ACC).

Table 5.1: Zero-shot OOD detection with MCM score based on CLIP-B/16 with various ID datasets.

ID Dataset	OOD Dataset								Average	
	iNaturalist		SUN		Places		Texture			
	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
CUB-200	9.83	98.24	4.93	99.10	6.65	98.57	6.97	98.75	7.09	98.66
Stanford-Cars	0.05	99.77	0.02	99.95	0.24	99.89	0.02	99.96	0.08	99.89
Food-101	0.64	99.78	0.90	99.75	1.86	99.58	4.04	98.62	1.86	99.43
Oxford-Pet	2.85	99.38	1.06	99.73	2.11	99.56	0.80	99.81	1.70	99.62
ImageNet-10	0.12	99.80	0.29	99.79	0.88	99.62	0.04	99.90	0.33	99.78
ImageNet-20	1.02	99.66	2.55	99.50	4.40	99.11	2.43	99.03	2.60	99.32
ImageNet-100	18.13	96.77	36.45	94.54	34.52	94.36	41.22	92.25	32.58	94.48

5.4.2 Main Results

MCM supports a diverse collection of tasks while being zero-shot. We first show that zero-shot OOD detection with MCM is effective across a wide variety of tasks—with just *one single pre-trained model*. To showcase the versatility of MCM, we consider the seven ID datasets here. To the best of our knowledge, this is among the first attempts to showcase the efficacy under an expansive and diverse collection of ID datasets. The zero-shot OOD detection performance is summarized in Table 5.1. A salient observation is that MCM can achieve superior detection performance on many tasks. For example, using Stanford-Cars as ID, MCM yields an average FPR95 of **0.08**%. Considering that there are no training samples or OOD information involved, these results are very encouraging.

It can be also seen from Table 5.1 that MCM is promising, especially when the number of samples per ID class is limited in the training set. For example, there are only around 40 samples per class for Stanford-Cars, 100 for Oxford-Pet, and 30 for CUB-200. The sample scarcity makes OOD detection methods that rely on fine-tuning difficult. For example, after fine-tuning on Food-101, while the ID accuracy is increased from 86.3% to $92.5\% \uparrow$, OOD detection based on MSP is on par with MCM (99.5% vs. 99.4% in AUROC).

Table 5.2: OOD detection performance for ImageNet-1k (Deng et al., 2009) as ID.

Method	OOD Dataset								Average	
	iNaturalist		SUN		Places		Texture			
	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
Requires training (or w. fine-tuning)										
MOS(BiT)	9.28	98.15	40.63	92.01	49.54	89.06	60.43	81.23	39.97	90.11
Fort et al. (ViT-B)	15.07	96.64	54.12	86.37	57.99	85.24	53.32	84.77	45.12	88.25
Fort et al. (ViT-L)	15.74	96.51	52.34	87.32	55.14	86.48	51.38	85.54	43.65	88.96
Energy (CLIP-B)	21.59	95.99	34.28	93.15	36.64	91.82	51.18	88.09	35.92	92.26
Energy (CLIP-L)	10.62	97.52	30.46	93.83	32.25	93.01	44.35	89.64	29.42	93.50
MSP (CLIP-B)	40.89	88.63	65.81	81.24	67.90	80.14	64.96	78.16	59.89	82.04
MSP (CLIP-L)	34.54	92.62	61.18	83.68	59.86	84.10	59.27	82.31	53.71	85.68
Zero-shot (no training required)										
MCM (CLIP-B)	30.91	94.61	37.59	92.57	44.69	89.77	57.77	86.11	42.74	90.77
MCM (CLIP-L)	28.38	94.95	29.00	94.14	35.42	92.00	59.88	84.88	38.17	91.49

MCM scales effectively to large datasets. To examine the scalability of MCM, we compare it with recent competitive OOD detection methods (Fort et al., 2021; Huang and Li, 2021) on the ImageNet-1k dataset (ID) in Table 5.2. We observe the following trends:

- Larger models lead to superior performance. Compared with CLIP-B, MCM based on CLIP-L reduces FPR95 by 4.57%. Zero-shot ID classification accuracy is also improved by 6.27% with the larger model, reaching 73.28% (see Appendix D.4). This suggests that larger models are endowed with a better representation quality, which benefits both ID classification and OOD detection with MCM. Our finding echos with the recent observations (Vaze et al., 2022) that higher ID classification accuracy is correlated with stronger OOD detection performance.

- MOS (Huang and Li, 2021) recently demonstrated competitive performance on ImageNet-1k, which requires model fine-tuning based on BiT (Kolesnikov et al., 2020). In contrast, we show that MCM (CLIP-L) outperforms MOS by 1.38% in AUROC while being zero-shot (training-free).

- MCM shares a softmax scaling function with the classic (visual)

confidence-based score MSP (Hendrycks and Gimpel, 2017). To implement MSP, we adopt the commonly used linear probe approach by fine-tuning a linear layer on frozen visual features of CLIP. After fine-tuning, ID accuracy significantly improves, reaching 84.12% (CLIP-L). Interestingly, the OOD detection performance of MSP is worse than MCM by 15.54% in FPR95. Under the same model fine-tuned with linear probing, we observe that the Energy score outperforms MSP, corroborating findings in (Liu et al., 2020). We investigate more in Section 5.5.

- Recently, Fort *et al.* (Fort et al., 2021) explore small-scale OOD detection by fine-tuning the full ViT model. When extended to large-scale tasks, we find that MCM still yields superior performance under the same image encoder configuration (ViT-B or ViT-L). This further highlights the advantage of utilizing vision-language joint embeddings for large-scale visual OOD detection.

MCM benefits hard OOD detection. Going beyond, we investigate whether MCM is still effective for hard OOD inputs. We consider the following two categories of hard OOD:

- **Semantically hard OOD**: OOD samples that are semantically similar to ID samples are particularly challenging for OOD detection algorithms (Winkens et al., 2020). To evaluate hard OOD detection tasks in realistic settings, here we consider ImageNet-10 (ID) vs. ImageNet-20 (OOD) and vice versa. The pair consists of high-resolution images with semantically similar categories such as dog versus wolf. As shown in Table 5.3, MCM outperforms Mahalanobis (Lee et al., 2018) by **73.32%** in FPR95 for ImageNet-10 (ID) vs. ImageNet-20 (OOD) and **30.12%** vice versa.

Table 5.3: Performance comparison on **hard OOD detection** tasks. MCM is competitive on all three hard OOD tasks without training involved. MSP (based on fine-tuned CLIP) does not further improve performance.

Method	**ID** **OOD**	ImageNet-10 ImageNet-20	ImageNet-20 ImageNet-10	Waterbirds Spurious OOD
		FPR95 / AUROC	FPR95 / AUROC	FPR95 / AUROC
MSP (fine-tuning)		9.38 / 98.31	12.51 / 97.70	39.57 / 90.99
Mahalanobis (visual only)		78.32 / 85.60	43.03 / 89.94	2.21 / 99.55
MCM (zero-shot)		5.00 / 98.71	12.91 / 98.09	5.87 / 98.36

- **Spurious OOD**: Modern neural networks can exploit spurious correlations for predictions (Beery et al., 2018). For example, in the Waterbirds dataset (Sagawa et al., 2019), there exist spurious correlations between the habitat (*e.g.*, water) and bird types. A recent work (Ming et al., 2022c) proposes a new type of hard OOD named spurious OOD and shows that most OOD detection methods perform much worse for spurious OOD inputs compared to non-spurious inputs. The spurious OOD inputs are created to share the same background (*i.e.*, water) as ID data but have different object labels (*e.g.*, a boat rather than a bird). See Appendix D.3 for illustrations. The results are shown in Table 5.3. It has been shown that CLIP representations are robust to distributional shifts (Radford et al., 2021). Therefore, while prior works (Ming et al., 2022c) show that spurious OOD inputs are challenging for methods based on ResNet (He et al., 2016), MCM and Mahalanobis scores based on pre-trained CLIP perform much better. On the other hand, fine-tuning exposes the model to the training set containing spurious correlations. As a result, MSP performs much worse than MCM (39.57% vs. 5.87% in FPR95).

MCM outperforms CLIP-based baselines. Two recent works also use CLIP embeddings for OOD detection (Esmaeilpour et al., 2022; Fort et al.,

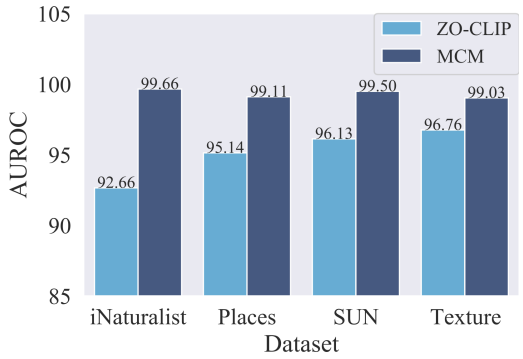


Figure 5.3: Comparison with a candidate label-based score ZO-CLIP on ImageNet-20, based on our implementation of (Esmaeilpour et al., 2022). Implementation details are deferred to Appendix D.5.1.

2021). However, fundamental limitations exist for both works. Fort *et al.* (Fort et al., 2021) assume that a candidate OOD label set $\mathcal{Y}_C$ is known, and used $\sum_{y\in\mathcal{Y}_C}\hat{p}(y|\mathbf{x})$ for OOD detection. Here the predictive probability $\hat{p}(y|\mathbf{x})$ is obtained by normalizing the inner products over $|\mathcal{Y}_{\text{in}}|+|\mathcal{Y}_C|$ classes. While applying softmax converts any vector to probabilities, as we show in Section 5.3, the converted probabilities do not necessarily correspond to $\mathbb{P}(\text{OOD}|\mathbf{x})$. Moreover, obtaining such an OOD label set is typically not feasible, which fundamentally limits its applicability. A recent work (Esmaeilpour et al., 2022) realizes this idea by training an extra text decoder on top of CLIP's image encoder to generate candidate labels. However, Esmaeilpour et al. (2022) cannot guarantee the generated labels are non-overlapping with the ID labels.

We enhance the baseline with a stronger decoder and a filter module (see Appendix D.5.1). As shown in Figure 5.3, MCM outperforms the enhanced baseline on all OOD datasets. Moreover, MCM is much simpler to use—alleviating the need for an OOD label set or training an additional caption generator. In contrast, the caption generator's performance largely affects OOD detection. Poor caption quality degenerates the OOD detection performance of candidate label-based methods. Moreover, ob-

taining a reliable caption generator for *any input image* can significantly increase the computational overhead.

5.5 Discussion: A Closer Look at MCM

Empirical verification on the role of softmax. In Section 5.3, we prove that softmax scaling on cosine similarity scores with a moderate τ improves the ID-OOD separability. Here we empirically verify our theoretical results. As shown in Figure 5.4, compared to directly using the maximum cosine similarity without softmax (leftmost figure), softmax scaling with a temperature $\tau = 1$ significantly improves the performance by 22.6% in FPR95, and further increasing τ (*e.g.*, $\tau = 10$) leads to similar performance. The results are based on ImageNet-100 (ID) versus iNaturalist (OOD).

Now, we verify if our theoretical bound (*c.f.* Theorem 5.2) is satisfied empirically as well in Figure 5.4. From the leftmost figure, we can estimate $\lambda^{\text{wo}} \approx 0.26$, $\delta \approx 0.03$, and $s_{\hat{y}_2} \approx 0.23$. By checking the third figure ($\tau = 1$ is the temperature value we use for most experiments), we approximate $\lambda \approx 0.011$. As $K = 100$, we plug in the values and obtain the lower bound

$$T = \frac{\lambda(K-1)\left(\lambda^{\text{wo}} + \delta - s_{\hat{y}_2}\right)}{K\lambda - 1} \approx 0.65$$

. Since $\tau = 1 > 0.65$, by Theorem 5.2, applying softmax scaling with $\tau = 1$ is provably superior to without softmax scaling for OOD detection.

Are vision-language features better than visual feature alone? MCM can be interpreted as a distance-based approach—images that are closer to one of the K class prototypes are more likely to be ID and vice versa. Here the class prototypes are defined based on a textual encoder. Alternatively, one can define the class prototypes based on visual features.

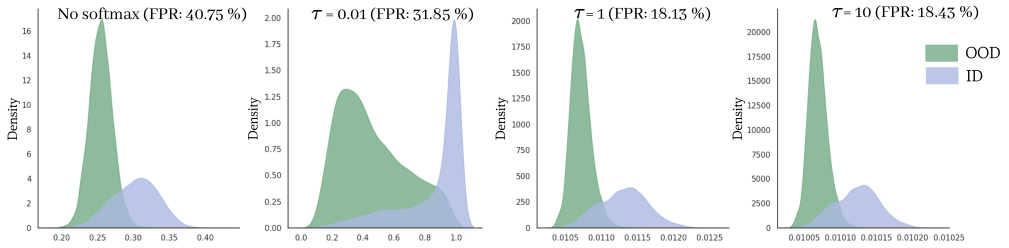


Figure 5.4: The influence of softmax scaling and temperature. We use ImgeNet-100 (ID) vs. iNaturalist (OOD). Softmax scaling with a moderate temperature significantly improves FPR95.

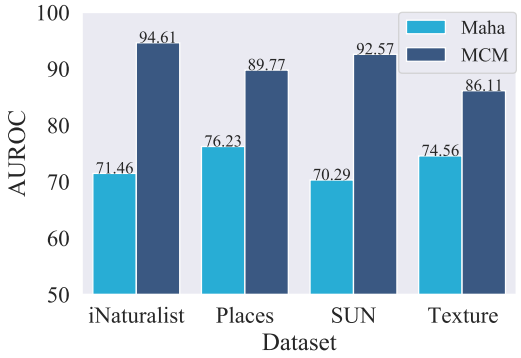


Figure 5.5: MCM vs. Mahalanobis (Maha) score on ImageNet-1k.

For example, Mahalanobis (Lee et al., 2018) defines a class prototype as the average of visual embeddings for images belonging to the same class. This raises the question whether MCM (with *multi-modal* vision-language features) is better than Mahalanobis (with *single-modal* visual feature). For a fair comparison, we use the same ViT image encoder from CLIP-B. Both MCM and Mahalanobis extract visual features from the penultimate layer. On ImageNet-1k, Mahalanobis displays a limited performance, with 73.14% AUROC averaged across four OOD test datasets (**90.77%** for MCM), as shown in Figure 5.5. From a practical perspective, Mahalanobis requires computing the inverse covariance matrix, which can be both computationally expensive and inaccurate when the number of samples is scarce and the number of ID classes grows. In contrast, MCM is easier to use and more robust.

Table 5.4: Zero-shot OOD detection of $S^{\text{wo}}_{\text{MCM}}$ based on CLIP-B/16.

ID Dataset	OOD Dataset								Average	
	iNaturalist		SUN		Places		Texture			
	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
Stanford-Cars	0.00	100	0.02	99.99	0.26	99.94	0.00	100	0.07	99.98
Food-101	0.56	99.86	0.09	99.95	0.49	99.88	8.33	97.44	2.37	99.28
Oxford-Pet	0.02	99.98	0.05	99.97	0.20	99.94	0.27	99.91	0.14	99.95
ImageNet-10	2.40	99.42	1.79	99.55	2.83	99.32	1.86	99.56	2.22	99.46
ImageNet-20	14.96	97.87	13.10	97.97	14.21	97.67	13.46	97.32	13.93	97.71
ImageNet-1k	61.66	89.31	64.39	87.43	63.67	85.95	86.61	71.68	69.08	83.59

MCM without softmax scaling. In Section 5.3, we provide theoretical justifications for the necessity of softmax scaling for CLIP-like models. To further verify our observations empirically, we show OOD detection performance based on the maximum cosine similarity score

$$S^{\text{wo}}_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) := \max_{i \in [K]} s_i(\mathbf{x}')$$

. The results are shown in Table 5.4. For easy tasks such as Food-101 (Krause et al., 2013), Stanford-Cars (Krause et al., 2013), and Oxford-Pet (Parkhi et al., 2012) as ID, the performance of maximum cosine similarity score is similar to MCM (see Table 5.1 and Table 5.2). However, for more challenging tasks such as ImageNet-20 and ImageNet-1k, MCM significantly outperforms that without softmax scaling. For example, the average FPR95 is improved by **11.33%** on ImageNet-20 and **26.34%** on ImageNet-1k, which highlights the necessity of a proper scaling function for CLIP-based OOD detection.

MCM for ResNet-based CLIP models. Our main results are based on the CLIP model with ViT image encoder. We additionally investigate the effectiveness of MCM on ResNet-based CLIP. Specifically, we use RN50x4 (178.3M), which shares a similar number of parameters as CLIP-B/16 (149.6M). The results are shown in Table 5.5. We can see that MCM still shows promising results with ResNet-based CLIP models, and the performance is comparable between RN50x4 and CLIP-B/16 (89.97 vs. 90.77

in AUROC).

Table 5.5: Comparison with ResNet-based CLIP models on ImageNet-1k (ID).

Model	OOD Dataset								Average	
	iNaturalist		SUN		Places		Texture			
	FPR95↓	**AUROC**↑	**FPR95**↓	**AUROC**↑	**FPR95**↓	**AUROC**↑	**FPR95**↓	**AUROC**↑	**FPR95**↓	**AUROC**↑
RN50x4	44.51	91.51	35.11	92.84	43.74	89.60	57.73	85.93	45.27	89.97
CLIP-B/16	30.91	94.61	37.59	92.57	44.69	89.77	57.77	86.11	42.74	90.77

Effect of prompt ensembling. We examine MCM's performance with prompt ensembling. For example, Radford *et al.* (Radford et al., 2021) create 80 possible prompts according to the image modalities and nuances in ImageNet. We experiment with the two prompt sets, one of size 80 as in (Radford et al., 2021), and our own set of 5 prompts. Ensembles are obtained by averaging the textual features. As expected, using ensembles increases the ID classification accuracy on ImageNet-1k (2% with CLIP-B and 3% with CLIP-L). For OOD detection, the average FPR95 is reduced from 38.17% with the default prompt to 35.23%↓ with an ensemble of five prompts shown in Table 5.6. In addition, the detection performance with 5 prompts is slightly better than with 80 prompts. Note that prompt ensembling does not increase the inference-time cost, as the textual embeddings (across many prompts) can be pre-calculated and averaged into a single embedding.

A photo of a <label>.
A blurry photo of a <label>.
A photo of many <label>.
A photo of the large <label>.
A photo of the small <label>.

Table 5.6: The five prompt templates.

5.6 Related Works

OOD detection in computer vision. For open-world multi-class classification, the goal of OOD detection is to derive a binary ID-OOD classifier

along with a multi-class classification model for visual inputs. A plethora of methods has been proposed for deep neural networks (Yang et al., 2021b), including generative model-based methods (Cai and Li, 2023; Ge et al., 2017; Kirichenko et al., 2020; Nalisnick et al., 2019; Neal et al., 2018; Oza and Patel, 2019; Ren et al., 2019; Serrà et al., 2020; Xiao et al., 2020), and discriminative-model based methods. For the latter category, an OOD score can be derived based on the softmax output (Bendale and Boult, 2016; DeVries and Taylor, 2018; Hein et al., 2019; Hendrycks and Gimpel, 2017; Hsu et al., 2020; Huang and Li, 2021; Liang et al., 2018; Yang et al., 2021a), energy-based score (Du et al., 2022b; Liu et al., 2020; Ming et al., 2022b; Sun et al., 2021; Sun and Li, 2022; Wang et al., 2021), gradient information (Huang et al., 2021a), or the feature embeddings (Du et al., 2022a; Lee et al., 2018; Sastry and Oore, 2020; Sehwag et al., 2021; Sun et al., 2022; Tack et al., 2020; Winkens et al., 2020) of a model. Morteza *et al.* (Morteza and Li, 2022), Fang *et al.* (Fang et al., 2022), and Bitterwolf *et al.* (Bitterwolf et al., 2022) provided theoretical analysis for OOD detection. Recent works (Roy et al., 2022; Wang et al., 2022c) also explored OOD detection for long-tailed distributions. Works insofar have mostly focused on OOD detection for a task-specific model using only visual information. In contrast, we explore a novel paradigm of zero-shot OOD detection that incorporates rich textual information and can perform a wide variety of tasks.

OOD detection in natural language processing. Distribution shifts can occur due to the change of topics and domains, unexpected user utterances, *etc*. Challenging benchmarks (Koh et al., 2021) and characterization of distributional shifts (Arora et al., 2021) have been proposed in recent years. Compared to early language models such as ConvNets and LSTM (Hochreiter and Schmidhuber, 1997), pre-trained language models are more robust to distribution shifts and more effective at identifying

OOD instances (Hendrycks et al., 2020; Podolskiy et al., 2021; Xu et al., 2021a). Various algorithmic solutions are proposed to handle OOD detection, including outlier exposure (Hu and Khan, 2021), model ensembling (Li et al., 2021d), data augmentation (Chen and Yu, 2021; Zhan et al., 2021; Zheng et al., 2020), contrastive learning (Jin et al., 2022; Zhou et al., 2021b), and an auxiliary module that incorporates domain labels (Shen et al., 2021). Tan *et al.* (Tan et al., 2019) also explore zero-shot OOD detection for text classification tasks. However, prior works focus on pure natural language processing (NLP) settings, while we explore utilizing textual embeddings for zero-shot *visual* OOD detection.

Vision-language models. Utilizing large-scale pre-trained vision-language models for multimodal downstream tasks has become an emerging paradigm with remarkable performance (Gu et al., 2020; Uppal et al., 2022). In general, two types of architectures exist: single-stream models like VisualBERT (Li et al., 2019) and ViLT (Kim et al., 2021b) feed the concatenated text and visual features into a single transformer-based encoder; dual-stream models such as CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and FILIP (Yao et al., 2021) use separate encoders for text and image and optimize with contrastive objectives to align semantically similar features in different modalities. In particular, CLIP enjoys popularity due to its simplicity and strong performance. CLIP-like models inspire numerous follow-up works (Li et al., 2022c; Zhang et al., 2021b; Zhou et al., 2022b), which aim to improve data efficiency and better adaptation to downstream tasks. This chapter adopts CLIP as the target pre-trained model, but our approach can be generally applicable to contrastive models that promote vision-language alignment.

Multi-modal OOD detection. Exploring textual information for visual OOD detection is a new area with limited existing works. Fort *et al.* (Fort et al., 2021) propose to feed the potential OOD labels to the textual en-

coder of CLIP (Radford et al., 2021). Recently, Esmaeilpour *et al.* (Esmaeilpour et al., 2022) propose to train a label generator based on the visual encoder of CLIP and use the generated labels for OOD detection. While both works rely on a set of candidate OOD labels, MCM is OOD-agnostic and alleviates the need for prior information on OOD. Moreover, prior works (Esmaeilpour et al., 2022; Radford et al., 2021) only focus on small-scale inputs. We largely expand the scope to a wide range of large-scale realistic datasets, and show new theoretical insights.

5.7 Conclusion

In this work, we delve into a new landscape for OOD detection, departing from the classic single-modal toward a multi-modal regime. By viewing the textual features as the "concept prototypes", we explore a new OOD detection approach MCM, based on the joint vision-language representations. Unlike the majority of OOD detection methods, MCM offers several compelling advantages: training-free, generalizable to many tasks, scalable to hundreds of classes, and does not require any prior information on OOD inputs. Moreover, we provide theoretical guarantees on how softmax scaling provably improves zero-shot OOD detection. We investigate the effectiveness of MCM on a wide range of large-scale realistic tasks, including several types of hard OOD datasets. Lastly, we demonstrate the advantage of vision-language features over pure visual features for OOD detection. We hope our work will inspire future research toward multi-modal OOD detection.

Chapter 6

How Does Fine-Tuning Impact Out-of-Distribution Detection for Vision-Language Models?

Publication Statement. This chapter is a joint work with Yixuan Li. The paper version of this chapter appeared in IJCV 2023 (Ming and Li, 2023).

Recent large vision-language models such as CLIP have shown remarkable out-of-distribution (OOD) detection and generalization performance. However, their zero-shot in-distribution (ID) accuracy is often limited for downstream datasets. Recent CLIP-based fine-tuning methods such as prompt learning have demonstrated significant improvements in ID classification and OOD generalization where OOD labels are available. Nonetheless, it remains unclear whether the model is reliable to semantic shifts without OOD labels. In this chapter, we aim to bridge the gap and present a comprehensive study to understand how fine-tuning impact OOD detection for few-shot downstream tasks. By framing OOD detection as multi-modal concept matching, we establish a connection between fine-tuning methods and various OOD scores. Our results suggest

that a proper choice of OOD scores is essential for CLIP-based fine-tuning. In particular, the maximum concept matching (MCM) score provides a promising solution consistently. We also show that prompt learning demonstrates the state-of-the-art OOD detection performance over the zero-shot counterpart.

6.1 Introduction

Machine learning (ML) is undergoing a paradigm shift with the rise of models that are trained on massive data and are adaptable to a wide range of downstream tasks. Popular pre-trained large vision-language models (Radford et al., 2021; Jia et al., 2021; Yao et al., 2021; Li et al., 2022c) demonstrate remarkable performance, and allow researchers without extensive computation power to benefit from these models. It is now the common practice of the ML community to adopt pre-trained models for transfer learning on downstream tasks rather than learning from scratch. Despite the promise, the safety risks of these large pre-trained models can be potentially inherited by all the fine-tuned models. Without appropriately understanding the safety risks, development on top of pre-trained models can exacerbate and propagate safety concerns writ large, causing profound impacts on society.

In response to these urgent challenges, the overall objective of this chapter is to systematically understand the out-of-distribution risks of learning with pre-trained vision-language models. This chapter seeks to address the research question that arises in building responsible and ethical AI models: *How does fine-tuning influence out-of-distribution (OOD) detection for large vision-language models?* Detecting OOD samples is crucial for machine learning models deployed in the open world, where samples from unseen classes naturally emerge, and failure to detect them can have severe consequences. Despite increasing attention (Yang et al.,

2021b), OOD detection research for large vision-language models has been scant. Among the most recent works, Ming et al. (2022a) investigated training-free OOD detection based on the pre-trained CLIP model. However, the impact of fine-tuning on OOD detection has been unexplored in the vision-language literature.

In this chapter, we bridge the gap by investigating how fine-tuning large vision-language models affects OOD detection. Parameter-efficient fine-tuning methods have been popularized in recent years. In particular, prompt learning (Zhou et al., 2022b,c) optimizes learnable word embeddings of the prompts, while adaptors directly optimize the internal feature representations (Gao et al., 2023; Zhang et al., 2022b). Both methods are parameter-efficient as image and text encoders are frozen during fine-tuning, and have shown significant improvement for few-shot in-distribution (ID) classification. Complementary to existing research, we focus on OOD detection for fine-tuned models using multi-modal concept matching. At the core of the concept matching framework, we use the few-shot ID training set and textual descriptions of the labels to derive a set of visual and textual features that represent the typical features for each ID class. We can measure OOD uncertainty based on the distance between the input feature and the nearest ID prototype.

Based on the concept matching framework, we then present a comprehensive and systematic study to explore how different parameter-efficient fine-tuning methods impact OOD detection performance, and contribute unexplored findings to the community. We disentangle various aspects such as adaptation methods and OOD scoring functions. Interestingly, we observe that parameter-efficient fine-tuning can significantly improve OOD reliability compared to zero-shot CLIP models. In particular, prompt learning methods exhibit very competitive performance when coupled with the maximum concept matching (MCM) score (Ming et al., 2022a).

Furthermore, we delve deeper into prompt learning and analyze how

the pre-trained features are modified during fine-tuning, and how it impacts OOD detection as a consequence. We study the impact of shots, architectures, and explore the effects of prompt learning on various downstream tasks, including the challenging ImageNet-1k (ID) benchmark. Our results demonstrate that prompt learning perturbs the pre-trained feature space that benefits both ID and OOD performance. More encouragingly, the trend holds consistently across different settings, highlighting its potential for reliable fine-tuning in vision-language modeling.

We summarize the contributions of this work as follows:

- We provide a timely and systematic study on how CLIP-based fine-tuning influences OOD detection in the few-shot setting. Our study disentangles various factors, including adaptation methods and OOD scoring functions.
- We present novel evidence that parameter-efficient fine-tuning does not deteriorate pre-trained features. Instead, they can improve both ID and OOD performance with a proper OOD scoring function, especially the MCM score. We show that prompt learning consistently demonstrates the state-of-the-art OOD detection performance over the zero-shot counterpart.
- We provide an in-depth analysis of prompt learning's impact on the feature space for OOD detection and conduct comprehensive ablations across datasets, architectures, and the number of shots with various OOD detection scores.

6.2 Related works

Parameter-efficient fine-tuning of vision-language models. Large-scale vision-language models have shown impressive performance on various downstream tasks (Radford et al., 2021; Jia et al., 2021; Yao et al., 2021; Li

et al., 2022c). These models learn transferable feature representations via pre-training on web-scale heterogeneous datasets. However, as downstream datasets can have a limited number of samples, adapting these large models in a parameter and data-efficient manner is crucial for effective knowledge transfer. Recent works propose various ways to tackle this challenge. Zhou et al. (2022c) propose to tune a set of soft prompts (Li and Liang, 2021; Lester et al., 2021) while freezing the encoders of CLIP. Zhou et al. (2022b) aims to improve the generalization ability of CoOp by introducing a meta-network that learns input-dependent tokens. Huang et al. (2022a) propose to learn prompts in an unsupervised manner while TPT (Manli et al., 2022) uses test-time prompt tuning to learn adaptive prompts on the fly. Beyond textual prompt learning, Bahng et al. (2022) propose to tune visual prompts for CLIP-based fine-tuning. Another line of work focuses on adaptor-style fine-tuning, where instead of tuning prompts, the feature embedding is directly optimized using an adaptor module (Gao et al., 2023; Zhang et al., 2022b; Udandarao et al., 2023). Prior works demonstrate significant improvement over zero-shot CLIP for few-shot ID classification and OOD generalization where OOD labels are given. However, it is unclear how reliable these parameter-efficient fine-tuning methods are for OOD detection tasks. Our work bridges this gap and explores how fine-tuning impacts OOD detection for few-shot downstream datasets.

OOD detection with vision-language representations. A plethora of OOD detection methods have been proposed on visual inputs (Lee et al., 2018; Liang et al., 2018; Hendrycks et al., 2019; Tack et al., 2020; Sun et al., 2022; Ming et al., 2022b; Du et al., 2022a; Wang et al., 2022b; Ming et al., 2023). With the rise of large-scale pre-trained models on vision language inputs, an increasing number of works utilize textual information for visual OOD detection and demonstrate promising performance. Fort et al.

(2021) propose a scheme where pre-trained CLIP models are provided with candidate OOD labels for each target dataset, and show that the output probabilities summed over the OOD labels effectively capture OOD uncertainty. Without the assumption of OOD labels, Esmaeilpour et al. (2022) propose to train a decoder based on the visual encoder of CLIP to generate candidate labels for OOD detection. However, training a high-quality decoder incurs significant computational costs and requires extra data. While both Esmaeilpour et al. (2022) and Radford et al. (2021) focus on small-scale inputs, Ming et al. (2022a) propose an OOD label-free method MCM which demonstrates promising results on a wide range of large-scale and challenging tasks (Ming et al., 2022c). However, Ming et al. (2022a) only investigate pre-trained CLIP models. For multi-modal OOD detection benchmarks, Bitterwolf et al. (2023) curate a new OOD test set for ImageNet-1k while Gu et al. (2023) provide new OOD datasets for document understanding. In contrast, our work focuses on the impact of parameter-efficient fine-tuning methods for OOD detection in few-shot downstream tasks, which has not been explored.

6.3 Preliminaries

Contrastive vision-language models. Recent large vision-language models have shown great potential for various computer vision tasks. In this chapter, we focus on CLIP-like models (Radford et al., 2021; Yao et al., 2021), which adopt a dual-stream architecture with one text encoder $f : t \to \mathbb{R}^d$ and one image encoder $g : \mathbf{x} \to \mathbb{R}^d$. CLIP is pre-trained on a massive web-scale image-caption dataset with a multi-modal contrastive loss that promotes the alignment of features from different modalities. CLIP learns transferable feature representations and demonstrates promising zero-shot generalization performance (Fort et al., 2021). Despite the promise, existing vision-language models perform zero-shot classification in a *closed-*

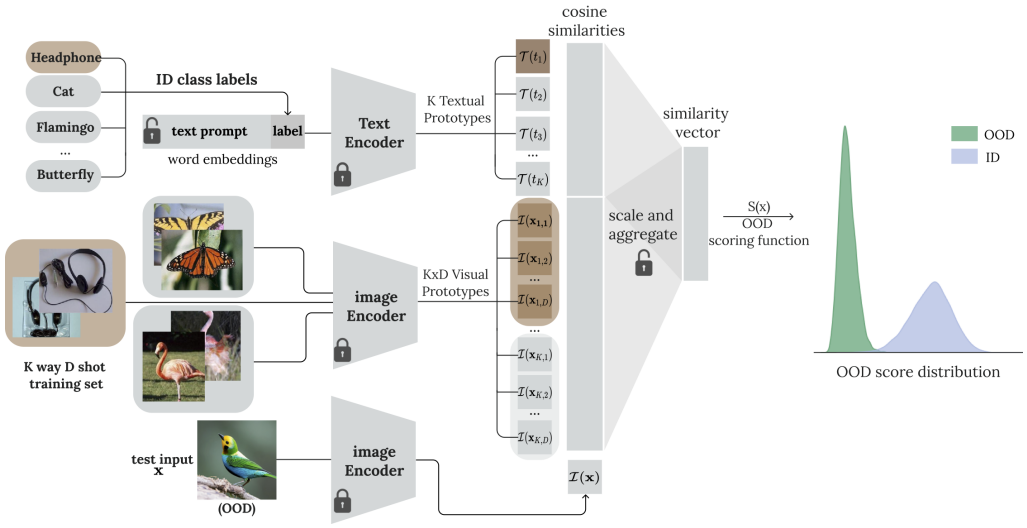


Figure 6.1: A unified pipeline for OOD detection with parameter-efficient fine-tuning of CLIP models on few-shot datasets. Given ID text labels $\mathcal{Y}_{\text{in}}$ and a few-shot training set, we view the textual and visual embeddings of ID classes as concept prototypes in the feature space. The OOD uncertainty of an input image can be characterized by the distance from its visual feature to the closest ID prototype from both modalities. See Section 6.4 for details.

world setting. That is, it will match an input into a fixed set of categories, even if it is irrelevant. For example, a bird in Figure 6.1 can be blindly predicted as one of the in-distribution classes $\mathcal{Y}_{\text{in}}$ ={headphone, cat, flamingo, butterfly}. This motivates the importance of OOD detection for vision-language models.

OOD detection for vision-language models. In the open-world setting, the goal of OOD detection is to detect samples that do not belong to ID classes $\mathcal{Y}_{\text{in}}$. Here ID classes are defined w.r.t. the classification task of interest, instead of the classes used in pre-training. Accordingly, OOD is defined w.r.t. the ID classes, not the data distribution during pre-training. Ming et al. (2022a) explore the zero-shot OOD detection for the pre-trained

CLIP model, without adapting to the ID dataset. Instead, we focus on the setting where CLIP models are fine-tuned on a few-shot dataset $\mathcal{D}_{\text{in}}$, and hence are better adapted to the downstream ID task. We evaluate the fine-tuned CLIP model on a combination of ID and OOD datasets $\mathcal{D}_{\text{in}} \cup \mathcal{D}_{\text{out}}$, where $\mathcal{D}_{\text{out}} = \{\mathbf{x}_i, y_i^{\text{out}}\}_{i=1}^{m}$ contains inputs with semantically different categories $y^{\text{out}} \notin \mathcal{Y}_{\text{in}}$. Formally, given an input $\mathbf{x}$, OOD detection can be formulated as:

$$G(\mathbf{x}; f, g) = \begin{cases} 1 & S(\mathbf{x}; f, g) \geq \lambda \\ -1 & S(\mathbf{x}; f, g) < \lambda \end{cases},$$

where $S(\cdot)$ is a scoring function that measures OOD uncertainty. In practice, λ is chosen so that a high fraction of ID data (*e.g.*, 95%) is above the threshold.

Parameter-efficient fine-tuning. To improve the performance on downstream tasks, parameter-efficient approaches are proposed to fine-tune CLIP on datasets of interest. Prompt learning and adaptor tuning have recently gained popularity and demonstrated improved results over zero-shot settings. In particular, prompt learning optimizes the word embeddings of the prompts, while adaptors directly optimize the internal feature representations. Both methods are parameter-efficient as image and text encoders are frozen during fine-tuning. In what follows, we introduce prompt-based and adaptor-based methods respectively.

For a downstream dataset with K in-distribution classes $\mathcal{Y}_{\text{in}} = \{y_1, y_2, ..., y_K\}$, prompt learning method such as `CoOp` (Zhou et al., 2022c) introduces M learnable context vectors $v_i \in \mathbb{R}^e$ to replace hand-engineered text prompts such as "`this is a photo of`", where e is the dimension of word embeddings. For each class y_k, we obtain its contextualized representation

$$t_k = [v_1, v_2, \cdots, v_M, w_k]$$

by concatenating the context vectors and the word embedding $w_k \in \mathbb{R}^e$ of the label (upper left, Figure 6.1). To avoid overfitting and improve generalization performance, `CoCoOp` (Zhou et al., 2022b) further introduces instance-conditional prompts via a meta-network which produces a meta token $m(\mathbf{x})$ given the visual feature of the input $\mathbf{x}$. The meta token is added to each context token $v_i(\mathbf{x}) = v_i + m(\mathbf{x})$ for $i \in \{1, 2, \cdots, M\}$. Therefore, the prompt for class k is conditioned on each input:

$$t_k(\mathbf{x}) = [v_1(\mathbf{x}), v_2(\mathbf{x}), \cdots, v_M(\mathbf{x}), w_k]$$

. To learn the context vectors, the cross-entropy loss is used in fine-tuning:

$$p(y_k \mid \boldsymbol{x}) = \frac{\exp\left(s_k(\mathbf{x})/\tau\right)}{\sum_{i=1}^{K} \exp\left(s_i(\mathbf{x})/\tau\right)}, \tag{6.1}$$

where $s_k(\mathbf{x}) = \frac{g(\mathbf{x}) \cdot f(t_k)}{\|g(\mathbf{x})\| \cdot \|f(t_k)\|}$ is the cosine similarity of input $\mathbf{x}$ with the k-th label, and τ is the temperature.

Alternatively, adaptor-based methods directly optimize the feature representations $g(\mathbf{x})$ instead of learning context vectors. Specifically, given a K-way-D-shot ID training set (consisting of K classes with D examples per class), Zhang et al. (2022b) propose a training-free adaptation method `TipAdaptor` which extracts all the visual features

$$W_g = [g(\mathbf{x}_{1,1}), g(\mathbf{x}_{1,2}), \cdots, g(\mathbf{x}_{K,D})] \in \mathbb{R}^{KD \times d}$$

from the few-shot training dataset. For each input $\mathbf{x}$, we can obtain $K \times D$ cosine similarities $s_{k,d}(\mathbf{x}) = \frac{g(\mathbf{x}) \cdot g(\mathbf{x}_{k,d})}{\|g(\mathbf{x})\| \cdot \|g(\mathbf{x}_{k,d})\|}$. The cosine similarities are scaled by an exponential function $\tilde{s} : s \mapsto \exp(-\beta + \beta s)$ with a hyperparameter β that modulates the sharpness. Therefore, we can obtain an average similarity vector for each class based on visual features, $\tilde{s}_k(\mathbf{x}) = \frac{1}{D} \sum_{d=1}^{D} \tilde{s}_{k,d}(\mathbf{x})$. The final similarity for class k is a weighted sum of similarities from the two modalities $\alpha \tilde{s}_k(\mathbf{x}) + s_k(\mathbf{x})$. To achieve bet-

ter few-shot ID performance, Zhang et al. (2022b) set visual features W_g as learnable parameters and denote the method as `TipAdaptorF`, where F stands for fine-tuning. Despite the stronger downstream classification performance, it remains unknown if fine-tuning leads to more reliable OOD detection at test time. We aim to provide a comprehensive understanding in this chapter.

6.4 Method

6.4.1 OOD detection with fine-tuning

We investigate OOD detection with parameter-efficient fine-tuning on downstream tasks. We present a unified framework in Figure 6.1, where the learnable part of the CLIP model is marked with an "unlock" icon while the frozen part is marked with a "lock" icon. For prompt learning methods such as `CoOp` and `CoCoOp`, the cosine similarity of the input feature with the k-th class $s_k(\mathbf{x}) = \frac{g(\mathbf{x})\cdot f(t_k)}{\|g(\mathbf{x})\|\cdot\|f(t_k)\|}$ is derived based on the adapted textual feature vector t_k. Alternatively, adaptor-based methods such as `TipAdaptor` and `TipAdaptorF` first scale the cosine similarities of visual prototypes and perform a weighted sum with the similarities of textual prototypes. Therefore, we can view `TipAdaptor` as an ensemble method that utilizes multi-modal prototypes.

To summarize, for each adaptation algorithm $\mathcal{A}$, OOD detection can be performed by:

$$G_{\mathcal{A}}(\mathbf{x}; f, g) = \begin{cases} \text{ID} & S(\mathbf{x}; f, g) \geq \lambda \\ \text{OOD} & S(\mathbf{x}; f, g) < \lambda \end{cases},$$

where $\mathcal{A}$ can be instantiated by an adaptation method such as `CoOp`, `CoCoOp`, `TipAdaptor`, or `TipAdaptorF`. Therefore, the OOD detector $G_{\mathcal{A}}(\cdot)$ can be viewed as a "safeguard" for the classification model. Next, we introduce

various OOD score functions $S(\mathbf{x}; f, g)$ assuming $G_{\mathcal{A}}(\mathbf{x}; f, g)$ is defined implicitly as each score function corresponds to an OOD detector G.

6.4.2 OOD score for vision-language models

Recently, Ming et al. (2022a) propose a conceptual framework of CLIP-based OOD detection via concept matching, where the textual feature $f(t_k)$ is viewed as the concept prototype for ID class $k \in \{1, 2, ..., K\}$. OOD uncertainty is then characterized by the distance from the visual feature of the input to the closest ID textual prototype. That is, images closer to one of the ID prototypes are more likely to be ID and vice versa. Ming et al. (2022a) suggest that softmax scaling with a proper temperature τ provably leads to state-of-the-art performance under the zero-shot (training-free) setting. Specifically, the maximum concept matching (MCM) score is defined as:

$$S_{\text{MCM}}(\mathbf{x}) = \max_{k \in [K]} \frac{e^{s_k(\mathbf{x})/\tau}}{\sum_{j=1}^{K} e^{s_j(\mathbf{x})/\tau}}, \tag{6.2}$$

where the temperature τ needs to be tuned on the downstream dataset. As a special case of MCM, we use MSP to denote the MCM score when the temperature τ_d is set as default for CLIP models at inference time (*e.g.*, 100 for CLIP-B/16).

Additionally, we consider a simpler scoring function based on the maximum similarity (MS) among ID prototypes before applying softmax scaling:

$$S_{\text{MS}}(\mathbf{x}) = \max_{k \in [K]} s_k(\mathbf{x}), \tag{6.3}$$

which does not require any hyperparameter tuning. We show in Section 6.5 that the MS score demonstrates strong OOD detection performance with fine-tuning, especially for fine-grained ID datasets. We now proceed to experiments where we investigate the impact of fine-tuning on

real-world tasks.

6.5 Experiments

6.5.1 Setup

Datasets. Following Ming et al. (2022a), we consider a wide range of real-world ID datasets with various semantics and number of classes: Caltech-101, Stanford-Cars (Krause et al., 2013), Food-101 (Bossard et al., 2014), Oxford-Pets (Parkhi et al., 2012) and ImageNet-1k (Deng et al., 2009). For each ID dataset, we follow Zhou et al. (2022b) and construct the training set with D random samples per class, while the original test set is used for testing. We use $D = 16$ by default and study the impact of shots as ablations in Section 6.5.3. For OOD test datasets, we use the same ones in Huang and Li (2021), including subsets of iNaturalist (Van Horn et al., 2018), SUN (Xiao et al., 2010), PLACES (Zhou et al., 2017), and TEXTURE (Cimpoi et al., 2014). For each OOD dataset, the categories do not overlap with the ID dataset. For ImageNet-1k as ID, we also consider two additional OOD datasets ImageNet-O (Hendrycks et al., 2021) and OpenImage-O (Wang et al., 2022b).

Models and training details. For pre-trained models, we use CLIP-B/16 as the default backbone for main experiments, which uses ViT-B/16 (Dosovitskiy et al., 2021) as the image encoder. The impact of backbones is included in the ablation studies. We use `ZOCLIP` to denote pre-trained CLIP without fine-tuning. For each method, we closely follow the original implementations. Specifically, for `CoOp` and `CoCoOp`, the context length is set to 4, and the context vectors are initialized using the pre-trained word embeddings of "a photo of a". `CoCoOp` is trained with a batch size of 1 for 10 epochs using SGD, while `CoOp` is trained for 100 epochs with a batch size of 32. `TipAdapterF` is trained with a batch size 256 using

AdamW (Loshchilov and Hutter, 2019) for 20 epochs. Cosine scheduling is used for all methods and the data preprocessing protocol consists of random re-sizing, cropping, and random horizontal flip.

Evaluation metrics. We consider the following evaluation metrics: (1) the false positive rate (FPR95) of OOD samples when the true positive rate of in-distribution samples is at 95%, (2) the area under the receiver operating characteristic curve (AUROC), and (3) ID classification accuracy (ID ACC).

6.5.2 Main results and discussions

In this section, we first present novel evidence that parameter-efficient fine-tuning generally improves OOD performance over the zero-shot counterpart with a simple OOD scoring function. Next, we investigate the effects of various OOD scoring functions in the parameter-efficient fine-tuning setting. In particular, we will show that the MCM score consistently demonstrates the most promising performance compared to alternative OOD scores when coupled with prompt learning.

How does parameter-efficient fine-tuning impact OOD detection? We evaluate the OOD detection performance on various ID datasets. The results are summarized in Table 6.1. We show that adapted CLIP models demonstrate nearly perfect OOD detection performance for ID datasets with fine-grained categories such as Stanford-Cars and Oxford-Pets. Moreover, when the ID dataset contains a diverse collection of categories such as Caltech-101[1], parameter-efficient fine-tuning still significantly improves the OOD detection performance on average compared to `ZOCLIP`. In particular, `CoCoOp` yields the best performance among other adaptation methods on Caltech-101 (ID). It achieves an average FPR95 of 5.94% using

[1]Similar trends also hold for ImageNet-1k as ID.

Table 6.1: OOD detection performance based on S_{MS} score (w.o. softmax scaling). When ID datasets contain finer-grained categories semantically different from OOD categories, the pre-trained CLIP model demonstrates nearly perfect OOD detection performance. More encouragingly, after adapting the model to downstream datasets, OOD detection performance remains competitive.

ID Dataset	Method	SUN		Places		Textures		iNaturalist		Average	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
						Training not required					
	ZOCLIP	0.04	99.92	0.12	99.93	4.63	98.29	0.15	99.87	1.24	99.50
	TipAdaptor	0.00	99.94	0.04	99.95	2.87	98.85	0.06	99.90	0.74	99.66
						Requires training					
Food-101	TipAdaptorF	0.00	99.94	0.03	99.95	3.16	98.77	0.05	99.91	0.81	99.64
	CoOp	0.01	99.97	0.00	99.98	1.45	99.68	0.00	99.97	0.36	99.90
	CoCoOp	0.00	99.98	0.00	99.98	1.97	99.51	0.01	99.97	0.49	99.86
						Training not required					
	ZOCLIP	0.03	99.99	0.14	99.96	0.12	99.95	0.00	100.00	0.07	99.97
	TipAdaptor	0.01	100.00	0.07	99.98	0.07	99.99	0.00	100.00	0.04	99.99
						Requires training					
Oxford-Pets	TipAdaptorF	0.02	100.00	0.07	99.98	0.09	99.98	0.00	100.00	0.04	99.99
	CoOp	0.02	100.00	0.18	99.97	0.25	99.92	0.00	100.00	0.11	99.97
	CoCoOp	0.03	99.99	0.19	99.96	0.11	99.96	0.00	100.00	0.08	99.98
						Training not required					
	ZOCLIP	0.02	99.99	0.24	99.94	0.00	100.00	0.00	100.00	0.07	99.98
	TipAdaptor	0.01	100.00	0.08	99.98	0.00	100.00	0.00	100.00	0.02	100.00
						Requires training					
Stanford-Cars	TipAdaptorF	0.01	100.00	0.06	99.98	0.00	100.00	0.00	100.00	0.02	100.00
	CoOp	0.01	100.00	0.07	99.97	0.00	100.00	0.00	100.00	0.02	99.99
	CoCoOp	0.01	100.00	0.07	99.97	0.00	100.00	0.00	100.00	0.02	99.99
						Training not required					
	ZOCLIP	32.03	94.06	33.01	93.39	54.66	89.29	32.14	94.30	37.96	92.76
	TipAdaptor	9.69	98.07	11.25	97.84	20.90	96.68	13.62	97.72	13.86	97.58
						Requires training					
Caltech-101	TipAdaptorF	10.20	97.76	11.60	97.42	23.32	95.54	14.01	97.36	14.78	97.02
	CoOp	5.53	98.56	9.88	97.50	13.10	97.10	4.89	98.76	8.35	97.98
	CoCoOp	2.86	99.19	6.42	98.37	8.81	98.09	5.68	98.68	**5.94**	**98.58**

S_{MS}, improving by 32.02% over `ZOCLIP`. While prior works suggest that parameter-efficient fine-tuning methods improve ID accuracy on few-shot datasets, our results complement their findings and show that fine-tuning also improves the OOD detection performance with proper OOD scoring functions.

Effects of OOD scoring functions. We investigate the effect of OOD scoring functions under fine-tuned vision-language models. In Table 6.2, we contrast the OOD detection performance using MCM (Ming et al., 2022a) vs. MS on Caltech-101 (ID). Our findings suggest that: (1) S_{MCM} performs on par with S_{MS} for fine-grained ID tasks across a wide range of

Table 6.2: OOD detection performance with S_{MS} and S_{MCM} score when the ID dataset contains diverse categories. Prompt learning methods display clear advantages over zero-shot models. The results are based on Caltech-101 (ID).

OOD Score	Method	SUN		Places		Textures		iNaturalist		Average	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
S_{MS}	ZOCLIP	32.03	94.06	33.01	93.39	54.66	89.29	32.14	94.30	37.96	92.76
	TipAdaptor	9.69	98.07	11.25	97.84	20.90	96.68	13.62	97.72	13.86	97.58
	TipAdaptorF	10.20	97.76	11.60	97.42	23.32	95.54	14.01	97.36	14.78	97.02
	CoOp	5.53	98.56	9.88	97.50	13.10	97.10	4.89	98.76	8.35	97.98
	CoCoOp	2.86	99.19	6.42	98.37	8.81	98.09	5.68	98.68	**5.94**	**98.58**
S_{MCM}	ZOCLIP	14.83	97.20	20.45	96.00	14.98	97.35	10.84	97.76	15.28	97.08
	TipAdaptor	5.12	98.83	8.05	98.34	4.65	99.05	6.94	98.77	6.19	**98.75**
	TipAdaptorF	4.83	98.79	8.09	98.07	6.41	98.11	4.94	98.98	6.07	98.49
	CoOp	3.62	99.01	8.15	97.89	6.29	98.62	7.57	98.35	6.41	98.47
	CoCoOp	4.26	98.94	6.76	98.00	4.33	98.88	4.71	98.68	**5.02**	98.62

Table 6.3: OOD detection performance based on S_{MCM} score.

ID Dataset	Method	SUN		Places		Textures		iNaturalist		Average	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
Food-101	ZOCLIP	1.75	99.46	2.04	99.35	5.54	98.05	2.80	99.17	3.03	99.01
	TipAdaptor	0.63	99.75	0.64	99.71	3.76	98.59	1.32	99.55	**1.59**	**99.40**
	TipAdaptorF	1.77	99.57	1.57	99.53	4.43	98.34	1.85	99.40	2.40	99.21
	CoOp	2.00	99.46	1.60	99.47	5.85	98.39	1.37	99.54	2.71	99.22
	CoCoOp	1.06	99.69	1.01	99.63	4.17	98.42	1.40	99.53	1.91	99.32
Oxford-Pets	ZOCLIP	1.18	99.73	3.37	99.28	1.37	99.73	6.17	98.84	3.02	99.40
	TipAdaptor	0.05	99.97	0.62	99.87	0.17	99.96	0.11	99.87	**0.24**	**99.92**
	TipAdaptorF	0.48	99.89	1.74	99.66	0.43	99.88	0.93	99.53	0.90	99.74
	CoOp	0.06	99.96	0.55	99.85	0.39	99.90	2.07	99.37	0.77	99.77
	CoCoOp	0.08	99.95	0.53	99.85	0.25	99.91	1.12	99.55	0.49	99.82
Stanford-Cars	ZOCLIP	0.02	99.96	0.31	99.89	0.02	99.96	0.10	99.74	0.11	99.89
	TipAdaptor	0.01	99.98	0.11	99.94	0.00	99.97	0.00	99.84	**0.03**	99.93
	TipAdaptorF	0.03	99.98	0.19	99.94	0.00	99.99	0.00	99.93	0.06	**99.96**
	CoOp	0.01	99.98	0.17	99.93	0.00	99.98	0.02	99.84	0.05	99.93
	CoCoOp	0.02	99.98	0.15	99.93	0.00	99.97	0.00	99.87	0.04	99.94
Caltech-101	ZOCLIP	14.83	97.20	20.45	96.00	14.98	97.35	10.84	97.76	15.28	97.08
	TipAdaptor	5.12	98.83	8.05	98.34	4.65	99.05	6.94	98.77	6.19	**98.75**
	TipAdaptorF	4.83	98.79	8.09	98.07	6.41	98.11	4.94	98.98	6.07	98.49
	CoOp	3.62	99.01	8.15	97.89	6.29	98.62	7.57	98.35	6.41	98.47
	CoCoOp	4.26	98.94	6.76	98.00	4.33	98.88	4.71	98.68	**5.02**	98.62

adaptation methods (Table 6.3). (2) However, when ID contains diverse categories, utilizing S_{MCM} generally leads to better performance compared to using S_{MS} for most adaptation methods (Table 6.2). (3) In particular, prompt learning methods such as `CoCoOp` demonstrate very competitive results with both OOD scores (an average FPR95 of 5.02% with S_{MCM} and 5.94% with S_{MS} in Table 6.2).

Effects of softmax scaling. Previously, Ming et al. (2022a) observed that the commonly used maximum softmax score (S_{MSP}) is suboptimal for zero-shot OOD detection with vision-language models. We investigate whether MSP is competitive for OOD detection with fine-tuned models. To better illustrate the effects, we plot the score distributions for StanfordCars (ID) vs. SUN (OOD) in Figure 6.2 when the model is fine-tuned with `CoOp`, `CoCoOp`, and `TipAdaptorF` respectively. For each fine-tuning method, we can clearly see that the S_{MS} leads to superior ID-OOD separability, while S_{MSP} displays significant overlapping. Quantitatively, compared to S_{MSP}, the average FPR95 is significantly decreased with S_{MS} (Table E.4). Our findings highlight that directly applying MSP is not competitive for fine-tuned vision-language models.

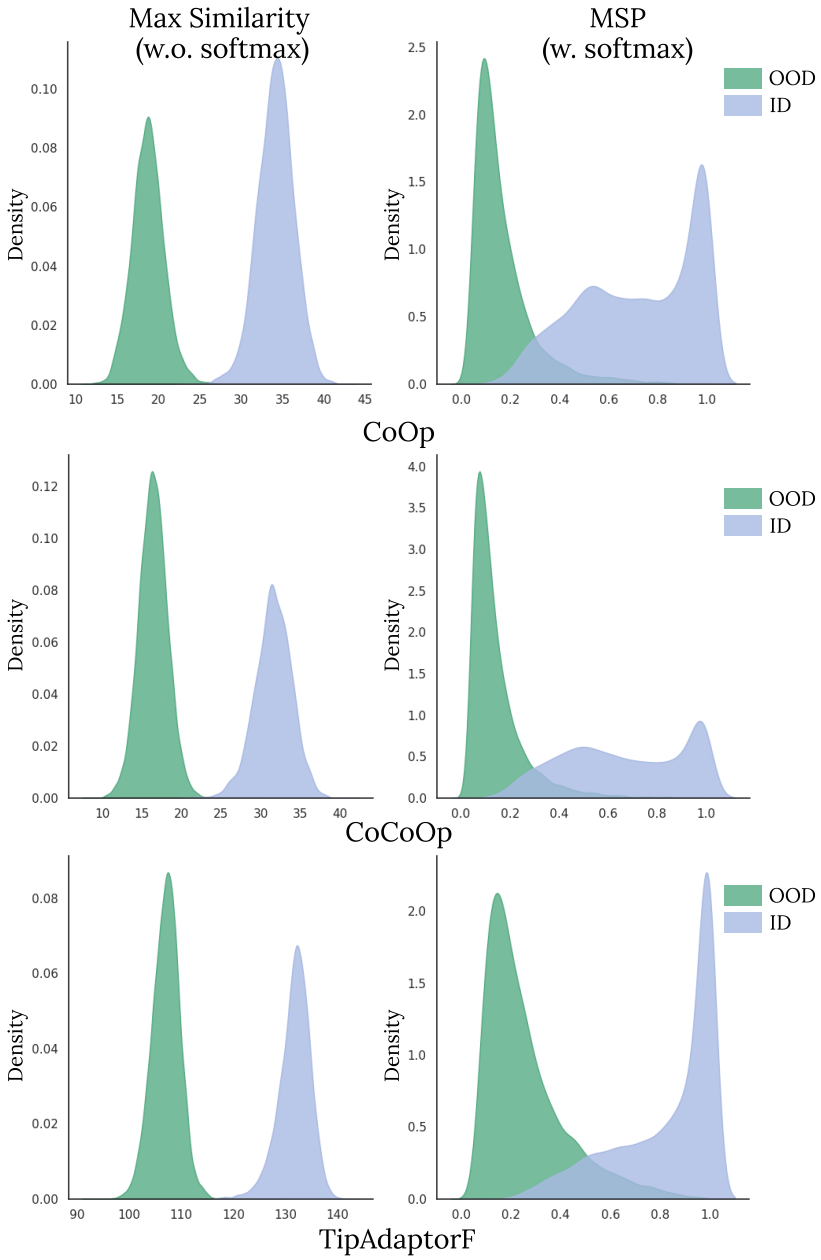


Figure 6.2: The impact of softmax scaling. We use Stanford-Cars (ID) vs. SUN (OOD) for illustration. Applying softmax scaling significantly decreases ID-OOD separability for `CoOp` (top row), `CoCoOp` (second row), and `TipAdaptorF` (last row), resulting in worse OOD detection performance.

6.5.3 Delving into parameter-efficient fine-tuning for OOD detection

The impact of fine-tuning on feature geometry. To better understand how fine-tuning leads to improved OOD detection performance, we examine the geometry of the feature representations. For illustration, we use the simple S_{MS} score as it provides an intuitive geometric interpretation. For each test input, S_{MS} captures the angular distance between its visual features and the closest ID prototype. Figure 6.3 shows S_{MS} for ID and each OOD test dataset, where radians are converted to degrees for better readability. Intuitively, one desires to learn compact ID clusters such that ID inputs are closer to the nearest ID prototypes than OOD inputs. We illustrate the effects of prompt learning in Figure 6.4. Compared to zero-shot CLIP, `CoOp` and `CoCoOp` decrease the angular distance for ID inputs to the nearest concept prototype while simultaneously increasing the angular distance for OOD inputs. In particular, `CoCoOp` decreases the angular distance for ID inputs more significantly, resulting in better ID-OOD separability. Although prompt learning methods introduce perturbations to the feature space, the overall effect is modest, with only a slight deviation of a few degrees from the pre-trained model[2]. Nonetheless, these perturbations play a crucial role in enhancing both ID classification and OOD detection performance.

Exploring prompt learning for OOD detection on challenging large-scale benchmarks In previous sections, we show that prompt learning with both S_{MS} and S_{MCM} scores display competitive performance. Next, we consider a more challenging large-scale benchmark ImageNet-1k (ID). The results in FPR95 and AUROC are shown in Figure 6.5 and Figure 6.6. While S_{MS} outperforms S_{MSP} score, we can clearly see that S_{MCM} is particularly advantageous compared to the simpler S_{MS} baseline. In partic-

[2]Similar observations can also be verified for adaptor-based methods.

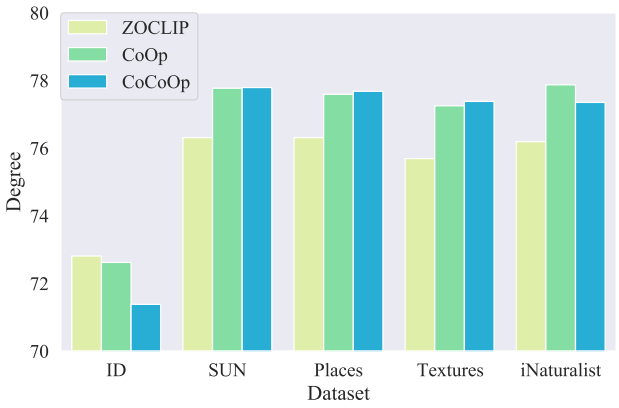


Figure 6.3: Average S_{MS} for ID (Caltech-101) and OOD test sets. Prompt learning methods decrease the angular distance for ID inputs while increasing the angular distance for OOD inputs to the nearest concept prototype, leading to better ID-OOD separability (Figure 6.4).

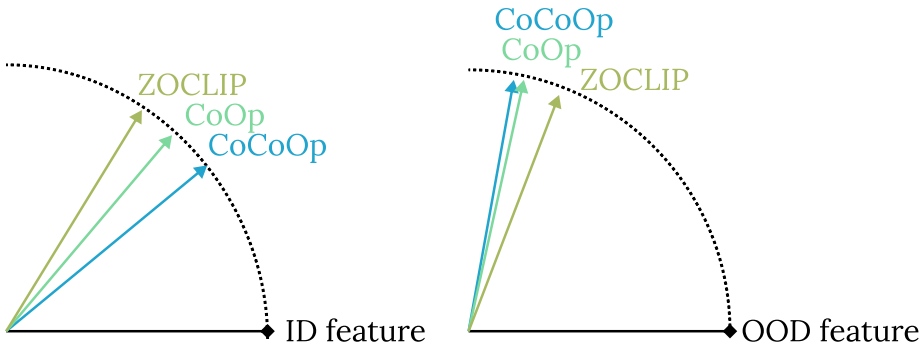


Figure 6.4: Illustration of how prompt learning methods impact the hyperspherical features. Left: feature of an ID sample and its nearest ID prototype; Right: feature of an OOD sample and its nearest ID prototype.

ular, S_{MCM} outperforms S_{MS} by 7.44% in FPR95 averaged across the four OOD test sets. Moreover, `CoOp` with S_{MCM} achieves an average FPR95 of 37.74% on the benchmark, surpassing the zero-shot performance of the large backbone CLIP-L/14 model which has an FPR95 of 38.17% (Ming et al., 2022a). These results further demonstrate the effectiveness of S_{MCM} in CLIP-based prompt learning for challenging scenarios.

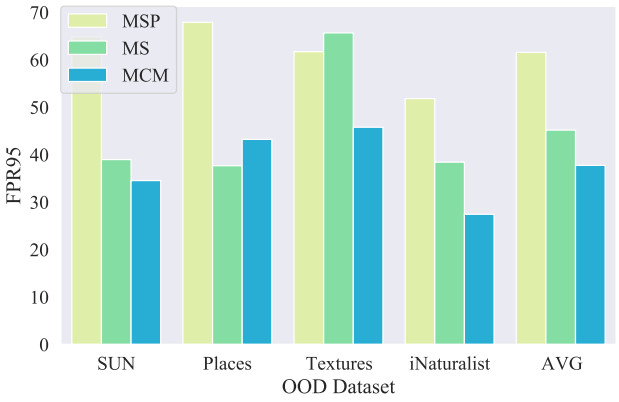


Figure 6.5: OOD detection performance (FPR95) on ImageNet-1k (ID). Using S_{MCM} score leads to significant improvement over S_{MSP}.

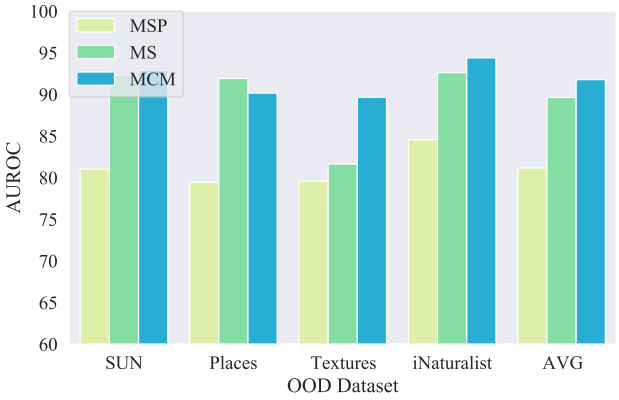


Figure 6.6: OOD detection performance (AUROC) on ImageNet-1k (ID). The trend is consistent with Fig 6.5.

The impact of shots. We investigate the impact of shots for CoOp and CoCoOp with various OOD detection scores. The results are shown in Figure 6.7 and Figure 6.8, where each point represents the average FPR95 over the four OOD test sets. We highlight two key findings. First, the OOD detection performance with both S_{MS} and S_{MCM} score improves as the number of shots increases. This trend is consistent with the ID classification accuracy reported in Zhou et al. (2022c), suggesting that using a suitable OOD uncertainty score can enhance the representation quality as more data is incorporated during prompt learning. Second, the performance of S_{MCM} is promising even with a low number of shots, demonstrating its effectiveness in resource-constrained settings.

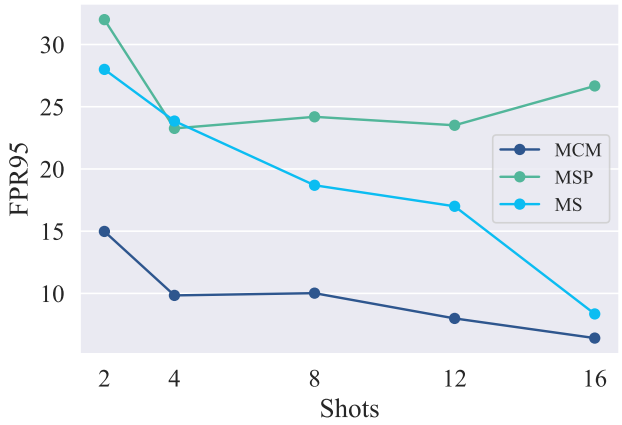


Figure 6.7: The effects of shots for CoOp with various OOD detection scores on Caltech-101 (ID). The performance is averaged over the four OOD test sets.

The impact of backbone architecture. We conduct another ablation study on the impact of model architectures. We consider CLIP with ResNet backbones (N50, RN101) and ViT backbones (CLIP-B/32, CLIP-L/14), where the vision encoder is based on ViT-B/32 and ViT-L/14, respectively. We train with CoOp with hyperparameters following the original implementation for each architecture (Zhou et al., 2022c). We evaluate the

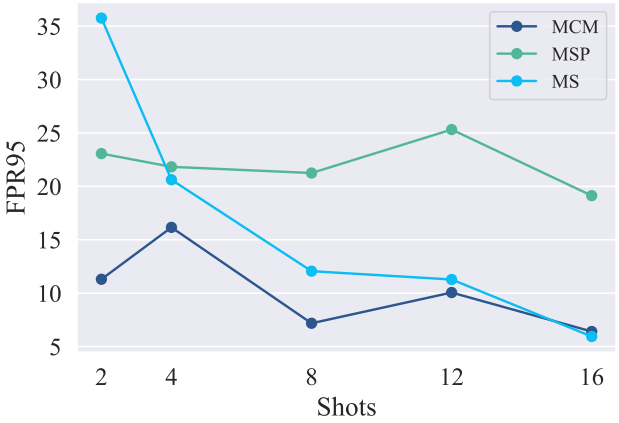


Figure 6.8: The effects of shots for `CoCoOp` with various OOD detection scores on Caltech-101 (ID). The performance is averaged over the four OOD test sets.

models using S_{MSP}, S_{MS}, and S_{MCM} score and summarize the results in Table 6.4 and Table 6.5. Interestingly, compared to S_{MSP}, S_{MS} brings more significant improvements under ViT backbones than ResNet backbones. In contrast, S_{MCM} score consistently demonstrates competitive performance for all the architectures considered. For instance, with CLIP-B/32, S_{MCM} achieves an average FPR95 of 6.17%, a 20.23% improvement over the S_{MSP} baseline. We observe similar improvements for RN101 (18.57%) and RN50 (22%). Moreover, larger backbones lead to superior performance when fixing the OOD detection score as MCM. For example, with CLIP-L/14, the average FPR95 is improved by 11.17% compared to RN50 and 2.67% compared to CLIP-B/32. A similar trend has been shown for ID classification (Radford et al., 2021), where larger models yield better feature representation.

Table 6.4: The impact of model architecture on ResNet backbones with CoOp on Caltech-101 (ID).

Arch	Score	OOD Dataset	FPR95↓	AUROC↑
RN50	S_{MSP}	SUN	29.93	93.95
		Places	37.64	91.96
		Textures	35.69	93.58
		iNaturalist	43.42	91.27
		AVG	36.67	92.69
	S_{MS}	SUN	6.02	98.45
		Places	9.02	97.79
		Textures	23.17	95.25
		iNaturalist	12.39	97.37
		AVG	12.65	97.22
	S_{MCM}	SUN	8.56	98.03
		Places	17.02	95.88
		Textures	12.09	97.56
		iNaturalist	21.00	95.93
		AVG	14.67	96.85
RN101	S_{MSP}	SUN	23.60	95.20
		Places	29.37	93.94
		Textures	21.29	96.24
		iNaturalist	34.18	94.05
		AVG	27.11	94.86
	S_{MS}	SUN	19.08	96.56
		Places	20.79	96.25
		Textures	36.97	94.39
		iNaturalist	30.89	95.41
		AVG	26.93	95.65
	S_{MCM}	SUN	6.19	98.42
		Places	11.57	97.16
		Textures	5.83	98.49
		iNaturalist	10.56	97.69
		AVG	8.54	97.94

6.6 Conclusion

In this chapter, we provide a timely study on the impact of parameter-efficient fine-tuning methods for OOD detection with large vision-language models. We focus on the few-shot setting without access to OOD labels, which has been largely unexplored in the literature. We show that parameter-efficient fine-tuning methods can improve both ID and OOD performance when coupled with a proper OOD score, with prompt learning-based methods showing the strongest performance under the MCM score. We analyze the feature space and provide insights into the effectiveness

Table 6.5: The impact of model architecture on ViT backbones with CoOp on Caltech-101 (ID).

Arch	Score	OOD Dataset	FPR95↓	AUROC↑
CLIP-B/32	S_{MSP}	SUN	24.20	96.02
		Places	27.94	94.99
		Textures	24.54	96.09
		iNaturalist	28.90	95.37
		AVG	26.40	95.62
	S_{MS}	SUN	13.81	97.41
		Places	16.49	96.48
		Textures	25.23	95.24
		iNaturalist	13.00	97.60
		AVG	17.13	96.68
	S_{MCM}	SUN	4.06	98.92
		Places	7.31	98.01
		Textures	4.61	98.81
		iNaturalist	8.70	98.17
		AVG	6.17	98.48
CLIP-L/14	S_{MSP}	SUN	7.73	98.36
		Places	10.96	97.71
		Textures	19.18	96.60
		iNaturalist	11.33	97.71
		AVG	15.85	97.41
	S_{MS}	SUN	13.81	97.41
		Places	16.49	96.48
		Textures	25.23	95.24
		iNaturalist	13.00	97.60
		AVG	12.30	97.59
	S_{MCM}	SUN	2.15	99.33
		Places	5.60	98.30
		Textures	2.32	99.31
		iNaturalist	3.94	99.06
		AVG	3.50	99.00

of such methods through the lens of multi-modal concept matching. We hope our findings will inspire and motivate future research on designing reliable fine-tuning methods for large vision-language models.

Chapter 7

Understanding Retrieval-Augmented Task Adaptation for Vision-Language Models

Pre-trained contrastive vision-language models have demonstrated remarkable performance across a wide range of tasks. However, they often struggle on fine-trained datasets with categories not adequately represented during pre-training, which makes adaptation necessary. Recent works have shown promising results by utilizing samples from web-scale databases for retrieval-augmented adaptation, especially in low-data regimes. Despite the empirical success, understanding how retrieval impacts the adaptation of vision-language models remains an open research question. In this chapter, we adopt a reflective perspective by presenting a systematic study to understand the roles of key components in retrieval-augmented adaptation. We unveil new insights on uni-modal and cross-modal retrieval and highlight the critical role of logit ensemble for effec-

tive adaptation. We further present theoretical underpinnings that directly support our empirical observations.

7.1 Introduction

Contrastive vision-language pre-training has emerged as a fundamental cornerstone for a wide array of tasks in natural language processing and computer vision Radford et al. (2021); Jia et al. (2021); Yang et al. (2022); Li et al. (2022c); Mu et al. (2022); Yu et al. (2022); Sun et al. (2023); Xu et al. (2024). These models excel in capturing the intricate relationships present in both visual and textual data, enabling them to understand context, semantics, and associations holistically. It is now a common practice to employ aligned multi-modal features from web-scale pre-training. However, a challenge arises when these pre-trained models encounter real-world downstream datasets, particularly in low-data (few-shot) scenarios. Such datasets often encompass fine-grained categories that were not adequately represented during the initial pre-training phase, posing a notable hurdle for the models in adapting to these nuanced distinctions.

In the low-data regime, retrieval-augmented adaptation has demonstrated promise, where a wealth of external resources is readily available on the Internet and can be retrieved efficiently to enhance adaptation. Recent works Udandarao et al. (2023); Zhang et al. (2023) showcase encouraging results by leveraging large-scale text and image databases Schuhmann et al. (2022). Retrieval-augmented adaptation involves two main steps: first retrieving the most relevant data from an external source, and then adapting to downstream task based on the retrieved samples. While existing works have primarily focused on developing new adaptation algorithms or integrating different knowledge sources, ***there remains a notable gap in understanding how retrieval augmentation impacts adaptation for vision-language models***. Such an understanding is imperative

to guide the future development of effective algorithms.

In this work, we adopt a reflective perspective by presenting a systematic study to understand retrieval-augmented adaptation, and establishing new theoretical underpinnings. Our empirical analysis reveals key insights revolving around two aspects: **(1)** the impact of the retrieval method, and **(2)** how retrieved samples help adaptation. First, we show that image-to-image (I2I) retrieval consistently outperforms text-to-image (T2I) retrieval for a wide range of downstream tasks. Under the same retrieval budget, these two retrieval methods differ by the query samples used: I2I employs a few seed images from the target data distribution, whereas T2I employs the textual description of each class label. While both I2I and T2I retrieval introduce distributional shifts *w.r.t.* the target data, we show that I2I achieves strong performance that matches more closely with the oracle when we directly retrieve from the target distribution (*i.e.*, no distributional shifts). Secondly, we show that ensembling the zero-shot prediction together with I2I retrieved samples is the key to improved adaptation performance. For a given test sample, the ensembling is achieved by taking a weighted average between the logit from the retrieved feature cache and the logit of the zero-shot inference. We empirically find that without ensembling, the performance of retrieval-augmented adaptation significantly degrades. This new observation complements previous studies that often attribute the success of retrieval to the diversity and quality of samples.

Going beyond empirical analysis, we provide theoretical insights that directly support our empirical observations above. We formalize T2I and I2I retrieval by characterizing the multi-modal feature space with each retrieval scheme. Under realistic assumptions, we analyze how retrieval impacts the modality gap and the shift between the retrieved and target distributions. In particular, we prove that I2I retrieval is superior to T2I retrieval (**Theorem 7.1**) and that logit ensemble is critical for improv-

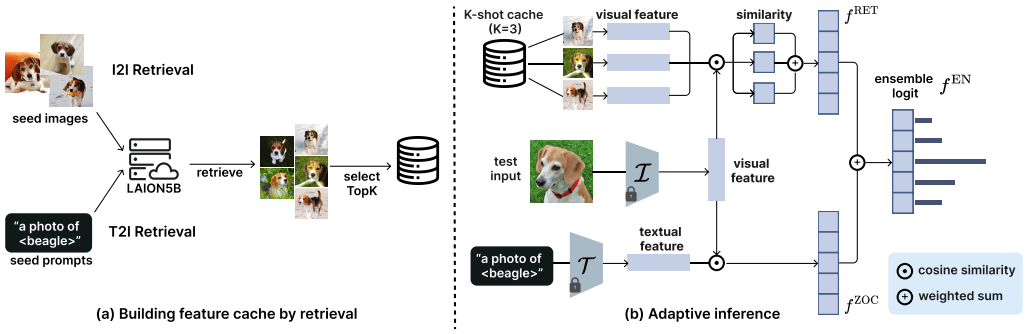


Figure 7.1: Illustration of the retrieval-augmented task adaptation framework for CLIP-like models. (a): Given a downstream target dataset, we first retrieve relevant samples from a web-scale database using seed prompts (T2I) or seed images (I2I). We can then build a K-shot cache by selecting the Top-K similar images per class based on CLIP embeddings. (b) At inference time, the final logit f^{EN} of a test input is an ensemble (weighted sum) of logits from the zero-shot model f^{ZOC} and the few-shot cache f^{RET}.

ing CLIP-based adaptation (**Theorem 7.2**) by better leveraging the knowledge encoded in different modalities. Our theoretical results shed light on the key factors in the design of effective retrieval-augmented adaptation algorithms for vision-language models.

Our main contributions are summarized as follows:

- We conduct a timely and systematic investigation into the retrieval-augmented adaptation of vision-language models, where we highlight key components such as the retrieval methods and logit ensemble.
- We provide a finer-grained empirical study with in-depth analysis. We unveil new insights on the critical role of uni-modal retrieval and logit ensemble for effective CLIP-based adaptation in low-data scenarios.
- We develop a novel theoretical framework for retrieval-augmented

adaptation and present theoretical results that directly support our empirical observations.

- We further provide a comprehensive ablation study and discuss alternative design choices such as the impact of model architectures, adaptation with a finetuned feature cache, and adaptation with data mixtures.

7.2 Retrieval-Augmented Task Adaptation

In this section, we first discuss the preliminaries of contrastive vision-language models as well as the external databases employed for retrieval (Section 7.2.1). Next, we illustrate the two main steps for retrieval-augmented task adaptation: building a feature cache by retrieving relevant samples from the external database (Section 7.2.2), and performing task adaptation based on retrieved samples (Section 7.2.3). An illustration of the pipeline is shown in Figure 7.1.

7.2.1 Preliminaries

Popular contrastive vision-language models such as CLIP (Radford et al., 2021) adopt a dual-stream architecture with one text encoder $\mathcal{T} : t \to \mathbb{R}^d$ and one image encoder $\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$. The model is pre-trained on a massive web-scale image-caption dataset with a multi-modal contrastive loss, which aligns features from different modalities. This alignment of multi-modal embeddings offers distinct advantages for contemporary large-scale multi-modal vector databases Schuhmann et al. (2022), enabling efficient retrieval based on semantic similarity.

Zero-shot inference. At inference time, given a test input $\mathbf{x}$, we can obtain the cosine similarity $f_c^{\text{ZOC}}(\mathbf{x}) = \text{sim}(\mathcal{I}(\mathbf{x}), \mathcal{T}(t_c))$ between the visual

embedding $\mathcal{I}(\mathbf{x})$ and contextualized representations $\mathcal{T}(t_c)$ for each label $c \in \{1, 2, ..., C\}$. Here the context t_c can be either a generic template such as "`a photo of <CLASS>`" or a textual description of the class. We denote the logit vector of the zero-shot model as $f^{\text{ZOC}}(\mathbf{x}) \in \mathbb{R}^C$, which consists of C cosine similarities. The class prediction can be made based on the maximum cosine similarity among C classes.

External web-scale knowledge base. Pre-trained CLIP models often struggle for downstream datasets with finer-grained categories, which are not well represented in the pre-training dataset. To adapt CLIP models to finer-grained datasets in a low-data scheme, recent works Liu et al. (2023) demonstrate promising performance by utilizing external resources such as LAION (Schuhmann et al., 2022), a web-scale knowledge base which consists of billions of image-text pairs $\mathcal{S}_L = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$ covering a diverse range of concepts in the real world. Given a fixed budget, we can efficiently build a few-shot cache by retrieving relevant samples from the knowledge base with approximate KNN search Johnson et al. (2019). We provide details as follows.

7.2.2 Building Feature Cache by Retrieval

Given a downstream dataset with C classes: $\mathcal{Y} = \{1, 2, ..., C\}$ and a budget size of KC, we can retrieve K samples per class to build a cache of size KC. For vision-language models, the retrieval methods be categorized as uni-modal and cross-modal retrieval, formalized as follows:

Uni-modal retrieval. We mainly consider image-to-image (I2I) retrieval due to its popularity. For I2I retrieval, we assume access to a small set of query images from the downstream dataset. The query set $\mathcal{Q}_I = \bigcup_{c=1}^C \mathcal{Q}_I^c$, where $\mathcal{Q}_I^c = \{\mathbf{x}_{c,1}, \mathbf{x}_{c,2}, \ldots, \mathbf{x}_{c,n_c}\}$ contains n_c seed images for each class

$c \in \mathcal{Y}$. We then retrieve top-K similar images from $\mathcal{S}_L$ per class:

$$\mathcal{R}^{\text{I2I}}(c) = \text{top}_K \left\{ \mathbf{x} \in \mathcal{S}_L : \text{sim}(\mathcal{I}(\mathbf{x}), \mathcal{I}(\mathbf{x}_{c,i})), \mathbf{x}_{c,i} \in \mathcal{Q}_I \right\},$$

where $\text{sim}(\mathcal{I}(\mathbf{x}), \mathcal{I}(\mathbf{x}_{c,i}))$ is the cosine similarity between the image embedding of $\mathbf{x}$ from retrieval database and the query image $\mathbf{x}_{c,i}$, and top_K denotes the operation of selecting the top-K items. We can build a K-shot cache for I2I retrieval by taking the union of these sets across all classes:

$$\mathcal{S}_R^{\text{I2I}} = \bigcup_{c \in \mathcal{C}} \left\{ (\mathbf{x}, t) \in \mathcal{S}_L : \mathbf{x} \in \mathcal{R}^{\text{I2I}}(c) \right\}.$$

Cross-modal retrieval. We mainly consider text-to-image (T2I) retrieval. We assume access to class names in the target dataset, also known as "name-only transfer" (Udandarao et al., 2023). The query set $\mathcal{Q}_T = \{t_c\}_{c=1}^{C}$, where t_c is a generic textual description of class c. The retrieved K samples for class c is:

$$\mathcal{R}^{\text{T2I}}(c) = \text{top}_K \left\{ \mathbf{x} \in \mathcal{S}_L : \text{sim}(\mathcal{I}(\mathbf{x}), \mathcal{T}(t_c)), t_c \in \mathcal{Q}_T \right\},$$

where $\text{sim}(\mathcal{I}(\mathbf{x}), \mathcal{T}(t_c))$ is the cosine similarity between the image embedding of $\mathbf{x}$ and the text embedding for class c. The K-shot cache for T2I retrieval is denoted as:

$$\mathcal{S}_R^{\text{T2I}} = \bigcup_{c \in \mathcal{C}} \left\{ (\mathbf{x}, t) \in \mathcal{S}_L : \mathbf{x} \in \mathcal{R}^{\text{T2I}}(c) \right\}.$$

7.2.3 Task Adaptation with Retrieved Samples

Given a K-shot cache ($\mathcal{S}_R^{\text{I2I}}$ or $\mathcal{S}_R^{\text{T2I}}$) and pre-trained CLIP image and text encoders $\mathcal{I}$ and $\mathcal{T}$, we can perform adaptation *w.r.t.* a fine-grained target dataset. To better understand the effects of retrieved samples, we consider zero-shot adaptation in Section 7.3, where the cache only consists of

retrieved samples. We discuss few-shot adaptation in Section 7.5, where the cache contains a mixture of samples in the target training set and retrieved samples.

Retrieval-based adaptation. A variety of cache-based adaptation methods have been recently proposed Zhang et al. (2022b, 2023); Udandarao et al. (2023). At the core, these methods typically obtain a logit ensemble for each test input based on two sources: (1) a logit from the zero-shot CLIP model, and (2) a logit from the cache. Without loss of generality, we consider a representative adaptation framework TipAdaptor (Zhang et al., 2022b). Specifically, given the cache of size CK (consisting of C classes with K retrieved samples per class), we denote the collection of the visual features as $\mathbf{K} = [\mathbf{k}_{1,1}, \mathbf{k}_{1,2}, \cdots, \mathbf{k}_{C,K}] \in \mathbb{R}^{d \times CK}$ where $\mathbf{k}_{c,i} = \mathcal{I}(\mathbf{x}_{c,i})$. For each test input $\mathbf{x}$, we can obtain CK cosine similarities $s_{c,i}(\mathbf{x}) = \text{sim}(\mathcal{I}(\mathbf{x}), \mathbf{k}_{c,i})$. The cosine similarities are then scaled by an exponential function $\tilde{s} : s \mapsto \exp(-\omega + \omega s)$ with a hyperparameter ω that modulates the sharpness. Accordingly, we can obtain an average similarity vector for each class based on visual features, $f_c^{\text{RET}}(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} \tilde{s}_{c,i}(\mathbf{x})$. The final logit of the test sample is an ensemble of logits from the feature cache and zero-shot CLIP prediction:

$$f^{\text{EN}}(\mathbf{x}) = \alpha f^{\text{ZOC}}(\mathbf{x}) + \gamma f^{\text{RET}}(\mathbf{x}),$$

where α, γ weigh the relative importance between two logits. Such a logit ensemble scheme has also been commonly adopted in recent works Zhang et al. (2023). For completeness, we also discuss learning-based adaptation by setting visual features in $\mathbf{K}$ as learnable parameters. We denote the method as `Ensemble(F)`, where F stands for fine-tuning.

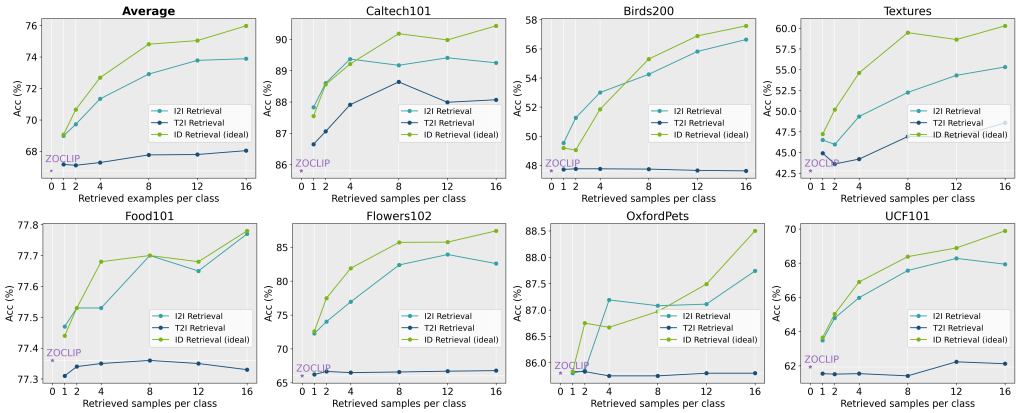


Figure 7.2: Comparison of adaptation performance (in accuracy) of different retrieval methods. Compared to the zero-shot model (purple star), I2I retrieval significantly improves the performance and consistently outperforms T2I retrieval across shots and datasets.

7.3 A Finer-Grained Analysis of Retrieval-Augmented Adaptation

Different from recent works on algorithm design and incorporation of new knowledge sources Zhang et al. (2023); Iscen et al. (2023); Udandarao et al. (2023), the goal of our work is to present a systematic analysis with theoretical insights on how retrieval augmentation impacts adaptation for vision-language models. In this section, we present empirical analysis focusing on the impact of two aspects: retrieval method (Section 7.3.2) and logit ensemble with retrieved samples (Section 7.3.3). We will provide theoretical analysis to support these empirical findings in Section 7.4. We discuss alternative design choices and ablation studies in Section 7.5.

7.3.1 Settings

Datasets. Following prior works Zhang et al. (2022b), we consider a wide range of real-world datasets that span both common and finer-grained categories: Caltech101 (Fei-Fei et al., 2004), Birds200 (Wah et al., 2011), Food101 (Bossard et al., 2014), OxfordPets (Parkhi et al., 2012), Flowers102 Nilsback and Zisserman (2008), Textures Cimpoi et al. (2014), and UCF101 Soomro et al. (2012).

Implementation details. We use LAION-5B Schuhmann et al. (2022) as the retrieval database, which consists of 5.85 billion image-text pairs. For T2I retrieval, the default query set contains class descriptions with a prompt template. For I2I retrieval, by default, we use 8 seed images per class as the query set. Based on the query set, we use the clip-retrieval tool[1] for efficient retrieval from LAION-5B. We vary the number of retrieved samples per class $K \in \{1, 2, 4, 8, 16\}$. For adaptation, we use pre-trained CLIP with RN50 backbone as the default. Unless otherwise specified, each reported result is averaged over three independent runs. The ensemble weights of two logits α, γ are tuned on the validation set. Ablation studies on the number of seed images and alternative backbones are in Section 7.5. Further implementation details can be seen in Appendix F.1.

7.3.2 Impact of Retrieval Method

I2I retrieval consistently outperforms T2I retrieval. To better understand the impact of the retrieval method, we compare the adaptation performance (in Accuracy) using I2I and T2I retrieval. The results are shown in Figure 7.2, where the horizontal axis indicates the number of retrieved samples for each class (shot). As both I2I and T2I retrieval introduce dis-

[1] https://github.com/rom1504/clip-retrieval

tributional shifts *w.r.t.* the target distribution, we also plot the oracle performance when retrieving samples from the target training set for reference, denoted as ID retrieval (green). Directly retrieving from the target training set can be viewed as performance upper bound.

We observe several salient trends: **(1)** I2I retrieval consistently outperforms T2I retrieval across all shots and datasets. In particular, the gap between I2I and T2I increases when increasing the shot. **(2)** Compared to the zero-shot inference without knowledge augmentation (purple star), I2I retrieval significantly improves the performance. Notably, the gap between I2I retrieval and ID-retrieval (ideal) can be as small as 1% on average (12 shots), highlighting the potential of utilizing retrieved samples in the extremely low-data scheme where one does not have training data in the target dataset. **(3)** While T2I retrieval obtains a diverse collection of samples, the performance gain compared to the zero-shot CLIP for multiple datasets can be marginal. We investigate the reasons by a detailed examination of retrieved samples next and provide theoretical understanding in Section 7.4 (Theorem 7.1). Similar trends also hold for training-based adaptation, where we finetune the cache features as in Zhang et al. (2022b) (see Figure F.3 in Appendix F.5).

A closer look at retrieved samples. To better understand the effects of retrieval, we examine the samples retrieved by T2I and I2I respectively. The results are shown in Figure 7.3. While T2I retrieval often results in a diverse collection of images corresponding to the class semantics, we find that such diversity may not always be desirable for target task adaptation. For example, when using the query `a photo of a cellphone`, we retrieve images with a broad range of cellphone types. However, the downstream dataset contains cellphones typical in the 2000s with physical keypads. The same phenomenon widely exists in the suite of datasets commonly used in the literature (see Appendix F.3 for more extensive

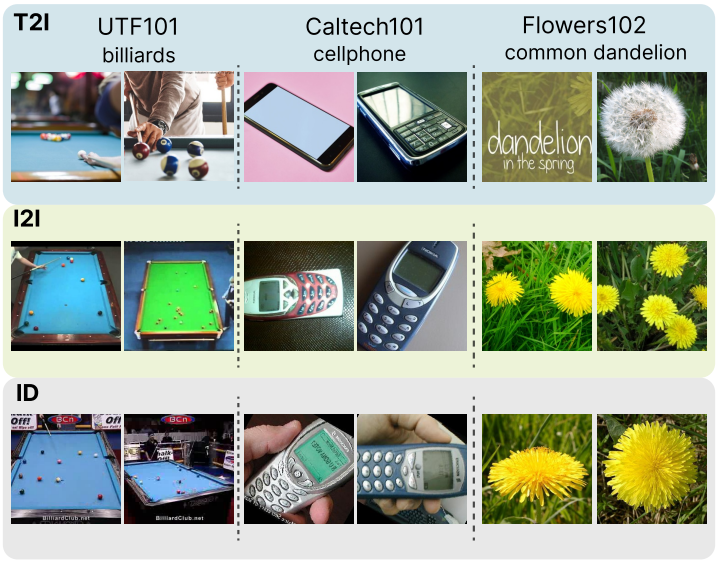


Figure 7.3: Samples from T2I and I2I retrieval. Top row: the main source of noise for T2I retrieval is semantic ambiguity, as the textual queries (*e.g.*, `a photo of a cellphone`) may not accurately describe the images from target distributions (*e.g.*, cellphones typical in the early 2000s). Middle row: samples retrieved by I2I matches more closely with ID data. Bottom row: images sampled from the target (ID) distribution. More examples can be seen in Appendix F.3.

examples) As a result, T2I retrieval can lead to undesirable performance due to semantic ambiguity. In contrast, I2I retrieval mitigates such ambiguity. For example, when using an image of a cellphone with smaller screens and physical keypads, one can retrieve images of older models of cellphones with similar layouts (middle row).

7.3.3 How Do Retrieved Samples Help Adaptation?

Ensemble with zero-shot prediction is the key. We show that ensembling the zero-shot prediction together with I2I-retrieved samples is the key to improved adaptation performance. The results are shown in Figure 7.4, where ensemble denotes using $f^{\text{EN}} = \alpha f^{\text{ZOC}} + \gamma f^{\text{RET}}$ with $\alpha, \gamma \in$

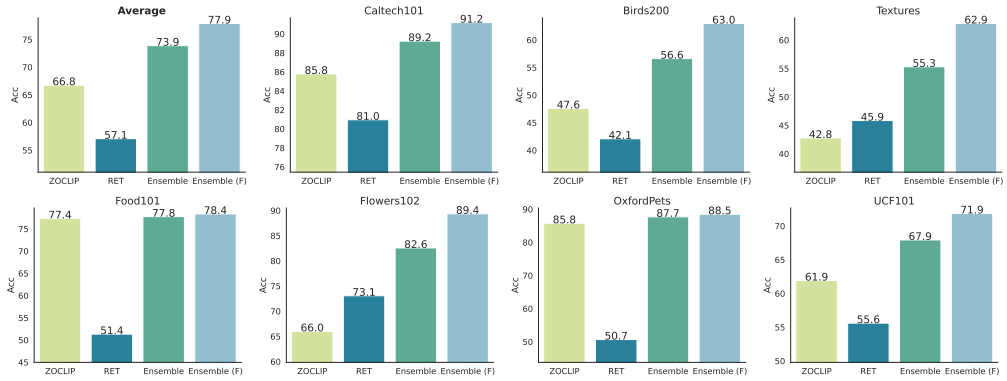


Figure 7.4: Importance of ensemble for I2I retrieval. Ensemble corresponds to the default logit ensemble: $f^{\text{EN}} = \alpha f^{\text{ZOC}} + \gamma f^{\text{RET}}$ with $\alpha, \gamma \in (0, 1)$. RET denotes only using f^{RET} $(\alpha = 0, \gamma = 1)$ and ZOCLIP denotes only using f^{ZOC} $(\alpha = 1, \gamma = 0)$. By ensembling the prediction with retrieved samples $(K = 16)$, the performance improvement over zero-shot prediction is significant for most datasets.

$(0, 1)$, RET denotes only using f^{RET} $(\alpha = 0, \gamma = 1)$, and ZOCLIP means only using f^{ZOC} $(\alpha = 1, \gamma = 0)$. This interesting phenomenon highlights the importance of logit ensembling for adapting vision-language models to downstream tasks. The benefits can also be seen by examining the class-wise performance of RET and Ensemble (see Figure F.1 in Appendix F.2). Similar trends also hold for training-based adaptation, denoted as Ensemble (F), where we finetune the cache features as in Zhang et al. (2022b). Next, we provide further theoretical explanations (Theorem 7.2).

7.4 Theoretical Understanding

We now provide theory to support our empirical observations and formally understand retrieval-augmented task adaptation. As an overview, **Theorem 7.1** shows why I2I retrieval is superior to T2I retrieval. We further prove that logit ensemble is the key for retrieval-augmented adapta-

tion in **Theorem 7.2**. These two theorems justify our empirical results in Section 7.3. Full proof is in Appendix F.4.

7.4.1 Problem Setup

Given a downstream task with C classes, let $[C] := \{1, 2, \cdots, C\}$. $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_C] \in \mathbb{R}^{d\times C}$ denotes the text embedding matrix for all classes, where $\mathbf{t}_c := \mathcal{T}(t_c) \in \mathbb{R}^d$ and t_c is a generic textual description of class c. Recall that $\mathbf{K} = [\mathbf{k}_{1,1}, \mathbf{k}_{1,2} \ldots, \mathbf{k}_{C,K}] \in \mathbb{R}^{d\times CK}$ denotes the embedding matrix for retrieved images, where $\mathbf{k}_{c,i} := \mathcal{I}(\mathbf{x}_{c,i}) \in \mathbb{R}^d$. For notational simplicity, we assume text and image features are ℓ_2 normalized. Let $\bar{\mathbf{K}} = \frac{\mathbf{K}\mathbf{V}^\top}{K} \in \mathbb{R}^{d\times C}$ contain the average retrieved feature for each class. $\mathbf{V} \in \mathbb{R}^{C\times CK}$ is a sparse matrix containing the one-hot labels for retrieved samples with entries $\mathbf{V}_{i,j} = \mathbb{1}\{i = \tilde{j}\}$ for $i \in [C], j \in [CK]$, where $\tilde{j} := \left\lceil \frac{j}{K} \right\rceil$ Zhang et al. (2022b). For example, when $K = 2, C = 3$, we have:

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

At inference time, let $(\mathbf{x}, y) \sim \mathcal{D}_T$ be a test sample from the target distribution $\mathcal{D}_T$ with label $y \in [C]$ and its visual feature $\mathbf{z} := \mathcal{I}(\mathbf{x})$. The final logit for the test sample can be represented as a weighted sum (ensemble) of logits from the zero-shot CLIP and the feature cache from retrieval:

$$f(\mathbf{x}) = (\alpha\mathbf{T} + \gamma\bar{\mathbf{K}})^\top \mathbf{z},$$

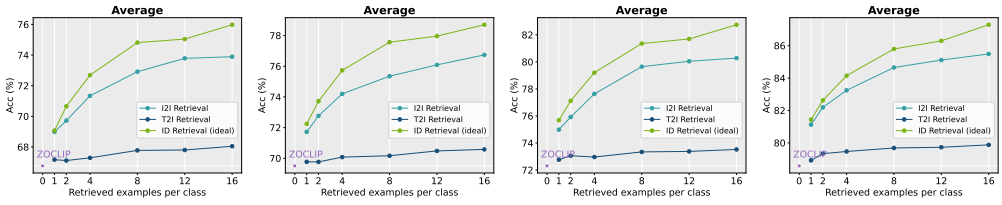
where $0 \leq \alpha, \gamma \leq 1$.

Given a loss function ℓ (e.g., cross-entropy), the risk on the downstream distribution is $\mathcal{L}(f) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T}[\ell(f(\mathbf{x}), y)]$. To simplify notations, we denote the risk as $\mathcal{R}(\mathbf{Q}) := \mathbb{E}\left[\ell(\mathbf{Q}^\top\mathbf{z}, y)\right]$ for some $\mathbf{Q} \in \mathbb{R}^{d\times C}$. For example, the risk of logit ensemble is $\mathcal{R}(\alpha\mathbf{T} + \gamma\bar{\mathbf{K}})$.

Modality gap and retrieval distribution shift. To understand the impact of retrieval, we characterize the distributional shift between the retrieved data and downstream data in the feature space. We define $\bar{\mathbf{s}}_c := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathcal{I}(\mathbf{x})|y = c]$ as the image representation of class $c \in [C]$ based on the downstream distribution. Let $\bar{\mathbf{S}} := [\bar{\mathbf{s}}_1, \ldots, \bar{\mathbf{s}}_C]$. We define the distributional shift between the retrieved data and target data for T2I and I2I retrieval as ξ_c^{T2I} and ξ_c^{I2I} for class c. Let $\xi_{\mathbf{t}} := \max_{c\in[C]} \xi_c^{\text{T2I}}$ and $\xi_{\mathbf{s}} := \max_{c\in[C]} \xi_c^{\text{I2I}}$ (Definition F.4). We can obtain an upper bound for $\xi_{\mathbf{s}}$ and a lower bound for $\xi_{\mathbf{t}}$ by Lemma F.10.

7.4.2 Main Results

Under realistic assumptions of T2I and I2I retrieval on the pre-trained feature space, we present two key results below. The detailed versions with full proof are in Appendix F.4.



(a) RN50 (b) ViT-B/32 (c) ViT-B/16 (d) ViT-L/14

Figure 7.5: Impact of architecture. We report the average performance (over all datasets) for I2I retrieval and T2I retrieval under different CLIP backbones and observe consistent trends. Results for individual datasets can be seen in Appendix F.6.

Theorem 7.1 (Benefit of uni-modal retrieval)**.** *With probability at least* $1-\delta$,

the following upper bound of the ensemble risk holds:

$$\begin{aligned}&\mathcal{R}(\alpha\mathbf{T}+\gamma\bar{\mathbf{K}})-\mathcal{R}(\bar{\mathbf{S}})\\ \leq& L\Bigg(\alpha\underbrace{\|(\mathbf{T}-\bar{\mathbf{S}})^\top\mathbf{z}\|_2}_{\textit{modality gap}}+\underbrace{\gamma\kappa\sqrt{\frac{8C}{K}\log\frac{C}{\delta}}}_{\textit{retrieval sample complexity}}+\underbrace{\gamma\sqrt{2C\xi}}_{\textit{retrieval shift}}\Bigg),\end{aligned}$$

where $L\leq\sqrt{\exp(2)+1}$, κ *characterizes the inner-class feature concentration (Definition F.1), and* ξ *is either* $\xi_{\mathbf{s}}$ *for I2I retrieval or* $\xi_{\mathbf{t}}$ *for T2I retrieval.*

Interpretations: The above upper bound consists of three terms: the gap between the textual and visual modality, the sample complexity of retrieved features which decreases as we increase K, and a term related to the distributional shift induced by the retrieval method. By Lemma F.10, we can further show that I2I provably outperforms T2I retrieval due to a smaller ξ.

Further, to understand the benefit of logit ensemble, we define the following three events:

$$\begin{aligned}E_1:=&\{(\mathbf{x},y)\sim\mathcal{D}_T: y\neq\arg\max_{c\in[C]}\mathbf{t}_c^\top\mathbf{z}\text{ and }y\neq\arg\max_{c\in[C]}\bar{\mathbf{k}}_c^\top\mathbf{z}\}\\ E_2:=&\{(\mathbf{x},y)\sim\mathcal{D}_T: y=\arg\max_{c\in[C]}\mathbf{t}_c^\top\mathbf{z}\text{ and }y\neq\arg\max_{c\in[C]}\bar{\mathbf{k}}_c^\top\mathbf{z}\}\\ E_3:=&\{(\mathbf{x},y)\sim\mathcal{D}_T: y\neq\arg\max_{c\in[C]}\mathbf{t}_c^\top\mathbf{z}\text{ and }y=\arg\max_{c\in[C]}\bar{\mathbf{k}}_c^\top\mathbf{z}\}\end{aligned}$$

Here E_1 indicates that both f^{ZOC} and f^{RET} incorrectly classify the test sample, while E_2 and E_3 denote the event where only one of them makes a correct prediction. We can see that $\mathcal{R}_{0-1}(f^{\text{ZOC}})=\Pr(E_1)+\Pr(E_3)$ and $\mathcal{R}_{0-1}(f^{\text{RET}})=\Pr(E_1)+\Pr(E_2)$.

Theorem 7.2 (Benefit of logit ensemble)**.** *Under realistic assumptions for I2I*

retrieval, when $\alpha = \gamma = \frac{1}{2}$, we can upper bound the 0-1 risk of logit ensemble:

$$\mathcal{R}_{0-1}(f) \leq \Pr(E_1) + C_1(\Pr(E_2) + \Pr(E_3)) + \rho_c$$

where $C_1 := \rho_d \max\{6\kappa - \nu, 2\kappa + \tau\}$ is a term related to modality gap, inner-class feature concentration, and inter-class separation. ρ_c characterizes the ratio of outliers. See Appendix F.4 for detailed definitions of $\kappa, \tau, \nu, \rho_c$, and ρ_d.

Interpretations: The above theorem characterizes the 0-1 risk upper bound by the modality gap and key properties of retrieved and target distributions. Moreover, logit ensemble utilizes knowledge encoded in different modalities to benefit each other. We can further show that under some conditions (detailed in Appendix F.4), logit ensemble leads to a lower 0-1 risk (*i.e.,* higher accuracy) than the zero-shot model.

7.5 Discussion of Design Choices

In this section, we discuss the impact of other design choices for retrieval-augmented adaptation.

Impact of model architecture. We conduct an ablation study on the impact of model architectures. We consider CLIP with ResNet (RN50) and ViT Dosovitskiy et al. (2021) backbones (CLIP-B/32, CLIP-B/16, CLIP-L/14), where the vision encoder is based on ViT-B/32 and ViT-L/14, respectively. The results are shown in Figure 7.5. We observe that a similar trend holds for CLIP with various backbones, where I2I retrieval consistently outperforms T2I retrieval. In particular, larger backbones such as CLIP-L/14 lead to overall superior performance compared to smaller backbones across the number of retrieved samples per class.

Impact of the number of seed images. To investigate the impact of seed images on I2I retrieval, we adjust the number of seed images per class from 2 to 8. The results are shown in Table 7.1 based on Textures ($K = 16$). We can see that increasing the number of seed images improves the adaptation performance because it is less prone to overfitting to limited retrieved samples. Similar trends also hold for other datasets in the test suite.

Seed #	Method			
	ZOCLIP	RET	Ensemble	Ensemble (F)
2	42.79	38.48	51.77	57.98
4	42.79	44.09	52.96	58.57
8	42.79	45.86	55.32	62.94

Table 7.1: The impact of the number of seed images (per class) for I2I retrieval. Results are based on RN50 backbone with $K = 16$.

Adaptation with a mixture of ID and retrieved samples. Previously, we have considered only using retrieved samples in the feature cache to better understand the effects of retrieval. When we have access to the few-shot (ID) training set, another practical scenario is to use a mixture of retrieved and ID samples. The results are shown in Figure 7.6. We report the average performance (over 7 datasets) for I2I retrieval ($K = 16$). EN denotes logit ensemble with only retrieved samples. MIX denotes logit ensemble with a mixture of ID samples and retrieved samples. EN (F) and MIX (F) stand for the finetuned variants. The mixture ratio is 1:1. We observe that mixing ID and retrieved samples further leads to improved performance compared to only using few-shot ID samples. Our observations are consistent with prior works Udandarao et al. (2023); Zhang et al. (2023) under different logit ensemble schemes, which highlight the potential of retrieval-augmented few-shot adaptation.

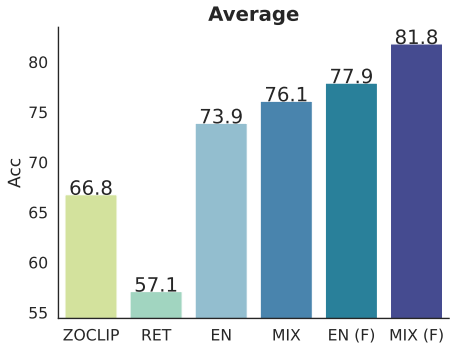


Figure 7.6: Impact of Mixture of retrieved samples with few-shot ID data. We report the average performance (over all datasets) for I2I retrieval ($K = 16$). EN denotes logit ensemble with only retrieved samples. MIX denotes logit ensemble with a mixture of ID samples and retrieved samples. The mixture ratio is 1:1.

Adaptation with finetuned feature cache. For completeness, we discuss learning-based adaptation by setting the visual features in the cache $\mathbf{K}$ as learnable parameters after initializing from the pre-trained CLIP model. We denote the variant as `Ensemble(F)`, where F stands for fine-tuning. We follow the hyperparameter tuning scheme in Zhang et al. (2022b) and show the results (averaged across all datasets) in Figure 7.7. We can see that a similar trend holds for training-based adaptation, where I2I retrieval significantly outperforms zero-shot CLIP and T2I retrieval. In the low-shot setting ($K = 1$ or 2), the performance is close to the ideal case (ID retrieval). Full results for individual datasets can be seen in Appendix F.5.

We provide additional ablation studies in the Appendix.

7.6 Related Works

Few-shot task adaptation for vision-language models. Recent years have witnessed the popularity of contrastive language-image pre-training (CLIP) Rad-

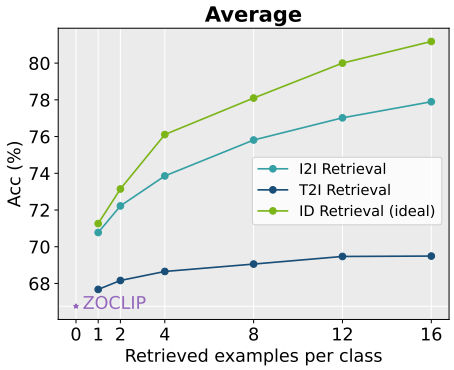


Figure 7.7: Adaptation with finetuned feature cache. We observe a similar trend as training-free adaptation.

ford et al. (2021); Jia et al. (2021); Yang et al. (2022); Li et al. (2022c); Mu et al. (2022); Yu et al. (2022); Zhai et al. (2022); Sun et al. (2023); Zhai et al. (2023); Xu et al. (2024), etc. While CLIP-like models learn aligned multi-modal features, they often struggle on fine-trained datasets with categories not adequately represented during pre-training, which makes adaptation necessary. Recent works propose various promising solutions for adapting the vision-language model in the low-data (few-shot) scheme such as tuning textual prompts Zhou et al. (2022b,c), visual prompts Bahng et al. (2022); Chen et al. (2023a), multi-modal prompts Khattak et al. (2023). Zhang et al. (2022c) use neural architecture search to optimize prompt modules. Lu et al. (2022) optimize prompts by learning prompt distributions. Alternatively, Yu et al. (2023) tune an additional task residual layer. Another line of work utilizes adaptor Zhang et al. (2022b); Gao et al. (2023); Zhang et al. (2023); Udandarao et al. (2023) by maintaining a memory cache that stores the features of few-shot data. Zhang et al. (2022b) uses an additive logit ensemble with a feature cache from the target training set. In contrast, we focus on the impact of retrieval and build the cache with retrieved samples, rather than the downstream dataset.

Knowledge-augmented adaptation for CLIP. A natural idea for task adaptation is to utilize external knowledge sources by retrieval or synthesis. Sampling from external datasets has shown promising performance in adapting vision models to fine-grained datasets Liu et al. (2022); Kim et al. (2023b). For CLIP-based adaptation, existing methods can be categorized into two regimes, based on the amount of external data utilized. In the high-data regime, Liu et al. (2023) demonstrates promising zero-shot performance by first constructing a large-scale dataset (10M) containing relevant samples retrieved from web-scale databases and then fine-tuning CLIP models on the retrieved dataset. Xie et al. (2023) propose a Retrieval Augmented Module to augment CLIP pre-training on 1.6M retrieved samples. Recently, Iscen et al. (2023) advocated uni-modal search but cross-modal fusion for CLIP adaptation, where the fusion model is trained on 10M samples. Long et al. (2022) demonstrate the promise of retrieval for long-tail visual recognition tasks. In the low-data regime, recent works also enhance the retrieval augmentation pipeline with synthetic samples from pre-trained generative models Udandarao et al. (2023); Zhang et al. (2023). Beyond augmenting the visual modality, Shen et al. (2022) leverage external text knowledge sources such as WordNet Miller (1995) and Wiktionary Meyer and Gurevych (2012) to augment captions with class-specific descriptions, while Pratt et al. (2023) perform augmentation by querying large language models. El Banani et al. (2023) use the language guidance to find similar visual nearest neighbors. Li et al. (2022a) establish a benchmark for evaluating the transfer learning performance of language-augmented visual models. In this work, we adopt a reflective perspective and provide a systematic study to understand retrieval-augmented adaptation in the low-data regime and establish new theoretical insights.

Theoretical understanding of multi-modal learning. A few works provide theoretical explanations for multi-modal learning Zadeh et al. (2020); Huang et al. (2021b); Fürst et al. (2022); Chen et al. (2023d). For CLIP models, Liang et al. (2022) demonstrate and provide a systematic analysis of the modality gap between the features of two modalities. Nakada et al. (2023) establish the connection between CLIP and singular value decomposition (SVD) under linear representations. Chen et al. (2023d) develop a theoretical framework to understand the zero-shot transfer mechanism of CLIP. Different from prior works, we focus on the theoretical understanding of retrieval-augmented task adaptation.

7.7 Conclusion

In this work, we present a timely and systematic investigation for retrieval-augmented adaptation of vision-language models in the low-data regime. Our work offers a finer-grained empirical study, unveiling insights into the impact of cross-modal and uni-modal retrieval. In addition, we highlight the necessity of logit ensemble. We also develop a novel theoretical framework that supports our empirical findings and provides a deeper understanding of retrieval-augmented adaptation. Additionally, our comprehensive ablation study explores various design choices in the retrieval augmentation pipeline. We hope our work will serve as a springboard for future research on algorithm design and theoretical understanding for effective adaptation of vision-language models.

7.8 Impact Statements

The main purpose of this work is to provide a systematic investigation of existing approaches with theoretical understanding. The work can help guide the development of effective and reliable algorithms for retrieval-

augmented adaptation of vision-language models. We conducted a thorough manual review to ensure that the retrieved samples do not contain illegal or inappropriate content, and we foresee no immediate negative ethical impact.

Part III

Extensions and Appendices

Chapter 8

A Critical Analysis of Document Out-of-Distribution Detection

Publication Statement. This chapter is a joint work with Jiuxiang Gu, Yi Zhou, Jason Kuen, Vlad I. Morariu, Handong Zhao, Ruiyi Zhang, Nikolaos Barmpalios, Anqi Liu, Yixuan Li, Tong Sun, and Ani Nenkova The paper version of this chapter appeared in EMNLP 2023-Findings (Gu et al., 2023).

Large-scale pre-training is widely used in recent document understanding tasks. During deployment, one may expect that models should trigger a conservative fallback policy when encountering out-of-distribution (OOD) samples, which highlights the importance of OOD detection. However, most existing OOD detection methods focus on single-modal inputs such as images or texts. While documents are multi-modal in nature, it is underexplored if and how multi-modal information in documents can be exploited for OOD detection. In this chapter, we first provide a systematic and in-depth analysis on OOD detection for document understanding models. We study the effects of model modality, pre-training, and fine-tuning across various types of OOD inputs. In particular, we find that spa-

tial information is critical for document OOD detection. To better exploit spatial information, we propose a spatial-aware adapter, which serves as a parameter-efficient add-on module to adapt transformer-based language models to the document domain. Extensive experiments show that adding the spatial-aware adapter significantly improves the OOD detection performance compared to directly using the language model and achieves superior performance compared to competitive baselines.

8.1 Introduction

The recent success of large-scale pre-training has propelled the widespread deployment of deep learning models in the document domain, where model predictions are used to help humans make decisions in various applications such as tax form processing and medical reports analysis. However, models are typically pre-trained on data collected from the web but deployed in an environment with distributional shifts Cui et al. (2021). For instance, the outbreak of COVID-19 has led to continually changing data distributions in machine-assisted medical document analysis systems Velavan and Meyer (2020). This motivates the need for reliable document understanding models against out-of-distribution (OOD) inputs.

The goal of OOD detection is to categorize in-distribution (ID) samples into one of the known categories and detect inputs that do not belong to any known classes at test time Bendale and Boult (2016). A plethora of OOD detection methods has been proposed for single-modal (image or text) inputs Ge et al. (2017); Nalisnick et al. (2019); Oza and Patel (2019); Tack et al. (2020); Hsu et al. (2020); Arora et al. (2021); Zhou et al. (2021b); Xiao et al. (2020); Xu et al. (2021a); Li et al. (2021d); Shen et al. (2021); Jin et al. (2022); Zhou et al. (2022d); Ming et al. (2022b,c); Podolskiy et al. (2021); Ren et al. (2023). Recent works Fort et al. (2021); Esmaeilpour et al. (2022); Ming et al. (2022a); Ming and Li (2023); Bit-

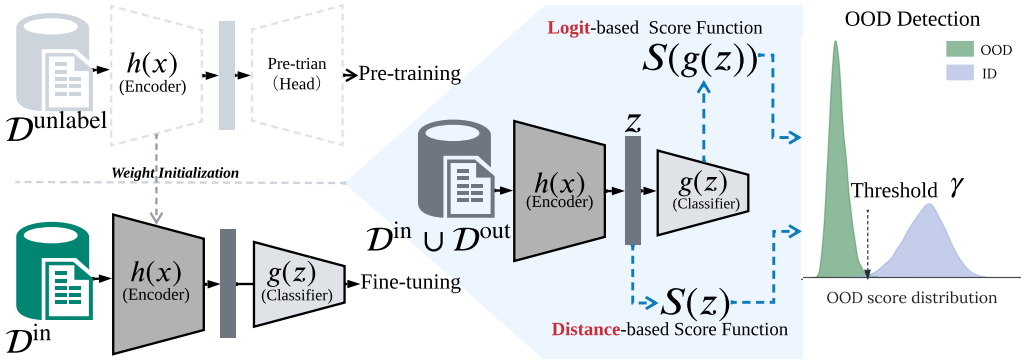


Figure 8.1: Illustration of OOD detection for document classification. The pre-training and fine-tuning pipelines are shown on the top left and bottom left, respectively. Right: During inference time, an OOD score can be derived based on logits $g(x)$ or feature embeddings $z := h(x)$. A document input x is identified as OOD if its OOD score is below some threshold γ.

terwolf et al. (2023) also demonstrate promising OOD detection performance based on large-scale models pre-trained on text-image pairs, as pre-training enables models to learn powerful and transferable feature representations Radford et al. (2021). However, it remains largely unexplored if existing findings in the OOD detection literature for images or texts can be naturally extended to the document domain.

Multiple unique challenges exist for document OOD detection. Unlike natural images, texts, or image-text pairs, no captions can describe a document and images in documents rarely contain natural objects. Moreover, the spatial relationship of text blocks further differentiates multimodal learning in documents from multimodal learning in the vision-language domain Lu et al. (2019); Li et al. (2020a). In addition, while recent pre-training methods have demonstrated remarkable performance in downstream document understanding tasks Xu et al. (2020, 2021b); Li et al. (2021c); Gu et al. (2022); Hong et al. (2022); Huang et al. (2022b); Li et al. (2022b); Wang et al. (2022f), existing pre-training datasets for documents are limited and lack diversity. This is in sharp contrast to common pre-

training datasets for natural images. It remains underexplored whether existing OOD detection methods are reliable in the document domain and how pre-training impacts OOD reliability.

In this work, we first present a comprehensive study to better understand OOD detection in the document domain through the following questions: (*1*) What is the role of document pre-training? How do pre-training datasets and tasks affect OOD detection performance? (*2*) Are existing OOD detection methods developed for natural images and texts transferrable to documents? (*3*) How does modality (textual, visual, and especially *spatial* information) affect OOD performance? In particular, we find that spatial information is critical for improving OOD reliability. Moreover, we propose a new spatial-aware adapter, a small learned module that can be inserted within a pre-trained language model such as RoBERTa Liu et al. (2019). Our module is computationally efficient and significantly improves both ID classification and OOD detection performance (Sec. 8.5.2). Our contributions are summarized as follows:

- We provide an extensive and in-depth study to investigate the impacts of pre-training, fine-tuning, model-modality, and OOD scoring functions on a broad spectrum of document OOD detection tasks. Our codebase will be open-sourced to facilitate future research.

- We present unique insights on document OOD detection. For example, we observe that distance-based OOD scores are consistently advantageous over logit-based scores, which is underexplored in the recent OOD detection literature on vision-language pre-trained models.

- We further propose a spatial-aware adapter module for transformer-based language models, facilitating easy adaptation of pre-trained language models to the document domain. Extensive experiments

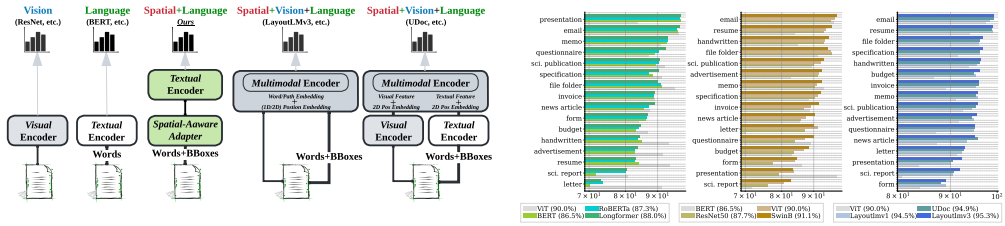
confirm the effectiveness of our module across diverse types of OOD data.

8.2 Preliminaries and Related Works

8.2.1 Document Models and Pre-Training

Large-scale pre-trained models gradually gain popularity in the document domain due to their success in producing generic representations from large-scale unlabeled corpora in vision and natural language processing (NLP) tasks Devlin et al. (2018); Lu et al. (2019); Su et al. (2019); Schiappa et al. (2022). As documents contain both visual and textual information distributed spatially in semantic regions, document-specific models and pre-training objectives are often necessary, which are distinct from vision or language domains.

We summarize common model structures for document pre-training in Fig. 8.2a. Specifically, LayoutLM Xu et al. (2020) takes a sequence of Optical Character Recognition (OCR) Smith (2007) words and word bounding boxes as inputs. It extends BERT to learn contextualized word representations for document images through multitask learning. LayoutLMv2 Xu et al. (2021b) improves on the prior work with new pre-training tasks to model the interaction among texts, layouts, and images. DocFormer Appalaraju et al. (2021) adopts a CNN model to extract image grid features, fusing the spatial information as an inductive bias for the self-attention module. LayoutLMv3 Huang et al. (2022b) further enhances visual and spatial characteristics with masked image modeling and word-patch alignment tasks. Another line of work focuses on various granularities of documents, such as region-level text/image blocks. Examples of such models include SelfDoc Li et al. (2021c), UDoc Gu et al. (2021), and MGDoc Wang et al. (2022g), which are pre-trained with a cross-modal encoder to capture the relationship between visual and tex-



(a) Illustration of common structures for document pre-training and classification.

(b) A detailed comparison of per-category accuracy on the RVL-CDIP test set.

Figure 8.2: (**Left**) Illustration of models for document pre-training and classification, with our proposed spatial-aware models in green blocks. Modality information is also shown atop each architecture. (**Right**) Evaluating fine-tuning performance for document classification of pre-trained models. Models are grouped into several categories (from left to right): language-only, vision-only, and multi-modal. For comparison, the performance of corresponding models in other groups is shown in gray. The average accuracy for each model is indicated in the parenthesis.

tual features. These models incorporate spatial information by fusing position embeddings at the output layer of their encoders, instead of the input layer. Additionally, OCR-free models Kim et al. (2022); Tang et al. (2023) tackle document understanding as a sequence generation problem, unifying multiple tasks through an image-to-sequence generation network.

While these pre-trained models demonstrate promising performance on downstream applications, their robustness to different types of OOD data, the influence of pre-training and fine-tuning, and the value of different modalities (*e.g.*, spatial, textual, and visual) for document OOD detection remain largely unexplored.

8.2.2 Out-of-Distribution Detection

OOD detection has been extensively studied for open-world multi-class classification with natural image and text inputs, where the goal is to de-

rive an OOD score that separates OOD from ID samples. A plethora of methods are proposed for deep neural networks, where the OOD scoring function is typically derived based on logits (without softmax scaling) Hendrycks et al. (2022), softmax outputs Liang et al. (2018); Hsu et al. (2020); Huang and Li (2021); Sun et al. (2021), gradients Huang et al. (2021a), and feature embeddings Tack et al. (2020); Fort et al. (2021); Ming et al. (2023). Despite their impressive performance on natural images and texts, it is underexplored if the results are transferrable to the document domain. A recent work Larson et al. (2022) studied OOD detection for documents but only explored a limited number of models and OOD detection methods. The impacts of pre-training, fine-tuning, and spatial information remain unknown. In this work, we aim to provide a comprehensive and finer-grained analysis to shed light on the key factors for OOD robustness in the document domain.

Notations. Following prior works on OOD detection with large-scale pre-trained models Ming et al. (2022a); Ming and Li (2023), the task of OOD detection is defined with respect to the downstream dataset, instead of the pre-training data which is often hard to characterize. In document classification, we use $\mathcal{X}^{\text{in}}$ and $\mathcal{Y}^{\text{in}} = \{1, \dots, K\}$ to denote the input and label space, respectively. Let $\mathcal{D}^{\text{in}} = \{(\boldsymbol{x}_i^{\text{in}}, y_i^{\text{in}})\}_{i=1}^{N}$ be the ID dataset, where $\boldsymbol{x} \in \mathcal{X}^{\text{in}}$, and $y^{\text{in}} \in \mathcal{Y}^{\text{in}}$. Let $\mathcal{D}^{\text{out}} = \{(\boldsymbol{x}_i^{\text{out}}, y_i^{\text{out}})\}_{i=1}^{M}$ denote an OOD test set where $y^{\text{out}} \in \mathcal{Y}^{\text{out}}$, and $\mathcal{Y}^{\text{out}} \cap \mathcal{Y}^{\text{in}} = \emptyset$. We express the neural network model $f := g \circ h$ as a composition of a feature extractor $h : \mathcal{X} \to \mathbb{R}^d$ and a classifier $g : \mathbb{R}^d \to \mathbb{R}^K$, which maps the feature embedding of an input to K real-valued numbers known as logits. During inference time, given an input $\boldsymbol{x}$, OOD detection can be formulated as:

$$G_\gamma(\boldsymbol{x}; h, g) = \begin{cases} \text{ID} & S(\boldsymbol{x}; h, g) \geq \gamma \\ \text{OOD} & S(\boldsymbol{x}; h, g) < \gamma \end{cases},$$

where $S(\cdot)$ is a scoring function that measures OOD uncertainty. In practice, the threshold $\mathrm{q}\gamma$ is often chosen so that a high fraction of ID data (*e.g.*, 95%) is above the threshold.

OOD detection scores. We focus on two major categories of computationally efficient OOD detection methods[1]: logit-based methods derive OOD scores from the logit layer of the model, while distance-based methods directly leverage feature embeddings, as shown in Fig. 8.1. We describe a few popular methods for each category as follows.

- **Logit-based:** Maximum Softmax Probability (MSP) score Hendrycks and Gimpel (2017) $S_{\text{MSP}} = \max_{i\in[K]} e^{f_i(\boldsymbol{x})} / \sum_{j=1}^{K} e^{f_j(\boldsymbol{x})}$ naturally arises as a classic baseline as models often output lower softmax probabilities for OOD data; Energy score Liu et al. (2020): $S_{\text{Energy}} = \log \sum_{i\in[K]} e^{f_i(\boldsymbol{x})}$ utilizes the Helmholtz free energy of the data and theoretically aligns with the logarithm of the ID density; the simple MaxLogit score Hendrycks et al. (2022): $S_{\text{Maxlogit}} = \max_{i\in[K]} f_i(\boldsymbol{x})$ has demonstrated promising performance on large-scale natural image datasets. We select the above scores due to their simplicity and computational efficiency. In addition, recent studies demonstrate that such simple scores are particularly effective with large-scale pre-trained models in vision Fort et al. (2021) and vision-language domains Ming et al. (2022a); Bitterwolf et al. (2023). We complement previous studies and investigate their effectiveness for documents.

- **Distance-based:** Distance-based methods directly leverage feature embeddings $\mathbf{z} = h(\mathbf{x})$ based on the idea that OOD inputs are relatively far away from ID clusters in the feature space, compared to ID

[1]We also investigate gradient-based methods such as GradNorm Huang et al. (2021a) in Appendix G.3.

inputs. Distance-based methods can be characterized as parametric and non-parametric. Parametric methods such as Mahalanobis score Lee et al. (2018); Sehwag et al. (2021) assume ID embeddings follow class-conditional Gaussian distributions and use the Mahalanobis distance as the distance metric. On the other hand, non-parametric methods such as KNN+ Sun et al. (2022) use cosine similarity as the distance metric.

Evaluation metrics. To evaluate OOD detection performance, we adopt the following commonly used metrics: the Area Under the Receiver Operating Characteristic (AUROC), False Positive Rate at 95% Recall (FPR95), and the multi-class classification accuracy (ID Acc).

8.3 Experimental Setup

Models. Fig. 8.2a summarizes common structures for document pre-training and classification models[2]. While documents typically come in the form of images Harley et al. (2015), an OCR system can be used to extract words and their coordinates from the input image. Therefore, models can use single-modal or multi-modal information. We categorize these models according to the input modalities into the following groups: (1) models using only visual features, (2) models using solely textual features, (3) models incorporating both visual and textual features, and (4) models integrating additional spatial (especially layout) information. Further details can be found in Appendix G.1.

- **Vision-only:** Document classification can be viewed as a standard image classification problem. We consider ResNet-50 He et al. (2016)

[2] Apart from document classification, in the Appendix G.2, we also investigate OOD detection for two entity-level tasks: document entity recognition and document object detection.

and ViT Fort et al. (2021) as exemplar document image classification models. We adopt two common pre-training settings: (1) only pre-trained on ImageNet Deng et al. (2009) and (2) further pre-trained on IIT-CDIP Lewis et al. (2006) with masked image modeling (MIM)[3]. After pre-training, we append a classifier for fine-tuning.

- **Text-only:** Alternatively, we can view document classification as text classification since documents often contain text blocks. To this end, we use RoBERTa Liu et al. (2019) and Longformer Beltagy et al. (2020) as the backbones. RoBERTa can handle up to 512 input tokens while Longformer can handle up to 4,096 input tokens. We pre-train the language models with masked language modeling (MLM) on IIT-CDIP extracted text corpus.

- **Text+Layout:** Layout information plays a crucial role in the document domain, as shown in Fig. 8.3. To investigate the effect of layout information, we adopt LayoutLM as the backbone. We will show that spatial-aware models demonstrate promising OOD detection performance. However, such specialized models can be computationally expensive. Therefore, we propose a new spatial-aware adapter, a small learned module that can be inserted within a pre-trained language model such as RoBERTa and transforms it into a spatial-aware model, which is computationally efficient and competitive for both ID classification and OOD detection (Sec. 8.5.2).

- **Vision+Text+Layout:** For comprehensiveness, we consider LayoutLMv3 and UDoc, which are large and computationally intensive. Both models are pre-trained on the full IIT-CDIP for fairness. These

[3]Note that the document classification dataset we used in this paper, RVL-CDIP Harley et al. (2015), is a subset of IIT-CDIP. Hence, unless otherwise specified, the IIT-CDIP pre-training data used in this paper excludes RVL-CDIP.

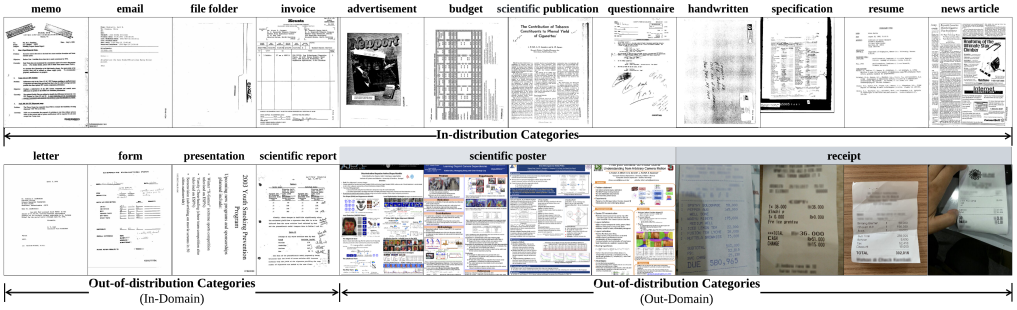


Figure 8.3: (**Top**) Examples of ID inputs sampled from RVL-CDIP (top). (**Bottom**) In-domain OOD from RVL-CDIP, and out-domain OOD from *Scientific Poster* and *Receipts*.

models utilize different input granularities and modalities, including textual, visual, and spatial information for document tasks.

Constructing ID and OOD datasets. We construct ID datasets from RVL-CDIP Harley et al. (2015), where 12 out of 16 classes are selected as ID classes. Dataset details are in Appendix G.1. We consider two OOD scenarios: in-domain and out-domain, based on the content (*e.g.*, , words, background) and layout characteristics.

- **In-domain OOD:** To determine the OOD categories, we analyzed the performance of recent document classification models on the RVL-CDIP test set. Fig. 8.2b shows the per-category test accuracy of various models. Naturally, for the classes the models perform poorly on, we may expect the models to detect such inputs as OOD instead of assigning a specific ID class with low confidence. We observe that the 4 categories (*letter*, *form*, *scientific report*, and *presentation*) result in the worst performance across most of the models with different modalities. We use these as OOD categories and construct the OOD datasets accordingly. The ID dataset is constructed from

the remaining 12 categories, which we refer to as *in-domain* OOD datasets, as they are also sourced from RVL-CDIP.

- **Out-domain OOD:** In the open-world setting, test inputs can have significantly different color schemes and layouts compared to ID samples. To mimic such scenarios, we use two public datasets as *out-domain* OOD test sets: NJU-Fudan Paper-Poster Dataset Qiang et al. (2019) and CORD Park et al. (2019). NJU-Fudan Paper-Poster Dataset contains scientific posters in digital PDF format[4]. CORD is a receipt understanding dataset with significantly different inputs compared to RVL-CDIP. As shown in Fig. 8.3, receipt images can be challenging and require models to handle not only textual but also visual and spatial information.

We further support our domain selection using OTDD Alvarez-Melis and Fusi (2020), a flexible geometric method for comparing probability distributions, which enables us to compare any two datasets regardless of their label sets. We observe a clear gap between in-domain and out-domain data, which aligns with our data selection. Further details can be found in Appendix G.1.1.

8.4 Analyzing OOD Reliability for Documents

8.4.1 OOD Detection Without Fine-Tuning

In this section, we begin by examining the influence of pre-training datasets on zero-shot OOD detection. For each model, we adopt the same pre-training objective while adjusting the amount of pre-training data. Specifically, we increase the data diversity by appending 10, 20, 40, and 100% of randomly sampled data from IIT-CDIP dataset (around 11M) and pre-

[4]Extracted using https://github.com/pymupdf/PyMuPDF

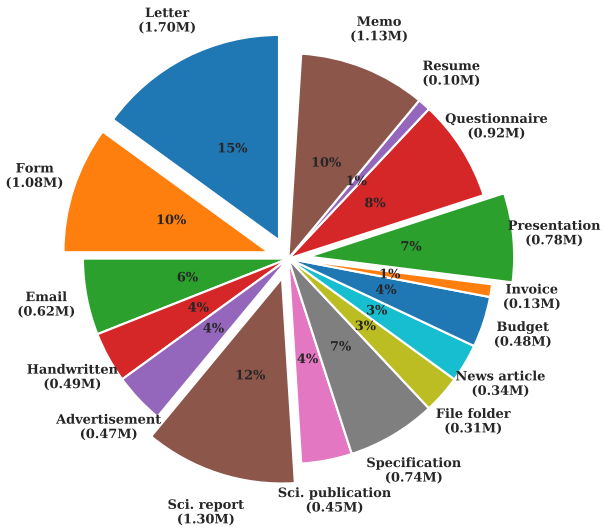
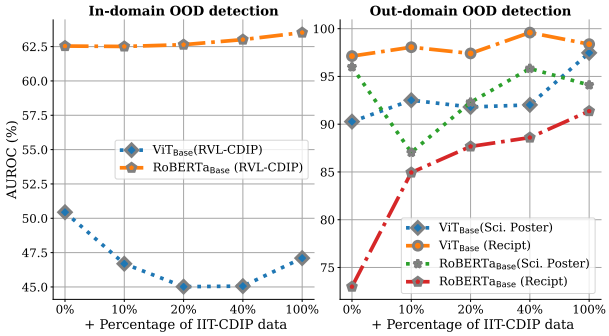


Figure 8.4: Analysis of IIT-CDIP.

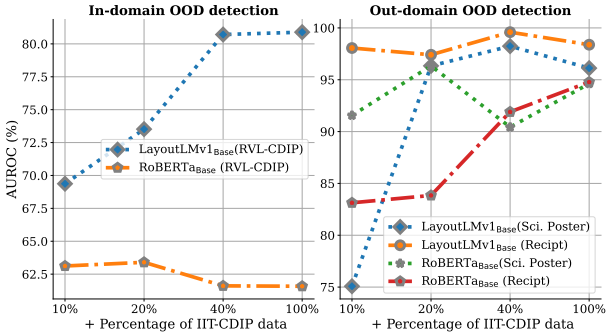
train each model. After pre-training, we measure the OOD detection performance with KNN+ score based on feature embeddings.

We observe that: (1) for out-domain OOD data (Fig. 8.5a, right), increasing the amount of pre-training data can significantly improve the zero-shot OOD detection performance (w.o. fine-tuning) for models across different modalities. Our hypothesis is that pre-training with diverse data is beneficial for coarse-grained OOD detection, such as inputs from different domains (*e.g.*, , color schemes). (2) For in-domain OOD inputs, even increasing the amount of pre-training data by over 40% provides negligible improvements (Fig. 8.5a, left). This suggests the necessity of fine-tuning for improving in-domain OOD detection performance (Fig. 8.7).

We further explore a more restricted setting for zero-shot OOD detection where potential OOD categories are removed from the pre-training dataset IIT-CDIP. First, we use LayoutLM fine-tuned on RVL-CDIP to predict labels for all documents in IIT-CDIP. Fig. 8.4 summarizes the distribution of the predicted classes on IIT-CDIP. Next, we remove the "OOD" categories from IIT-CDIP and pre-train two models (RoBERTa and Lay-



(a) Pre-train on IIT-CDIP.



(b) Pre-train on IIT-CDIP^{-}.

Figure 8.5: The impact of pre-training data on zero-shot OOD detection performance. IIT-CDIP^{-} denotes the filtered pre-training data after removing the "*OOD*" categories.

outLM) with 10, 20, 40, and 100% of randomly sampled data from the filtered IIT-CDIP (dubbed III-CDIP^{-}), respectively. The zero-shot OOD performance for in-domain and out-domain OOD is shown in Fig. 8.5b[5]. For RoBERTa, we observe similar trends as in Fig. 8.5a, where increasing the amount of pre-training data improves zero-shot OOD detection performance for *out-domain* data. However, the zero-shot performance of LayoutLM benefits from a larger pre-training dataset. In particular, given the same amount of pre-training data, LayoutLM consistently out-

[5]Note that we do not show 0% in Fig. 8.5b since we pre-train LayoutLM from scratch.

performs RoBERTa for both in-domain and out-domain OOD detection, which suggests that *spatial information* can be essential for boosting the OOD reliability in the document domain. Motivated by the above observations, we dive deeper and analyze spatial-aware models next.

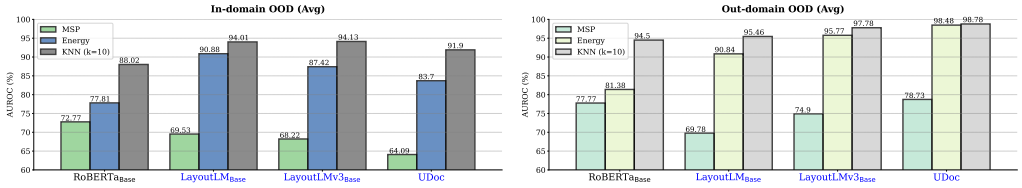


Figure 8.6: Comparison between representative feature-based scores and logit-based scores for spatial-aware and non-spatial-aware models. Spatial-aware models are colored in blue.

While pre-trained models exhibit the capability to differentiate data from various domains as a result of being trained on a diverse range of data. We observe that achieving more precise separation for in-domain OOD inputs remains difficult. Given this observation, we further analyze the impacts of fine-tuning for OOD detection with fixed pre-training datasets in the next section. By combining pre-trained models with a simple classifier and fine-tuning on RVL-CDIP (ID), we find that fine-tuning is advantageous in enhancing the OOD detection performance for both types of OOD samples.

8.4.2 The Impact of Fine-Tuning on Document OOD Detection

Recent document models are often pre-trained on a large-scale dataset and adapted to the target task via fine-tuning. To better understand the role of fine-tuning, we explore the following questions: 1) *How does fine-tuning impact OOD reliability for in-domain and out-domain OOD inputs?* 2) *How does model modality impact the performance?*

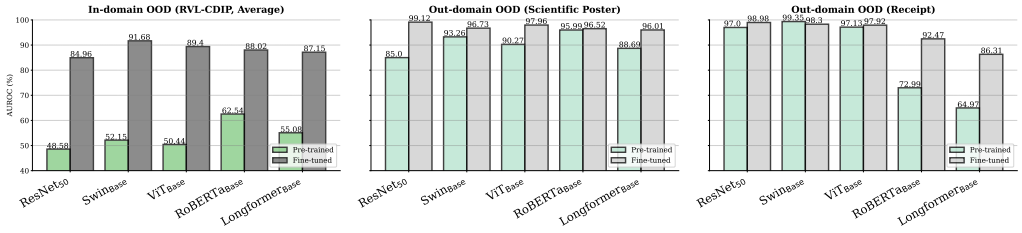


Figure 8.7: OOD detection performance for pre-trained models w. and w.o. fine-tuning. We use a distance-based method KNN+ as the OOD scoring function. Fine-tuning significantly improves performance for both in and out-domain OOD data.

We consider a wide range of models pre-trained on pure-text/image data (*e.g.*, , ImageNet and Wikipedia) described in Appendix G.1.3. During fine-tuning, we combine pre-trained models with a simple classifier and fine-tune on RVL-CDIP (ID). For models before and after fine-tuning, we extract the final feature embeddings and use a distance-based method KNN+ Sun et al. (2022) for OOD detection. The results are shown in Fig. 8.7. We observe the following trends. First, fine-tuning largely improves OOD detection performance for both in-domain and out-domain OOD data. The same trend holds broadly across models with different modalities. Second, the improvement of fine-tuning is less significant for out-domain OOD data. For example, on Receipt (out-domain OOD), the AUROC for pre-trained ViT model is 97.13, whereas fine-tuning only improves by 0.79%. This suggests that pre-trained models do have the potential to separate data from different domains due to the diversity of data used for pre-training, while it remains hard for pre-trained models to perform finer-grained separation for in-domain OOD inputs. Therefore, fine-tuning is beneficial for improving OOD detection performance for both types of OOD samples. To further validate our conclusion, we consider two additional in-domain OOD settings for our analysis: (1) selecting the classes the model performs well on, as in-domain OOD categories; (2) randomly selecting classes as OOD categories (Appendix G.1.2). We

find that fine-tuning improves OOD detection for both settings, further verifying our observations.

Next, we take a closer look at the impact of model modality on out-domain OOD detection. As shown in Fig. 8.7 (mid and right), both vision and text-based models demonstrate strong reliability against scientific posters (OOD). However, vision-based models display stronger performance than text-based models for Receipts (OOD). This can be explained by the fact that ViT was first pre-trained on ImageNet while scientific posters and receipts contain diverse visual information such as colors and edges for vision models to utilize (see Fig. 8.3). On the other hand, although fine-tuning text-based models largely improves the detection performance compared to pre-trained counterparts, utilizing only textual information can be inherently limited for out-domain OOD detection.

8.5 The Importance of Spatial-Awareness

In previous sections, we mainly focus on mainstream text-based and vision-based models for in- and out-domain OOD detection. Next, we consider models tailored to document processing, which we refer to as *spatial-aware models*, such as LayoutLMv3 and UDoc. Given fine-tuned models, we compare the performance of logit-based and distance-based OOD scores.

8.5.1 Analysis of Spatial-Aware Models

We summarize key comparisons in Fig. 8.6, where we use MSP and Energy as exemplar logit-based scores and KNN+ as the distance-based score. Full results are in Appendix G.3. We can see that the simple KNN-based score (KNN+) consistently outperforms logit-based scores for both in-domain and out-domain OOD data across different models with different modalities. This is *in contrast with* recent works that investigate large-scale pre-trained models in the vision-language domain, where logit-based

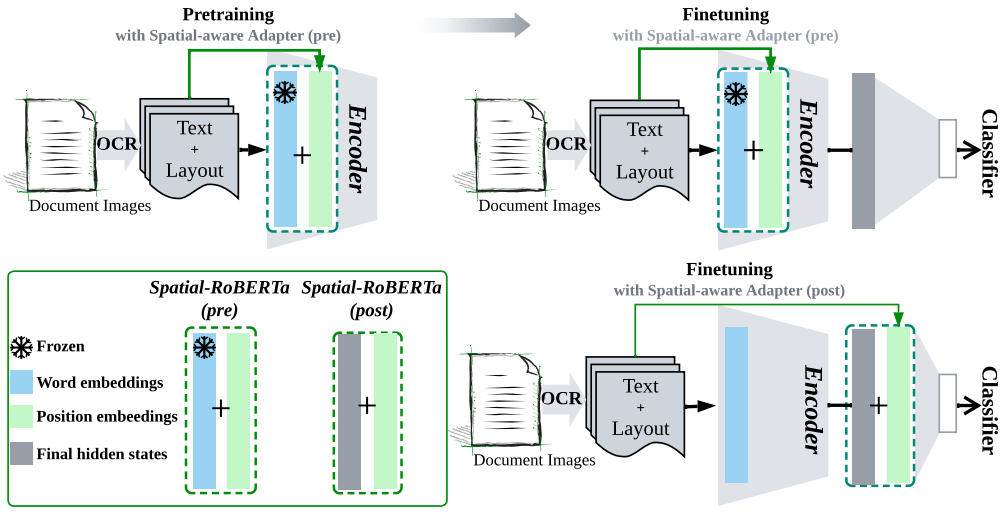


Figure 8.8: Illustration of our spatial-aware adapter for language models. We present 2 adapter designs (marked in green box): (*1*) insert the adapter into the word embedding layer during pre-training and fine-tuning; (2) insert the adapter into the output layer for fine-tuning only. For the first design, we freeze the word embedding layer and learn the adapter and transformer layers.

scores demonstrate strong OOD detection performance Fort et al. (2021). As documents are distinct from natural image-text pairs, observations in the vision-language domain do not seamlessly translate to the document domain. Moreover, spatial-aware models demonstrate stronger OOD detection performance for both in and out-domain OOD. For example, with the best scoring function (KNN+), LayoutLMv3 improves the average AUROC by 7.09% for out-domain OOD and 7.54% for in-domain OOD data compared to RoBERTa. This further highlights the value of spatial information for improving OOD robustness for documents.

Despite the impressive improvements brought by spatial-aware models, acquiring a large-scale pre-training dataset that includes spatial information remains challenging. In contrast, there is a growing abundance of pre-trained language models that are based on textual data. This

motivates us to explore the possibility of leveraging these pre-trained language models by training an adapter on a small dataset containing document-specific information. By adopting this approach, we can effectively utilize existing models while minimizing the time and cost required for training.

8.5.2 Towards Effective Spatial-Aware Adapter

During our investigation into the effects of model modality, pre-training, and fine-tuning on various types of OOD inputs, we find that spatial/layout information plays a critical role in the document domain. However, existing pre-training models such as LayoutLM series, SelfDoc, and UDoc do not fully leverage the benefits of well-pre-trained language models. This raises the question of whether a large-scale language model, such as RoBERTa, can be adapted to detect OOD documents effectively. In this section, we demonstrate that incorporating an adapter module that accounts for spatial information with transformer-based pre-trained models can achieve strong performance with minimal changes to the code. To the best of our knowledge, this is the first study to apply the adapter idea to documents.

Spatial-aware adapter. Given a pre-trained language model such as RoBERTa, we propose an adapter that utilizes spatial information. We consider two potential designs: 1) the adapter is appended to the word embedding layer, denoted as Spatial-RoBERTa (pre), which requires both pre-training and fine-tuning. This architecture is illustrated in the top row of Fig. 8.8. 2) The adapter is appended to the final layer of the text encoder, denoted as Spatial-BoBERTa (post), which only requires fine-tuning as the model can utilize the pre-trained textual encoder, as shown in the bottom row of Fig. 8.8.

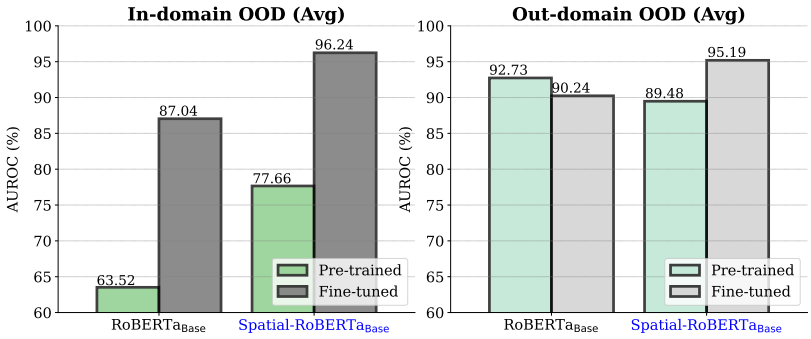


Figure 8.9: Comparison of OOD detection performance of Spatial-RoBERTa and RoBERTa. All models are initialized with public pre-trained checkpoints trained on purely textual data and further pre-trained on IIT-CDIP. The only difference is that Spatial-RoBERTa has an additional spatial-ware adapter and takes word bounding boxes as additional inputs.

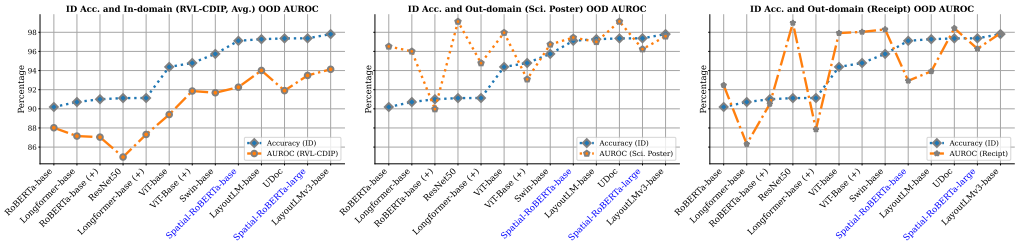


Figure 8.10: Correlation between ID accuracy and OOD detection performance. For most models, ID accuracy is positively correlated with OOD detection performance. Language models with spatial-aware adapters (highlighted in blue) achieve significantly higher ID accuracy and stronger OOD robustness (in AUROC) compared to language models without adapters. Here, (+) represents further pre-training on the IIT-CDIP dataset.

For Spatial-RoBERTa (pre), we freeze the word embedding layer during pre-training for several considerations: 1) word embeddings learned from large-scale corpus already cover most of those words from documents; 2) pre-training on documents without strong language depen-

dency may not help improve word embeddings. For example, in semi-structured documents (*e.g.*, , forms, receipts), language dependencies are not as strong as in text-rich documents (*e.g.*, , letters, resumes), which may degenerate the learned word representations. In practice, each word has a normalized bounding box (x_0, y_0, x_1, y_1), where (x_0, y_0) / (x_1, y_1) corresponds to the position of the upper left / lower right in the bounding box. To encode positional information, we employ four position embedding layers, where each layer= encodes one coordinate (*e.g.*, , x_0) and produces a corresponding position embedding. The special tokens ([CLS], [SEP], and [PAD]) are attached with an empty bounding box $(0, 0, 0, 0)$. As depicted in the top row of Fig. 8.8, the spatial-aware word embeddings are formed by adding position embeddings to their corresponding word embeddings.

For Spatial-RoBERTa (post), position embeddings are added through late fusion in the final hidden states during fine-tuning without affecting the pre-trained encoder. Our experiments demonstrate that introducing spatial-aware adapters during pre-training yields better results than only adding position embeddings during fine-tuning. For additional details[6], please refer to Appendix G.3. In the following, we focus on analyzing Spatial-RoBERTa (pre) and comparing both ID and OOD performance with that of the pure-text pre-trained RoBERTa.

Spatial-RoBERTa significantly outperforms RoBERTa. To verify the effectiveness of Spatial-RoBERTa, we compare the OOD detection performance of pre-trained and fine-tuned models. The results are shown in Fig. 8.9, where OOD performance is based on KNN+ (K=10). Full results can be seen in Table G.6. Spatial-RoBERTa significantly improves the OOD detection performance, especially after fine-tuning. For exam-

[6]Spatial-RoBERTa$_{\text{Base}}$ (pre) incorporates position information during both pre-training and fine-tuning, while Spatial-RoBERTa$_{\text{Base}}$ (post) only inserts the adapter into the output layer for fine-tuning.

ple, compared to RoBERTa (base), Spatial-RoBERTa (base) improves AUROC significantly by 4.24% averaged over four in-domain OOD datasets. This further confirms the importance of spatial information for OOD detection in the document domain.

Spatial-RoBERTa is competitive for both ID classification and OOD detection. Beyond OOD detection performance, we also examine the multi-class ID classification accuracy and plot the two metrics for all models with different modalities in Fig. 8.10. We can clearly observe a positive correlation between ID accuracy and OOD detection performance (measured by AUROC) for both in-domain and out-domain OOD data. Moreover, spatial-aware models display superior ID accuracy and OOD robustness compared to text-only and vision-only models. Overall, Spatial-RoBERTa greatly improves upon RoBERTa and matches the performance of models with more complex and specialized architectures such as LayoutLM. Specifically, Spatial-RoBERTa$_{\text{Large}}$ achieves 97.37 ID accuracy, which is even higher than LayoutLM (97.28) and UDoc (97.36).

To summarize, our spatial-aware adapter effectively adapts pre-trained transformer-based text models to the document domain, improving both ID and OOD performance. In addition, by freezing the original word embeddings during pre-training, the models (Spatial-RoBERTa$_{\text{Base}}$ and Spatial-RoBERTa$_{\text{Large}}$) are parameter-efficient and thus reduce the training cost.

8.6 Conclusions

In this work, we provide a comprehensive and in-depth study on the impacts of pre-training, fine-tuning, model-modality, and OOD scores on a broad variety of document OOD detection tasks. We present novel insights on document OOD detection, which are under-explored or in con-

trast with OOD detection works based on vision-language models. In particular, we highlight that spatial information is critical for OOD detection in documents. We further propose a spatial-aware adapter as an add-on module to transformer-based models. Our module adapts pre-trained language models to the document domain. Extensive experiments on a broad range of datasets verify the effectiveness of our design. We hope our work will inspire future research toward improving OOD robustness for reliable document understanding.

Appendix A

On the Impact of Spurious Correlation for Out-of-Distribution Detection

Publication Statement. This chapter is a joint work with Hang Yin and Yixuan Li. The paper version of this chapter appeared in AAAI 2022 (Ming et al., 2022c).

Modern neural networks can assign high confidence to inputs drawn from outside the training distribution, posing threats to models in real-world deployments. While much research attention has been placed on designing new out-of-distribution (OOD) detection methods, the precise definition of OOD is often left in vagueness and falls short of the desired notion of OOD in reality. In this chapter, we present a new formalization and model the data shifts by taking into account both the invariant and environmental (spurious) features. Under such formalization, we systematically investigate how spurious correlation in the training set impacts OOD detection. Our results suggest that the detection performance is severely worsened when the correlation between spurious features and

labels is increased in the training set. We further show insights on detection methods that are more effective in reducing the impact of spurious correlation, and provide theoretical analysis on why reliance on environmental features leads to high OOD detection error. Our work aims to facilitate a better understanding of OOD samples and their formalization, as well as the exploration of methods that enhance OOD detection. Code is available at https://github.com/deeplearning-wisc/Spurious_OOD.

A.1 Introduction

Modern deep neural networks have achieved unprecedented success in known contexts for which they are trained, yet they do not necessarily know what they don't know (Nguyen et al., 2015). In particular, neural networks have been shown to produce high posterior probability for test inputs from out-of-distribution (OOD), which should not be predicted by the model. This gives rise to the importance of OOD detection, which aims to identify and handle unknown OOD inputs so that the algorithm can take safety precautions.

Before we attempt any solution, an important yet often overlooked problem is: what do we mean by out-of-distribution data? While the research community lacks a consensus on the precise definition, a common evaluation protocol views data with non-overlapping semantics as OOD inputs (Hendrycks and Gimpel, 2017). For example, an image of a `cow` can be viewed as an OOD *w.r.t* a model tasked to classify `cat` vs. `dog`. However, such an evaluation scheme is often oversimplified and may not capture the nuances and complexity of the problem in reality.

We begin with a motivating example where a neural network can rely on statistically informative yet *spurious* features in the data. Indeed, many prior works showed that modern neural networks can spuriously rely on the biased features (e.g., background or textures) instead of features of

the object to achieve high accuracy (Beery et al., 2018; Geirhos et al., 2019; Sagawa et al., 2019). In Figure A.1, we illustrate a model that exploits the spurious correlation between the `water background` and label `waterbird` for prediction. Consequently, a model that relies on spurious features can produce a high-confidence prediction for an OOD input with the same background (*i.e.*, water) but a different semantic label (*e.g.*, boat). This can manifest in downstream OOD detection, yet unexplored in prior works. In this paper, we systematically investigate how spurious correlation in the training set impacts OOD detection. We first provide a new formalization and explicitly model the data shifts by taking into account both **invariant** features and **environmental** features (Section A.2). Invariant features can be viewed as essential cues directly related to semantic labels, whereas environmental features are non-invariant and can be spurious. Our formalization encapsulates two types of OOD data: (1) *spurious OOD*—test samples that contain environmental (non-invariant) features but no invariant features; (2) *non-spurious OOD*—inputs that contain neither the environmental nor invariant features, which is more in line with the conventional notion of OOD. We provide an illustration of both types of OOD in Figure A.1.

Under the new formalization, we conduct extensive experiments and investigate the detection performance under both spurious and non-spurious OOD inputs (Section A.3). Our results suggest that spurious correlation in the training data poses a significant challenge to OOD detection. For both spurious and non-spurious OOD samples, the detection performance is severely worsened when the correlation between spurious features and labels is increased in the training set. Further, we comprehensively evaluate common OOD detection approaches, and show that feature-based methods have a competitive edge in improving non-spurious OOD detection, while detecting spurious OOD remains challenging (Section A.4). To further understand this, we provide theoretical insights on

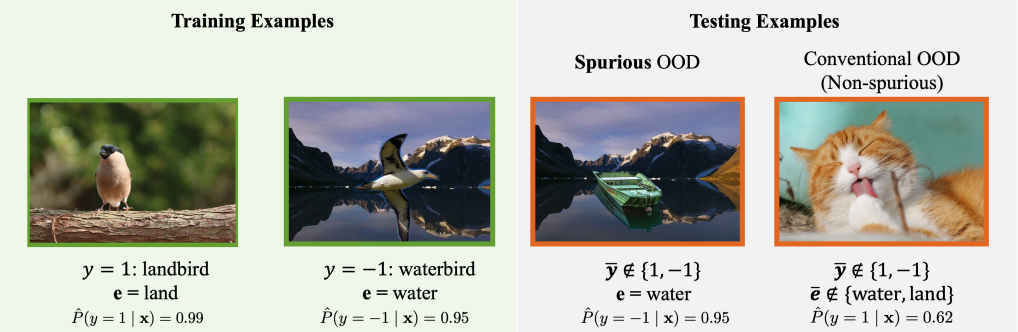


Figure A.1: **Left** (train): The training examples $\mathbf{x}$ are generated by a combination of invariant features, dependent on the label y; and environmental features, dependent on the environment e. In Waterbirds dataset (Sagawa et al., 2019), $y \in \{\texttt{waterbird}, \texttt{landbird}\}$ is correlated with the environment $e \in \{\text{water}, \text{land}\}$. **Right** (test): During test time, we consider two types of OOD inputs. Spurious OOD inputs contain the environmental features, but no signals related to the in-distribution classes. Non-spurious OOD inputs have neither environmental features nor invariant features. Confidence scores are computed from a ResNet-18 model trained on Waterbirds (Sagawa et al., 2019).

why reliance on non-invariant features leads to high OOD detection error (Section A.5). We provably show the existence of spurious OOD inputs with arbitrarily high confidence, which can fail to be distinguished from the ID data. Our **key contributions** are as follows:

- We provide a new formalization of OOD detection by explicitly taking into account the separation between invariant features and environmental features. Our formalization encapsulates both spurious and non-spurious OOD. Our work, therefore, provides a complementary perspective in the evaluation of OOD detection.

- We provide systematic investigations on how the extent of spurious correlation in the training set impacts OOD detection. We further show insights on OOD detection solutions that are more effective in mitigating the impact of spurious correlation, with up to 46.73% reduction of FPR95 in detecting non-spurious OOD data.

- We provide theoretical analysis, provably showing that detecting spurious OOD samples remains challenging due to the model's reliance on the environmental features.

Our study provides strong implications for future research on out-of-distribution detection. Our study signifies the importance for future works to evaluate OOD detection algorithms on spurious OOD examples besides standard benchmarks (most of which are non-spurious) to test the limits of the approaches. We hope that our work will inspire future research on the formalization of the OOD detection problem and algorithmic solutions.

A.2 A New Formalization of Out-of-distribution Data

Data Model. We consider a supervised multi-class classification problem, where $\mathcal{X} = \mathbb{R}^d$ denotes the input space and $\mathcal{Y} = \{1, 2, ..., K\}$ denotes the label space. We assume that the data is drawn from a set of E environments (domains) $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$. The inputs $\mathbf{x}$ is generated by a combination of invariant features $\mathbf{z}_{\text{inv}} \in \mathbb{R}^s$, dependent on the label y; and environmental feature $\mathbf{z}_e \in \mathbb{R}^{d_e}$, dependent on the environment e:

$$\mathbf{x} = \tau(\mathbf{z}_{\text{inv}}, \mathbf{z}_e),$$

where τ is a function transformation from the latent features $[\mathbf{z}_{\text{inv}}, \mathbf{z}_e]^\top$ to the pixel-space $\mathcal{X}$. The signal $\mathbf{z}_{\text{inv}}$ are the cues essential for the recognition of $\mathbf{x}$ as y; examples include the color, the shape of beaks and claws, and fur patterns of birds for classifying `waterbird` vs. `landbird`. Environmental features $\mathbf{z}_e$, on the other hand, are cues not essential for the recognition but correlated with the target y. For example, many waterbird images are taken in water habitat, so water scenes can be considered as $\mathbf{z}_e$. Under the

OOD Type	Test Set	r=0.5		r=0.7		r=0.9	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC↑	FPR95 ↓	AUROC ↑
Spurious OOD		59.89 ± 12.40	88.54 ± 4.81	74.22 ± 13.12	80.98 ± 4.45	74.39 ± 12.50	79.81 ± 8.43
Non-spurious OOD	iSUN	19.69 ± 10.66	91.88 ± 4.52	43.22 ± 12.50	91.81 ± 3.32	57.40 ± 15.54	82.45 ± 7.98
	LSUN	22.60 ± 12.08	90.80 ± 3.33	43.30 ± 16.66	90.09 ± 4.51	52.68 ± 13.70	84.56 ± 8.56
	SVHN	15.32 ± 5.05	95.71 ± 2.20	25.53 ± 8.11	95.60 ± 2.45	43.89 ± 23.80	93.27 ± 6.90

Table A.1: OOD detection performance of models trained on **Waterbirds** (Sagawa et al., 2019). Increased spurious correlation in the training set results in worsen performance for both non-spurious and spurious OOD samples. In particular, spurious OOD is more challenging than non-spurious OOD samples. Results (mean and std) are estimated over 4 runs for each setting.

data model, we have a joint distribution $P(\mathbf{x}, y, e)$. Each $g = (y, e) \in \mathcal{Y} \times \mathcal{E}$ group has its own distribution over features $[\mathbf{z}_{\text{inv}}, \mathbf{z}_e] \in \mathbb{R}^{s+d_e}$. Furthermore, let $\mathcal{D}_{\text{in}}^e$ denote the marginal distribution on $\mathcal{X}$ for environment e. The union of distributions $\mathcal{D}_{\text{in}}^e$ over all environments is the in-distribution $\mathcal{D}_{\text{in}}$.

Out-of-distribution Data. In practice, OOD refers to samples from an irrelevant distribution whose label set has no intersection with $\mathcal{Y}$, and therefore should not be predicted by the model. Under our data model, we define data distributional shifts by explicitly taking into account the separation between invariant features and environmental features. Concretely, our formalization encapsulates two types of OOD data defined below.

- **Spurious OOD** is a particularly challenging type of inputs, which contain the *environmental feature, but no invariant feature essential for the label*. Formally, we denote by $\mathbf{x} = \tau(\mathbf{z}_{\bar{Y}}, \mathbf{z}_e)$, where $\mathbf{z}_{\bar{Y}}$ is from an out-of-class label $\bar{Y} \notin \mathcal{Y}$. For example, this can be seen in Figure A.1 (middle right), where the OOD example contains the semantic feature $\texttt{boat} \notin \{\texttt{waterbird}, \texttt{landbird}\}$, yet it has the environmental feature of water background.

- **Non-spurious (conventional) OOD** are inputs that contain *neither the environmental nor the invariant features*, i.e., $\mathbf{x} = \tau(\mathbf{z}_{\bar{Y}}, \mathbf{z}_{\bar{e}})$. In particular, $\mathbf{z}_{\bar{Y}}$ is sampled from an out-of-class label $\bar{Y} \notin \mathcal{Y}$, and $\mathbf{z}_{\bar{e}}$ is sampled from a different environment $\bar{e} \notin \mathcal{E}$. For example, an input of an `indoor cat` falls into this category, where both the semantic label `cat` and environment `indoor` are distinct from the in-distribution data of waterbirds and landbirds.

Out-of-distribution Detection. OOD detection can be viewed as a binary classification problem. Let $f : \mathcal{X} \to \mathbb{R}^K$ be a neural network trained on samples drawn from the data distribution defined above. During inference time, OOD detection can be performed by exercising a thresholding mechanism:

$$G_\lambda(\mathbf{x}; f) = \begin{cases} \text{in} & S(\mathbf{x}; f) \geq \lambda \\ \text{out} & S(\mathbf{x}; f) < \lambda \end{cases}, \tag{A.1}$$

where samples with higher scores $S(\mathbf{x}; f)$ are classified as ID and vice versa. The threshold λ is typically chosen so that a high fraction of ID data (*e.g.*, 95%) is correctly classified.

A.3 How does spurious correlation impact OOD detection?

During training, a classifier may learn to rely on the association between environmental features and labels to make its predictions. Moreover, we hypothesize that such a reliance on environmental features can cause failures in the downstream OOD detection. To verify this, we begin with the most common training objective empirical risk minimization (ERM). Given a loss function ℓ, ERM finds the model w that minimizes the aver-

age training loss:

$$\hat{\mathcal{R}}(w) = \mathbb{E}_{(\mathbf{x},y,e)\sim\hat{P}}[\ell(w;(\mathbf{x},y,e))]. \tag{A.2}$$

We now describe the datasets we use for model training and OOD detection tasks. We consider three tasks that are commonly used in the literature. We start with a natural image dataset Waterbirds, and then move onto the CelebA dataset (Liu et al., 2015). Due to space constraints, a third evaluation task on ColorMNIST is in the Supplementary.

Evaluation Task 1: Waterbirds. Introduced in (Sagawa et al., 2019), this dataset is used to explore the spurious correlation between the image background and bird types, specifically $\mathcal{E} \in \{\texttt{water}, \texttt{land}\}$ and $\mathcal{Y} \in \{\texttt{waterbirds}, \texttt{landbirds}\}$. We also control the correlation between y and e during training as $r \in \{0.5, 0.7, 0.9\}$. The correlation r is defined as $r = P(e = \texttt{water} \mid y = \texttt{waterbirds}) = P(e = \texttt{land} \mid y = \texttt{landbirds})$. For spurious OOD, we adopt a subset of images of land and water from the Places dataset (Zhou et al., 2017). For non-spurious OOD, we follow the common practice and use the `SVHN` (Netzer et al., 2011), `LSUN` (Yu et al., 2015), and `iSUN` (Xu et al., 2015) datasets.

Evaluation Task 2: CelebA. In order to further validate our findings beyond background spurious (environmental) features, we also evaluate on the CelebA (Liu et al., 2015) dataset. The classifier is trained to differentiate the hair color (grey vs. non-grey) with $\mathcal{Y} = \{\texttt{grey hair}, \texttt{nongrey hair}\}$. The environments $\mathcal{E} = \{\texttt{male}, \texttt{female}\}$ denote the gender of the person. In the training set, "Grey hair" is highly correlated with "Male", where 82.9% ($r \approx 0.8$) images with grey hair are male. Spurious OOD inputs consist of `bald male`, which contain environmental features (gender) without invariant features (hair). The non-spurious OOD test suite is the same as above (`SVHN`, `LSUN`, and `iSUN`). Figure A.2 illustates ID samples, spuri-

Figure A.2: For CelebA, the classifier is trained to differentiate the hair color (grey vs. non-grey). **Left**: Training environments. 82.9% images with grey hair are male, whereas 82.9% images with non-grey hair are female. **Middle**: Spurious OOD inputs contain the environmental feature (male) without invariant features (hair). **Right**: Non-spurious OOD samples consist of images with diverse semantics without human faces.

ous and non-spurious OOD test sets. We also subsample the dataset to ablate the effect of r; see results are in the Supplementary.

Results and Insights. We train on ResNet-18 (He et al., 2016) for both tasks. See Appendix for details on hyperparameters and in-distribution performance. We summarize the OOD detection performance in Table A.1 (Waterbirds), Table A.2 (CelebA) and Table A.4 (ColorMNIST).

There are several salient observations. **First**, for both spurious and non-spurious OOD samples, the detection performance is severely worsened when the correlation between spurious features and labels is increased in the training set. Take the Waterbirds task as an example, under correlation $r = 0.5$, the average false positive rate (FPR95) for spurious OOD samples is 59.89%, and increases to 74.39% when $r = 0.9$. Similar trends also hold for other datasets. **Second**, spurious OOD is much more challenging to be detected compared to non-spurious OOD. From TableA.1, under correlation $r = 0.7$, the average FPR95 is 37.35% for non-spurious OOD, and increases to 74.22% for spurious OOD. Similar observations hold under different correlation and different training datasets. **Third**, for non-spurious OOD, samples that are more semantically dissimilar to ID are easier to detect. Take Waterbirds as an example, images

containing scenes (e.g. LSUN and iSUN) are more similar to the training samples compared to images of numbers (e.g. SVHN), resulting in higher FPR95 (e.g. 43.22% for iSUN compared to 25.53% for SVHN under $r = 0.7$).

OOD Type	Test Set	FPR95 ↓	AUROC ↑
Spurious OOD		71.28 ± 4.12	82.04 ± 2.64
Non-spurious OOD	iSUN	17.35 ± 2.97	97.03 ± 0.30
	LSUN	18.85 ± 2.44	96.90 ± 0.17
	SVHN	5.63 ± 2.60	98.64 ± 0.21

Table A.2: OOD detection performance of models trained on **CelebA** (Liu et al., 2015) with $r \approx 0.8$. Spurious OOD test data incurs much higher FPR than non-spurious OOD data. Results (mean and std) are estimated over 4 runs for each setting.

Our results suggest that spurious correlation poses a significant threat to the model. In particular, a model can produce high-confidence predictions on the spurious OOD, due to the reliance on the environmental feature (*e.g.*, background information) rather than the invariant feature (*e.g.*, bird species). To verify that the spurious feature causes poor detection performance, we show that the classifier frequently predicts the spurious OOD as the ID class with the same environmental feature. For Waterbirds, on average 93.9% of OOD samples with water background is classified as waterbirds, and 80.7% of OOD samples with land background is classified as land birds. For the CelebA dataset, on average 86.5% of spurious OOD samples (bold male) are classified as grey hair. Note that our results here are based on the energy score (Liu et al., 2020), which is one competitive detection method derived from the model output (logits) and has shown superior OOD detection performance over directly using the predictive confidence score. Next, we provide an expansive evaluation using a broader suite of OOD scoring functions in Section A.4.

Scoring Func	MSP (Hendrycks and Gimpel, 2017)				ODIN (Liang et al., 2018)				Mahalanobis (Lee et al., 2018)				Energy (Liu et al., 2020)				Gram (Sastry and Oore, 2020)			
Metric	FPR95↓		AUROC↑		FPR95↓		AUROC↑		FPR95↓		AUROC↑		FPR95↓		AUROC↑		FPR95↓		AUROC↑	
In-distribution Data	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP	SP	NSP
ColorMNIST	42.99	3.15	77.75	99.13	38.06	1.88	78.78	99.01	14.97	0.04	88.65	99.54	30.45	7.65	86.74	97.54	4.33	0.05	96.89	99.40
Waterbirds	74.68	47.53	79.22	92.34	77.25	34.06	81.04	93.48	69.35	0.80	82.73	99.51	74.22	37.35	80.98	92.50	58.25	0.65	87.33	99.71
CelebA	83.70	22.60	68.22	90.21	81.07	11.49	75.22	89.11	78.75	2.33	83.12	98.93	71.28	13.94	82.04	97.51	81.21	3.11	68.58	98.96

Table A.3: Performance for different post-hoc OOD detection methods when the spurious correlation is high in the training set. We choose $r = 0.45$ for ColorMNIST, $r = 0.7$ for Waterbirds, and $r = 0.8$ for CelebA. SP stands for Spurious OOD test set. NSP denotes non-spurious OOD, where the results are averaged over 3 OOD test sets (see details in Section A.3).

A.4 How to reduce the impact of spurious correlation for OOD detection?

The results in the previous section naturally prompt the question: how can we better detect spurious and non-spurious OOD inputs when the training dataset contains spurious correlation? In this section, we comprehensively evaluate common OOD detection approaches, and show that feature-based methods have a competitive edge in improving non-spurious OOD detection, while detecting spurious OOD remains challenging (which we further explain theoretically in Section A.5).

Feature-based vs. Output-based OOD Detection. Section A.3 suggests that OOD detection becomes challenging for output-based methods especially when the training set contains high spurious correlation. However, the efficacy of using representation space for OOD detection remains unknown. In this section, we consider a suite of common scoring functions including maximum softmax probability (MSP) (Hendrycks and Gimpel, 2017), ODIN score (Liang et al., 2018; Hsu et al., 2020), Mahalanobis distance-based score (Lee et al., 2018), energy score (Liu et al., 2020), and Gram matrix-based score (Sastry and Oore, 2020)—all of which can be derived *post hoc*[1] from a trained model. Among those, Mahalanobis and

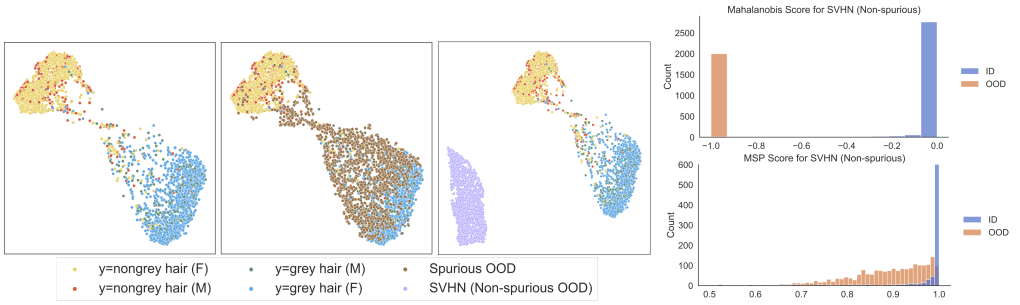
[1]Note that Generalized-ODIN requires modifying the training objective and model retraining. For fairness, we primarily consider strict post-hoc methods based on the

Gram Matrices can be viewed as feature-based methods. For example, Lee et al. (2018) estimates class-conditional Gaussian distributions in the representation space and then uses the maximum Mahalanobis distance as the OOD scoring function. Data points that are sufficiently far away from all the class centroids are more likely to be OOD.

Results. The performance comparison is shown in Table A.3. Several interesting observations can be drawn. **First**, we can observe a significant performance gap between *spurious OOD* (SP) and *non-spurious OOD* (NSP), irrespective of the OOD scoring function in use. This observation is in line with our findings in Section A.3. **Second**, the OOD detection performance is generally improved with the feature-based scoring functions such as Mahalanobis distance score (Lee et al., 2018) and Gram Matrix score (Sastry and Oore, 2020), compared to scoring functions based on the output space (*e.g.*, MSP, ODIN, and energy). The improvement is substantial for non-spurious OOD data. For example, on Waterbirds, FPR95 is reduced by 46.73% with Mahalanobis score compared to using MSP score. For spurious OOD data, the performance improvement is most pronounced using the Mahalanobis score. Noticeably, using the Mahalanobis score, the FPR95 is reduced by 28.02% on the ColorMNIST dataset, compared to using the MSP score. Our results suggest that feature space preserves useful information that can more effectively distinguish between ID and OOD data.

Analysis and Visualizations. To provide further insights on why the feature-based method is more desirable, we show the visualization of embeddings in Figure A.3a. The visualization is based on the CelebA task. From Figure A.3a (left), we observe a clear separation between the two class labels. Within each class label, data points from both environments

standard cross-entropy loss.



(a) Visualization of Feature Embedding

(b) Histograms of Different Scores

Figure A.3: (a) **Left**: Feature for in-distribution data only. (a) **Middle**: Feature for both ID and spurious OOD data. (a) **Right**: Feature for ID and non-spurious OOD data (SVHN). M and F in parentheses stand for male and female respectively. (b) Histogram of Mahalanobis score and MSP score for ID and SVHN (Non-spurious OOD). Full results for other non-spurious OOD datasets (iSUN and LSUN) are in the Supplementary.

are well mixed (e.g., see the green and blue dots). In Figure A.3a (middle), we visualize the embedding of ID data together with spurious OOD inputs, which contain the environmental feature (`male`). Spurious OOD (bold male) lies between the two ID clusters, with some portion overlapping with the ID samples, signifying the hardness of this type of OOD. This is in stark contrast with non-spurious OOD inputs shown in Figure A.3a (right), where a clear separation between ID and OOD (purple) can be observed. This shows that feature space contains useful information that can be leveraged for OOD detection, especially for conventional non-spurious OOD inputs. Moreover, by comparing the histogram of Mahalanobis distance (top) and MSP score (bottom) in Figure A.3b, we can further verify that ID and OOD data is much more separable with the Mahalanobis distance. Therefore, our results suggest that feature-based methods show promise for improving non-spurious OOD detection when the training set contains spurious correlation, while there still exists large room for improvement on spurious OOD detection.

A.5 Why is it hard to detect spurious OOD?

Given the results above, a natural question arises: why is it hard to detect spurious OOD inputs? To better understand this issue, we now provide theoretical insights. In what follows, we first model the ID and OOD data distributions and then derive mathematically the model output of invariant classifier, where the model aims not to rely on the environmental features for prediction.

Setup. We consider a binary classification task where $y \in \{-1, 1\}$, and is drawn according to a fixed probability $\eta := P(y = 1)$. We assume both the invariant features $\mathbf{z}_{\text{inv}}$ and environmental features $\mathbf{z}_e$ are drawn from Gaussian distributions:

$$\mathbf{z}_{\text{inv}} \sim \mathcal{N}\left(y \cdot \boldsymbol{\mu}_{\text{inv}}, \sigma_{\text{inv}}^2 I\right), \quad \mathbf{z}_e \sim \mathcal{N}\left(y \cdot \boldsymbol{\mu}_e, \sigma_e^2 I\right)$$

where $\boldsymbol{\mu}_e \in \mathbb{R}^{d_e}$, $\boldsymbol{\mu}_{\text{inv}} \in \mathbb{R}^s$, and I is the identity matrix. Note that the parameters $\boldsymbol{\mu}_{\text{inv}}$ and σ_{inv}^2 are the same for all environments. In contrast, the environmental parameters $\boldsymbol{\mu}_{\text{e}}$ and σ_e^2 are different across e, where the subscript is used to indicate the dependence on the environment and the index of the environment. In what follows, we present the results, with detailed proof deferred in the Appendix.

Lemma A.1. (*Bayes optimal classifier*) *For any feature vector which is a linear combination of the invariant and environmental features* $\Phi_e(\mathbf{x}) = M_{inv}\mathbf{z}_{inv} + M_e\mathbf{z}_e$*, the optimal linear classifier for an environment* e *has the corresponding coefficient* $2\Sigma_\Phi^{-1}\boldsymbol{\mu}_\Phi$*, where:*

$$\boldsymbol{\mu}_\Phi = M_{inv}\boldsymbol{\mu}_{inv} + M_e\boldsymbol{\mu}_e$$
$$\Sigma_\Phi = M_{inv}M_{inv}^T\sigma_{inv}^2 + M_eM_e^T\sigma_e^2$$

Note that the Bayes optimal classifier uses environmental features which are informative of the label but non-invariant. Rather, we hope to rely *only* on invariant features while ignoring environmental features. Such a predictor is also referred to as *optimal invariant predictor* (Rosenfeld et al., 2021), which is specified in the following. Note that this is a special case of Lemma A.1 with $M_{\text{inv}} = I$ and $M_e = 0$.

Proposition A.2. (*Optimal invariant classifier using invariant features*) *Assume the featurizer recovers the invariant feature* $\Phi_e(\mathbf{x}) = [\mathbf{z}_{inv}]\ \forall e \in \mathcal{E}$, *the optimal invariant classifier has the corresponding coefficient* $2\boldsymbol{\mu}_{inv}/\sigma^2_{inv}$.[2]

The optimal invariant classifier explicitly ignores the environmental features. However, an invariant classifier learned does not necessarily depend only on the invariant features. Next Lemma shows that *it can be possible to learn an invariant classifier that relies on the environmental features while achieving lower risk than the optimal invariant classifier*.

Lemma A.3. (*Invariant classifier using non-invariant features*) *Suppose* $E \leq d_e$, *given a set of environments* $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ *such that all environmental means are linearly independent. Then there always exists a unit-norm vector* $\mathbf{p}$ *and positive fixed scalar* β *such that* $\beta = \mathbf{p}^T\boldsymbol{\mu}_e/\sigma^2_e\ \forall e \in \mathcal{E}$. *The resulting optimal classifier weights are*

$$\hat{w} = \begin{bmatrix} \beta_{inv} \\ 2\beta \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{\mu}_{inv}/\sigma^2_{inv} \\ 2\mathbf{p}^\top\boldsymbol{\mu}_e/\sigma^2_e \end{bmatrix}.$$

Note that the optimal classifier weight 2β is a constant, which does not depend on the environment (and neither does the optimal coefficient for $\mathbf{z}_{\text{inv}}$). The projection vector $\mathbf{p}$ acts as a "short-cut" that the learner can use to yield an insidious surrogate signal $\mathbf{p}^\top\mathbf{z}_e$. Similar to $\mathbf{z}_{\text{inv}}$, this insidious

[2]The constant term in the classifier weights is $\log \eta/(1-\eta)$, which we omit here and in the sequel.

signal can also lead to an invariant predictor (across environments) admissible by invariant learning methods. In other words, despite the varying data distribution across environments, the optimal classifier (using non-invariant features) is the same for each environment. We now show our main results, where OOD detection can fail under such an invariant classifier.

Theorem A.4. (*Failure of OOD detection under invariant classifier*) *Consider an out-of-distribution input which contains the environmental feature:* $\Phi_{out}(\mathbf{x}) = M_{inv}\mathbf{z}_{out} + M_e\mathbf{z}_e$*, where* $\mathbf{z}_{out} \perp \boldsymbol{\mu}_{inv}$*. Given the invariant classifier* (*cf. Lemma* 2)*, the posterior probability for the OOD input is*

$$p(y = 1 \mid \Phi_{out}) = \sigma\left(2\mathbf{p}^\top \mathbf{z}_e \beta + \log \eta/(1-\eta)\right)$$

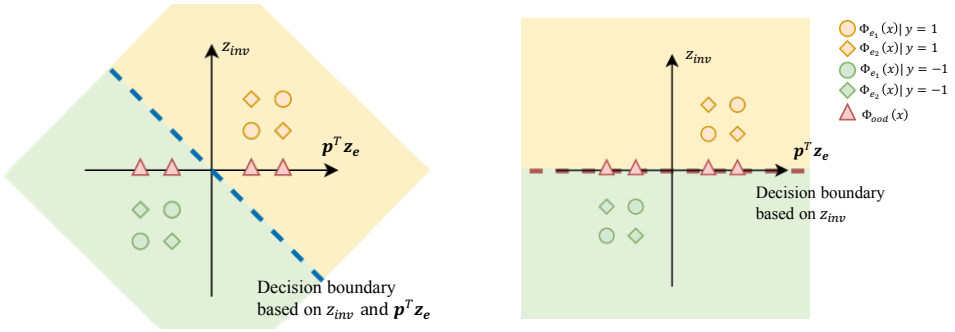
, where σ *is the logistic function. Thus for arbitrary confidence* $0 < c := P(y = 1 \mid \Phi_{out}) < 1$*, there exists* $\Phi_{out}(\mathbf{x})$ *with* $\mathbf{z}_e$ *such that*

$$\mathbf{p}^\top \mathbf{z}_e = \frac{1}{2\beta} \log \frac{c(1-\eta)}{\eta(1-c)}$$

.

Our theorem above signifies the existence of OOD inputs that can trigger high-confidence predictions on in-distribution classes yet contain no meaningful feature related to the labels in $\mathcal{Y} = \{1, -1\}$ at all. An OOD detector can fail to detect these inputs with predictions that are indistinguishable from ID data. We provide a simple toy example to explain this phenomenon further.

An Intuitive Example. An illustrative example with two environments is provided in Figure A.4a. The feature representations for examples in environments 1 and 2 are shown as circle and diamond, respectively. In-distribution samples with different colors correspond to different labels:



(a) Feature based on $\mathbf{z}_{\text{inv}}$ and $\mathbf{p}^\top \mathbf{z}_e$

(b) Feature based on $\mathbf{z}_{\text{inv}}$

Figure A.4: The ID data is comprised of two classes $y = 1$ (yellow) and $y = -1$ (green). Two environments are shown as circle and diamond, respectively. (a) The invariant decision boundary (blue dashed line) is based on both the invariant feature z_{inv} and environmental features z_e. Illustration of the existence of OOD inputs (red triangles) that can be predicted as in-distribution with high confidence, therefore can fail to be detected by OOD methods (e.g., using predictive confidence threshold). (b) An ideal case when the invariant decision boundary is purely based on z_{inv} (red dashed line). The OOD inputs lie on the decision boundary and will be predicted as $y = 1$ or $y = -1$ with a probability 0.5.

yellow indicates $y = 1$ and green indicates $y = -1$. The decision boundary of classification is denoted by the dashed line, which relies on both the invariant features $\mathbf{z}_{\text{inv}}$ and environmental features $\mathbf{z}_e$. It can be seen that if the feature representation relies on environmental features $\mathbf{p}^\top \mathbf{z}_e$, spurious OOD samples (red triangles) can trick the classifier into recognizing OOD samples as one of the in-distribution classes with high confidence, posing severe threats to OOD detection.

In contrast, under an ideal case when the invariant classifier only uses invariant features $\mathbf{z}_{\text{inv}}$, the optimal decision boundary is a horizontal dashed line (see Figure A.4b). OOD inputs (red triangles) will be predicted with a probability of 0.5 since they lie on the decision boundary.

Remark. As a special case, if the representation consists purely of environmental features, i.e., $\Phi_e(\mathbf{x}) = [\mathbf{z}_e]$, the resulting optimal classifier

weights are $2\mathbf{p}^\top \mu_e/\sigma_e^2 = 2\beta$, a fixed scalar that is still invariant across environments. Lemma A.5 below shows that such a predictor can yield low risks under certain conditions. Our main theorem above still holds under such a predictor.

Lemma A.5. (*Existence of purely environmental predictors with low risks* (*Rosenfeld et al., 2021*)) *There exists a representation constructed purely relying on environmental features based on the short-cut direction* $\mathbf{p}$ *that achieves lower risk than the optimal invariant predictor on every environment* e *such that* $\sigma_e\beta > \sigma_{inv}^{-1} \|\mu_{inv}\|_2$ *and* $2\sigma_e\beta\sigma_{inv}^{-1} \|\mu_{inv}\|_2 \geq |\log \eta/(1-\eta)|$.

Summary. To summarize, the theoretical analysis demonstrates the difficulty of recovering the invariant classifier without using environmental features. In particular, there exists an invariant classifier that uses non-invariant features, and achieves lower risks than the classifiers only based on invariant features. As a result, spurious OOD samples can utilize environmental clues to deteriorate the OOD detection performance. Our main theorem provably shows the existence of OOD inputs with arbitrarily high confidence, and can fail to be distinguished from the ID data.

Extension: Empirical Validation of Theoretical Analysis. To further validate our analysis above, we comprehensively evaluate the OOD detection performance of models that are trained with recent prominent domain invariance learning objectives (Arjovsky et al., 2019; Bahng et al., 2020; Krueger et al., 2021; Ganin et al., 2016; Li et al., 2018c; Sagawa et al., 2019) (Section A.12 in Appendix). The results align with our theoretical analysis.

A.6 Discussion and related works

Out-of-distribution Uncertainty Estimation. The phenomenon of neural networks' overconfidence to out-of-distribution data is revealed by Nguyen *et al.* Nguyen et al. (2015). Early works attempt to improve the OOD uncertainty estimation by proposing the ODIN score (Liang et al., 2018) and Mahalanobis distance-based confidence score (Lee et al., 2018). Recent work by Liu *et al.* (Liu et al., 2020) proposed using an energy score for OOD detection, which demonstrated advantages over the softmax confidence score both empirically and theoretically. Huang and Li Huang and Li (2021) proposed a group-based OOD detection method that scales effectively to large-scale dataset ImageNet. Recent work by Lin *et al.* Lin et al. (2021) also proposed dynamic OOD inference framework that improved the computational efficiency of OOD detection. However, previous methods primarily focused on convention non-spurious OOD. We introduce a new formalization of OOD detection that encapsulates both spurious and non-spurious OOD data. A parallel line of approaches resorts to generative models Goodfellow et al. (2014); Kingma and Dhariwal (2018) that directly estimate in-distribution density (Nalisnick et al., 2019; Ren et al., 2019; Serrà et al., 2020; Xiao et al., 2020; Kirichenko et al., 2020). In particular, Ren et al. (2019) addressed distinguishing between background and semantic content under unsupervised generative models. Generative approaches yield limiting performance compared with supervised discriminative models due to the lack of label information and typically suffer from high computational complexity. Notably, none of the previous works systematically investigate the influence of spurious correlation for OOD detection. Our work presents a novel perspective for defining OOD data and investigates the impact of spurious correlation in the training set. Moreover, our formulation is more general and broader than the image background (for example, gender bias in our CelebA experiments is another type of contextual bias beyond image background).

Near-ID Evaluations. Our proposed spurious OOD can be viewed as a form of near-ID evaluation. Orthogonal to our work, previous works (Winkens et al., 2020; Roy et al., 2022) considered the near-ID cases where the *semantics* of OOD inputs are similar to that of ID data (*e.g.*, CIFAR-10 vs. CIFAR-100). In our setting, spurious OOD inputs may have very different semantic labels but are statistically close to the ID data due to shared environmental features (*e.g.*, boat vs. waterbird in Figure 1). While other works have considered domain shift (Hsu et al., 2020) or covariate shift (Ovadia et al., 2019), they are more relevant for evaluating model generalization and robustness performance—in which case the goal is to make the model classify accurately into the ID classes and should not be confused with OOD detection task. We emphasize that semantic label shift (i.e., change of invariant feature) is more akin to OOD detection task, which concerns model reliability and detection of shifts where the inputs have disjoint labels from ID data and therefore should not be predicted by the model.

Out-of-distribution Generalization. Recently, various works have been proposed to tackle the issue of domain generalization, which aims to achieve high classification accuracy on new test environments consisting of inputs *with invariant features*, and does not consider the change of invariant features at test time (i.e., label space $\mathcal{Y}$ remains the same)—a key difference from our focus. Literature in OOD detection is commonly concerned about model reliability and detection of shifts where the OOD inputs have disjoint labels and therefore should not be predicted by the model. In other words, we consider samples *without invariant features*, regardless of the presence of environmental features or not.

A plethora of algorithms are proposed: learning invariant representation across domains (Ganin et al., 2016; Li et al., 2018c; Sun and Saenko, 2016; Li et al., 2018b), minimizing the weighted combination of risks from

training domains (Sagawa et al., 2019), using different risk penalty terms to facilitate invariance prediction (Arjovsky et al., 2019; Krueger et al., 2021), causal inference approaches (Peters et al., 2016), and forcing the learned representation different from a set of pre-defined biased representations (Bahng et al., 2020), mixup-based approaches (Zhang et al., 2018; Wang et al., 2020; Luo et al., 2020), etc. A recent study (Gulrajani and Lopez-Paz, 2021) shows that no domain generalization methods achieve superior performance than ERM across a broad range of datasets.

Contextual Bias in Recognition. There has been a rich literature studying the classification performance in the presence of contextual bias (Torralba, 2003; Beery et al., 2018; Barbu et al., 2019). The reliance on contextual bias such as image backgrounds, texture, and color for object detection are investigated in (Zhu et al., 2017; Baker et al., 2018; Geirhos et al., 2019; Zech et al., 2018; Xiao et al., 2021; Sagawa et al., 2019). However, the contextual bias for OOD detection is underexplored. In contrast, our study systematically investigates the impact of spurious correlation on OOD detection and how to mitigate it.

A.7 Conclusion

Out-of-distribution detection is an essential task in open-world machine learning. However, the precise definition is often left in vagueness, and common evaluation schemes can be too primitive to capture the nuances of the problem in reality. In this paper, we present a new formalization where we model the data distributional shifts by considering the invariant and non-invariant features. Under such formalization, we systematically investigate the impact of spurious correlation in the training set on OOD detection and further show insights on detection methods that are more effective in mitigating the impact of spurious correlation. Moreover, we

provide theoretical analysis on why reliance on environmental features leads to high OOD detection error. We hope that our work will inspire future research on the understanding and formalization of OOD samples, new evaluation schemes of OOD detection methods, and algorithmic solutions in the presence of spurious correlation.

A.8 Proofs for Theoretical Results

Lemma A.6. (*Bayes optimal classifier*) *For any feature vector which is a linear combination of the invariant and environmental features* $\Phi_e(\mathbf{x}) = M_{inv}\mathbf{z}_{inv} + M_e\mathbf{z}_e$*, the optimal linear classifier for an environment* e *has the corresponding coefficient* $2\Sigma_\Phi^{-1}\boldsymbol{\mu}_\Phi$*, where:*

$$\boldsymbol{\mu}_\Phi = M_{inv}\boldsymbol{\mu}_{inv} + M_e\boldsymbol{\mu}_e$$
$$\Sigma_\Phi = M_{inv}M_{inv}^T\sigma_{inv}^2 + M_eM_e^T\sigma_e^2$$

Proof. Since the feature vector $\Phi_e(\mathbf{x}) = M_{\text{inv}}\mathbf{z}_{\text{inv}} + M_e\mathbf{z}_e$ is a linear combination of two independent Gaussian densities, $\Phi_e(\mathbf{x})$ is also Gaussian with the following density:

$$M_{\text{inv}}\mathbf{z}_{\text{inv}} + M_e\mathbf{z}_e \mid y \sim \mathcal{N}(y \cdot \underbrace{(M_{\text{inv}}\boldsymbol{\mu}_{\text{inv}} + M_e\boldsymbol{\mu}_e)}_{\boldsymbol{\mu}_\Phi}, \underbrace{M_{\text{inv}}M_{\text{inv}}^T\sigma_{\text{inv}}^2 + M_eM_e^T\sigma_e^2}_{\Sigma_\Phi}). \tag{A.3}$$

The conditional density is given by:

$$p(\Phi_e(\mathbf{x}) = \phi \mid y) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_\Phi|}}\exp(-\frac{1}{2}(\phi - y \cdot \boldsymbol{\mu}_\Phi)^\top\Sigma_\Phi^{-1}(\phi - y \cdot \boldsymbol{\mu}_\Phi)) \tag{A.4}$$

Then, the probability of $y = 1$ conditioned on $\Phi_e(\mathbf{x}) = \phi$ can be expressed

as:

$$
\begin{aligned}
p\left(y=1 \mid \Phi_e=\phi\right) &= \frac{p(\Phi_e=\phi \mid y=1)p(y=1)}{p(\Phi_e=\phi \mid y=1)p(y=1)+p(\Phi_e=\phi \mid y=-1)p(y=-1)} \\
&= \frac{1}{1+\frac{p(\Phi_e=\phi|y=-1)p(y=-1)}{p(\Phi_e=\phi|y=1)p(y=1)}} \\
&= \frac{1}{1+\exp(-y\cdot 2\phi^\top \Sigma_\Phi^{-1}\mu_\Phi - \log \eta/(1-\eta))} \\
&= \sigma\left(y\cdot 2\phi^\top \Sigma_\Phi^{-1}\mu_\Phi + \log \eta/(1-\eta)\right),
\end{aligned}
$$

where $\sigma(\cdot)$ is the sigmoid function. The log odds of y are linear w.r.t. the feature representation Φ_e. Thus given feature $\begin{bmatrix} \Phi_e(\mathbf{x}) \\ 1 \end{bmatrix} = \begin{bmatrix} \phi \\ 1 \end{bmatrix}$ (appended with constant 1), the optimal classifier weights are $\begin{bmatrix} 2\Sigma_\Phi^{-1}\mu_\Phi \\ \log \eta/(1-\eta) \end{bmatrix}$. Note that the Bayes optimal classifier uses environmental features which are informative of the label but non-invariant. □

Lemma A.7. (*Invariant classifier using non-invariant features*) *Suppose $E \leq d_e$, given a set of environments $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ such that all environmental means are linearly independent. Then there always exists a unit-norm vector $\mathbf{p}$ and positive fixed scalar β such that $\beta = \mathbf{p}^\top \boldsymbol{\mu}_e/\sigma_e^2 \ \forall e \in \mathcal{E}$. The resulting optimal classifier weights are*

$$
\hat{w} = \begin{bmatrix} \beta_{inv} \\ 2\beta \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{\mu}_{inv}/\sigma_{inv}^2 \\ 2\mathbf{p}^\top \boldsymbol{\mu}_e/\sigma_e^2 \end{bmatrix}.
$$

Proof. Suppose $M_{\text{inv}} = \begin{bmatrix} I_{s\times s} \\ 0_{1\times s} \end{bmatrix}$, and $M_e = \begin{bmatrix} 0_{s\times e} \\ \mathbf{p}^\top \end{bmatrix}$ for some unit-norm vector $\mathbf{p} \in \mathbb{R}^{d_e}$, then $\Phi_e(\mathbf{x}) = \begin{bmatrix} \mathbf{z}_{\text{inv}} \\ \mathbf{p}^\top \mathbf{z}_e \end{bmatrix}$. By plugging into the results of

Lemma A.1, we can obtain the optimal classifier weights as $\begin{bmatrix} 2\mu_{\text{inv}}/\sigma_{\text{inv}}^2 \\ 2\mathbf{p}^\top \boldsymbol{\mu}_e/\sigma_e^2 \end{bmatrix}$.[3] If the total number of environments is insufficient (i.e., $E \leq d_E$, which is a practical consideration because datasets with diverse environmental features w.r.t. a specific class of interest are often very computationally expensive to obtain), a short-cut direction $\mathbf{p}$ that yields invariant classifier weights satisfies the system of linear equations $A\mathbf{p} = \mathbf{b}$, where $A = \begin{bmatrix} \boldsymbol{\mu}_1^\top \\ \cdots \\ \boldsymbol{\mu}_E^\top \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} \sigma_1^2 \\ \cdots \\ \sigma_E^2 \end{bmatrix}$. As A has linearly independent rows and $E \leq d_e$, there always exists feasible solutions, among which the minimum-norm solution is given by $\mathbf{p} = A^\top(AA^\top)^{-1}\mathbf{b}$. Thus $\beta = 1/\|A^\top(AA^\top)^{-1}\mathbf{b}\|_2$. □

Theorem A.8. (*Failure of OOD detection under invariant classifier*) *Consider an out-of-distribution input which contains the environmental feature:* $\Phi_{out}(\mathbf{x}) = M_{inv}\mathbf{z}_{out} + M_e\mathbf{z}_e$, *where* $\mathbf{z}_{out} \perp \boldsymbol{\mu}_{inv}$. *Given the invariant classifier* (*cf. Lemma 2*), *the posterior probability for the OOD input is*

$$p(y = 1 \mid \Phi_{out}) = \sigma\left(2\mathbf{p}^\top \mathbf{z}_e \beta + \log \eta/(1-\eta)\right)$$

, *where* σ *is the logistic function. Thus for arbitrary confidence* $0 < c := P(y = 1 \mid \Phi_{out}) < 1$, *there exists* $\Phi_{out}(\mathbf{x})$ *with* $\mathbf{z}_e$ *such that*

$$\mathbf{p}^\top \mathbf{z}_e = \frac{1}{2\beta} \log \frac{c(1-\eta)}{\eta(1-c)}$$

.

Proof. Consider an out-of-distribution input $\mathbf{x}_{\text{out}}$ with $M_{\text{inv}} = \begin{bmatrix} I_{s\times s} \\ 0_{1\times s} \end{bmatrix}$, and $M_e = \begin{bmatrix} 0_{s\times e} \\ \mathbf{p}^\top \end{bmatrix}$, then the feature representation is $\Phi_e(\mathbf{x}) = \begin{bmatrix} \mathbf{z}_{\text{out}} \\ \mathbf{p}^\top \mathbf{z}_e \end{bmatrix}$,

[3] The constant term is $\log \eta/(1-\eta)$, as in Proposition A.2.

where $\mathbf{p}$ is the unit-norm vector defined in Lemma A.3. By Bayes' rule, the posterior probability of $y = 1$ can be expressed as:

$$\begin{aligned} P\left(y = 1 \mid \mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e\right) &= \frac{P\left(\mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e, y = 1\right)}{P\left(\mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e\right)} \\ &= \frac{P\left(\mathbf{z}_{\text{out}} \mid y = 1\right) P(\mathbf{p}^\top \mathbf{z}_e \mid y = 1) P(y = 1)}{P\left(\mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e\right)} \\ &= \frac{1}{1 + \frac{P(\mathbf{z}_{\text{out}} | y=-1) P(\mathbf{p}^\top \mathbf{z}_e | y=-1) P(y=-1)}{P(\mathbf{z}_{\text{out}} | y=1) P(\mathbf{p}^\top \mathbf{z}_e | y=1) P(y=1)}} \end{aligned} \tag{A.5}$$

Recall that the conditional density is given by:

$$p(\mathbf{z}_{\text{out}} \mid y) = \frac{1}{\sqrt{(2\pi)^s |\sigma_{\text{inv}}^2 I|}} \exp(-\frac{1}{2}(\mathbf{z}_{\text{out}} - y \cdot \boldsymbol{\mu}_{\text{inv}})^\top \frac{1}{\sigma_{\text{inv}}^2} \cdot I \cdot (\mathbf{z}_{\text{out}} - y \cdot \boldsymbol{\mu}_{\text{inv}})). \tag{A.6}$$

Canceling common terms, we get

$$\begin{aligned} P\left(y = 1 \mid \mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e\right) &= \frac{1}{1 + \frac{\exp\left(-\boldsymbol{\mu}_{\text{inv}}^\top \mathbf{z}_{\text{out}} / \sigma_{\text{inv}}^2 - \mathbf{p}^\top \mathbf{z}_e \beta\right)(1-\eta)}{\exp\left(\boldsymbol{\mu}_{\text{inv}}^\top \mathbf{z}_{\text{out}} / \sigma_{\text{inv}}^2 + \mathbf{p}^\top \mathbf{z}_e \beta\right)\eta}} \\ &= \frac{1}{1 + \exp\left(-\left(2\mathbf{p}^\top \mathbf{z}_e \beta + \log \eta/(1-\eta)\right)\right)} \end{aligned} \tag{A.7}$$

Then we have

$$P(y = 1 \mid \Phi_{\text{out}}) = P(y = 1 \mid \mathbf{z}_{\text{out}}, \mathbf{p}^\top \mathbf{z}_e) = \sigma\left(2\mathbf{p}^\top \mathbf{z}_e \beta + \log \eta/(1-\eta)\right)$$

, where σ is the logistic function. Thus for arbitrary confidence $0 < c := P(y = 1 \mid \Phi_{\text{out}}) < 1$, there exists $\Phi_{\text{out}}(\mathbf{x})$ with $\mathbf{z}_e$ such that

$$\mathbf{p}^\top \mathbf{z}_e = \frac{1}{2\beta} \log \frac{c(1-\eta)}{\eta(1-c)}$$

. □

Remark: In a more general case, $\mathbf{z}_{\text{out}}$ can be modeled as a random vector that is independent of the in-distribution labels $y = 1$ and $y = -1$ and environmental features: $\mathbf{z}_{\text{out}} \perp\!\!\!\perp y$ and $\mathbf{z}_{\text{out}} \perp\!\!\!\perp \mathbf{z}_e$. Thus in Eq. A.5 we have $P(\mathbf{z}_{\text{out}} \mid y = 1) = P(\mathbf{z}_{\text{out}} \mid y = -1) = P(\mathbf{z}_{\text{out}})$. Then $P(y = 1 \mid \Phi_{\text{out}}) = \sigma\left(2\mathbf{p}^\top \mathbf{z}_e \beta + \log \eta/(1-\eta)\right)$, same as in Eq. A.7. Therefore our main theorem still holds under more general cases.

A.9 Extension: Color Spurious Correlation

To further validate our findings beyond background and gender spurious (environmental) features, we provide additional experimental results with the ColorMNIST dataset, as shown in Figure A.5.

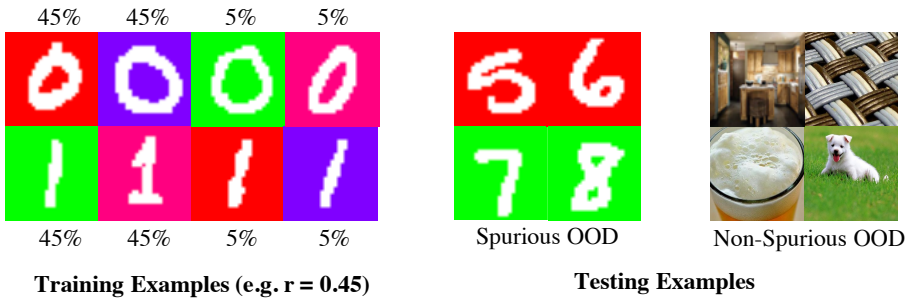


Figure A.5: **Left**: Training environments of ColorMNIST. The digit 0 correlates both red and purple background with probability r, whereas digit 1 correlates with green and pink with probability r. **Right**: Spurious OOD inputs contain the shared environmental feature (color background) yet with different digit labels (*e.g.*, not 0 or 1). Non-spurious (conventional) OOD samples share neither the digit semantics nor colors in the training set.

Evaluation Task 3: ColorMNIST. The ColorMNIST dataset is modified from MNIST (LeCun et al., 1998), which composes colored backgrounds on digit images. In this dataset, $\mathcal{E} = \{\texttt{red}, \texttt{green}, \texttt{purple}, \texttt{pink}\}$ denotes the background color and we use $\mathcal{Y} = \{0, 1\}$ as in-distribution classes.

The correlation between the background color e and the digit y is explicitly controlled, with $r \in \{0.25, 0.35, 0.45\}$. That is, r denotes the probability of $P(e = \texttt{red} \mid y = 0) = P(e = \texttt{purple} \mid y = 0) = P(e = \texttt{green} \mid y = 1) = P(e = \texttt{pink} \mid y = 1)$, while $0.5 - r = P(e = \texttt{green} \mid y = 0) = P(e = \texttt{pink} \mid y = 0) = P(e = \texttt{red} \mid y = 1) = P(e = \texttt{purple} \mid y = 1)$. Note that the maximum correlation r (reported in Table A.4) is 0.45. As ColorMNIST is relatively simpler compared to Waterbirds and CelebA, further increasing the correlation results in less interesting environments where the learner can easily pick up the contextual information. For spurious OOD, we use digits $\{5, 6, 7, 8, 9\}$ with background color `red` and `green`, which contain overlapping environmental features as the training data. For non-spurious OOD, following common practice (Hendrycks and Gimpel, 2017), we use the `Textures` (Cimpoi et al., 2014), `LSUN` (Yu et al., 2015) and `iSUN` (Xu et al., 2015) datasets. We train on ResNet-18 (He et al., 2016), which achieves 99.9% accuracy on the in-distribution test set. The OOD detection performance is shown in Table A.4.

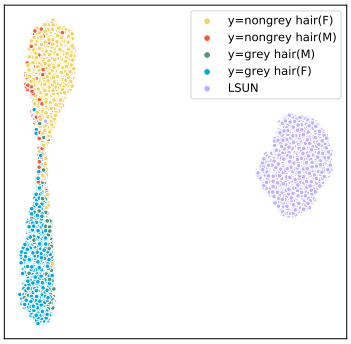
		r=0.25		r=0.35		r=0.45	
OOD Type	**Test Set**	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC**↑	**FPR95** ↓	**AUROC** ↑
Spurious OOD		5.40 ± 1.81	98.25 ± 0.89	13.5 ± 1.90	94.91 ± 0.86	30.45 ± 10.42	86.74 ± 7.76
Non-spurious OOD	Texture	0.03 ± 0.03	99.42 ± 0.35	0.43 ± 0.49	99.41 ± 0.17	9.93 ± 5.26	95.94 ± 0.88
	iSUN	0.18 ± 0.19	99.43 ± 0.27	0.25 ± 0.10	99.39 ± 0.14	6.68 ± 4.23	98.16 ± 1.25
	LSUN	0.3 ± 0.28	99.55 ± 0.17	0.63 ± 0.40	99.40 ± 0.19	6.33 ± 4.93	98.51 ± 0.80

Table A.4: OOD detection performance of models trained on **ColorMNIST**. Increased spurious correlation in the training set results in worsen performance for both non-spurious and spurious OOD samples. For any fixed spurious correlation, spurious OOD is more challenging than non-spurious OOD samples. Results (mean and std) are estimated over 4 runs for each setting.

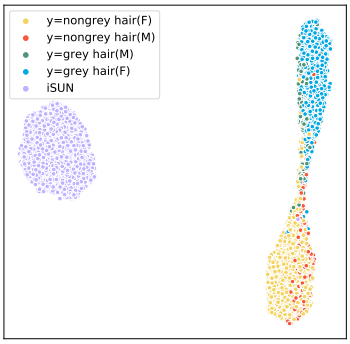
A.10 Visualization and Histograms

Visualization. As an extension of Section A.4, here we present the visualization of embeddings for ID samples and samples from non-spurious

OOD test sets LSUN (Figure A.6a) and iSUN (Figure A.6b) based on the CelebA task. We can observe that for both non-spurious OOD test sets, the feature representations of ID and OOD are separable, similar to observations in Section A.4.



(a) Feature for ID data and LSUN (Non-spurious).



(b) Feature for ID data and iSUN (Non-spurious).

Figure A.6: Visualization of feature embedding for in-distribution samples and non-spurious OOD samples: LSUN (Left) and iSUN (right).

Histograms. We also present histograms of the Mahalanobis distance score and MSP score for non-spurious OOD test sets iSUN and LSUN based on the CelebA task. As shown in Figure A.7, for both non-spurious OOD datasets, the observations are similar to what we describe in Section A.4 where ID and OOD are more separable with Mahalanobis score than MSP score. This further verifies that feature-based methods such as Mahalanobis score is promising to mitigate the impact of spurious correlation in the training set for non-spurious OOD test sets compared to output-based methods such as MSP score.

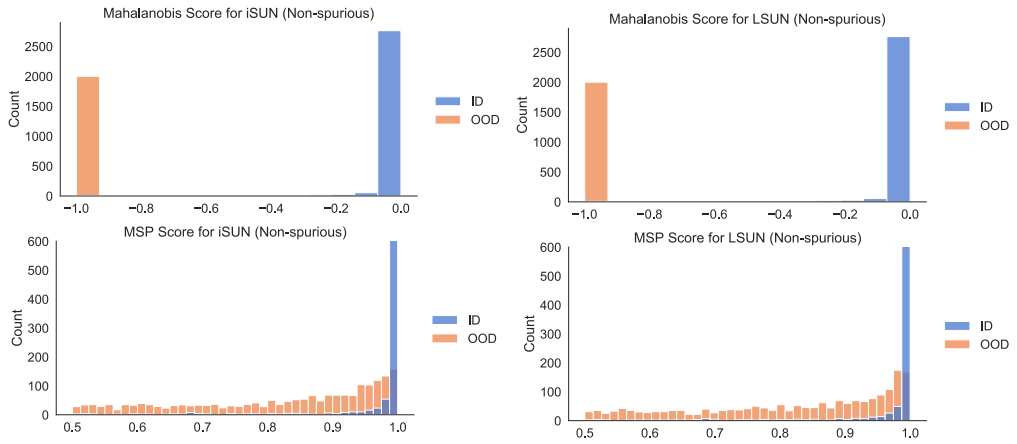


Figure A.7: **Left**: Histograms of the Mahalanobis score and MSP score for iSUN (Non-spurious OOD). **Right**: Histograms of the Mahalanobis score and MSP score for LSUN (Non-spurious OOD).

A.11 Adjusting Spurious Correlation in the Training Set for CelebA

To further validate if our observations on the impact of the extent of spurious correlation in the training set still hold beyond the Waterbirds and ColorMNIST tasks, here we subsample the CelebA dataset (described in Section A.3) such that the spurious correlation is reduced to $r = 0.7$. Note that we do not further reduce the correlation for CelebA because that will result in a small size of total training samples in each environment which may make the training unstable. The results are shown in Table A.5. The observations are similar to what we describe in Section A.3 where increased spurious correlation in the training set results in worsened performance for both non-spurious and spurious OOD samples. For example, the average FPR95 is reduced by 3.37% for LSUN, and 2.07% for iSUN when $r = 0.7$ compared to $r = 0.8$. In particular, spurious OOD is more challenging than non-spurious OOD samples under both spurious correlation settings.

		r=0.7		r=0.8	
OOD Type	**Test Set**	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC**↑ ↑
Spurious OOD		70.18 ± 1.76	83.30 ± 0.68	71.28 ± 4.12	82.04 ± 2.64
Non-spurious OOD	iSUN	15.28 ± 3.18	97.37 ± 0.14	17.35 ± 2.97	97.03 ± 0.30
	LSUN	15.48 ± 3.57	97.56 ± 0.20	18.85 ± 2.44	96.90 ± 0.17
	SVHN	5.39 ± 4.30	98.89 ± 0.90	5.63 ± 2.60	98.64 ± 0.21

Table A.5: OOD detection performance of models trained on **CelebA** (Liu et al., 2015). The observations are similar to the Waterbirds and ColorMNIST tasks. Increased spurious correlation in the training set results in worsen performance for both non-spurious and spurious OOD samples. In particular, spurious OOD is more challenging than non-spurious OOD samples. Results (mean and std) are estimated over 4 runs for each setting.

A.12 Extension: Training with Domain Invariance Objectives

In this section, we provide empirical validation of our analysis in Section A.5, where we evaluate the OOD detection performance based on models that are trained with recent prominent domain invariance learning objectives where the goal is to find a classifier that does not overfit to environment-specific properties of the data distribution. Note that OOD **generalization** aims to achieve high classification accuracy on new test environments consisting of inputs *with invariant features*, and does not consider the absence of invariant features at test time—a key difference from our focus. In the setting of spurious OOD **detection**, we consider test samples in environments *without invariant features*. We begin by describing the more popular objectives and include a more expansive list of invariant learning approaches in our study.

Invariant Risk Minimization (IRM). IRM (Arjovsky et al., 2019) assumes the existence of a feature representation Φ such that the optimal classifier on top of these features is the same across all environments. To learn this Φ, the IRM objective solves the following bi-level optimization

problem:

$$\min_{\Phi,\hat{w}} \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}}\mathcal{R}^e(\Phi,\hat{w}) \quad \text{s.t.} \quad \hat{w}\in\arg\min_{w}\mathcal{R}^e(\Phi,w) \quad \forall e\in\mathcal{E} \tag{A.8}$$

The authors also propose a practical version named IRMv1 as a surrogate to the original challenging bi-level optimization formula (A.8) which we adopt in our implementation:

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}}\sum_{e\in\mathcal{E}_{\text{tr}}} R^e(\Phi)+\lambda\cdot\left\|\nabla_{w|w=1.0}R^e(w\cdot\Phi)\right\|^2 \tag{A.9}$$

where an empirical approximation of the gradient norms in IRMv1 can be obtained by a balanced partition of batches from each training environment.

Group Distributionally Robust Optimization (GDRO). GDRO (Sagawa et al., 2019) minimizes the worst-group risk:

$$\min_{w}\max_{g\in\mathcal{G}}\mathbb{E}_{(\mathbf{x},y)\sim\hat{P}_g}[\ell(w;(x,y))], \tag{A.10}$$

where each example belongs to a group $g\in\mathcal{G}=\mathcal{Y}\times\mathcal{E}$, with $g=(y,e)$. The model learns the correlation between label y and environment e in the training data would do poorly on minority group where the correlation does not hold. Hence, by minimizing the worst-group risk, the model is discouraged from relying on spurious features. The authors show that objective (A.10) can be rewritten as:

$$\min_{w}\sup_{q\in\Delta m}\sum_{g=1}^{m} q_g\mathbb{E}_{(x,y)\sim P_g}[\ell(w;(x,y))] \tag{A.11}$$

Then Algorithm 1 in (Sagawa et al., 2019) can be used for optimization where stochastic gradient descent on w is interleaved with exponentiated

gradient ascent on q. For further details and convergence analysis, we encourage interested readers to refer to (Sagawa et al., 2019).

Alternative Objectives. IRM is motivated by the existence of a feature representation Φ such that $\mathbb{E}[y|\Phi(\mathbf{x})]$ is invariant across environments. Follow-up works proposed several variations, based on different notions of invariance. In particular, (Krueger et al., 2021) proposed Risk Extrapolation (**REx**), which aims to achieve stronger invariance $p(y|\Phi(\mathbf{x}))$ by penalizing the variance of risks of environments. Other approaches have proposed to remove the predictability of $p(e|\Phi(\mathbf{x}))$ through domain adversarial losses such as **DANN** (Ganin et al., 2016) and **CDANN** (Li et al., 2018c) (adapted for domain generalization). For completeness, we include all the aforementioned methods in our study[4].

Training Objective	FPR95 ↓	AUROC ↑
ERM (Vapnik, 1992)	71.28 ± 4.12	82.04 ± 2.64
IRM (Arjovsky et al., 2019)	70.09 ± 3.67	82.66 ± 3.28
GDRO (Sagawa et al., 2019)	68.77 ± 4.56	83.39 ± 3.12
REx (Krueger et al., 2021)	72.43 ± 3.21	81.88 ± 3.19
DANN (Ganin et al., 2016)	70.01 ± 7.47	82.30 ± 9.89
CDANN (Li et al., 2018c)	69.87 ± 4.19	82.93 ± 4.55

Table A.6: Spurious OOD detection performance on CelebA (Liu et al., 2015) where $r \approx 0.8$. The models are trained with domain invariance learning objectives. The results verify that detecting spurious OOD data is challenging as no training objectives significantly outperform ERM.

Results. Table A.6 summarizes the OOD detection performance for spurious OOD samples based on models trained with various invariance learning objectives. All methods are trained on the CelebA dataset described in Section A.3 where "Grey hair" is highly correlated with "Male' in the

[4]Our implementation for most of the training objects are based on: `https://github.com/facebookresearch/DomainBed`.

Training Objective	FPR95 ↓	AUROC ↑
ERM (Vapnik, 1992)	74.22 ± 13.12	80.98 ± 4.45
IRM (Arjovsky et al., 2019)	72.41 ± 13.27	81.29 ± 5.24
GDRO (Sagawa et al., 2019)	70.79 ± 11.51	82.94 ± 4.59
REx (Krueger et al., 2021)	73.83 ± 15.26	81.25 ± 4.99
DANN (Ganin et al., 2016)	72.81 ± 13.47	81.11 ± 6.21
CDANN (Li et al., 2018c)	72.37 ± 14.20	82.13 ± 3.53

Table A.7: Spurious OOD detection performance on Waterbirds (Sagawa et al., 2019) where $r = 0.7$. The models are trained with domain invariance learning objectives. The results are similar to what we observe for CelebA, where detecting spurious OOD data is challenging.

training set ($r \approx 0.8$). We then compute the energy score (Liu et al., 2020) from the model output $f(\mathbf{x})$ as OOD uncertainty measurement for OOD detection. From the table, we can observe that despite being motivated by invariance learning, many objectives do not significantly outperform the ERM baseline. For example, DGRO only mildly improves over ERM (1.35% improvement in terms of AUROC). Moreover, invariance learning methods generally display larger variances across runs compared to ERM. Similar observations still hold for Waterbirds, where we choose $r = 0.7$, as shown in Table A.7. A recent study (Gulrajani and Lopez-Paz, 2021) shows that ERM remains competitive in OOD generalization tasks compared with various domain invariance learning methods across a broad range of real-world datasets. While our results suggest that given high spurious correlation in the training set, detecting spurious OOD remains challenging, even for models trained with domain invariance objectives.

A.13 Experiment Details and In-distribution Classification Performance

Software and Hardware. Our code is implemented with Python 3.8.0 and PyTorch 1.6.0. All experiments are run on NVIDIA GeForce RTX 2080Ti.

Experiment Details. Following the common setup, the validation set is randomly selected from 20% of the training set. We perform grid search over learning rate $\gamma \in \{0.0001, 0.005, 0.001, 0.01\}$ and l_2 penalties $\lambda \in \{0.0001, 0.001, 0.01, 0.05\}$. We train for 30 epochs with SGD on ResNet-18. For ColorMNIST, we train from scratch while we start training with pre-trained ResNet for Waterbirds and CelebA, as in Sagawa et al. (2019).

In-distribution Classification Performance. Table A.8, Table A.9, and Table A.10 present the in-distribution data classification accuracy for models trained with ERM and other domain invariance learning objectives for different tasks respectively (averaged over 4 runs).

Training Objective	r=0.25	r=0.35	r=0.45
ERM (Vapnik, 1992)	99.98 ± 0.03	99.99 ± 0.02	99.97 ± 0.03
IRM (Arjovsky et al., 2019)	99.98 ± 0.02	100.00 ± 0.00	99.99 ± 0.02
GDRO (Sagawa et al., 2019)	99.97 ± 0.04	99.98 ± 0.03	99.98 ± 0.02
REx (Krueger et al., 2021)	100.00 ± 0.00	99.99 ± 0.02	99.99 ± 0.02
DANN (Ganin et al., 2016)	99.97 ± 0.02	99.99 ± 0.02	99.99 ± 0.02
CDANN (Li et al., 2018c)	99.97 ± 0.02	99.99 ± 0.02	99.98 ± 0.02

Table A.8: In-Distribution data classification accuracy on ColorMNIST.

Training Objective	r=0.5	r=0.7	r=0.9
ERM (Vapnik, 1992)	96.93 ± 0.05	96.64 ± 0.06	94.67 ± 0.25
IRM (Arjovsky et al., 2019)	96.48 ± 0.24	96.67 ± 0.13	94.70 ± 0.44
GDRO (Sagawa et al., 2019)	96.82 ± 0.00	96.63 ± 0.04	94.55 ± 0.12
REx (Krueger et al., 2021)	97.11 ± 0.07	96.67 ± 0.14	94.68 ± 0.10
DANN (Ganin et al., 2016)	96.65 ± 0.12	96.08 ± 0.08	93.57 ± 0.48
CDANN (Li et al., 2018c)	96.57 ± 0.09	96.19 ± 0.13	94.17 ± 0.21

Table A.9: In-Distribution data classification accuracy on Waterbirds (Sagawa et al., 2019).

Training Objective	r=0.8
ERM (Vapnik, 1992)	95.78 ± 0.48
IRM (Arjovsky et al., 2019)	95.97 ± 0.62
GDRO (Sagawa et al., 2019)	95.74 ± 0.54
REx (Krueger et al., 2021)	95.49 ± 0.77
DANN (Ganin et al., 2016)	96.27 ± 0.25
CDANN (Li et al., 2018c)	94.74 ± 0.63

Table A.10: In-Distribution data classification accuracy on CelebA (Liu et al., 2015).

Appendix B

Appendix for How to Exploit Hyperspherical Embeddings for Out-of-Distribution Detection

B.1 Algorithm Details and Discussions

The training scheme of our compactness and dispersion regularized (CIDER) learning framework is shown in Algorithm 1. We jointly optimize: (1) a *compactness loss* to encourage samples to be close to their class prototypes, and (2) a *dispersion loss* to encourage larger angular distances among different class prototypes.

Remark 1: The prototype update rule. The class prototypes are only updated by exponential moving average (EMA). Since the prototypes are not learnable parameters, the gradients of the dispersion loss have no direct impact on their updates. EMA-style techniques have been used in prior works Li et al. (2020b), and can be rigorously interpreted from a clustering-based Expectation-Maximization (EM) perspective. Alternatively, the prototypes can also be updated via gradients without EMA.

Algorithm 1: Pseudo-code of CIDER.

Input: Training dataset $\mathcal{D}$, neural network encoder f, projection head h, classifier g, class prototypes $\boldsymbol{\mu}_j$ $(1 \leq j \leq C)$, weights of loss terms λ_d, and λ_c, temperature τ

for $epoch = 1, 2, \ldots,$ **do**

  for $iter = 1, 2, \ldots,$ **do**

    sample a mini-batch $B = \{\mathbf{x}_i, y_i\}_{i=1}^{b}$

    obtain augmented batch $\tilde{B} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^{2b}$ by applying two random augmentations to $\mathbf{x}_i \in B$ $\forall i \in \{1, 2, \ldots, b\}$

    for $\tilde{\mathbf{x}}_i \in \tilde{B}$ **do**

      `// obtain normalized embedding`

      $\tilde{\mathbf{z}}_i = h(f(\tilde{\mathbf{x}}_i))$, $\mathbf{z}_i = \tilde{\mathbf{z}}_i / \|\tilde{\mathbf{z}}_i\|_2$

      `// update class-prototypes`

      $\boldsymbol{\mu}_c := \text{Normalize}(\alpha\boldsymbol{\mu}_c + (1-\alpha)\mathbf{z}_i),\ \forall c \in \{1, 2, \ldots, C\}$

    `// calculate compactness loss`

    $\mathcal{L}_{\text{comp}} = -\sum_{i=1}^{b} \log \frac{\exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)}/\tau\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_j/\tau\right)}$

    `// calculate dispersion loss`

    $\mathcal{L}_{\text{dis}} = \frac{1}{C} \sum_{i=1}^{C} \log \frac{1}{C-1} \sum_{j=1}^{C} \mathbb{1}\{j \neq i\} e^{\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau}$

    `// calculate overall loss`

    $\mathcal{L} = \mathcal{L}_{\text{dis}} + \lambda_c \mathcal{L}_{\text{comp}}$

    `// update the network weights`

    update the weights in the encoder and the projection head

We provide an ablation study of CIDER based on different prototype update rules in Appendix B.3.

Remark 2: CIDER vs. Wang and Isola (2020). The notion of alignment and uniformity for contrastive losses were proposed in Wang and Isola (2020) for the *unsupervised setting* where both metrics are based on individual samples. CIDER is a contrastive loss designed for the *supervised* setting. In particular, the uniform loss in Wang and Isola (2020) is defined based on randomly sampled pairs of data and promotes an *instance-to-instance* uniformity on the hypersphere. The notion of uniformity is fun-

damentally different from CIDER, which promotes *prototype-to-prototype* dispersion.

Remark 3: CIDER vs. Cross Entropy. When a model is trained with the cross-entropy (CE) loss, the weight matrix of the last fully connected layer can be interpreted as the set of class prototypes. However, CE is suboptimal for OOD detection for two main reasons: (1) CE does not explicitly optimize for the intra-class compactness and inter-class dispersion in the feature space. As a consequence, the embeddings obtained by CE loss display insufficient compactness and dispersion (Table 3.4). Compared to CE, CIDER is more structured by exploiting the hyperspherical embeddings and explicitly optimizing towards the desirable properties for OOD detection and ID classification. (2) the feature space obtained by CE loss is Euclidean instead of hyperspherical. As shown in Tack et al. (2020), ID data tend to have a larger norm than OOD data. As a result, the Euclidean distance between ID features can be larger than the distance from OOD to ID data. A recent work (Sun et al., 2022) verified that Euclidean embedding without feature normalization leads to suboptimal OOD detection performance. Instead, CIDER is designed to optimize hyperspherical embeddings, which benefit OOD detection.

Remark 4: On the measurement of embedding quality. In Section 3.4.3, we provide measurements of embedding quality (inter-class dispersion and intra-class compactness) via prototypes. In practice, one can replace the class prototypes in the dispersion and compactness metrics to be one random sample (or the average of a random subset) from the corresponding class. For example, on CIFAR-10, as the embeddings of CIDER are compact, we observe that the two metrics give similar results (e.g., the ID-OOD Separability is 42.5 with prototype-based metrics vs. 38.3 with instance-based metrics). We choose to use prototypes because (1) the definition directly maps to our loss function design, and (2) prototypes are

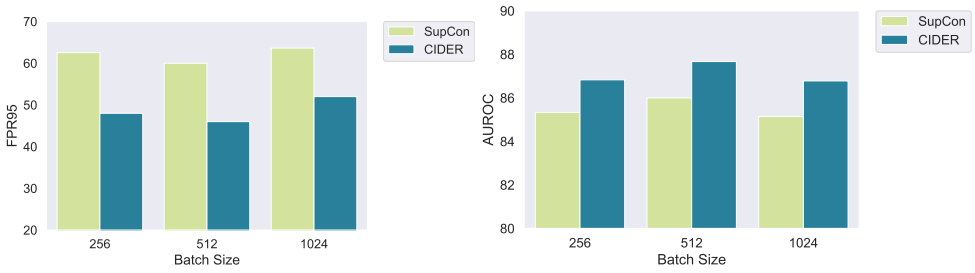
calculated as the averaged feature for each class, which helps to mitigate the sampling bias.

B.2 Experimental Details

Software and hardware. All methods are implemented in Pytorch 1.10. We run all the experiments on NVIDIA GeForce RTX-2080Ti GPU for small to medium batch size and on NVIDIA A100 GPU for large batch size and larger network encoder.

Architecture. As shown in Figure 3.1, the overall architecture of `CIDER` consists of a projection head h on top of a deep neural network encoder f. Following common practice and fair comparison with prior works (Khosla et al., 2020; Sehwag et al., 2021), we fix the output dimension of the projection head to be 128. We use a two-layer non-linear projection head for CIFAR-10 and CIFAR-100 as in Sun et al. (2022).

Training. For methods based on pre-trained models such as MSP (Hendrycks and Gimpel, 2017), ODIN (Liang et al., 2018), Mahalanobis (Lee et al., 2018), and Energy (Liu et al., 2020), we follow the configurations in Sun et al. (2022) for CIFAR-10 and train with the cross-entropy loss for 100 epochs. The initial learning rate is 0.1 and decays by a factor of 10 at epochs 50, 75, and 90 respectively. For the more challenging dataset CIFAR-100, we train 200 epochs. We use stochastic gradient descent with momentum 0.9, and weight decay 10^{-4}. For fair comparison, methods involving contrastive learning (Winkens et al., 2020; Tack et al., 2020; Sehwag et al., 2021) are trained for 500 epochs on CIFAR-10 and CIFAR-100. For `CIDER`, we adopt the same key hyperparameters for contrastive losses such as initial learning rate (0.5), temperature (0.1), and batch size (512) as `SSD+` (Sehwag et al., 2021) in main experiments to demonstrate the effectiveness and simplicity of CIDER. For the prototype update factor α,



(a) FPR95 under different batch sizes (b) AUROC under different batch sizes

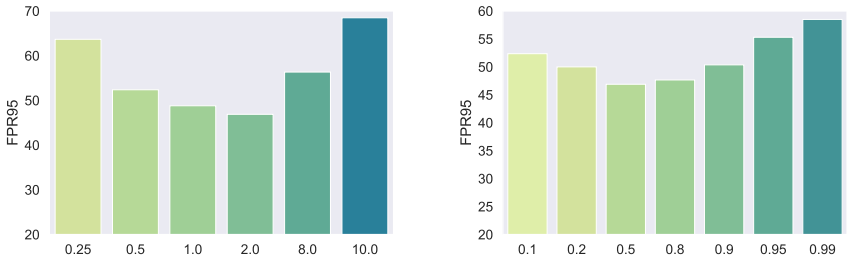
Figure B.1: Ablation on `CIDER` v.s. `SupCon` loss under different batch sizes. The results are averaged across the 5 OOD test sets based on ResNet-34. `CIDER` outperforms `SupCon` across different batch sizes, suggesting the effectiveness of explicitly facilitating prototype-wise dispersion.

we set the default value as 0.95 for simplicity. We observed that $\alpha = 0.95$ on CIFAR-10 with ResNet-18 and $\alpha = 0.5$ on CIFAR-100 with ResNet-34 provide stronger performance.

OOD detection score. By default, we use the non-parametric KNN score (Sun et al., 2022). We use a larger $K = 300$ for CIFAR-100 and a smaller $K = 100$ for CIFAR-10 for simplicity. Adjusting $K \in \{10, 20, 50, 100, 200, 300, 500\}$ yields similar performance. In practice, K can be tuned using a validation method (Sun et al., 2022) to further improve the performance.

B.3 Additional Ablation Studies

CIDER is effective under various batch sizes. Figure B.1a and B.1b also indicate that `CIDER` remains competitive under different batch size configurations compared to `SupCon`. To explain this, the standard `SupCon` loss requires *instance-to-instance* distance measurement, whereas compactness loss reduces the complexity to *instance-to-prototype*. The class-conditional prototypes are updated during training, which capture the average statistics of each class and alleviate the dependency on the batch size. This



(a) weight λ_c (b) Mov-avg α

Figure B.2: Ablation on (a) weight λ_c of the compactness loss; (b) prototype update discount factor α. The results are based on CIFAR-100 (ID) averaged over 5 OOD test sets.

leads to an overall memory-efficient solution for OOD detection.

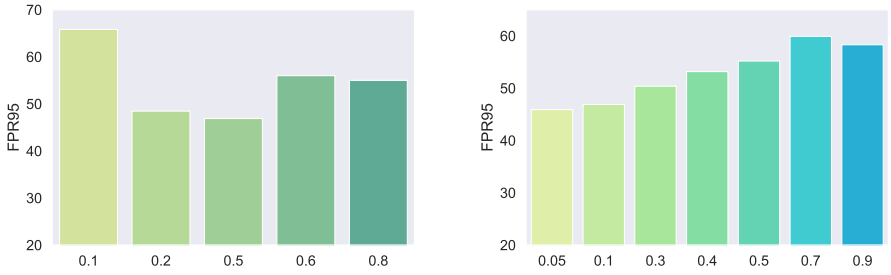
Ablation on the loss weights. In the main results (Table 3.1), we demonstrate the effectiveness of `CIDER` where the loss weight λ_c is simply set to balance the initial scale between the $\mathcal{L}_{\text{dis}}$ and $\mathcal{L}_{\text{comp}}$. In fact, `CIDER` can be further improved by adjusting λ_c. As shown in Figure B.2a, the performance of `CIDER` is relatively stable for moderate adjustments of λ_c (*e.g.* 0.5 to 2), with the best performance at around $\lambda_c \in [1, 2]$. This indicates `CIDER` provides a simple and effective solution for improving OOD detection, without much need for hyperparameter tuning on the loss scale.

Adjusting prototype update factor α improves CIDER. We show in Figure B.2b the performance by varying the moving-average discount factor α in Eq. 3.8. We can observe that the detection performance (averaged over 5 test sets) is still competitive across a wide range of α. In particular, for CIFAR-100, $\alpha = 0.5$ results in the best performance with average FPR95 of 46.89% under KNN score. For CIFAR-10, we observe that a larger α (*e.g.* 0.95 to 0.99) results in stronger performance.

Ablation on the learning rate. Prior works (Khosla et al., 2020; Sehwag et al., 2021) use a default initial learning rate (lr) of 0.5 to train contrastive

losses, which is also the default setting of CIDER. We further investigate the impact of the initial learning rate on OOD detection. As shown in Figure B.3a, a relatively higher initial lr is indeed desirable for competitive performance while too small lr (*e.g.* 0.1) would lead to performance degradation.

Small temperature τ leads to better performance. Figure B.3b demonstrates the detection performance as we vary the temperature parameter τ. We observe that the OOD detection performance is desirable at a relatively small temperature. Complementary to our finding, a relatively small temperature is shown to be desirable for ID classification (Khosla et al., 2020; Wang and Liu, 2021) which penalizes hard negative samples with larger gradients and leads to separable features.



(a) init lr (b) temperature τ

Figure B.3: Ablation on (a) initial learning rate and (b) temperature. The results are based on CIFAR-100 (ID) averaged over 5 OOD test sets.

Ablation on network capacity. We verify the effectiveness of CIDER under networks with other architectures such as ResNet-50 for CIFAR-100. The results are shown in Figure B.4. The trend is similar to what we observed with ResNet-34. Specifically, as a result of the improved representation, training with CIDER improves the FPR95 for various test sets compared to training with the SupCon loss.

Ablation on gradient-based prototype update. We examine the effect of updating prototypes via gradients. Compared to CIDER with EMA, CIDER with learnable prototypes (LP) can be more sensitive to initialization. We report the average performance of CIDER (EMA) and CIDER (LP) across 3 independent runs for CIFAR-10 in Table B.1. All training and evaluation configurations (e.g., learning rate and batch size) are the same. We can see that CIDER with EMA improves the average FPR95 by 5.08% with smaller standard deviation. Therefore, we empirically verify that updating prototypes via EMA is a better option with stronger training stability in practice.

Table B.1: Ablation on prototype update rules. OOD detection performance for ResNet-18 trained on CIFAR-10 with EMA-style updates denoted as CIDER (EMA) vs. learnable prototypes denoted as CIDER (LP). `CIDER` with EMA demonstrates strong OOD detection performance. Results are averaged over 3 independent runs.

Method	**OOD Dataset** SVHN		Places365		LSUN		iSUN		Texture		**Average**	
	FPR↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑
CIDER (LP)	$2.17^{\pm 1.50}$	$99.55^{\pm 0.48}$	$28.13^{\pm 1.59}$	$94.53^{\pm 0.12}$	$5.23^{\pm 2.68}$	$98.16^{\pm 0.99}$	$36.47^{\pm 8.93}$	$94.59^{\pm 0.92}$	$16.25^{\pm 1.07}$	$97.38^{\pm 0.19}$	$17.65^{\pm 2.36}$	$96.84^{\pm 0.40}$
CIDER (EMA)	$3.04^{\pm 1.38}$	$99.50^{\pm 0.30}$	$26.60^{\pm 2.47}$	$94.64^{\pm 0.51}$	$4.10^{\pm 1.68}$	$99.14^{\pm 0.19}$	$15.94^{\pm 4.56}$	$97.10^{\pm 0.54}$	$13.19^{\pm 0.82}$	$97.39^{\pm 0.48}$	$12.57^{\pm 1.31}$	$97.56^{\pm 0.33}$

Stability of CIDER. To verify that CIDER consistently provides strong performance, we train with 3 independent seeds for each ID dataset. Table B.2 shows the OOD detection performance of CIDER with ResNet-18 trained on CIFAR-10 and ResNet-34 trained on CIFAR-100. Comparing Table 3.1 and Table B.3, we can see that CIDER yields consistently strong performance. Code and checkpoints are provided in `https://github.com/deeplearning-wisc/cider`.

Table B.2: Ablation on stability. OOD detection performance of CIDER for CIFAR-10 and CIFAR-100. Results are averaged over 3 independent runs.

ID Dataset	OOD Dataset										Average	
	SVHN		Places365		LSUN		iSUN		Texture			
	FPR↓	**AUROC↑**	**FPR↓**	**AUROC↑**	**FPR↓**	**AUROC↑**	**FPR↓**	**AUROC↑**	**FPR↓**	**AUROC↑**	**FPR↓**	**AUROC↑**
CIFAR-10	$3.04^{\pm 1.38}$	$99.50^{\pm 0.30}$	$26.60^{\pm 2.47}$	$94.64^{\pm 0.51}$	$4.10^{\pm 1.68}$	$99.14^{\pm 0.19}$	$15.94^{\pm 4.56}$	$97.10^{\pm 0.54}$	$13.19^{\pm 0.82}$	$97.39^{\pm 0.48}$	$12.57^{\pm 1.31}$	$97.56^{\pm 0.33}$
CIFAR-100	$23.67^{\pm 2.28}$	$95.07^{\pm 0.13}$	$79.37^{\pm 1.84}$	$72.97^{\pm 3.90}$	$22.04^{\pm 5.12}$	$96.01^{\pm 1.80}$	$62.16^{\pm 8.48}$	$83.70^{\pm 2.92}$	$44.96^{\pm 6.01}$	$90.25^{\pm 0.97}$	$46.45^{\pm 2.01}$	$87.60^{\pm 1.03}$

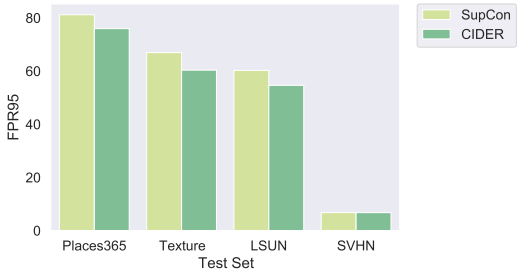
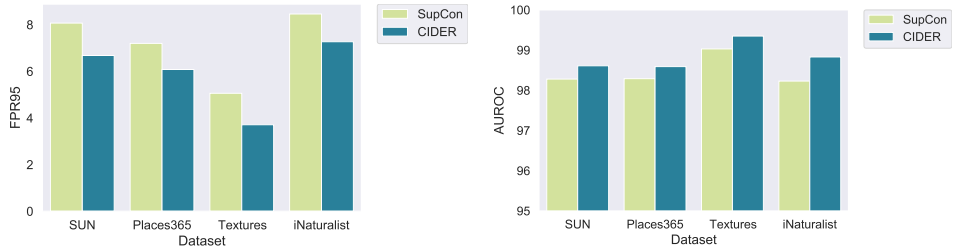


Figure B.4: Ablation on architecture. Results are based on ResNet-50.

B.4 Results on Large-scale Datasets

In recent years, there has been a paradigm shift towards fine-tuning pre-trained models, as opposed to training from scratch. Given this trend, it is important to explore whether CIDER remains effective based on pre-trained models. Specifically, we fine-tune ImageNet pre-trained ResNet-34 on ImageNet-100 with CIDER and SupCon losses for 10 epochs. For each loss, we update the weights of the last residual block and the nonlinear projection head, while freezing the parameters in the first three residual blocks. At test time, we use the same detection score (KNN) to evaluate representation quality. FPR95 and AUROC for each OOD test set are shown in Figure B.5a and B.5b, respectively. The results suggest that CIDER remains very competitive, which highlight the benefits of promoting inter-class dispersion and intra-class compactness.



(a) FPR95 for different OOD test sets

(b) AUROC for different OOD test sets

Figure B.5: OOD detection performance of fine-tuning with CIDER v.s. SupCon for ImageNet-100 (ID). With the same detection score (KNN), CIDER consistently outperforms SupCon across all OOD test datasets.

B.5 Results on CIFAR-10

In the main paper, we mainly focus on the more challenging task CIFAR-100. In this section, we additionally evaluate on CIFAR-10, a commonly used benchmark in literature. For methods involving contrastive losses, we use the same network encoder and embedding dimension, while only varying the training objective. The Mahalanobis score is used for OOD detection in `SSD+` (Sehwag et al., 2021), `CE+SimCLR` (Winkens et al., 2020), `SupCon`, and `CIDER`. As CIFAR-10 is much less challenging compared to CIFAR-100, recent methods with contrastive losses yield similarly strong performance. For methods trained with cross-entropy loss, we use the publicly available checkpoints in Sun et al. (2022) for better consistency. The results are shown in Table B.3. Similar trends also hold as we describe in Section 3.4.2: (1) `CIDER` achieves superior OOD detection performance in CIFAR-10 as a result of better inter-class dispersion and intra-class compactness. For example, compared to the Mahalanobis baseline (Lee et al., 2018), `CIDER` reduces the FPR95 by 24.99% averaged over 5 diverse test sets; (2) Although the ID classification accuracy of `CIDER` is similar to another proxy-based loss `ProxyAnchor` (Table B.4), `CIDER` significantly improves the OOD detection performance by 21.05% in FPR95 due to the

addition of explicit inter-class dispersion which we show is critical for OOD detection in Section 3.4.3. The significant improvements highlight the importance of representation learning for OOD detection.

Method	OOD Dataset SVHN		Places365		LSUN		iSUN		Texture		Average	
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
Without Contrastive Learning												
MSP	59.66	91.25	62.46	88.64	45.21	93.80	54.57	92.12	66.45	88.50	57.67	90.86
Energy	54.41	91.22	42.77	91.02	10.19	98.05	27.52	95.59	55.23	89.37	38.02	93.05
ODIN	53.78	91.30	43.40	90.98	10.93	97.93	28.44	95.51	55.59	89.47	38.43	93.04
GODIN	18.72	96.10	55.25	85.50	11.52	97.12	30.02	94.02	33.58	92.20	29.82	92.97
Mahalanobis	9.24	97.80	83.50	69.56	67.73	73.61	6.02	98.63	23.21	92.91	37.94	86.50
With Contrastive Learning												
CE + SimCLR	6.98	99.22	54.39	86.70	64.53	85.60	59.62	86.78	16.77	96.56	40.46	90.97
CSI	37.38	94.69	38.31	93.04	10.63	97.93	10.36	98.01	28.85	94.87	25.11	95.71
SSD+	2.47	99.51	22.05	95.57	10.56	97.83	28.44	95.67	9.27	98.35	14.56	97.38
ProxyAnchor	39.27	94.55	43.46	92.06	21.04	97.02	23.53	96.56	42.70	93.16	34.00	94.67
KNN+	2.70	99.61	23.05	94.88	7.89	98.01	24.56	96.21	10.11	97.43	13.66	97.22
CIDER	2.89	99.72	23.88	94.09	5.45	99.01	20.21	96.64	12.33	96.85	12.95	97.26

Table B.3: Results on CIFAR-10. OOD detection performance for ResNet-18 trained on CIFAR-10 with and without contrastive loss. `CIDER` achieves strong OOD detection performance and ID classification accuracy (Table B.4).

B.6 ID Classification Accuracy

The ID classification accuracy on CIFAR-10 and CIFAR-100 can be seen in Table B.4 and Table B.5, where for contrastive losses such as KNN+, SSD+, and CIDER, we follow the common practice as in Khosla et al. (2020) and use linear probe on normalized features.

Table B.4: ID classification accuracy on CIFAR-10 (%)

Method	ID ACC
w.o. contrastive loss	
MSP	94.21
ODIN	94.21
GODIN	93.64
Energy	94.21
Mahalanobis	94.21
w. contrastive loss	
CE + SimCLR	93.12
SSD+	94.53
ProxyAnchor	94.21
KNN+	94.53
CIDER	94.58

Table B.5: ID classification accuracy on CIFAR-100 (%)

Method	ID ACC
w.o. contrastive loss	
MSP	74.59
ODIN	74.59
GODIN	74.92
Energy	74.59
Mahalanobis	74.59
w. contrastive loss	
CE + SimCLR	73.54
SSD+	75.11
ProxyAnchor	74.21
KNN+	75.11
CIDER	75.35

Appendix C

Appendix for Hyperspherical Out-of-Distribution Generalization

C.1 Pseudo Algorithm

The training scheme of HYPO is shown below. We jointly optimize for (1) *low variation*, by encouraging the feature embedding of samples to be close to their class prototypes; and (2) *high separation*, by encouraging different class prototypes to be far apart from each other.

C.2 Broader Impacts

Our work facilitates the theoretical understanding of OOD generalization through prototypical learning, which encourages low variation and high separation in the hyperspherical space. In Section 4.5.2, we qualitatively and quantitatively verify the low intra-class variation of the learned embeddings and we discuss in Section 4.6 that the variation estimate determines the general upper bound on the generalization error for a learn-

Algorithm 2: Hyperspherical Out-of-Distribution Generalization

Input: Training dataset $\mathcal{D}$, deep neural network encoder h, class prototypes $\boldsymbol{\mu}_c$ $(1 \leq j \leq C)$, temperature τ
for $epoch = 1, 2, \ldots,$ **do**
  for $iter = 1, 2, \ldots,$ **do**
    sample a mini-batch $B = \{\mathbf{x}_i, y_i\}_{i=1}^{b}$
    obtain augmented batch $\tilde{B} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^{2b}$ by applying two random augmentations to $\mathbf{x}_i \in B \; \forall i \in \{1, 2, \ldots, b\}$
    for $\tilde{\mathbf{x}}_i \in \tilde{B}$ **do**
      `// obtain normalized embedding`
      $\tilde{\mathbf{z}}_i = h(\tilde{\mathbf{x}}_i)$, $\mathbf{z}_i = \tilde{\mathbf{z}}_i / \|\tilde{\mathbf{z}}_i\|_2$
      `// update class-prototypes`
      $\boldsymbol{\mu}_c := \text{Normalize}(\alpha \boldsymbol{\mu}_c + (1-\alpha)\mathbf{z}_i), \; \forall c \in \{1, 2, \ldots, C\}$
    `// calculate the loss for low variation`
    $\mathcal{L}_{\text{var}} = -\frac{1}{N} \sum_{e \in \mathcal{E}_{\text{avail}}} \sum_{i=1}^{|\mathcal{D}^e|} \log \frac{\exp\left(\mathbf{z}_i^{e\top} \boldsymbol{\mu}_{c(i)} / \tau\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{z}_i^{e\top} \boldsymbol{\mu}_j / \tau\right)}$
    `// calculate the loss for high separation`
    $\mathcal{L}_{\text{sep}} = \frac{1}{C} \sum_{i=1}^{C} \log \frac{1}{C-1} \sum_{j \neq i, j \in \mathcal{Y}} \exp\left(\boldsymbol{\mu}_i^{\top} \boldsymbol{\mu}_j / \tau\right)$
    `// calculate overall loss`
    $\mathcal{L} = \mathcal{L}_{\text{var}} + \mathcal{L}_{\text{sep}}$
    `// update the network weights`
    update the weights in the deep neural network

able OOD generalization task. This provable framework may serve as a foothold that can be useful for future OOD generalization research via representation learning.

From a practical viewpoint, our research can directly impact many real applications, when deploying machine learning models in the real world. Out-of-distribution generalization is a fundamental problem and is commonly encountered when building reliable ML systems in the industry. Our empirical results show that our approach achieves consistent improvement over the baseline on a wide range of tasks. Overall, our

work has both theoretical and practical impacts.

C.3 Theoretical Analysis

Notations. We first set up notations for theoretical analysis. Recall that $\mathbb{P}^e_{XY}$ denotes the joint distribution of X, Y in domain e. The label set $\mathcal{Y} := \{1, 2, \cdots, C\}$. For an input $\mathbf{x}$, $\mathbf{z} = h(\mathbf{x})/\|h(\mathbf{x})\|_2$ is its feature embedding. Let $\mathbb{P}^{e,y}_X$ denote the marginal distribution of X in domain e with class y. Similarly, $\mathbb{P}^{e,y}_Z$ denotes the marginal distribution of Z in domain e with class y. Let $E := |\mathcal{E}_{\text{train}}|$ for abbreviation. As we do not consider the existence of spurious correlation in this work, it is natural to assume that domains and classes are uniformly distributed: $\mathbb{P}_X := \frac{1}{EC}\sum_{e,y}\mathbb{P}^{e,y}_X$. We specify the distance metric to be the Wasserstein-1 distance *i.e.,* $\mathcal{W}_1(\cdot,\cdot)$ and define all notions of variation under such distance.

Next, we proceed with several lemmas that are particularly useful to prove our main theorem.

Lemma C.1. *With probability at least* $1-\delta$*,*

$$
\begin{aligned}
&-\mathbb{E}_{(\mathbf{x},c)\sim\mathbb{P}_{XY}}\boldsymbol{\mu}_c^\top\frac{h(\mathbf{x})}{\|h(\mathbf{x})\|_2}+\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}\\
&\le\mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^{N}\sigma_i\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}]+\beta\sqrt{\frac{\ln(2/\delta)}{N}}.
\end{aligned}
$$

where β *is a universal constant and* $\sigma_1,\ldots,\sigma_N$ *are Rademacher variables.*

Proof. By Cauchy-Schwarz inequality,

$$
|\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}|\le\left\|\boldsymbol{\mu}_{c(i)}\right\|_2\left\|\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}\right\|_2=1
$$

Define $\mathcal{G}=\{\langle\frac{h(\cdot)}{\|h(\cdot)\|_2},\cdot\rangle : h\in\mathcal{H}\}$. Let $S=(\mathbf{u}_1,\ldots,\mathbf{u}_N)\sim\mathbb{P}_N$ where

$\mathbf{u}_i = \begin{pmatrix} \mathbf{x}_i \\ \boldsymbol{\mu}_{c(i)} \end{pmatrix}$ and N is the sample size. The Rademacher complexity of $\mathcal{G}$ is

$$\mathcal{R}_N(\mathcal{G}) := \mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\sup_{g\in\mathcal{G}}\sum_{i=1}^{N}\sigma_i g(\mathbf{u}_i)].$$

We can apply the standard Rademacher complexity bound (Theorem 26.5 in Shalev-Shwartz and Ben-David) to $\mathcal{G}$, then we have that,

$$\begin{aligned}
&-\mathbb{E}_{(\mathbf{x},c)\sim\mathbb{P}_{XY}}\boldsymbol{\mu}_c^\top\frac{h(\mathbf{x})}{\|h(\mathbf{x})\|_2}+\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}\\
&\leq \mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{g\in\mathcal{G}}\sum_{i=1}^{N}\sigma_i g(\mathbf{u}_i)]+\beta\sqrt{\frac{\ln(2/\delta)}{N}}\\
&= \mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^{N}\sigma_i\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}]+\beta\sqrt{\frac{\ln(2/\delta)}{N}},
\end{aligned}$$

where β is a universal positive constant.

Remark C.2. *The above lemma indicates that when samples are sufficiently aligned with their class prototypes on the hyperspherical feature space,* i.e.*,*

$$\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}\geq 1-\epsilon$$

for some small constant $\epsilon > 0$, we can upper bound $-\mathbb{E}_{(\mathbf{x},c)\sim\mathbb{P}_{XY}}\boldsymbol{\mu}_c^\top\frac{h(\mathbf{x})}{\|h(\mathbf{x})\|_2}$. This result will be useful to prove Thm 4.6.

Lemma C.3. *Suppose $\mathbb{E}_{(\mathbf{z},c)\sim\mathbb{P}_{ZY}}\boldsymbol{\mu}_c^\top\mathbf{z}\geq 1-\gamma$. Then, for all $e\in\mathcal{E}_{\text{train}}$ and $y\in[C]$, we have that*

$$\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}\boldsymbol{\mu}_c^\top\mathbf{z}\geq 1-CE\gamma.$$

Proof. Fix $e' \in \mathcal{E}_{\text{train}}$ and $y' \in [C]$. Then,

$$\begin{aligned}
1-\gamma &\leq \mathbb{E}_{(\mathbf{z},c)\sim\mathbb{P}_{ZY}} \boldsymbol{\mu}_c^\top \mathbf{z} \\
&= \frac{1}{CE} \sum_{e\in\mathcal{E}_{\text{train}}} \sum_{y\in[C]} \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}} \mathbf{z}^\top \boldsymbol{\mu}_y \\
&= \frac{1}{CE} \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y'}} \mathbf{z}^\top \boldsymbol{\mu}_{y'} + \frac{1}{CE} \sum_{(e,y)\in\mathcal{E}_{\text{train}}\times[C]\setminus\{(e',y')\}} \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}} \mathbf{z}^\top \boldsymbol{\mu}_y \\
&\leq \frac{1}{CE} \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y'}} \mathbf{z}^\top \boldsymbol{\mu}_{y'} + \frac{CE-1}{CE}
\end{aligned}$$

where the last line holds by $|\mathbf{z}^\top \boldsymbol{\mu}_c| \leq 1$ and we also used the assumption that the domains and classes are uniformly distributed. Rearranging the terms, we have

$$1 - CE\gamma \leq \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y'}} \mathbf{z}^\top \boldsymbol{\mu}_{y'}$$

Lemma C.4. *Fix $y \in [C]$ and $e \in \mathcal{E}_{\text{train}}$. Fix $\eta > 0$. If*

$$\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}} \mathbf{z}^\top \boldsymbol{\mu}_y \geq 1 - CE\gamma,$$

then

$$\mathbb{P}_Z^{e,y}(\|\mathbf{z} - \boldsymbol{\mu}_y\|_2 \geq \eta) \leq \frac{2CE\gamma}{\eta^2}.$$

Proof. Note that

$$\begin{aligned}
\|\mathbf{z} - \boldsymbol{\mu}_y\|_2^2 &= \|\mathbf{z}\|_2^2 + \|\boldsymbol{\mu}_y\|_2^2 - 2\mathbf{z}^\top \boldsymbol{\mu}_y \\
&= 2 - 2\mathbf{z}^\top \boldsymbol{\mu}_y.
\end{aligned}$$

Taking the expectation on both sides and applying the hypothesis, we

have that

$$\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}} \|\mathbf{z} - \boldsymbol{\mu}_c\|_2^2 \leq 2CE\gamma.$$

Applying Chebyschev's inequality to $\|\mathbf{z} - \boldsymbol{\mu}_y\|_2$, we have that

$$\begin{aligned}
\mathbb{P}_Z^{e,y}(\|\mathbf{z} - \boldsymbol{\mu}_y\|_2 \geq \eta) &\leq \frac{\text{Var}(\|\mathbf{z} - \boldsymbol{\mu}_y\|_2)}{\eta^2} \\
&\leq \frac{\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}(\|\mathbf{z} - \boldsymbol{\mu}_y\|_2^2)}{\eta^2} \\
&\leq \frac{2CE\gamma}{\eta^2}
\end{aligned}$$

Lemma C.5. *Fix $y \in [C]$. Fix $e, e' \in \mathcal{E}_{train}$. Suppose $\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}} \mathbf{z}^\top \boldsymbol{\mu}_c \geq 1 - CE\gamma$. Fix $\mathbf{v} \in S^{d-1}$. Let P denote the distribution of $\mathbf{v}^\top \mathbf{z}_e$ and Q denote the distribution $\mathbf{v}^\top \mathbf{z}_{e'}$. Then,*

$$\mathcal{W}_1(P, Q) \leq 10(CE\gamma)^{1/3}$$

where $\mathcal{W}_1(P, Q)$ is the Wassersisten-1 distance.

Proof. Consider the dual formulation of Wasserstein-1 distance:

$$\mathcal{W}(P, Q) = \sup_{f:\|f\|_{\text{lip}}\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_X^{e,y}}[f(\mathbf{v}^\top \mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_X^{e',y}}[f(\mathbf{v}^\top \mathbf{x})]$$

where $\|f\|_{\text{lip}}$ denotes the Lipschitz norm. Let $\kappa > 0$. There exists f_0 such that

$$\mathcal{W}(P, Q) \leq \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})] + \kappa.$$

We assume that without loss of generality $f_0(\boldsymbol{\mu}_y^\top \mathbf{v}) = 0$. Define $f'(\cdot) =$

$f_0(\cdot) - f_0(\boldsymbol{\mu}_y^\top \mathbf{v})$. Then, note that $f'(\boldsymbol{\mu}_y^\top \mathbf{v}) = 0$ and

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f'(\mathbf{v}^\top \mathbf{z})] - \mathbb{E}_{z\sim\mathbb{P}_Z^{e',y}}[f'(\mathbf{v}^\top \mathbf{z})] \\
&= \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})] + f'(\boldsymbol{\mu}_y^\top v) - f'(\boldsymbol{\mu}_y^\top v) \\
&= \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})],
\end{aligned}$$

proving the claim.

Now define $B := \{\mathbf{u} \in S^{d-1} : \|\mathbf{u} - \boldsymbol{\mu}_y\|_2 \leq \eta\}$. Then, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})] \\
&= \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \in B\}] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \in B\}] \\
&+ \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \notin B\}] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \notin B\}]
\end{aligned}$$

Note that if $\mathbf{z} \in B$, then by $\|f\|_{\text{lip}} \leq 1$,

$$\begin{aligned}
|f_0(\mathbf{v}^\top \mathbf{z}) - f_0(\mathbf{v}^\top \boldsymbol{\mu}_y)| &\leq |\mathbf{v}^\top(\mathbf{z} - \boldsymbol{\mu}_y)| \\
&\leq \|\mathbf{v}\|_2 \|\mathbf{z} - \boldsymbol{\mu}_y\|_2 \\
&\leq \eta.
\end{aligned}$$

Therefore, $|f_0(\mathbf{v}^\top \mathbf{z})| \leq \eta$ and we have that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \in B\}] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top \mathbf{z})\mathbb{1}\{\mathbf{z} \in B\}] \\
&\leq 2\eta(\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[\mathbb{1}\{\mathbf{z} \in B\}] + \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[\mathbb{1}\{\mathbf{z} \in B\}]) \\
&\leq 2\eta.
\end{aligned}$$

Now, note that $\max_{\mathbf{u}\in S^{d-1}} |f(\mathbf{u}^\top \mathbf{v})| \leq 2$ (repeat the argument from

above but use $\|\mathbf{u} - \boldsymbol{\mu}_y\|_2 \leq 2$. Then,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[f_0(\mathbf{v}^\top\mathbf{z})\mathbb{1}\{\mathbf{z}\notin B\}] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[f_0(\mathbf{v}^\top\mathbf{z})\mathbb{1}\{\mathbf{z}\notin B\}] \\
&\leq 2[\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}[\mathbb{1}\{\mathbf{z}\notin B\}] + \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e',y}}[\mathbb{1}\{\mathbf{z}\notin B\}]] \\
&\leq \frac{8CE\gamma}{\eta}
\end{aligned}$$

where in the last line, we used the hypothesis and Lemma C.4. Thus, by combining the above, we have that

$$\mathcal{W}(P,Q) \leq 2\eta + \frac{8CE\gamma}{\eta^2} + \kappa.$$

Choosing $\eta = (CE\gamma)^{1/3}$, we have that

$$\mathcal{W}(P,Q) \leq 10(CE\gamma)^{1/3} + \kappa.$$

Since $\kappa > 0$ was arbitrary, we can let it go to 0, obtaining the result.

Next, we are ready to prove our main results. For completeness, we state the theorem here.

Theorem C.6 (Variation upper bound (Thm 4.1))**.** *Suppose samples are aligned with class prototypes such that* $\frac{1}{N}\sum_{j=1}^N \boldsymbol{\mu}_{c(j)}^\top \mathbf{z}_j \geq 1-\epsilon$ *for some* $\epsilon \in (0,1)$*, where* $\mathbf{z}_j = \frac{h(\mathbf{x}_j)}{\|h(\mathbf{x}_j)\|_2}$*. Then* $\exists\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\mathcal{V}^{\text{sup}}(h, \Sigma_{\text{avail}}) \leq O(\epsilon^{1/3} + (\mathbb{E}_{\mathcal{D}}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^N \sigma_i \mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)}])^{1/3} + (\frac{\ln(2/\delta)}{N})^{1/6}),$$

where $\sigma_1,\ldots,\sigma_N$ *are Rademacher random variables and* $O(\cdot)$ *suppresses dependence on constants and* $|\mathcal{E}_{\text{avail}}|$*.*

Proof of Theorem 4.6. Suppose $\frac{1}{N}\sum_{j=1}^N \boldsymbol{\mu}_{c(j)}^\top \mathbf{z}_j = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\mu}_{c(i)}^\top \frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2} \geq 1-\epsilon$.

Then, by Lemma C.1, with probability at least $1-\delta$, we have

$$\begin{aligned}
-\mathbb{E}_{(\mathbf{x},c)\sim\mathbb{P}_{XY}}\boldsymbol{\mu}_c^\top\frac{h(\mathbf{x})}{\|h(\mathbf{x})\|_2} &\leq \mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^N\sigma_i\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}] \\
&\quad+\beta\sqrt{\frac{\ln(2/\delta)}{N}}-\frac{1}{N}\sum_{i=1}^N\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2} \\
&\leq \mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^N\sigma_i\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}] \\
&\quad+\beta\sqrt{\frac{\ln(2/\delta)}{N}}+\epsilon-1
\end{aligned}$$

where $\sigma_1,\ldots,\sigma_N$ denote Rademacher random variables and β is a universal positive constant. Define

$$\gamma=\epsilon+\mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N}\sup_{h\in\mathcal{H}}\sum_{i=1}^N\sigma_i\boldsymbol{\mu}_{c(i)}^\top\frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}]+\beta\sqrt{\frac{\ln(2/\delta)}{N}}.$$

Then, we have

$$\mathbb{E}_{(\mathbf{z},c)\sim\mathbb{P}_{ZY}}\boldsymbol{\mu}_c^\top\mathbf{z}\geq 1-\gamma.$$

Then, by Lemma C.3, for all $e\in\mathcal{E}_{\text{train}}$ and $y\in[C]$,

$$\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_Z^{e,y}}\boldsymbol{\mu}_y^\top\mathbf{z}\geq 1-CE\gamma.$$

Let $\alpha>0$ and $\mathbf{v}_0$ such that

$$\mathcal{V}^{\text{sup}}(h,\mathcal{E}_{\text{train}})=\sup_{\mathbf{v}\in S^{d-1}}\mathcal{V}(\mathbf{v}^\top h,\mathcal{E}_{\text{train}})\leq\mathcal{V}(\mathbf{v}_0^\top h,\mathcal{E}_{\text{train}})+\alpha$$

Let $Q_{\mathbf{v}_0}^{e,y}$ denote the distribution of $\mathbf{v}_0^\top\mathbf{z}$ in domain e under class y. From

Lemma C.5, we have that

$$\mathcal{W}_1(Q^{e,y}_{\mathbf{v}_0}, Q^{',y}_{\mathbf{v}_0}) \leq 10(CE\gamma)^{1/3}$$

for all $y \in [C]$ and $e, e' \in \mathcal{E}_{\text{train}}$.

We have that

$$\begin{aligned}
\sup_{\mathbf{v}\in S^{d-1}} \mathcal{V}(\mathbf{v}^\top h, \mathcal{E}_{\text{train}}) &= \sup_{\mathbf{v}\in S^{d-1}} \mathcal{V}(\mathbf{v}^\top h, \mathcal{E}_{\text{train}}) \\
&= \max_y \sup_{e,e'} \mathcal{W}_1(Q^{e,y}_{\mathbf{v}_0}, Q^{e',y}_{\mathbf{v}_0}) + \alpha \\
&\leq 10(CE\gamma)^{1/3} + \alpha.
\end{aligned}$$

Noting that α was arbitrary, we may send it to 0 yielding

$$\sup_{\mathbf{v}\in S^{d-1}} \mathcal{V}(\mathbf{v}^\top h, \mathcal{E}_{\text{train}}) \leq 10(CE\gamma)^{1/3}.$$

Now, using the inequality that for $a, b, c \geq 0$, $(a+b+c)^{1/3} \leq a^{1/3} + b^{1/3} + c^{1/3}$, we have that

$$\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{train}}) \leq O(\epsilon^{1/3} + (\mathbb{E}_{S\sim\mathbb{P}_N}[\frac{1}{N}\mathbb{E}_{\sigma_1,\ldots,\sigma_N} \sup_{h\in\mathcal{H}} \sum_{i=1}^{N} \sigma_i \boldsymbol{\mu}^\top_{c(i)} \frac{h(\mathbf{x}_i)}{\|h(\mathbf{x}_i)\|_2}])^{1/3} + \beta(\frac{\ln(2/\delta)}{N})^{1/6})$$

Remark C.7. *As our loss promotes alignment of sample embeddings with their class prototypes on the hyperspherical space, the above Theorem implies that when such alignment holds, we can upper bound the intra-class variation with three main factors: the optimization error ϵ, the Rademacher complexity of the given neural network, and the estimation error $(\frac{\ln(2/\delta)}{N})^{1/6}$.*

C.3.1 Extension: From Low Variation to Low OOD Generalization Error

Ye et al. (2021) provide OOD generalization error bounds based on the notation of variation. Therefore, bounding intra-class variation is critical to bound OOD generalization error. For completeness, we reinstate the main results in Ye et al. (2021) below, which provide both OOD generalization error upper and lower bounds based on the variation w.r.t. the training domains. Interested readers shall refer to Ye et al. (2021) for more details and illustrations.

Definition C.8 (Expansion Function (Ye et al., 2021)). *We say a function $s : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+ \cup \{0, +\infty\}$ is an expansion function, iff the following properties hold: 1) $s(\cdot)$ is monotonically increasing and $s(x) \geq x, \forall x \geq 0$; 2) $\lim_{x \to 0^+} s(x) = s(0) = 0$.*

As it is impossible to generalize to an arbitrary distribution, characterizing the relation between $\mathcal{E}_{\text{avail}}$ and $\mathcal{E}_{\text{all}}$ is essential to formalize OOD generalization. Based on the notion of expansion function, the learnability of OOD generalization is defined as follows:

Definition C.9 (OOD-Learnability (Ye et al., 2021)). *Let Φ be the feature space and ρ be a distance metric on distributions. We say an OOD generalization problem from $\mathcal{E}_{\text{avail}}$ to $\mathcal{E}_{\text{all}}$ is* learnable *if there exists an expansion function $s(\cdot)$ and $\delta \geq 0$, such that: for all $\phi \in \Phi$[1] satisfying $\mathcal{I}_\rho(\phi, \mathcal{E}_{\text{avail}}) \geq \delta$, we have $s(\mathcal{V}_\rho(\phi, \mathcal{E}_{\text{avail}})) \geq \mathcal{V}_\rho(\phi, \mathcal{E}_{\text{all}})$. If such $s(\cdot)$ and δ exist, we further call this problem $(s(\cdot), \delta)$-learnable.*

For learnable OOD generalization problems, the following two theorems characterize OOD error upper and lower bounds based on variation.

[1] ϕ referred to as feature h in theoretical analysis.

Theorem C.10 (OOD Error Upper Bound (Ye et al., 2021)). *Suppose we have learned a classifier with loss function $\ell(\cdot,\cdot)$ such that $\forall e \in \mathcal{E}_{\text{all}}$ and $\forall y \in \mathcal{Y}$, $p_{h^e|Y^e}(h|y) \in L^2(\mathbb{R}^d)$. $h(\cdot) \in \mathbb{R}^d$ denotes the feature extractor. Denote the characteristic function of random variable $h^e|Y^e$ as $\hat{p}_{h^e|Y^e}(t|y) = \mathbb{E}[\exp\{i\langle t, h^e\rangle\}|Y^e = y]$. Assume the hypothetical space $\mathcal{F}$ satisfies the following regularity conditions that $\exists \alpha, M_1, M_2 > 0, \forall f \in \mathcal{F}, \forall e \in \mathcal{E}_{\text{all}}, y \in \mathcal{Y}$,*

$$\int_{h\in\mathbb{R}^d} p_{h^e|Y^e}(h|y)|h|^\alpha \mathrm{d}h \le M_1 \quad \text{and} \quad \int_{t\in\mathbb{R}^d} |\hat{p}_{h^e|Y^e}(t|y)||t|^\alpha \mathrm{d}t \le M_2. \tag{C.1}$$

If $(\mathcal{E}_{\text{avail}}, \mathcal{E}_{\text{all}})$ is $\Big(s(\cdot), \mathcal{I}^{inf}(h, \mathcal{E}_{\text{avail}})\Big)$-learnable under Φ with Total Variation ρ[2], then we have

$$\text{err}(f) \le O\Big(s\Big(\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}})\Big)^{\frac{\alpha^2}{(\alpha+d)^2}}\Big), \tag{C.2}$$

where $O(\cdot)$ depends on d, C, α, M_1, M_2.

Theorem C.11 (OOD Error Lower Bound (Ye et al., 2021)). *Consider 0-1 loss: $\ell(\hat{y}, y) = \mathbb{I}(\hat{y} \neq y)$. For any $\delta > 0$ and any expansion function satisfying 1) $s'_+(0) \triangleq \lim_{x\to 0^+} \frac{s(x)-s(0)}{x} \in (1, +\infty)$; 2) exists $k > 1, t > 0$, s.t. $kx \le s(x) < +\infty, x \in [0, t]$, there exists a constant C_0 and an OOD generalization problem $(\mathcal{E}_{\text{avail}}, \mathcal{E}_{\text{all}})$ that is $(s(\cdot), \delta)$-learnable under linear feature space Φ w.r.t symmetric KL-divergence ρ, s.t. $\forall \varepsilon \in [0, \frac{t}{2}]$, the optimal classifier f satisfying $\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}}) = \varepsilon$ will have the OOD generalization error lower bounded by*

$$\text{err}(f) \ge C_0 \cdot s(\mathcal{V}^{\text{sup}}(h, \mathcal{E}_{\text{avail}})). \tag{C.3}$$

C.4 Additional Experimental Details

Software and hardware. Our method is implemented with PyTorch 1.10. All experiments are conducted on NVIDIA GeForce RTX 2080 Ti GPUs for

[2]For two distribution $\mathbb{P}, \mathbb{Q}$ with probability density function p, q, $\rho(\mathbb{P}, \mathbb{Q}) = \frac{1}{2}\int_x |p(x) - q(x)|\mathrm{d}x$.

small to medium batch sizes and NVIDIA A100 and RTX A6000 GPUs for large batch sizes.

Architecture. In our experiments, we use ResNet-18 for CIFAR-10, ResNet-34 for ImageNet-100, ResNet-50 for PACS, VLCS, Office-Home and Terra Incognita. Following common practice in prior works (Khosla et al., 2020), we use a non-linear MLP projection head to obtain features in our experiments. The embedding dimension is 128 of the projection head for ImageNet-100. The projection head dimension is 512 for PACS, VLCS, Office-Home, and Terra Incognita.

Additional implementation details. In our experiments, we follow the common practice that initializing the network with ImageNet pre-trained weights for PACS, VLCS, Office-Home, and Terra Incognita. We then fine-tune the network for 50 epochs. For the large-scale experiments on ImageNet-100, we fine-tune ImageNet pre-trained ResNet-34 with our method for 10 epochs for computational efficiency. We set the temperature $\tau = 0.1$, prototype update factor $\alpha = 0.95$ as the default value. We use stochastic gradient descent with momentum 0.9, and weight decay 10^{-4}. The search distribution in our experiments for the learning rate hyperparameter is: lr $\in \{0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001, 0.00005\}$. The search space for the batch size is bs $\in \{32, 64\}$. The loss weight λ for balancing our loss function ($\mathcal{L} = \lambda \mathcal{L}_{\text{var}} + \mathcal{L}_{\text{sep}}$) is selected from $\lambda \in \{1.0, 2.0, 4.0\}$. For multi-source domain generalization, hard negatives can be incorporated by a simple modification to the denominator of the variation loss:

$$\mathcal{L}_{\text{var}} = -\frac{1}{N} \sum_{e \in \mathcal{E}_{\text{avail}}} \sum_{i=1}^{|\mathcal{D}^e|} \log \frac{\exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)} / \tau\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_j / \tau\right) + \sum_{j=1}^{N} \mathbb{I}(y_j \neq y_i, e_i = e_j) \exp\left(\mathbf{z}_i^\top \mathbf{z}_j / \tau\right)} \tag{C.4}$$

Details of datasets. We provide a detailed description of the datasets used in this work:

CIFAR-10 (Krizhevsky et al., 2009) is consist of $60,000$ color images with 10 classes. The training set has $50,000$ images and the test set has $10,000$ images.

ImageNet-100 is composed by randomly sampled 100 categories from ImageNet-1K.This dataset contains the following classes: n01498041, n01514859, n01582220, n01608432, n01616318, n01687978, n01776313, n01806567, n01833805, n01882714, n01910747, n01944390, n01985128, n02007558, n02071294, n02085620, n02114855, n02123045, n02128385, n02129165, n02129604, n02165456, n02190166, n02219486, n02226429, n02279972, n02317335, n02326432, n02342885, n02363005, n02391049, n02395406, n02403003, n02422699, n02442845, n02444819, n02480855, n02510455, n02640242, n02672831, n02687172, n02701002, n02730930, n02769748, n02782093, n02787622, n02793495, n02799071, n02802426, n02814860, n02840245, n02906734, n02948072, n02980441, n02999410, n03014705, n03028079, n03032252, n03125729, n03160309, n03179701, n03220513, n03249569, n03291819, n03384352, n03388043, n03450230, n03481172, n03594734, n03594945, n03627232, n03642806, n03649909, n03661043, n03676483, n03724870, n03733281, n03759954, n03761084, n03773504, n03804744, n03916031, n03938244, n04004767, n04026417, n04090263, n04133789, n04153751, n04296562, n04330267, n04371774, n04404412, n04465501, n04485082, n04507155, n04536866, n04579432, n04606251, n07714990, n07745940.

CIFAR-10-C is generated based on the previous literature (Hendrycks and Dietterich, 2019), applying different corruptions on CIFAR-10 data. The corruption types include gaussian noise, zoom blur, impulse noise, defocus blur, snow, brightness, contrast, elastic transform, fog, frost, gaussian blur, glass blur, JEPG compression, motion blur, pixelate, saturate, shot noise, spatter, and speckle noise.

ImageNet-100-C is algorithmically generated with Gaussian noise based on (Hendrycks and Dietterich, 2019) for the ImageNet-100 dataset.

PACS (Li et al., 2017a) is commonly used in OoD generalization. This

dataset contains $9,991$ examples of resolution 224×224 and four domains with different image styles, namely photo, art painting, cartoon, and sketch with seven categories.

VLCS (Gulrajani and Lopez-Paz, 2021) comprises four domains including Caltech101, LabelMe, SUN09, and VOC2007. It contains $10,729$ examples of resolution 224×224 and 5 classes.

Office-Home (Gulrajani and Lopez-Paz, 2021) contains four different domains: art, clipart, product, and real. This dataset comprises $15,588$ examples of resolution 224×224 and 65 classes.

Terra Incognita (Gulrajani and Lopez-Paz, 2021) comprises images of wild animals taken by cameras at four different locations: location100, location38, location43, and location46. This dataset contains $24,788$ examples of resolution 224×224 and 10 classes.

C.5 Detailed Results on CIFAR-10

In this section, we provide complete results of the different corruption types on CIFAR-10. In Table C.1, we evaluate HYPO under various common corruptions. Results suggest that HYPO achieves consistent improvement over the ERM baseline for all 19 different corruptions. We also compare our loss (HYPO) with more recent competitive algorithms: EQRM (Eastwood et al., 2022) and SharpDRO (Huang et al., 2023), on the CIFAR10-C dataset (Gaussian noise). The results on ResNet-18 are presented in Table C.13.

C.6 Additional Evaluations on Other OOD Generalization Tasks

In this section, we provide detailed results on more OOD generalization benchmarks, including Office-Home (Table C.3), VLCS (Table C.4), and

Table C.1: Main results for verifying OOD generalization performance on the 19 different covariate shifts datasets. We train on CIFAR-10 as ID, using CIFAR-10-C as the OOD test dataset. Acc. denotes the accuracy on the OOD test set.

Method	Corruptions	Acc.	Corruptions	Acc.	Corruptions	Acc.	Corruptions	Acc.
CE	Gaussian noise	78.09	Zoom blur	88.47	Impulse noise	83.60	Defocus blur	94.85
HYPO	Gaussian noise	85.21	Zoom blur	93.28	Impulse noise	87.54	Defocus blur	94.90
CE	Snow	90.19	Brightness	94.83	Contrast	94.11	Elastic transform	90.36
HYPO	Snow	91.10	Brightness	94.87	Contrast	94.53	Elastic transform	91.64
CE	Fog	94.45	Frost	90.33	Gaussian blur	94.85	Glass blur	56.99
HYPO	Fog	94.57	Frost	92.28	Gaussian blur	94.91	Glass blur	63.66
CE	JEPG compression	86.95	Motion blur	90.69	Pixelate	92.67	Saturate	92.86
HYPO	JEPG compression	89.24	Motion blur	93.07	Pixelate	93.95	Saturate	93.66
CE	Shot noise	85.86	Spatter	92.20	Speckle noise	85.66	**Average**	88.32
HYPO	Shot noise	89.87	Spatter	92.46	Speckle noise	89.94	**Average**	**90.56**

Terra Incognita (Table C.5). We observe that our approach achieves strong performance on these benchmarks. We compare our method with a collection of OOD generalization baselines such as `IRM` (Arjovsky et al., 2019), `DANN` (Ganin et al., 2016), `CDANN` (Li et al., 2018c), `GroupDRO` (Sagawa et al., 2019), `MTL` (Blanchard et al., 2021), `I-Mixup` (Zhang et al., 2018), `MMD` (Li et al., 2018b), `VREx` (Krueger et al., 2021), `MLDG` (Li et al., 2018a), `ARM` (Zhang et al., 2021a), `RSC` (Huang et al., 2020), `Mixstyle` (Zhou et al., 2021a), `ERM` (Vapnik, 1999), `CORAL` (Sun and Saenko, 2016), `SagNet` (Nam et al., 2021), `SelfReg` (Kim et al., 2021a), `GVRT` Min et al. (2022), `VNE` (Kim et al., 2023a). These methods are all loss-based and optimized using standard SGD. On the Office-Home, our method achieves an improved OOD generalization performance of **1.6**% compared to a competitive baseline (Sun and Saenko, 2016).

We also conduct experiments coupling with SWAD and achieve superior performance on OOD generalization. As shown in Table C.6, Table C.7, Table C.8, our method consistently establish superior results on different benchmarks including VLCS, Office-Home, Terra Incognita, showing the effectiveness of our method via hyperspherical learning.

Algorithm	Art painting	Cartoon	Photo	Sketch	Average Acc. (%)
IRM (Arjovsky et al., 2019)	84.8	76.4	96.7	76.1	83.5
DANN (Ganin et al., 2016)	86.4	77.4	97.3	73.5	83.7
CDANN (Li et al., 2018c)	84.6	75.5	96.8	73.5	82.6
GroupDRO (Sagawa et al., 2019)	83.5	79.1	96.7	78.3	84.4
MTL (Blanchard et al., 2021)	87.5	77.1	96.4	77.3	84.6
I-Mixup (Wang et al., 2020)	86.1	78.9	97.6	75.8	84.6
MMD (Li et al., 2018b)	86.1	79.4	96.6	76.5	84.7
VREx (Krueger et al., 2021)	86.0	79.1	96.9	77.7	84.9
MLDG (Li et al., 2018a)	85.5	80.1	97.4	76.6	84.9
ARM (Zhang et al., 2021a)	86.8	76.8	97.4	79.3	85.1
RSC (Huang et al., 2020)	85.4	79.7	97.6	78.2	85.2
Mixstyle (Zhou et al., 2021a)	86.8	79.0	96.6	78.5	85.2
ERM (Vapnik, 1999)	84.7	80.8	97.2	79.3	85.5
CORAL (Sun and Saenko, 2016)	88.3	80.0	97.5	78.8	86.2
SagNet (Nam et al., 2021)	87.4	80.7	97.1	80.0	86.3
SelfReg (Kim et al., 2021a)	87.9	79.4	96.8	78.3	85.6
GVRT Min et al. (2022)	87.9	78.4	98.2	75.7	85.1
VNE (Kim et al., 2023a)	88.6	79.9	96.7	82.3	86.9
HYPO (Ours)	87.2	82.3	98.0	84.5	**88.0**

Table C.2: Comparison with state-of-the-art methods on the PACS benchmark. All methods are trained on ResNet-50. The model selection is based on a training domain validation set. To isolate the effect of loss functions, all methods are optimized using standard SGD. *Results based on retraining of PCL with SGD using official implementation. PCL with SWAD optimization is further compared in Table 4.2. We run HYPO 3 times and report the average and std. $\pm x$ denotes the standard error, rounded to the first decimal point.

C.7 Experiments on ImageNet-100 and ImageNet-100-C

In this section, we provide additional large-scale results on the ImageNet benchmark. We use ImageNet-100 as the in-distribution data and use ImageNet-100-C with Gaussian noise as OOD data in the experiments. In Figure C.1, we observe our method improves OOD accuracy compared to the ERM baseline.

Algorithm	Art	Clipart	Product	Real World	Average Acc. (%)
IRM (Arjovsky et al., 2019)	58.9	52.2	72.1	74.0	64.3
DANN (Ganin et al., 2016)	59.9	53.0	73.6	76.9	65.9
CDANN (Li et al., 2018c)	61.5	50.4	74.4	76.6	65.7
GroupDRO (Sagawa et al., 2019)	60.4	52.7	75.0	76.0	66.0
MTL (Blanchard et al., 2021)	61.5	52.4	74.9	76.8	66.4
I-Mixup (Wang et al., 2020)	62.4	54.8	76.9	78.3	68.1
MMD (Li et al., 2018b)	60.4	53.3	74.3	77.4	66.4
VREx (Krueger et al., 2021)	60.7	53.0	75.3	76.6	66.4
MLDG (Li et al., 2018a)	61.5	53.2	75.0	77.5	66.8
ARM (Zhang et al., 2021a)	58.9	51.0	74.1	75.2	64.8
RSC (Huang et al., 2020)	60.7	51.4	74.8	75.1	65.5
Mixstyle (Zhou et al., 2021a)	51.1	53.2	68.2	69.2	60.4
ERM (Vapnik, 1999)	63.1	51.9	77.2	78.1	67.6
CORAL (Sun and Saenko, 2016)	65.3	54.4	76.5	78.4	68.7
SagNet (Nam et al., 2021)	63.4	54.8	75.8	78.3	68.1
SelfReg (Kim et al., 2021a)	63.6	53.1	76.9	78.1	67.9
GVRT Min et al. (2022)	66.3	55.8	78.2	80.4	70.1
VNE (Kim et al., 2023a)	60.4	54.7	73.7	74.7	65.9
HYPO (Ours)	68.3	57.9	79.0	81.4	**71.7**

Table C.3: Comparison with state-of-the-art methods on the Office-Home benchmark. All methods are trained on ResNet-50. The model selection is based on a training domain validation set. To isolate the effect of loss functions, all methods are optimized using standard SGD.

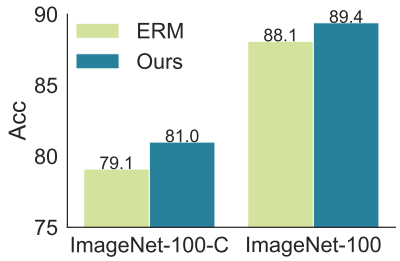


Figure C.1: Experiments on ImageNet-100 (ID) vs. ImageNet-100-C (OOD).

C.8 Ablation of Different Loss Terms

Ablations on separation loss. In Table C.9, we demonstrate the effectiveness of the first loss term (variation) empirically. We compare the OOD performance of our method (with separation loss) vs. our method (without separation loss). We observe our method without separation loss term can still achieve strong OOD accuracy–average 87.2% on the

Algorithm	Caltech101	LabelMe	SUN09	VOC2007	Average Acc. (%)
IRM (Arjovsky et al., 2019)	98.6	64.9	73.4	77.3	78.6
DANN (Ganin et al., 2016)	99.0	65.1	73.1	77.2	78.6
CDANN (Li et al., 2018c)	97.1	65.1	70.7	77.1	77.5
GroupDRO (Sagawa et al., 2019)	97.3	63.4	69.5	76.7	76.7
MTL (Blanchard et al., 2021)	97.8	64.3	71.5	75.3	77.2
I-Mixup (Wang et al., 2020)	98.3	64.8	72.1	74.3	77.4
MMD (Li et al., 2018b)	97.7	64.0	72.8	75.3	77.5
VREx (Krueger et al., 2021)	98.4	64.4	74.1	76.2	78.3
MLDG (Li et al., 2018a)	97.4	65.2	71.0	75.3	77.2
ARM (Zhang et al., 2021a)	98.7	63.6	71.3	76.7	77.6
RSC (Huang et al., 2020)	97.9	62.5	72.3	75.6	77.1
Mixstyle (Zhou et al., 2021a)	98.6	64.5	72.6	75.7	77.9
ERM (Vapnik, 1999)	97.7	64.3	73.4	74.6	77.5
CORAL (Sun and Saenko, 2016)	98.3	66.1	73.4	77.5	78.8
SagNet (Nam et al., 2021)	97.9	64.5	71.4	77.5	77.8
SelfReg (Kim et al., 2021a)	96.7	65.2	73.1	76.2	77.8
GVRT Min et al. (2022)	98.8	64.0	75.2	77.9	79.0
VNE (Kim et al., 2023a)	97.5	65.9	70.4	78.4	78.1
HYPO (Ours)	98.1	65.3	73.1	76.3	78.2

Table C.4: Comparison with state-of-the-art methods on the VLCS benchmark. All methods are trained on ResNet-50. The model selection is based on a training domain validation set. To isolate the effect of loss functions, all methods are optimized using standard SGD.

PACS dataset. This ablation study indicates the first term (variation) of our method plays a more important role in practice, which aligns with our theoretical analysis in Section 4.6 and Appendix C.3.

Ablations on hard negative pairs. To verify that hard negative pairs help multiple training domains, we conduct ablation by comparing ours (with hard negative pairs) vs. ours (without hard negative pairs). We can see in Table C.10 that our method with hard negative pairs improves the average OOD performance by 0.4% on the PACS dataset. Therefore, we empirically demonstrate that emphasizing hard negative pairs leads to better performance for multi-source domain generalization tasks.

Comparing EMA update and learnable prototype. We conduct an ablation study on the prototype update rule. Specifically, we compare our

Algorithm	Location100	Location38	Location43	Location46	Average Acc. (%)
IRM (Arjovsky et al., 2019)	54.6	39.8	56.2	39.6	47.6
DANN (Ganin et al., 2016)	51.1	40.6	57.4	37.7	46.7
CDANN (Li et al., 2018c)	47.0	41.3	54.9	39.8	45.8
GroupDRO (Sagawa et al., 2019)	41.2	38.6	56.7	36.4	43.2
MTL (Blanchard et al., 2021)	49.3	39.6	55.6	37.8	45.6
I-Mixup (Wang et al., 2020)	59.6	42.2	55.9	33.9	47.9
MMD (Li et al., 2018b)	41.9	34.8	57.0	35.2	42.2
VREx (Krueger et al., 2021)	48.2	41.7	56.8	38.7	46.4
MLDG (Li et al., 2018a)	54.2	44.3	55.6	36.9	47.8
ARM (Zhang et al., 2021a)	49.3	38.3	55.8	38.7	45.5
RSC (Huang et al., 2020)	50.2	39.2	56.3	40.8	46.6
Mixstyle (Zhou et al., 2021a)	54.3	34.1	55.9	31.7	44.0
ERM (Vapnik, 1999)	49.8	42.1	56.9	35.7	46.1
CORAL (Sun and Saenko, 2016)	51.6	42.2	57.0	39.8	47.7
SagNet (Nam et al., 2021)	53.0	43.0	57.9	40.4	48.6
SelfReg (Kim et al., 2021a)	48.8	41.3	57.3	40.6	47.0
GVRT Min et al. (2022)	53.9	41.8	58.2	38.0	48.0
VNE (Kim et al., 2023a)	58.1	42.9	58.1	43.5	50.6
HYPO (Ours)	58.8	46.6	58.7	42.7	**51.7**

Table C.5: Comparison with state-of-the-art methods on the Terra Incognita benchmark. All methods are trained on ResNet-50. The model selection is based on a training domain validation set. To isolate the effect of loss functions, all methods are optimized using standard SGD.

Algorithm	Art	Clipart	Product	Real World	Average Acc. (%)
SWAD (Cha et al., 2021)	66.1	57.7	78.4	80.2	70.6
PCL+SWAD (Yao et al., 2022)	67.3	59.9	78.7	80.7	71.6
VNE+SWAD (Kim et al., 2023a)	66.6	58.6	78.9	80.5	71.1
HYPO+SWAD (Ours)	68.4	61.3	81.8	82.4	**73.5**

Table C.6: Results with SWAD-based optimization on the Office-Home benchmark.

Algorithm	Caltech101	LabelMe	SUN09	VOC2007	Average Acc. (%)
SWAD (Cha et al., 2021)	98.8	63.3	75.3	79.2	79.1
PCL+SWAD (Yao et al., 2022)	95.8	65.4	74.3	76.2	77.9
VNE+SWAD (Kim et al., 2023a)	99.2	63.7	74.4	81.6	79.7
HYPO+SWAD (Ours)	98.9	67.8	74.3	77.7	**79.7**

Table C.7: Rresults with SWAD-based optimization on the VLCS benchmark.

method with exponential-moving-average (EMA) (Li et al., 2020b; Wang et al., 2022a; Ming et al., 2023) prototype update versus learnable pro-

Algorithm	Location100	Location38	Location43	Location46	Average Acc. (%)
SWAD (Cha et al., 2021)	55.4	44.9	59.7	39.9	50.0
PCL+SWAD (Yao et al., 2022)	58.7	46.3	60.0	43.6	52.1
VNE+SWAD (Kim et al., 2023a)	59.9	45.5	59.6	41.9	51.7
HYPO+SWAD (Ours)	56.8	61.3	54.0	53.2	**56.3**

Table C.8: Results with SWAD-based optimization on the Terra Incognita benchmark.

Algorithm	Art painting	Cartoon	Photo	Sketch	Average Acc. (%)
Ours (w/o separation loss)	86.2	81.2	97.8	83.6	87.2
Ours (w separation loss)	87.2	82.3	98.0	84.5	**88.0**

Table C.9: Ablations on separation loss term.

Algorithm	Art painting	Cartoon	Photo	Sketch	Average Acc. (%)
Ours (w/o hard negative pairs)	87.8	82.9	98.2	81.4	87.6
Ours (w hard negative pairs)	87.2	82.3	98.0	84.5	**88.0**

Table C.10: Ablation on hard negative pairs. OOD generalization performance on the PACS dataset.

totypes (LP). The results on PACS are summarized in Table C.11. We observe our method with EMA achieves better average OOD accuracy 88.0% compared to learnable prototype update rules 86.7%. We empirically verify EMA-style method is a suitable prototype updating rule to facilitate gradient-based prototype update in practice.

Algorithm	Art painting	Cartoon	Photo	Sketch	Average Acc. (%)
Ours (LP)	88.0	80.7	97.5	80.7	86.7
Ours (EMA)	87.2	82.3	98.0	84.5	**88.0**

Table C.11: Ablation on prototype update rules. Comparing EMA update and learnable prototype (LP) on the PACS benchmark.

Quantitative verification of the ϵ factor in Theorem 4.6. We calculate the average intra-class variation over data from all environments $\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\mu}_{c(j)}^{\top}\mathbf{z}_j$ (Theorem 4.6) models trained with HYPO. Then we obtain $\hat{\epsilon} := 1 -$

$\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\mu}_{c(j)}^{\top}\mathbf{z}_j$. We evaluated PACS, VLCS, and OfficeHome and summarized the results in Table C.12. We observe that training with HYPO significantly reduces the average intra-class variation, resulting in a small epsilon ($\hat{\epsilon} < 0.1$) in practice. This suggests that the first term $O(\epsilon^{\frac{1}{3}})$ in Theorem 4.6 is indeed small for models trained with HYPO.

Dataset	$\hat{\epsilon}$
PACS	0.06
VLCS	0.08
OfficeHome	0.09

Table C.12: Empirical verification of intra-class variation in Theorem 4.6.

Method	OOD Acc. (%)
EQRM (Eastwood et al., 2022)	77.06
SharpDRO (Huang et al., 2023)	81.61
HYPO (ours)	85.21

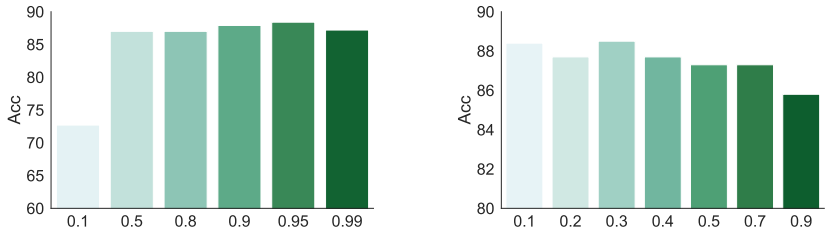
Table C.13: Comparison with more recent competitive baselines. Models are trained on CIFAR-10 using ResNet-18 and tested on CIFAR10-C (Gaussian noise).

C.9 Analyzing the Effect of τ and α

In Figure C.2a, we present the OOD generalization performance by adjusting the prototype update factor α. The results are averaged over four domains on the PACS dataset. We observe the generalization performance is competitive across a wide range of α. In particular, our method achieves the best performance when $\alpha = 0.95$ on the PACS dataset with an average of 88.0% OOD accuracy.

We show in Figure C.2b the OOD generalization performance by varying the temperature parameter τ. The results are averaged over four different domains on PACS. We observe a relative smaller τ results in stronger

OOD performance while too large τ (e.g., 0.9) would lead to degraded performance.



(a) Moving average α (b) Temperature τ

Figure C.2: Ablation on (a) prototype update discount factor α and (b) temperature τ. The results are averaged over four domains on the PACS dataset.

C.10 Theoretical Insights on Inter-class Separation

To gain theoretical insights into inter-class separation, we focus on the learned prototype embeddings of the separation loss with a simplified setting where we directly optimize the embedding vectors.

Definition C.12. (*Simplex ETF (Sustik et al., 2007)*). *A set of vectors* $\{\boldsymbol{\mu}_i\}_{i=1}^C$ *in* $\mathbb{R}^d$ *forms a simplex Equiangular Tight Frame (ETF) if* $\|\boldsymbol{\mu}_i\| = 1$ *for* $\forall i \in [C]$ *and* $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = -1/(C-1)$ *for* $\forall i \neq j$.

Next, we will characterize the optimal solution for the separation loss defined as:

$$\mathcal{L}_{\text{sep}} = \underbrace{\frac{1}{C}\sum_{i=1}^{C} \log \frac{1}{C-1} \sum_{j\neq i, j=1}^{C} \exp\left(\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau\right)}_{\uparrow \text{ separation}} := \frac{1}{C}\sum_{i=1}^{C} \log \mathcal{L}_{\text{sep}}(i)$$

Lemma C.13. (*Optimal solution of the separation loss*) *Assume the number of classes* $C \leq d+1$, $\mathcal{L}_{sep}$ *is minimized when the learned class prototypes* $\{\boldsymbol{\mu}_i\}_{i=1}^C$ *form a simplex ETF.*

Proof.

$$\mathcal{L}_{\text{sep}}(i) = \frac{1}{C-1} \sum_{j\neq i, j=1}^{C} \exp\left(\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau\right) \tag{C.5}$$

$$\geq \exp\left(\frac{1}{C-1} \sum_{j\neq i, j=1}^{C} \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau\right) \tag{C.6}$$

$$= \exp\left(\frac{\boldsymbol{\mu}_i^\top \boldsymbol{\mu} - \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_i}{\tau(C-1)}\right) \tag{C.7}$$

$$= \exp\left(\frac{\boldsymbol{\mu}_i^\top \boldsymbol{\mu} - 1}{\tau(C-1)}\right) \tag{C.8}$$

where we define $\boldsymbol{\mu} = \sum_{i=1}^{C} \boldsymbol{\mu}_i$ and (C.6) follows Jensen's inequality. Therefore, we have

$$\begin{aligned}
\mathcal{L}_{\text{sep}} &= \frac{1}{C}\sum_{i=1}^{C} \log \mathcal{L}_{\text{sep}}(i) \\
&\geq \frac{1}{C}\sum_{i=1}^{C} \log \exp\left(\frac{\boldsymbol{\mu}_i^\top \boldsymbol{\mu} - 1}{\tau(C-1)}\right) \\
&= \frac{1}{\tau C(C-1)}\sum_{i=1}^{C}(\boldsymbol{\mu}_i^\top \boldsymbol{\mu} - 1) \\
&= \frac{1}{\tau C(C-1)}\boldsymbol{\mu}^\top \boldsymbol{\mu} - \frac{1}{\tau(C-1)}
\end{aligned}$$

It suffices to consider the following optimization problem,

$$\begin{aligned}
\text{minimize} \quad & \mathcal{L}_1 = \boldsymbol{\mu}^\top \boldsymbol{\mu} \\
\text{subject to} \quad & \|\boldsymbol{\mu}_i\| = 1 \quad \forall i \in [C]
\end{aligned}$$

where $\boldsymbol{\mu}^\top \boldsymbol{\mu} = (\sum_{i=1}^{C} \boldsymbol{\mu}_i)^\top (\sum_{i=1}^{C} \boldsymbol{\mu}_i) = \sum_{i=1}^{C} \sum_{j \neq i} \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j + C$

However, the problem is non-convex. We first consider a convex relaxation and show that the optimal solution to the original problem is the same as the convex problem below,

$$\begin{aligned}
\text{minimize} \quad & \mathcal{L}_2 = \sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_j \\
\text{subject to} \quad & \|\boldsymbol{\mu}_i\| \leq 1 \quad \forall i \in [C]
\end{aligned}$$

Note that the optimal solution $\mathcal{L}_1^* \geq \mathcal{L}_2^*$. Next, we can obtain the Lagrangian form:

$$\mathcal{L}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C, \lambda_1, \ldots, \lambda_C) = \sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_j + \sum_{i=1}^{C} \lambda_i (\|\boldsymbol{\mu}_i\|^2 - 1)$$

where λ_i are Lagrange multipliers. Taking the gradient of the La-

grangian with respect to $\boldsymbol{\mu}_k$ and setting it to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = 2\sum_{i\neq k}^{C} \boldsymbol{\mu}_i + 2\lambda_k \boldsymbol{\mu}_k = 0$$

Simplifying the equation, we have:

$$\boldsymbol{\mu} = \boldsymbol{\mu}_k(1-\lambda_k)$$

Therefore, the optimal solution satisfies that (1) either all feature vectors are co-linear (*i.e.* $\boldsymbol{\mu}_k = \alpha_k \boldsymbol{v}$ for some vector $\boldsymbol{v} \in \mathbb{R}^d \; \forall k \in [C]$) or (2) the sum $\boldsymbol{\mu} = \sum_{i=1}^{C} \boldsymbol{\mu}_i = \mathbf{0}$. The Karush-Kuhn-Tucker (KKT) conditions are:

$$\begin{aligned} \boldsymbol{\mu}_k(1-\lambda_k) &= \mathbf{0} \quad \forall k \\ \lambda_k(\|\boldsymbol{\mu}_k\|^2 - 1) &= 0 \quad \forall k \\ \lambda_k &\geq 0 \quad \forall k \\ \|\boldsymbol{\mu}_k\| &\leq 1 \quad \forall k \end{aligned}$$

When the learned class prototypes $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ form a simplex ETF, $\boldsymbol{\mu}_k^\top \boldsymbol{\mu} = 1 + \sum_{i\neq k} \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_k = 1 - \frac{C-1}{C-1} = 0$. Therefore, we have $\boldsymbol{\mu} = \mathbf{0}$, $\lambda_k = 1$, $\|\boldsymbol{\mu}_k\| = 1$ and KKT conditions are satisfied. Particularly, $\|\boldsymbol{\mu}_k\| = 1$ means that all vectors are on the unit hypersphere and thus the solution is also optimal for the original problem $\mathcal{L}_1$. The solution is optimal for $\mathcal{L}_{\text{sep}}$ as Jensen's inequality (C.6) becomes equality when $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ form a simplex ETF. The above analysis provides insights on why $\mathcal{L}_{\text{sep}}$ promotes inter-class separation.

Appendix D

Appendix for Delving into Out-of-Distribution Detection with Vision-Language Representations

D.1 Theoretical Justification: Softmax Scaling for Zero-Shot OOD Detection

In this section, we provide the proof for Theorem 5.2 in Section 5.3, which states the benefits of applying softmax scaling to inner products for OOD detection. We begin with a review of notations.

Notations. We denote the text encoder of a pre-trained CLIP-like model as $\mathcal{T} : t \to \mathbb{R}^d$ and the image encoder $\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$. For a given task with label set $\mathcal{Y}_{\text{in}} = \{y_1, y_2, ..., y_K\}$, we construct a collection of concept vectors $\mathcal{T}(t_i)$. For a given input $\mathbf{x}'$, we denote the cosine similarity *w.r.t.* concept vectors as $s_i(\mathbf{x}') = \frac{\mathcal{I}(\mathbf{x}')\cdot\mathcal{T}(t_i)}{\|\mathcal{I}(\mathbf{x}')\|\cdot\|\mathcal{T}(t_i)\|} \forall i \in [K]$, where $|s_i(\mathbf{x}')| \leq B$ for

all $\mathbf{x}' \in \mathcal{X}$.[1]. We define the maximum concept matching (MCM) score as: $S_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) = \max_{i \in [K]} \frac{e^{s_i(\mathbf{x}')/\tau}}{\sum_{j=1}^{K} e^{s_j(\mathbf{x}')/\tau}}$. We denote the maximum inner product without applying softmax scaling as $S^{\text{wo}}_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) = \max_{i \in [K]} s_i(\mathbf{x}')$. By convention, the OOD detection functions are given by:

$$G^{\text{wo}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) = \begin{cases} 1 & S^{\text{wo}}_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) \geq \lambda^{\text{wo}} \\ 0 & S^{\text{wo}}_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) < \lambda^{\text{wo}} \end{cases},$$

$$G(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) = \begin{cases} 1 & S_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) \geq \lambda \\ 0 & S_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) < \lambda \end{cases},$$

Remarks: By convention, 1 represents the positive class (ID) and 0 indicates OOD; λ and λ^{wo} are typically chosen such that the true positive rate is at 95%.

For convenience, we paste the assumptions and the theorem in Section 5.3 below,

Assumption D.1. *Let $z := \mathbb{1}\{y \in \mathcal{Y}_{in}\}$ and $Q_{\mathbf{x}}$ denotes the out-of-distribution $\mathbb{P}_{\mathbf{x}|z=0}$ (marginal distribution of $\mathbf{x}$ conditioned on $z = 0$). Assume $\exists\, \delta > 0$ such that*

$$Q_{\mathbf{x}}\left(\frac{1}{K-1}\sum_{i \neq \hat{y}}[s_{\hat{y}_2}(\mathbf{x}) - s_i(\mathbf{x})] < \delta\right) = 1,$$

where $\hat{y} := \text{argmax}_{i \in [K]} s_i(\mathbf{x})$ and $\hat{y}_2 := \text{argmax}_{i \neq \hat{y}, i \in [K]} s_i(\mathbf{x})$ denote the indices of the largest and second largest cosine similarities for an OOD input $\mathbf{x}$.

Theorem D.2. *Given a pre-trained CLIP-like model $(\mathcal{T}, \mathcal{I})$ and a task with label set $\mathcal{Y}_{in} = \{y_1, y_2, ..., y_K\}$. If $Q_{\mathbf{x}}$ satisfy Assumption D.1, Then there exists a*

[1] In practice, we observe that $s_i \in [0.1, 0.3]$ for CLIP with high probability.

constant $T = \frac{\lambda(K-1)\left(\lambda^{wo}+\delta-s_{\hat{y}_2}\right)}{K\lambda-1}$ *such that for any temperature* $\tau > T$*, we have:*

$$\mathrm{FPR}(\tau,\lambda) \leq \mathrm{FPR}^{wo}(\lambda^{wo}),$$

where $\mathrm{FPR}(\tau,\lambda)$ *is the false positive rate based on softmax scaling with temperature* τ *and threshold* λ*;* $\mathrm{FPR}^{wo}(\lambda^{wo})$ *is the false positive rate without softmax scaling based on threshold* λ^{wo}*. This suggests that applying softmax scaling with temperature results in superior OOD detection performance compared to without softmax scaling.*

Proof. By definition, we express the false positive rate $\mathrm{FPR}(\tau,\lambda)$ as follows,

$$\begin{aligned}
\mathrm{FPR}(\tau,\lambda) &= \mathbb{P}\left(G(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) = 1 \mid z = 0\right) \\
&= Q_{\mathbf{x}'}\left(G(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) = 1\right) \\
&= Q_{\mathbf{x}'}\left(p_{\hat{y}}\left(\mathbf{x}';\tau\right) > \lambda\right) \\
&= Q_{\mathbf{x}'}\left(\frac{e^{s_{\hat{y}}(\mathbf{x}')/\tau}}{\sum_{j=1}^{K} e^{s_j(\mathbf{x}')/\tau}} > \lambda\right) \\
&= Q_{\mathbf{x}'}\left(\frac{1}{\lambda} > \sum_{i=1}^{K} \exp\left(\frac{s_i(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')}{\tau}\right)\right)
\end{aligned}$$

By inequality $e^x \geq 1 + x$, we have,

$$Q_{\mathbf{x}'}\left(\frac{1}{\lambda} > \sum_{i=1}^{K} \exp\left(\frac{s_i(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')}{\tau}\right)\right) \leq Q_{\mathbf{x}'}\left(\frac{1}{\lambda} > \sum_{i=1}^{K}\left[1 + \frac{s_i(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')}{\tau}\right]\right)$$

This indicates

$$\begin{aligned}
&Q_{\mathbf{x}'}\left(\frac{1}{\lambda} > \sum_{i=1}^{K} \exp\left(\frac{s_i(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')}{\tau}\right)\right) \leq Q_{\mathbf{x}'}\left(\frac{1}{\lambda} > \sum_{i=1}^{K}\left[1 + \frac{s_i(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')}{\tau}\right]\right) \\
&= Q_{\mathbf{x}'}\left(\sum_{i=1}^{K}\left(s_{\hat{y}}(\mathbf{x}') - s_i(\mathbf{x}')\right) > \left(K - \frac{1}{\lambda}\right)\tau\right)
\end{aligned}$$

Since

$$\begin{aligned}\sum_{i=1}^{K}\left(s_{\hat{y}}(\mathbf{x}')-s_i(\mathbf{x}')\right) &= \sum_{i\neq\hat{y}}\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')+s_{\hat{y}_2}(\mathbf{x}')-s_i(\mathbf{x}')\right)\\ &= \sum_{i\neq\hat{y}}\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')\right)+\sum_{i\neq\hat{y}}\left(s_{\hat{y}_2}(\mathbf{x}')-s_i(\mathbf{x}')\right)\\ &= (K-1)\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')\right)+\sum_{i\neq\hat{y}}\left(s_{\hat{y}_2}(\mathbf{x}')-s_i(\mathbf{x}')\right)\end{aligned}$$

By Assumption 5.1, we have

$$Q_{\mathbf{x}'}\left(\sum_{i=1}^{K}\left(s_{\hat{y}}(\mathbf{x}')-s_i(\mathbf{x}')\right)<(K-1)\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')\right)+(K-1)\delta\right)=1.$$

Therefore,

$$\begin{aligned}Q_{\mathbf{x}'}\left(\sum_{i=1}^{K}\left(s_{\hat{y}}(\mathbf{x}')-s_i(\mathbf{x}')\right)>\left(K-\frac{1}{\lambda}\right)\tau\right) &\leq Q_{\mathbf{x}'}\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')>-\delta_2+\left(K-\frac{1}{\lambda}\right)\frac{\tau}{K-1}\right)\\ &= Q_{\mathbf{x}'}\left(s_{\hat{y}}(\mathbf{x}')-s_{\hat{y}_2}(\mathbf{x}')>-\delta_2+\lambda'\right)\\ &= Q_{\mathbf{x}'}\left(s_{\hat{y}}(\mathbf{x}')>s_{\hat{y}_2}(\mathbf{x}')-\delta_2+\lambda'\right),\end{aligned}$$

where $\lambda'=\left(K-\frac{1}{\lambda}\right)\frac{\tau}{K-1}$ is a monotonic function of λ (*i.e.*, minimizing false positive rate *w.r.t.* λ is equivalent to minimizing *w.r.t.* λ'.)

For $\tau>0$, we can rewrite the MCM score as

$$\begin{aligned}S_{\text{MCM}}(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I}) &= \max_{i\in[K]}\frac{e^{s_i(\mathbf{x}')/\tau}}{\sum_{j=1}^{K}e^{s_j(\mathbf{x}')/\tau}}=\frac{e^{s_{\hat{y}}(\mathbf{x}')/\tau}}{\sum_{j=1}^{K}e^{s_j(\mathbf{x}')/\tau}}\\ &= \frac{1}{1+\sum_{j=1,j\neq\hat{y}}^{K}e^{\left(s_j(\mathbf{x}')-s_{\hat{y}}(\mathbf{x}')\right)/\tau}}\end{aligned}$$

As $\hat{y}:=\text{argmax}_{i\in[K]}s_i(\mathbf{x})$, $s_j(\mathbf{x}')-s_{\hat{y}}(\mathbf{x}')\leq 0$, $S_{\text{MCM}}(\mathbf{x}';\mathcal{Y}_{\text{in}},\mathcal{T},\mathcal{I})$ is a mono-

tonically decreasing function of τ, we have:

$$S_{\text{MCM}}(\mathbf{x}'; \mathcal{Y}_{\text{in}}, \mathcal{T}, \mathcal{I}) > \lim_{\tau \to \infty} \frac{1}{1 + \sum_{j=1, j \neq \hat{y}}^{K} e^{\left(s_j(\mathbf{x}') - s_{\hat{y}}(\mathbf{x}')\right)/\tau}} = \frac{1}{K}$$

Therefore by the definition of λ, we have $\lambda > \frac{1}{K}$, $\lambda' = \left(K - \frac{1}{\lambda}\right) \frac{\tau}{K-1} > 0$

For moderately large $\tau > T$ where $T = \frac{\lambda(K-1)\left(\lambda^{\text{wo}} + \delta - s_{\hat{y}_2}\right)}{K\lambda - 1}$, we always have $s_{\hat{y}_2}(\mathbf{x}') - \delta + \lambda' > \lambda^{\text{wo}}$. Therefore, we obtain the following inequality,

$$\text{FPR}(\tau, \lambda) \leq Q_{\mathbf{x}'}\left(s_{\hat{y}}(\mathbf{x}') > s_{\hat{y}_2}(\mathbf{x}') - \delta_2 + \lambda'\right) \leq Q_{\mathbf{x}'}\left(s_{\hat{y}}(\mathbf{x}') > \lambda^{\text{wo}}\right) := \text{FPR}^{\text{wo}}(\lambda^{\text{wo}}),$$

which means that the FPR without softmax scaling is larger than that with softmax scaling and a moderately large temperature. We show in Section 5.5 that the bound is indeed satisfied in practice with a large-scale ID dataset. □

D.2 Experimental Details

D.2.1 Software and Hardware

All methods are implemented in Pytorch 1.10. We run all OOD detection experiments on NVIDIA GeForce RTX-2080Ti GPU and use NVIDIA A100 GPU for fine-tuning CLIP and ViT.

D.2.2 Hyperparameters

The only hyperparameter in MCM is the (temperature) scaling factor τ. We use $\tau = 1$ by default unless otherwise specified. Our experiments suggest that MCM is insensitive to the scaling factor, where τ in a wide range of $[0.5, 100]$ shares similar performance.

D.2.3 Datasets

ImageNet-10 We create ImageNet-10 that mimics the class distribution of CIFAR-10 but with high-resolution images. It contains the following categories (with class ID): warplane (n04552348), sports car (n04285008), brambling bird, (n01530575), Siamese cat (n02123597), antelope (n02422699), Swiss mountain dog (n02107574), bull frog (n01641577), garbage truck (n03417042), horse (n02389026), container ship (n03095699).

ImageNet-20 For hard OOD evaluation with realistic datasets, we curate ImageNet-20, which consists of 20 classes semantically similar to ImageNet-10 (*e.g.*, dog (ID) vs. wolf (OOD)). The categories are selected based on the distance in the WordNet synsets (Miller, 1995). Specifically, it contains the following categories: sailboat (n04147183), canoe (n02951358), balloon (n02782093), tank (n04389033), missile (n03773504), bullet train (n02917067), starfish (n02317335), spotted salamander (n01632458), common newt (n01630670), zebra (n01631663), frilled lizard (n02391049), green lizard (n01693334), African crocodile (n01697457), Arctic fox (n02120079),

timber wolf (n02114367), brown bear (n02132136), moped (n03785016), steam locomotive (n04310018), space shuttle (n04266014), snowmobile (n04252077).

We hope the above two datasets will help future research on large-scale hard OOD detection. We provide a script for generating the datasets at https://github.com/deeplearning-wisc/MCM.

ImageNet-100 We randomly sample 100 classes from ImageNet-1k to curate ImageNet-100. To facilitate reproducibility, the script for generating the dataset and the class list are provided at https://github.com/deeplearning-wisc/MCM.

Conventional (non-spurious) OOD datasets Huang *et al.* (Huang and Li, 2021) curate a diverse collection of subsets from iNaturalist (Van Horn et al., 2018), SUN (Xiao et al., 2010), Places (Zhou et al., 2017), and Texture (Cimpoi et al., 2014) as large-scale OOD datasets for ImageNet-1k, where the classes of the test sets do not overlap with ImageNet-1k. We provide a brief introduction to each dataset as follows.

iNaturalist contains images in the natural world (Van Horn et al., 2018). It has 13 super-categories and 5,089 sub-categories covering plants, insects, birds, mammals, and so on. We use the subset that contains 110 plant classes not overlapping with ImageNet-1k.

SUN stands for the Scene UNderstanding Dataset (Xiao et al., 2010). SUN contains 899 categories that cover more than indoor, urban, and natural places with or without human beings appearing. We use the subset which contains 50 natural objects not showing in ImageNet-1k.

Places is a large scene photographs dataset (Zhou et al., 2017). It contains photos that are labeled with scene semantic categories from three macro-classes: Indoor, Nature, and Urban. The subset we use is sampled from 50 categories that are not present in ImageNet-1k.

Texture stands for the Describable Textures Dataset (Cimpoi et al.,

2014). It contains images of textures and abstracted patterns. As no categories overlap with ImageNet-1k, we use the entire dataset as in (Huang and Li, 2021).

D.2.4 Baselines and sources of model checkpoints

For the Mahalanobis score (Lee et al., 2018), we use the feature embeddings without l_2 normalization as Gaussian distributions naturally do not fit hyperspherical features. Alternatively, one can normalize the embeddings first and then apply the Mahalanobis score.

For Fort *et al.* (Fort et al., 2021) in Table 5.2, we fine-tune the whole ViT model on the ID dataset. Specifically, we use the publicly available checkpoints from Hugging Face where the ViT model is pre-trained on ImageNet-21k and fine-tuned on ImageNet-1k.

For CLIP models, our reported results are based on checkpoints provided by Hugging Face for CLIP-B and CLIP-L. Similar results can be obtained with checkpoints in the codebase by OpenAI `https://github.com/openai/CLIP`. Note that for CLIP (RN50x4), which is not available in Hugging Face, we use the checkpoint provided by OpenAI.

D.3 Spurious OOD Datasets

In general, spurious attributes refer to statistically informative features that co-exist with the majority of ID samples but do not necessarily capture cues related to the labels such as color, texture, background, etc (Barbu et al., 2019; Beery et al., 2018; Geirhos et al., 2019; Xiao et al., 2021; Zhu et al., 2017). A recent work (Ming et al., 2022c) investigated a new type of hard OOD samples (called spurious OOD) that contain spurious or environmental features, but no object features related to the ID classes. A concrete example is shown in Figure D.1, where images of birds co-occur frequently with either the land background or water background. Mod-

ern neural networks can spuriously rely on the image background (*e.g.*, water or land) for classification instead of learning to recognize the actual object (Ribeiro et al., 2016). Ming et al. (2022c) show that spurious OOD samples remain challenging for most common OOD detection methods based on pure vision models such as ResNet (He et al., 2016).

For ID dataset, we use Waterbirds (Sagawa et al., 2019), which combines bird photographs from CUB-200 (Wah et al., 2011) with water or land background images from Places (Zhou et al., 2017). For the spurious OOD dataset, we use the one created in Ming et al. (2022c) consisting of land and water background from Places (Zhou et al., 2017).

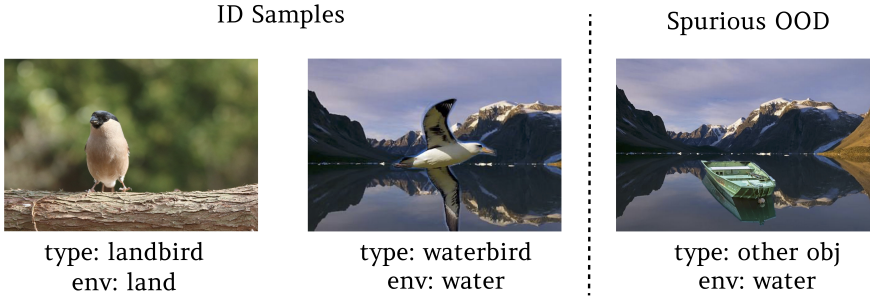


Figure D.1: Illustration of spurious OOD samples for Waterbirds (Sagawa et al., 2019). Images are taken from Ming et al. (2022c).

D.4 ID Classification Accuracy

Table D.1 shows the multi-class classification accuracy on ImageNet-1k for methods in Table 5.2.

Table D.1: ID classification accuracy on ImageNet-1k (%)

Method	ID ACC
zero-shot	
MCM (CLIP-B/16)	67.01
MCM (CLIP-L/14)	73.28
w. fine-tuning	
MSP (CLIP-B/16)	79.39
MSP (CLIP-L/14)	84.12
Energy (Liu et al., 2020) (CLIP-B/16)	79.39
Energy (Liu et al., 2020) (CLIP-L/14)	84.12
Fort et al. (Fort et al., 2021) (ViT-B/16)	81.25
Fort et al. (Fort et al., 2021) (ViT-L/14)	84.05
MOS (Huang and Li, 2021) (BiT)	75.16

D.5 Implementation of CLIP-Based Baselines

D.5.1 Overview of Baselines

We review two previous works on CLIP-based OOD detection (Esmaeilpour et al., 2022; Fort et al., 2021) in Figure D.2, which derive the scoring function based on candidate OOD labels. For a given task with ID label set $\mathcal{Y}_{\text{in}} = \{y_1, y_2, ..., y_K\}$ and candidate labels $\mathcal{Y}_{\text{C}} = \{y_{K+1}, y_{K+2}, ..., y_{K+L}\}$, where ideally $\mathcal{Y}_{\text{in}} \cap \mathcal{Y}_{\text{C}} = \emptyset$, they construct a collection of text embeddings $\mathcal{T}(t_i), i \in \{1, 2, ..., K+L\}$. Here, t_i is the text prompt "`this is a photo of a` $\langle y_i \rangle$" for a label y_i. For any test input image $\mathbf{x}$, we can calculate the label-wise matching score based on the cosine similarity between the image and text features: $s_i(\mathbf{x}) = \frac{\mathcal{I}(\mathbf{x}) \cdot \mathcal{T}(t_i)}{\|\mathcal{I}(\mathbf{x}')\| \cdot \|\mathcal{T}(t_i)\|}$. Therefore, a detection score can be derived as:

$$S(\mathbf{x}; \mathcal{Y}_{\text{in}}, \mathcal{Y}_{\text{C}}, \mathcal{T}, \mathcal{I}) = \sum_{i=1}^{K} \frac{e^{s_i(\mathbf{x})/\tau}}{\sum_{j=1}^{K+L} e^{s_j(\mathbf{x})/\tau}},$$

where $\tau > 0$ is the temperature scaling hyperparameter.

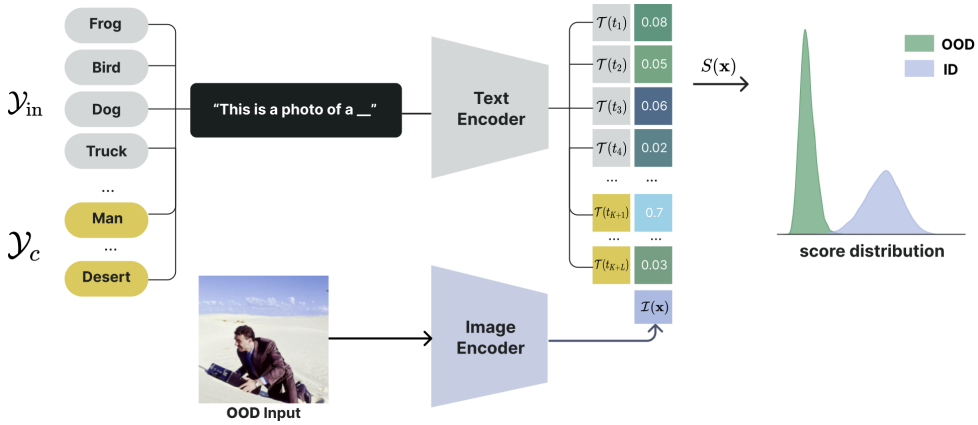


Figure D.2: Zero-shot OOD detection with candidate OOD labels. The ID classification task is defined by a set of class labels $\mathcal{Y}_{\text{in}}$. With an additional set of candidate labels $\mathcal{Y}_{\text{C}}$ that describes the contents of the input image, the OOD detection scoring function can be calculated by normalizing over the expanded space of cosine similarities.

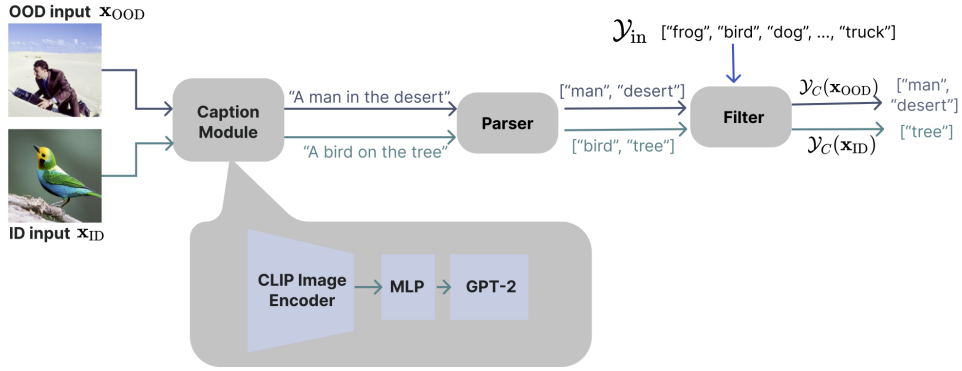


Figure D.3: Improved pipeline to generate candidate OOD labels. It consists of three main components: a caption generator, a syntactic parser, and a filtering module to remove candidate labels that overlap with the ID label set.

D.5.2 Obtaining OOD Candidate Labels

For the baseline methods, obtaining OOD candidate labels is a major challenge and limitation. Recently, (Esmaeilpour et al., 2022) propose ZO-CLIP, where a transformer (decoder) based on the image encoder of CLIP

is used to generate candidate labels. The transformer is trained from scratch on the COCO dataset (Lin et al., 2014) with simple teacher forcing algorithms. Although the decoder trained on COCO may work well on CIFAR (ID), it does not scale up to large-scale datasets such as ImageNet (Deng et al., 2009) where categories are not covered in COCO. As a result, (Esmaeilpour et al., 2022) only test on small-scale datasets with common classes such as CIFAR (ID).

We improve the baseline by using a high-quality caption generator pre-trained on much larger datasets, which not only saves computational overhead but can potentially improve the quality of generated labels. The pipeline involves three components (see Figure D.3):

- A caption generator. Given an input image, it generates a caption serving as the textual description of the input. In this work, we consider ClipCap (Mokady et al., 2021), which uses GPT-2 (Radford et al., 2019) to generate captions based on CLIP's image encoder. ClipCap is pre-trained on a much larger dataset Conceptual Captions (Ng et al., 2020) compared to COCO, which can be viewed as an enhanced version of the ZO-CLIP baseline (Esmaeilpour et al., 2022). The checkpoints are publicly available[2].

- A syntactic parser. Given a caption, we extract noun objects using a parsing toolkit released by spaCy [3]. Those nouns can be used as candidate labels $\mathcal{Y}_C$ of the input image.

- A filter module. Unlike (Esmaeilpour et al., 2022), we further enhance the baseline by adopting a filtering technique to remove overlapping categories in $\mathcal{Y}_C$ with ID labels $\mathcal{Y}_{\text{in}}$, which we detail below.

[2] https://github.com/rmokady/CLIP_prefix_caption
[3] https://spacy.io/models/en

D.5.3 Label Filtering

Example. To illustrate the effects of filtering, we begin with a concrete example where ID labels are ["frog","bird"... "truck"], as shown in Figure D.3. The generated labels (without filtering) of an ID input of a bird sitting on a tree are ["bird", "tree"]. Therefore, $\mathcal{Y}_{\text{in}} \cup \mathcal{Y}_{\text{C}}$ =["frog","bird"... "truck","bird", "tree"]. Ideally, the softmax probability distribution over the concatenated labels would be $[0, 0.5, 0, \ldots, 0.5, 0]$ and by definition $S(\mathbf{x}) \approx 0.5$. However, if we filter the generated labels to eliminate nouns with similar meanings as ID, our concatenated labels would be ["frog","bird"... "truck","tree"] and the probability vector would be $[0, 1, 0, \ldots, 0]$, which leads to a much higher score $S(\mathbf{x}) = 1$. In contrast, the generated labels for an OOD input with a caption "man in the desert" would be ["man", "desert"]. The resulting probability vector would be $[0, 0, 0, \ldots, 1, 0]$ and the score $S(\mathbf{x}) = 0$. Therefore, filtering makes it easier to separate ID inputs from OOD inputs (*c.f.* Figure D.4).

String-based filtering. To implement the idea of filtering, we need a measurement of the similarity between the generated labels and ID labels. The simplest way is string-based filtering where a generated label is filtered if it matches any ID labels (in the string format), as in the case above.

D.6 Alternative Scoring Functions

We explore the effectiveness of several alternative scoring functions:

- Entropy: the (negative) entropy of softmax scaled cosine similarities denoted as S_{entropy};
- Var: the variance of the cosine similarities denoted as S_{var};

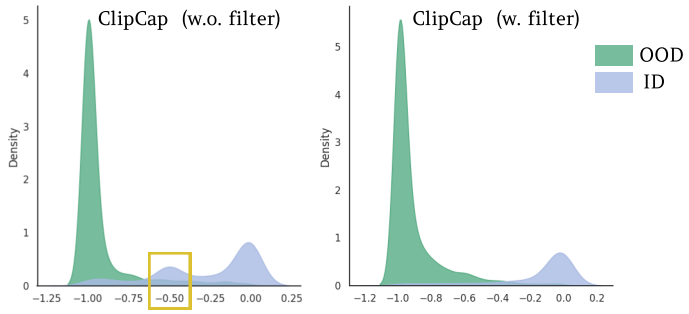


Figure D.4: Score distributions for ImageNet-10 (ID) and iNaturalist (OOD) inputs. Simple string-based filtering alleviates the overlap between OOD inputs and ID inputs especially with scores around 0.5 (yellow rectangle), resulting in better ID-OOD separability.

- Scaled: the scaled difference between the largest and second-largest cosine similarities $S_{\text{scaled}} := e^{s_{\hat{y}}(\mathbf{x}) - s_{\hat{y}_2}(\mathbf{x})}$ where $\hat{y} := \text{argmax}_{i \in [K]} s_i(\mathbf{x})$ and $\hat{y}_2 := \text{argmax}_{i \neq \hat{y}, i \in [K]} s_i(\mathbf{x})$.

As shown in Table D.2, MCM still gives the most promising results compared to the other three alternative scores across most OOD test sets.

Table D.2: Comparison with other scaling functions (applied to inner products) on the large-scale benchmark ImageNet-1k (ID). We use CLIP-B/16 as the backbone.

Method	**OOD Dataset**								**Average**	
	iNaturalist		SUN		Places		Texture			
	FPR↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑	**FPR**↓	**AUROC**↑
Entropy	84.44	63.50	93.79	62.54	94.10	64.15	97.16	58.98	92.37	62.29
Var	87.42	63.87	68.71	81.02	76.28	75.38	80.04	71.90	78.11	73.04
Scaled	89.06	72.26	89.06	70.81	89.08	69.66	89.56	68.17	89.19	70.22
MCM	30.91	94.61	37.59	92.57	44.69	89.77	57.77	86.11	42.74	90.77

Appendix E

Appendix for How Does Fine-Tuning Impact Out-of-Distribution Detection for Vision-Language Models

E.1 Dataset Details

Details on ID and OOD dataset construction For ID datasets, we follow the same construction as in previous works (Zhang et al., 2022b; Zhou et al., 2022b,c). Detailed instructions on dataset installation can be found in https://github.com/KaiyangZhou/CoOp/blob/main/DATASETS.md. For OOD datasets, Huang and Li (2021) curate a collection of subsets from iNaturalist Van Horn et al. (2018), SUN Xiao et al. (2010), Places Zhou et al. (2017), and Texture Cimpoi et al. (2014) as large-scale OOD datasets for ImageNet-1k, where the classes of the test sets do not overlap with ImageNet-1k. Detailed instructions can be found in https://github.com/deeplearning-wisc/large_scale_ood.

E.2 Additional Results

E.2.1 ID accuracy

While we primarily focus on the OOD detection performance of CLIP-based fine-tuning methods, we present the results of the ID accuracy for each dataset based on CLIP-B/16 in Table E.1 for completeness. Further results on the ID accuracy with various datasets and architectures can be seen in Zhou et al. (2022b), Zhou et al. (2022c), and Zhang et al. (2022b).

Table E.1: ID accuracy on the downstream datasets for CLIP-based fine-tuning methods with CLIP-B/16.

ID Dataset	Method	ID Acc
Caltech-101	ZOCLIP	92.90
	TipAdaptor	95.01
	TipAdaptorF	95.66
	CoOp	95.30
	CoCoOp	95.00
Food-101	ZOCLIP	86.10
	TipAdaptor	86.49
	TipAdaptorF	87.43
	CoOp	85.50
	CoCoOp	87.30
Stanford-Cars	ZOCLIP	65.27
	TipAdaptor	75.29
	TipAdaptorF	83.40
	CoOp	78.50
	CoCoOp	72.30
Oxford-Pets	ZOCLIP	89.10
	TipAdaptor	91.85
	TipAdaptorF	92.91
	CoOp	93.40
	CoCoOp	93.30
ImageNet-1k	ZOCLIP	68.77
	TipAdaptor	70.26
	TipAdaptorF	73.70
	CoOp	71.63
	CoCoOp	71.20

E.2.2 OOD detection performance based on visual features alone

In this section, we explore several commonly used OOD detection scores solely based on the visual branch of CLIP models. Specifically, we consider the Mahalanobis score (Lee et al., 2018) on the penultimate layer of the visual encoder and MSP (Hendrycks and Gimpel, 2017), Energy (Liu et al., 2020), and KL Matching (Hendrycks et al., 2022) scores on the logit layer after linear probing the visual encoder. The results are summarized in Table E.2, based on 16-shot Caltech-101 (ID). We can see that the Mahalanobis score does not yield promising performance because 1) the feature embeddings from the visual encoder of CLIP may not follow class-conditional Gaussian distributions, 2) it is challenging to estimate the mean and especially covariance matrix when the number of samples is much smaller than the feature dimension in the few-shot setting. On the other hand, the OOD scores based on fine-tuned logit layer result in worse performance compared to the MCM score. One major reason is that fine-tuning CLIP in the few-shot setting is prone to overfitting the downstream ID dataset, making the model less reliable. This further highlights the importance of choosing OOD detection scores fitted to parameter-efficient fine-tuning methods.

E.2.3 Additional results on ImageNet-1k

In this section, we consider two additional OOD test sets ImageNet-O (Hendrycks et al., 2021) and OpenImage-O (Wang et al., 2022b) for ImageNet-1k (ID). OpenImage-O is a subset curated from the test set of OpenImage-V3 (Krasin et al., 2017) containing a diverse set of categories. ImageNet-O is a challenging OOD dataset that contains naturally adversarial examples for ImageNet-1k. The results are shown in Table E.3. The model (CLIP-B/16) is trained with CoOp. We can see that: 1) The performance on ImageNet-O is gen-

Table E.2: Additional results for OOD scores based on visual encoder only. ID dataset is Caltech-101 (16 shot).

OOD Score	OOD Dataset	FPR95↓	AUROC↑
Maha	SUN	34.15	95.20
	Places	20.50	96.21
	Textures	64.10	92.43
	iNaturalist	66.62	92.97
	AVG	46.34	94.20
Energy	SUN	15.02	97.05
	Places	21.10	95.75
	Textures	15.60	97.00
	iNaturalist	33.77	95.49
	AVG	21.37	96.32
KL Matching	SUN	4.56	98.21
	Places	8.92	97.52
	Textures	42.64	94.47
	iNaturalist	9.70	97.35
	AVG	16.46	96.89
MSP	SUN	16.23	96.59
	Places	20.98	95.97
	Textures	7.15	98.33
	iNaturalist	11.79	97.31
	AVG	14.04	97.05

erally worse than the rest of OOD test sets (iNaturalist, Textures, SUN, Places) in Section 6.5.3, suggesting that this task remains challenging in the context of few-shot prompt learning. 2) MCM score still performs the best compared to MS and MSP on both OOD test sets, consistent with our previous observations, which further highlights the importance of softmax and temperature scaling for OOD detection with fine-tuning.

Table E.3: OOD detection performance on two OOD additional test sets for ImageNet-1k (ID). We train CLIP-B/16 with CoOp.

OOD Dataset	OOD Score	FPR95↓	AUROC↑
ImageNet-O	S_{MSP}	77.20	74.01
	S_{MS}	70.75	82.30
	S_{MCM}	61.50	84.13
OpenImage-O	S_{MSP}	56.89	83.73
	S_{MS}	39.18	91.48
	S_{MCM}	36.68	92.76

E.2.4 Alternative OOD scores

In this section, we investigate the performance with several alternative OOD scoring functions based on the cosine similarities of input $\mathbf{x}$ with the k-th label $s_k(\mathbf{x})$, $k \in \{1, 2, ..., K\}$ (defined in Section 6.4.2). Specifically, we consider the energy and the KL matching score for each adaptation method and summarize the results based on Caltech-101 (ID) in Table E.5. We observe that 1) using the energy score, all adaptation methods significantly enhance the performance over the zero-shot baseline (ZOCLIP). 2) the general performance vastly improves when utilizing the KL Matching score. However, even the highest achieved performance (FPR95 at 7.91 with CoCoOp) falls short when compared to the MCM score (FPR95 at 5.02 with CoCoOp).

Table E.4: OOD detection performance based on S_{MSP} score. The average performance for most adaptation methods is much worse than using S_{MS} (Table 6.1) and S_{MCM} (Table 6.3).

ID Dataset	Method	SUN		Places		Textures		iNaturalist		Average	
		FPR95↓	**AUROC↑**	**FPR95↓**	**AUROC↑**	**FPR95↓**	**AUROC↑**	**FPR95↓**	**AUROC↑**	**FPR95↓**	**AUROC↑**
		FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
Food-101	ZOCLIP	11.48	97.76	13.11	97.48	15.04	96.08	16.65	96.73	14.07	97.01
	TipAdaptor	7.32	98.51	9.03	98.31	11.88	96.94	14.47	97.21	**10.68**	**97.74**
	TipAdaptorF	15.08	97.26	15.38	97.24	17.57	95.99	20.95	96.18	17.25	96.67
	CoOp	19.66	96.20	21.15	95.95	28.33	93.62	23.80	95.51	23.23	95.32
	CoCoOp	8.67	98.28	10.56	98.03	14.77	96.23	14.33	97.26	12.08	97.45
Oxford-Pets	ZOCLIP	24.67	94.72	28.54	93.71	19.01	96.42	39.77	93.01	28.00	94.47
	TipAdaptor	15.66	97.11	18.83	96.45	12.50	97.92	25.19	95.90	18.04	96.84
	TipAdaptorF	16.79	96.77	20.33	96.04	12.22	97.90	26.62	95.80	18.99	96.63
	CoOp	8.46	98.50	10.75	98.13	11.21	98.09	32.13	94.08	15.64	97.20
	CoCoOp	9.06	98.31	10.43	98.13	7.39	98.70	27.97	95.11	**13.71**	**97.56**
Stanford-Cars	ZOCLIP	6.99	98.49	10.33	97.68	8.24	98.39	32.85	92.56	14.60	96.78
	TipAdaptor	1.94	99.58	3.30	99.31	1.97	99.56	12.52	97.80	**4.93**	**99.06**
	TipAdaptorF	15.39	97.19	14.01	97.32	8.39	98.49	21.88	95.90	14.92	97.22
	CoOp	9.88	98.05	14.07	97.12	10.71	97.71	36.73	91.51	17.85	96.10
	CoCoOp	9.99	97.81	11.87	97.15	10.46	97.69	31.58	92.59	15.97	96.31
Caltech-101	ZOCLIP	16.17	96.47	22.45	94.96	17.89	96.33	15.01	96.96	17.88	96.18
	TipAdaptor	12.98	97.40	17.79	96.77	13.74	97.72	20.08	96.65	**16.15**	**97.13**
	TipAdaptorF	17.94	96.68	22.92	95.74	15.16	97.40	24.18	96.01	20.05	96.46
	CoOp	24.07	96.11	29.91	94.59	26.29	95.72	26.35	95.92	26.66	95.58
	CoCoOp	14.92	97.32	20.67	95.91	19.20	96.56	21.74	96.33	19.13	96.53

Table E.5: Comparison with additional OOD scores on Caltech-101 (ID). S_{KL} stands for the KL matching score (Hendrycks et al., 2022) and S_{Energy} denotes the energy score (Liu et al., 2020).

OOD Score	Method	SUN		Places		Textures		iNaturalist		Average	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
S_{MS}	ZOCLIP	32.03	94.06	33.01	93.39	54.66	89.29	32.14	94.30	37.96	92.76
	TipAdaptor	9.69	98.07	11.25	97.84	20.90	96.68	13.62	97.72	13.86	97.58
	TipAdaptorF	10.20	97.76	11.60	97.42	23.32	95.54	14.01	97.36	14.78	97.02
	CoOp	5.53	98.56	9.88	97.50	13.10	97.10	4.89	98.76	8.35	97.98
	CoCoOp	2.86	99.19	6.42	98.37	8.81	98.09	5.68	98.68	5.94	98.58
S_{MCM}	ZOCLIP	14.83	97.20	20.45	96.00	14.98	97.35	10.84	97.76	15.28	97.08
	TipAdaptor	5.12	98.83	8.05	98.34	4.65	99.05	6.94	98.77	6.19	**98.75**
	TipAdaptorF	4.83	98.79	8.09	98.07	6.41	98.11	4.94	98.98	6.07	98.49
	CoOp	3.62	99.01	8.15	97.89	6.29	98.62	7.57	98.35	6.41	98.47
	CoCoOp	4.26	98.94	6.76	98.00	4.33	98.88	4.71	98.68	**5.02**	98.62
S_{Energy}	ZOCLIP	53.83	90.22	50.51	90.21	74.10	83.20	56.00	90.13	58.61	88.44
	TipAdaptor	11.71	97.72	12.20	97.61	30.48	95.73	16.42	97.30	17.70	97.09
	TipAdaptorF	11.57	97.46	11.89	97.30	29.38	94.70	16.18	96.90	17.26	96.59
	CoOp	6.58	98.29	11.16	97.20	18.19	96.32	5.92	98.53	10.46	97.59
	CoCoOp	5.22	98.87	8.80	98.13	17.30	96.87	11.28	97.95	10.65	97.95
S_{KL}	ZOCLIP	5.51	97.57	9.48	96.61	7.41	97.64	11.43	96.22	14.02	97.31
	TipAdaptor	5.54	97.63	7.69	97.13	5.74	97.96	8.00	97.37	6.74	97.52
	TipAdaptorF	8.52	96.89	13.00	95.92	7.02	98.02	10.71	97.11	9.81	96.98
	CoOp	7.15	98.06	12.37	96.60	8.74	97.62	9.33	98.00	9.40	97.57
	CoCoOp	4.07	98.95	9.61	97.59	5.30	98.77	12.67	97.57	7.91	98.22

Appendix F

Appendix for Understanding Retrieval-Augmented Task Adaptation for Vision-Language Models

F.1 Experimental Details

Hardware and software. We run all experiments on NVIDIA GeForce RTX-A6000 GPU. To retrieve samples from the LAION5B database, we build a semantics-based retrieval system with `clip-retrieval` (`https://github.com/rom1504/clip-retrieval`) for fast T2I and I2I retrieval based on textual and visual embeddings of pre-trained CLIP. Our implementation is based on PyTorch 1.12.

Retrieval dataset. We adopt LAION5B as the database for retrieval for three main reasons: (1) Scale: in contrast to prior works that use smaller-scale datasets such as WebVision Li et al. (2017b), Conceptual Captions Sharma et al. (2018), and ImageNet-21k Ridnik et al. (2021), LAION5B is a web-scale open-source dataset that contains 5,85 billion CLIP-filtered image-

text pairs covering a wide range of concepts in the real world. The diverse concept coverage makes it a reliable source for retrieval Udandarao et al. (2023). (2) Multi-modal retrieval: one major advantage of LAION is that it computes the textual and visual embeddings of the text-image pairs based on pre-trained CLIP. This provides the foundation for us to conduct a systematic study on both T2I and I2I retrieval. (3) Retrieval efficiency: the development of distributed inference tools such as `clip-retrieval` enable fast index building and efficient retrieval from LAION5B based on approximate KNN search. Such community support for LAION5B makes retrieval more practical compared to alternatives.

Prompts for T2I retrieval. In this work, we use dataset-specific prompts in T2I retrieval to mitigate semantic ambiguity. For example, for Bird200 (Wah et al., 2011), the prompt for T2I retrieval is `A photo of a <CLS>, a type of bird`. The prompts for other datasets can be seen in Table F.1. In a recent work (Udandarao et al., 2023), language model-based prompts are used for retrieval. However, external knowledge encoded in pre-trained language models can introduce additional biases especially when the target dataset contains characteristics not captured by the class names. As a result, we observed similar issues as in Figure 7.3 when using language model-based prompts in our initial experiments.

Dataset	Prompt
Caltech101 (Fei-Fei et al., 2004)	`A photo of a <CLS>`
Birds200 (Wah et al., 2011)	`A photo of a <CLS>, a type of bird`
Food101 (Bossard et al., 2014)	`A photo of <CLS>, a type of food`
OxfordPets (Parkhi et al., 2012)	`A photo of a <CLS> pet`
Flowers102 Nilsback and Zisserman (2008)	`A photo of a <CLS> flower`
Textures Cimpoi et al. (2014)	`A photo of <CLS> texture`
UCF101 Soomro et al. (2012)	`A photo of <CLS> in action`

Table F.1: Default prompts for T2I retrieval. In this work, we use dataset-specific prompts to mitigate semantic ambiguity.

Fine-tuning details. As our work focuses on the impact of retrieval, we adopt the fine-tuning scheme in Zhang et al. (2022b) for training-based

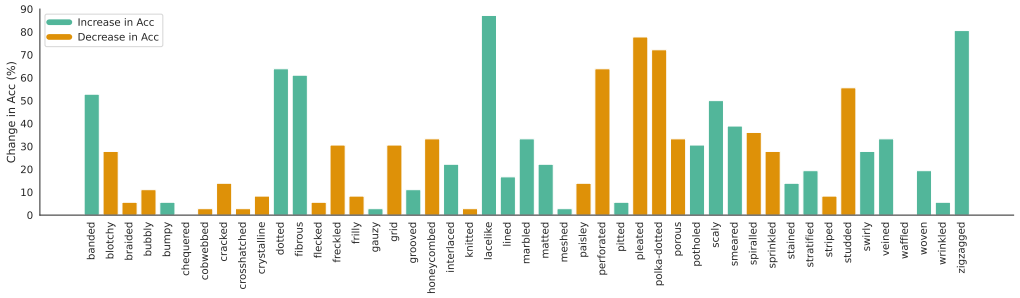
adaptation, where we set features in the retrieval cache as learnable. For each target dataset, the train, validation, and test split also follow (Zhang et al., 2022b). Specifically, we use AdamW Loshchilov and Hutter (2019) as the optimizer with a cosine scheduler. The initial learning rate is set as 0.001 and we finetune for 20 epochs. The hyperparameters such as α, ω, γ are determined based on the validation split of each target dataset.

F.2 A Closer Look at Logit Ensemble via Classwise Performance

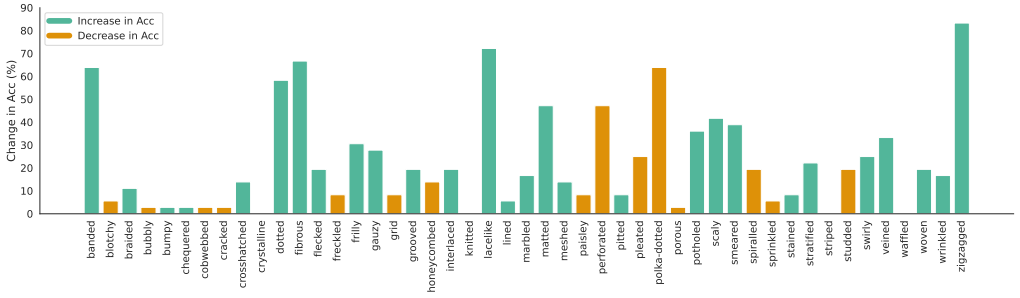
In Section 7.3.3, we have shown that logit ensemble is essential to CLIP-based adaptive inference with the few-shot cache obtained by retrieval. In this section, we take a finer-grained view by examining the change of accuracy for each class before and after logit ensemble. For better visualization, we use Textures Cimpoi et al. (2014), a dataset with 47 classes. The results are shown in Figure F.1, where green indicates an increase in accuracy while orange denotes a decrease in accuracy. The result for RET vs. ZOCLIP (*i.e.*, before ensemble) is shown in Figure F.1a and Ensemble vs. ZOCLIP is shown in Figure F.1b. We can clearly observe that (1) before ensemble, RET is inferior to ZOCLIP for multiple classes such as `blotchy` and `freckled`, and `pleated`, as a result of retrieval ambiguity. (2) Logit ensemble significantly mitigates such issue and results in an overall larger proportion of green bars compared to orange bars, as shown in Figure F.1b.

F.3 Qualitative Analysis of Retrieved Samples

In Section 7.3.2, we examined the retrieved samples from I2I and T2I retrieval to identify the main sources of errors. Here, we present additional retrieved samples for diverse datasets. The results are depicted in Fig-



(a) RET over ZOCLIP (average improvement in accuracy: 3.1%)



(b) Ensemble over ZOCLIP (average improvement in accuracy: 12.5%)

Figure F.1: Change of classwise accuracy before and after logit ensemble. For better visualization, the results are based on Textures Cimpoi et al. (2014), a dataset with 47 classes. We use I2I retrieval to obtain the few-shot feature cache. We plot the change of accuracy over ZOCLIP for each class before (top row) and after logit ensemble (bottom row). Blue bars indicate an increase in accuracy while orange denotes a decrease in accuracy. (a) Comparison of RET versus ZOCLIP. On average, RET achieves a 3.1% improvement in accuracy compared to ZOCLIP. (b) Comparison of Ensemble versus ZOCLIP. On average, Ensemble achieves a 12.5% improvement in accuracy compared to ZOCLIP. This further highlights the importance of logit ensemble for retrieval-augmented adaptation.

ure F.2, where we contrast samples from T2I retrieval (top row), I2I retrieval (middle row), and the downstream dataset (bottom row). We have two salient observations: (1) As discussed in Section 7.3.2, T2I retrieval often yields a diverse set of images that match the class semantics. However, this diversity may not always be beneficial for adapting to

the target dataset, especially in the few-shot retrieval setting where one is under a limited budget. For example, using the query `a photo of a lobster`, we may not retrieve images of cooked lobsters that often appear in the target dataset. (2) Since T2I retrieval utilizes the class name in the query, it occasionally retrieves images with text on them, rather than images of the actual object. For instance, we retrieve images that feature the text "summer tanager" or "dandelion" (as seen in the 4th and 3rd columns of Figures F.2 and 7.3, respectively). This occurs because the cosine similarity between pairs of `(class name, image of the actual object)` and `(class name, image with the text <class name>)` is similar, based on pre-trained CLIP models. This highlights a prevalent challenge in web-scale cross-modal retrieval systems, such as LAION5B. Conversely, this type of misalignment is rarely encountered in I2I retrieval. Therefore, samples from T2I retrieval can introduce undesirable inductive biases, resulting in limited performance gains over the zero-shot model.

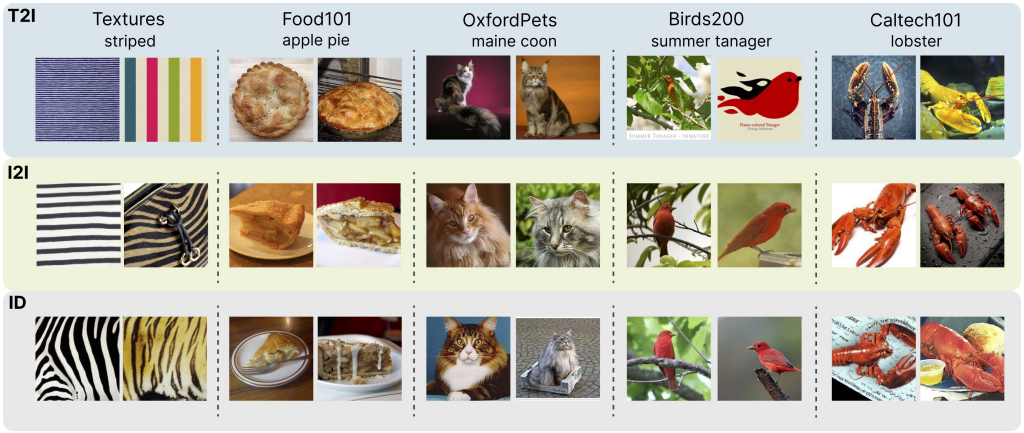


Figure F.2: More samples from T2I and I2I retrieval. Top row: the main source of noise for T2I retrieval is semantic ambiguity, as the textual queries (*e.g.*, `striped texture`) may not accurately describe the images from target distributions (bottom row). Middle row: samples retrieved by I2I matches more closely with ID data. Bottom row: images sampled from the target (ID) distribution.

F.4 Theoretical Understanding

In this section, we provide details on the problem setup, introduce relevant definitions and lemmas, and provide the complete proof for our theoretical results discussed in Section 4.6. Common notations can be seen in Table F.2.

Notation	Description
$[C]$	The set $\{1, 2, \ldots, C\}$
$\mathbb{1}$[condition]	Indicator function, equals 1 if the condition is true, 0 otherwise
$\mathcal{T}$	$\mathcal{T} : t \to \mathbb{R}^d$ is the text encoder of CLIP
$\mathcal{I}$	$\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$ is the image encoder of CLIP

Table F.2: Common notations.

F.4.1 Problem Setup

We consider a pre-trained CLIP model (Radford et al., 2021) with one text encoder $\mathcal{T} : t \to \mathbb{R}^d$ and one image encoder $\mathcal{I} : \mathbf{x} \to \mathbb{R}^d$. We use $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_C] \in \mathbb{R}^{d \times C}$ to denote the text embedding matrix for all classes, where $\mathbf{t}_c := \mathcal{T}(\mathbf{t}_c) \in \mathbb{R}^d$ and t_c is a generic textual description of class c such as "`a photo of <CLASS c>`". For theoretical analysis, we consider training-free adaptation based on retrieved samples. We use the terms "downstream" and "target" dataset interchangeably which refer to the dataset a pre-trained CLIP model is adapted to.

Building feature cache by retrieval. Given a downstream dataset with C classes: $\mathcal{Y} = \{1, 2, ..., C\}$ and a retrieval budget size of KC, we can retrieve K samples per class to build a cache of size KC. Recall that $\mathbf{K} = [\mathbf{k}_{1,1}, \mathbf{k}_{1,2} \ldots, \mathbf{k}_{C,K}] \in \mathbb{R}^{d \times CK}$ denotes the embedding matrix for retrieved images, where $\mathbf{k}_{c,i} := \mathcal{I}(\mathbf{x}_{c,i}) \in \mathbb{R}^d$. For notational simplicity, we assume

text and image features are ℓ_2 normalized Radford et al. (2021). In other words, we have $\|\mathbf{z}\|_2 = \|\mathbf{t}_c\|_2 = 1$ for any $\mathbf{z} = \mathcal{I}(\mathbf{x})$ and $\mathbf{t}_c = \mathcal{T}(t_c)$.

Let $\tilde{\mathbf{K}} = \frac{\mathbf{K}\mathbf{V}^\top}{K} = [\tilde{\mathbf{k}}_1, \tilde{\mathbf{k}}_2, \ldots, \tilde{\mathbf{k}}_C] \in \mathbb{R}^{d\times C}$ contain the average retrieved feature for each class. $\mathbf{V} \in \mathbb{R}^{C\times CK}$ is a sparse matrix containing the one-hot labels for retrieved samples with entries $\mathbf{V}_{i,j} = \mathbb{1}\{i = \tilde{j}\}$ for $i \in [C], j \in [CK]$, where $\tilde{j} := \left\lceil \frac{j}{K} \right\rceil$ Zhang et al. (2022b). For example, when $K = 2, C = 3$, we have:

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

We define $\bar{\mathbf{K}} := [\bar{\mathbf{k}}_1, \bar{\mathbf{k}}_2, \ldots, \bar{\mathbf{k}}_C]$ as the normalized version where $\bar{\mathbf{k}}_i = \frac{\tilde{\mathbf{k}}_i}{\|\tilde{\mathbf{k}}_i\|_2}$, which will be used in the rest of the section. Note that here the notations are slightly different from Section 7.4.1 and are more rigorous.

Task adaptation with retrieved cache. At inference time, let $(\mathbf{x}, y) \sim \mathcal{D}_T$ be a test sample from the target distribution $\mathcal{D}_T$ with label $y \in [C]$ and its visual feature $\mathbf{z} := \mathcal{I}(\mathbf{x})$. In some cases, beyond retrieved samples, one also has access to a cache consisting of few-shot training samples from the target distribution. For theoretical analysis, we consider one-shot and denote the feature cache as $\mathbf{S} := [\mathbf{s}_1, \ldots, \mathbf{s}_C] \in \mathbb{R}^{d\times C}$. The final logit for the test sample can be represented as a weighted sum (ensemble) of logits from the zero-shot CLIP and the feature cache from retrieved and training samples[1]:

$$f(\mathbf{x}) = (\alpha\mathbf{T} + \beta\mathbf{S} + \gamma\bar{\mathbf{K}})^\top \mathbf{z},$$

where $0 \le \alpha, \beta, \gamma \le 1$. Without loss of generality, we assume $\alpha+\beta+\gamma = 1$.

In particular, the zero-shot logit $f^{\text{ZOC}}(\mathbf{x}) := \mathbf{T}^\top\mathbf{z}$ and the retrieval logit

[1]For theoretical analysis, we omit the exponential scaling function to better focus on the effects of ensembling.

$f^{\text{RET}}(\mathbf{x}) := \bar{\mathbf{K}}^\top \mathbf{z}$. In the main paper, we mainly focus on $\beta = 0$ (*i.e.*, one only has access to retrieved samples). We denote the corresponding ensemble logit as $f^{\text{EN}}(\mathbf{x}) = (\alpha\mathbf{T} + \gamma\bar{\mathbf{K}})^\top \mathbf{z}$.

Evaluation metric. Given a loss function $\ell(\mathbf{v}, y)$ such as the cross-entropy:

$$\ell(\mathbf{v}, y) = -\log \frac{\exp(\mathbf{v}_y)}{\sum_{i\in[C]} \exp(\mathbf{v}_i)},$$

the population risk on the target distribution is:

$$\mathcal{L}(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T}[\ell(f(\mathbf{x}), y)].$$

To simplify notations, we denote the risk as $\mathcal{R}(\mathbf{Q}) := \mathbb{E}\left[\ell(\mathbf{Q}^\top \mathbf{z}, y)\right]$ for some $\mathbf{Q} \in \mathbb{R}^{d\times C}$. For example, the risk of logit ensemble is $\mathcal{R}(\alpha\mathbf{T} + \gamma\bar{\mathbf{K}})$. We also have the error risk $\mathcal{R}_{0-1}$ defined as:

$$\mathcal{R}_{0-1}(f) = 1 - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T}\left[\mathbb{1}\{\arg\max_{i\in[C]} f(\mathbf{x})_i = y\}\right].$$

F.4.2 Definitions and Assumptions

Before presenting the main theoretical results, we first introduce the following definitions and assumptions to formalize the retrieval augmented adaptation process based on pre-trained CLIP models.

For class $i \in [C]$, we define $\tilde{\mathbf{s}}_i := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathcal{I}(\mathbf{x})|y = i]$, which is the image representation of class i based on the downstream distribution and $\bar{\mathbf{s}}_i = \frac{\tilde{\mathbf{s}}_i}{\|\tilde{\mathbf{s}}_i\|_2}$ the ℓ_2 normalized version[2]. Let $\bar{\mathbf{S}} := [\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \ldots, \bar{\mathbf{s}}_C]$.

Definition F.1 (Inner-class concentration and inter-class separation)**.** *We define the inter-class feature separation as* $\nu := 1 - \max_{i\neq j} \bar{\mathbf{s}}_i^\top \bar{\mathbf{s}}_j$. *We use* ρ_c *to*

[2]For any two non-zero vectors $\mathbf{v}_1, \mathbf{v}_2$ with unit norms, we have $\|\mathbf{v}_1 - \mathbf{v}_2\|_2 = \sqrt{2 - 2\mathbf{v}_1^\top \mathbf{v}_2}$.

denote the inner-class feature concentration:

$$\rho_c := \max_{i \in [C]} \Pr\left(\left\|\mathcal{I}(\mathbf{x}) - \bar{\mathbf{s}}_i\right\|_2 \geq \kappa | y = i\right)$$

for some positive constant κ*.*

Definition F.2. *Let* $\bar{\mathbf{Z}} = [\bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_C] \in \mathbb{R}^{d \times C}$*. We define the optimal representations as*

$$\bar{\mathbf{Z}}^* = \underset{\bar{\mathbf{Z}} \in \mathbb{R}^{d \times C}; \forall i \in [C], \|\bar{\mathbf{z}}_i\| = 1}{\operatorname{argmin}} \mathbb{E}[\ell(\bar{\mathbf{Z}}^{*\top} \mathbf{z}, y)].$$

Definition F.3 (Modality gap)**.** *We define the modality gap between the pretrained text distribution and the target distribution (in the visual modality) as* $\tau := \max_{i \neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top \bar{\mathbf{s}}_i$*, where* $i, j \in [C]$*.*

Definition F.4 (Retrieval distribution shift)**.** *We denote the retrieval distribution based on the (text or image) query (denote* $\mathbf{t}_c$ *or* $\mathbf{s}_c$ *as* $\mathbf{q}_c$*) from class* c *as* $\mathcal{D}_{R|\mathbf{q}_c}$*.* $\tilde{\mathbf{k}}_{\mathbf{q}_c} := \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{R|\mathbf{q}_c}}[\mathcal{I}(\mathbf{x})]$ *is the average retrieved feature from class* c*.* $\bar{\mathbf{k}}_{\mathbf{q}_c} := \frac{\tilde{\mathbf{k}}_{\mathbf{q}_c}}{\|\tilde{\mathbf{k}}_{\mathbf{q}_c}\|_2}$ *denotes the normalized version. We define the distributional shift between target data and T2I and I2I retrieval data for class* c *as* $\xi_c^{T2I} := 1 - \bar{\mathbf{k}}_{\mathbf{t}_c}^\top \bar{\mathbf{s}}_c$ *and* $\xi_c^{I2I} := 1 - \bar{\mathbf{k}}_{\mathbf{s}_c}^\top \bar{\mathbf{s}}_c$*. Let,* $\xi_{\mathbf{t}} := \max_{c \in [C]} \xi_c^{T2I}$ *and* $\xi_{\mathbf{s}} := \max_{c \in [C]} \xi_c^{I2I}$*.*

Remarks: Note that $\tilde{\mathbf{k}}_{\mathbf{q}_c}$ is the expected version, while $\tilde{\mathbf{k}}_c$ (defined in Appendix F.4.1) is the empirical mean of retrieved samples for class $c \in [C]$.

At inference time, for a test sample $(\mathbf{x}, y) \sim \mathcal{D}_T$ with image feature

$\mathbf{z} = \mathcal{I}(\mathbf{x})$, one of the following four events can happen:

$$
\begin{aligned}
E_1 :=& \{(\mathbf{x}, y) \sim \mathcal{D}_T : y \neq \arg\max_{i \in [C]} \mathbf{t}_i^\top \mathbf{z} \text{ and } y \neq \arg\max_{i \in [C]} \bar{\mathbf{k}}_i^\top \mathbf{z}\} \\
E_2 :=& \{(\mathbf{x}, y) \sim \mathcal{D}_T : y = \arg\max_{i \in [C]} \mathbf{t}_i^\top \mathbf{z} \text{ and } y \neq \arg\max_{i \in [C]} \bar{\mathbf{k}}_i^\top \mathbf{z}\} \\
E_3 :=& \{(\mathbf{x}, y) \sim \mathcal{D}_T : y \neq \arg\max_{i \in [C]} \mathbf{t}_i^\top \mathbf{z} \text{ and } y = \arg\max_{i \in [C]} \bar{\mathbf{k}}_i^\top \mathbf{z}\} \\
E_4 :=& \{(\mathbf{x}, y) \sim \mathcal{D}_T : y = \arg\max_{i \in [C]} \mathbf{t}_i^\top \mathbf{z} \text{ and } y = \arg\max_{i \in [C]} \bar{\mathbf{k}}_i^\top \mathbf{z}\}.
\end{aligned}
$$

We can see that $\mathcal{R}_{0-1}(f^{\text{ZOC}}) = \Pr(E_1) + \Pr(E_3)$ and $\mathcal{R}_{0-1}(f^{\text{RET}}) = \Pr(E_1) + \Pr(E_2)$. Next, we formalize the intuitions in Figure 7.3 as the following definition:

Definition F.5 (Knowledge encoded in different modalities)**.** *For a vector $\mathbf{v} \in \mathbb{R}^C$ and a scalar $i \in [C]$, We define $\phi(\mathbf{v}, i, z) := \{j | \mathbf{v}_i - \mathbf{v}_j \leq z\}$. Consider $(\mathbf{x}, y) \sim \mathcal{D}_T$ and $\mathbf{z} = \mathcal{I}(\mathbf{x})$. We define the conditional probability $\rho_d(z)$ as*

$$
\rho_d(z) = \Pr\left(\phi(\mathbf{T}^\top \mathbf{z}, y, z) \cap \phi(\bar{\mathbf{K}}^\top \mathbf{z}, y, z) \neq \{y\} \middle| E_2 \text{ or } E_3\right).
$$

Remarks: $\phi(\mathbf{v}, i, z)$ identifies elements in vector $\mathbf{v}$ that are within a threshold z of the i-th element of $\mathbf{v}$. $\rho_d(z)$ represents the likelihood that, given events E_2 or E_3, the transformed data $\mathbf{z}$ is associated with an incorrect class by both $\mathbf{T}$ and $\bar{\mathbf{K}}$. In practical scenarios, $\rho_d(z)$ is typically small. This is because different modalities usually represent knowledge in distinct ways and, as a result, have different patterns of confusion or error.

Assumption F.6 (Sample representativeness)**.** *We assume that the sample for each class is relatively representative, i.e., $\forall i \in [C], \|\mathbf{s}_i - \bar{\mathbf{s}}_i\|_2 \leq \kappa$ for some constant κ.*

Assumption F.7 (Retrieved data distribution)**.** *We assume that for each class the distribution of retrieved samples is composed of clusters, which exhibit ν separation and κ concentration as defined in Definition F.1. We assume that the*

retrieval process for a query sample is uniformly sampling from its closest retrieval cluster.

F.4.3 Main Results and Analysis

Lemma F.8. *We can upper bound the risk $\mathcal{R}(\bar{\mathbf{S}})$ as follows:*

$$\mathcal{R}(\bar{\mathbf{S}}) \leq (1-\rho_c)\log\left(1+(C-1)\exp\left(2\kappa-\nu\right)\right)+\rho_c\log\left(1+(C-1)\exp\left(2\right)\right)$$

where ρ_c, κ, ν defined in Definition F.1 characterize the inner-class concentration and inter-class separation.

Proof. For a test sample $(\mathbf{x}, y) \sim \mathcal{D}_T$ with $\mathbf{z} = \mathcal{I}(\mathbf{x})$. Let $\mathbf{z} = \mathbf{v} + \bar{\mathbf{s}}_y$. By the definition of inner-class feature concentration, we have $\Pr\left(\|\mathbf{v}\|_2 \geq \kappa\right) \leq \rho_c$. Thus, we have

$$\begin{aligned}
\mathcal{R}(\bar{\mathbf{S}}) =& \mathbb{E}\left[\ell(\bar{\mathbf{S}}^\top \mathbf{z}, y)\right] && \text{(F.1)}\\
=& \mathbb{E}\left[-\log\frac{\exp\left(\bar{\mathbf{s}}_y^\top \mathbf{z}\right)}{\sum_{i\in[C]}\exp\left(\bar{\mathbf{s}}_i^\top \mathbf{z}\right)}\right] && \text{(F.2)}\\
=& \mathbb{E}\left[\log\left(1+\sum_{i\neq y}\exp\left(\bar{\mathbf{s}}_i^\top \mathbf{z}-\bar{\mathbf{s}}_y^\top \mathbf{z}\right)\right)\right] && \text{(F.3)}\\
=& \mathbb{E}\left[\log\left(1+\sum_{i\neq y}\exp\left(\bar{\mathbf{s}}_i^\top \left(\mathbf{v}+\bar{\mathbf{s}}_y\right)-\bar{\mathbf{s}}_y^\top \left(\mathbf{v}+\bar{\mathbf{s}}_y\right)\right)\right)\right] && \text{(F.4)}\\
\leq& (1-\rho_c)\mathbb{E}\left[\log\left(1+\sum_{i\neq y}\exp\left(\bar{\mathbf{s}}_i^\top \mathbf{v}+1-\nu-\bar{\mathbf{s}}_y^\top \mathbf{v}-1\right)\right)\middle|\|\mathbf{v}\|_2\leq\kappa\right] && \text{(F.5)}\\
&+\rho_c\mathbb{E}\left[\log\left(1+\sum_{i\neq y}\exp\left(2\right)\right)\middle|\|\mathbf{v}\|_2\geq\kappa\right] && \text{(F.6)}\\
\leq& (1-\rho_c)\mathbb{E}\left[\log\left(1+\sum_{i\neq y}\exp\left(2\|\mathbf{v}\|_2-\nu\right)\right)\middle|\|\mathbf{v}\|_2\leq\kappa\right] && \text{(F.7)}\\
&+\rho_c\log\left(1+(C-1)\exp\left(2\right)\right) && \text{(F.8)}\\
\leq& (1-\rho_c)\log\left(1+(C-1)\exp\left(2\kappa-\nu\right)\right)+\rho_c\log\left(1+(C-1)\exp\left(2\right)\right). && \text{(F.9)}
\end{aligned}$$

□

Remarks: Lemma F.8 is a tight upper bound. We give a simple toy example here for illustration: consider binary classification on two data points $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$. Suppose $\mathbf{z}_1 = \mathcal{I}(\mathbf{x}_1) = -\mathbf{z}_2 = -\mathcal{I}(\mathbf{x}_2)$, we can see that $\mathcal{R}(\bar{\mathbf{S}}) = \mathcal{R}(\bar{\mathbf{Z}}^*) = \log(1 + \exp(-2))$, where $C = 2, \rho_c = \kappa = 0, \nu = 2$.

Lemma F.9. *For a test sample $(\mathbf{x}, y) \sim \mathcal{D}_T$ and its image feature $\mathbf{z} = \mathcal{I}(\mathbf{x})$, with probability at least $1 - \rho_c$, we have*

$$\max_{i \neq y} \mathbf{s}_i^\top \mathbf{z} - \mathbf{s}_y^\top \mathbf{z} \leq 4\kappa - \nu.$$

Proof of Lemma F.9. Let $\mathbf{z} = \mathbf{v} + \bar{\mathbf{s}}_y$. By Definition F.1 and Assumption F.6, we have $\Pr(\|\mathbf{v}\|_2 \geq \kappa) \leq \rho_c$. Thus, we have with probability at least $1 - \rho_c$ such that

$$\begin{aligned}
\max_{i \neq y} \mathbf{s}_i^\top \mathbf{z} - \mathbf{s}_y^\top \mathbf{z} =& \max_{i \neq y} (\mathbf{s}_i - \bar{\mathbf{s}}_i + \bar{\mathbf{s}}_i)^\top (\mathbf{v} + \bar{\mathbf{s}}_y) - (\mathbf{s}_y - \bar{\mathbf{s}}_y + \bar{\mathbf{s}}_y)^\top (\mathbf{v} + \bar{\mathbf{s}}_y) && \text{(F.10)} \\
=& \max_{i \neq y} \mathbf{s}_i^\top \mathbf{v} + (\mathbf{s}_i - \bar{\mathbf{s}}_i)^\top \bar{\mathbf{s}}_y + \bar{\mathbf{s}}_i^\top \bar{\mathbf{s}}_y && \text{(F.11)} \\
& - \mathbf{s}_y^\top \mathbf{v} - (\mathbf{s}_y - \bar{\mathbf{s}}_y)^\top \bar{\mathbf{s}}_y - \bar{\mathbf{s}}_y^\top \bar{\mathbf{s}}_y && \text{(F.12)} \\
\leq& \max_{i \neq y} \kappa + \kappa + 1 - \nu + \kappa + \kappa - 1 && \text{(F.13)} \\
=& 4\kappa - \nu. && \text{(F.14)}
\end{aligned}$$

□

Remarks: From the above lemma, we can see that if $4\kappa < \nu$, the accuracy of $f^{\text{RET}}(\cdot)$ is at least $1 - \rho_c$.

Lemma F.10 (Retrieval distribution shift bound)**.** *Under Assumption F.6 and Assumption F.7 and suppose that $\mathbf{s}_i$ is in the support of $\mathcal{D}_R$, we have $\xi_{\mathbf{s}} \leq 2\kappa^2$. Furthermore, when the retrieval cluster for $\mathbf{t}_i$ and $\mathbf{s}_i$ are different for any $i \in [C]$, we have $\xi_{\mathbf{t}} \geq \nu - 2\kappa$.*

Proof of Lemma F.10. By Assumption F.6 and Assumption F.7, for any $i \in [C]$, we have

$$\begin{aligned}
\xi_i^{\text{I2I}} =& 1 - \bar{\mathbf{k}}_{\mathbf{s}_i}^\top \bar{\mathbf{s}}_i && \text{(F.15)}\\
=& \frac{1}{2} \left\| \bar{\mathbf{s}}_i - \bar{\mathbf{k}}_{\mathbf{s}_i} \right\|_2^2 && \text{(F.16)}\\
\leq& 2\kappa^2. && \text{(F.17)}
\end{aligned}$$

Furthermore, when the retrieval clusters for $\mathbf{t}_i$ and $\mathbf{s}_i$ are different, by Assumption F.7, we have

$$\begin{aligned}
\xi_i^{\text{T2I}} =& 1 - \bar{\mathbf{k}}_{\mathbf{t}_i}^\top \bar{\mathbf{s}}_i && \text{(F.18)}\\
=& 1 - \left(\bar{\mathbf{k}}_{\mathbf{t}_i}\right)^\top \left(\bar{\mathbf{s}}_i - \bar{\mathbf{k}}_{\mathbf{s}_i} + \bar{\mathbf{k}}_{\mathbf{s}_i}\right) && \text{(F.19)}\\
=& 1 - \left(\bar{\mathbf{k}}_{\mathbf{t}_i}\right)^\top \left(\bar{\mathbf{s}}_i - \bar{\mathbf{k}}_{\mathbf{s}_i}\right) - \bar{\mathbf{k}}_{\mathbf{t}_i}^\top \bar{\mathbf{k}}_{\mathbf{s}_i} && \text{(F.20)}\\
\geq& \nu - \left\| \bar{\mathbf{s}}_i - \bar{\mathbf{k}}_{\mathbf{s}_i} \right\|_2 && \text{(F.21)}\\
=& \nu - \sqrt{2\xi_i^{\text{I2I}}} && \text{(F.22)}\\
\geq& \nu - 2\kappa. && \text{(F.23)}
\end{aligned}$$

□

Theorem F.11 (Benefit of uni-modal retrieval)**.** *Assume the same condition as Lemma F.10, with probability at least $1-\delta$, the following upper bound of the ensemble risk holds:*

$$\mathcal{R}(\alpha\mathbf{T} + \gamma\bar{\mathbf{K}}) - \mathcal{R}(\bar{\mathbf{S}}) \leq L\left(\alpha \underbrace{\|(\mathbf{T} - \bar{\mathbf{S}})^\top \mathbf{z}\|_2}_{\textit{modality gap}} + \underbrace{\gamma\kappa\sqrt{\frac{8C}{K}\log\frac{C}{\delta}}}_{\textit{retrieval sample complexity}} + \underbrace{\gamma\sqrt{2C\xi}}_{\textit{retrieval shift}}\right), \tag{F.24}$$

where $L = \sqrt{\exp(2)+1}$, κ characterizes the inner-class feature concentration (Definition F.1), and ξ is either $\xi_{\mathbf{s}}$ for I2I retrieval or $\xi_{\mathbf{t}}$ for T2I retrieval.

Proof of Theorem F.11. By Lemma F.13 and Lemma F.14, let $L = \sqrt{\exp(2)+1}$, we have:

$$\mathcal{R}(\alpha\mathbf{T} + \gamma\bar{\mathbf{K}}) - \mathcal{R}(\bar{\mathbf{S}}) \leq L\left(\alpha\|(\mathbf{T}-\bar{\mathbf{S}})^\top\mathbf{z}\|_2 + \gamma\|(\bar{\mathbf{K}}-\bar{\mathbf{S}})^\top\mathbf{z}\|_2\right). \quad \text{(F.25)}$$

By the vector Bernstein inequality in Lemma F.15 and the union bound, with probability at least $1-\delta$, for any $c \in [C]$:

$$\|\bar{\mathbf{k}}_c - \bar{\mathbf{k}}_{\mathbf{q}_c}\|_2 \leq \kappa\sqrt{\frac{8}{K}\log\frac{C}{\delta}}, \quad \text{(F.26)}$$

This bound characterizes the retrieval sample complexity. Moreover, from the definition of the retrieval distributional shift, we have $\|\bar{\mathbf{k}}_{\mathbf{q}_c} - \bar{\mathbf{s}}_c\|_2 = \sqrt{2 - 2\bar{\mathbf{k}}_{\mathbf{q}_c}^\top\bar{\mathbf{s}}_c} = \sqrt{2\xi_c}$, where $\mathbf{q}_c = \mathbf{s}_c$ for I2I retrieval and $\mathbf{q}_c = \mathbf{t}_c$ for T2I retrieval. Therefore, we obtain an upper bound of $\|(\bar{\mathbf{K}}-\bar{\mathbf{S}})^\top\mathbf{z}\|_2$ as:

$$\|(\bar{\mathbf{K}}-\bar{\mathbf{S}})^\top\mathbf{z}\|_2 \leq \kappa\sqrt{\frac{8C}{K}\log\frac{C}{\delta}} + \sqrt{2C\xi} \quad \text{(F.27)}$$

We obtain the final bound by putting together Eq. (F.25) and Eq. (F.27). □

Remarks: The above upper bound consists of three terms: the gap between the textual and visual modality, the sample complexity of retrieved features which decreases as we increase K, and a term related to the distributional shift induced by the retrieval method. By Lemma F.10, we can see the superiority of I2I over T2I retrieval by comparing $\xi_{\mathbf{s}}$ and $\xi_{\mathbf{t}}$.

Theorem F.12 (Benefit of logit ensemble)**.** *Assume the same condition as Lemma F.10. For I2I retrieval with $\alpha = \gamma = \frac{1}{2}, \beta = 0$, we have*

$$\mathcal{R}_{0-1}(f) \leq \Pr(E_1) + (\Pr(E_2) + \Pr(E_3))\rho_d(\max\{6\kappa - \nu, 2\kappa + \tau\}) + \rho_c \quad \text{(F.28)}$$

Proof of Theorem F.12. We define the events

$$E_c = \{\|\mathcal{I}(\mathbf{x}) - \bar{\mathbf{s}}_i\|_2 \geq \kappa \text{ and } y = i, \forall i \in [C]\}.$$

We also define events

$$E_d(z) = \left\{\phi(\mathbf{T}^\top \mathbf{z}, y, z) \cap \phi(\bar{\mathbf{K}}^\top \mathbf{z}, y, z) = \{y\}\right\}$$

. Note that we have

$$\Pr\left(E_d(z) | E_2 \text{ or } E_3\right) = 1 - \rho_d(z)$$

. By Definition F.5, we have

$$\begin{aligned}
\max_{(\mathbf{x},y)\in E_c, y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} = & \max_{(\mathbf{x},y)\in E_c, y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top (\mathbf{z} - \bar{\mathbf{s}}_i + \bar{\mathbf{s}}_i) && \text{(F.29)} \\
= & \max_{(\mathbf{x},y)\in E_c, y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top (\mathbf{z} - \bar{\mathbf{s}}_i) + (\mathbf{t}_j - \mathbf{t}_i)^\top \bar{\mathbf{s}}_i \\
& && \text{(F.30)} \\
\leq & 2\kappa + \tau. && \text{(F.31)}
\end{aligned}$$

By Lemma F.9 and Assumption F.6 and Assumption F.7, conditional on E_c, we have the logits gap $\max_{i\neq y} \bar{\mathbf{k}}_i^\top \mathbf{z} - \bar{\mathbf{k}}_y^\top \mathbf{z} \leq 6\kappa - \nu$. Let $\mathrm{ACC}(f) =$

$1 - \mathcal{R}_{0-1}(f)$. Then, we get

$$\begin{aligned}
\mathrm{ACC}(f) &= \Pr\left(y = \arg\max_i \frac{1}{2}\mathbf{t}_i^\top \mathbf{z} + \frac{1}{2}\bar{\mathbf{k}}_i^\top \mathbf{z}\right) && \text{(F.32)}\\
&\geq \Pr\left(E_4\right) + \Pr(E_c \cap E_2) \Pr\left(y = \arg\max_i \mathbf{t}_i^\top \mathbf{z} + \bar{\mathbf{k}}_i^\top \mathbf{z} \middle| E_c \cap E_2\right) && \text{(F.33)}\\
&\quad + \Pr(E_c \cap E_3) \Pr\left(y = \arg\max_i \mathbf{t}_i^\top \mathbf{z} + \bar{\mathbf{k}}_i^\top \mathbf{z} \middle| E_c \cap E_3\right) && \text{(F.34)}\\
&= \Pr\left(E_4\right) + \Pr(E_c \cap E_2) \Pr\left(\max_{y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < 0 \middle| E_c \cap E_2\right) && \text{(F.35)}\\
&\quad + \Pr(E_c \cap E_3) \Pr\left(\max_{y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < 0 \middle| E_c \cap E_3\right). && \text{(F.36)}
\end{aligned}$$

Now, we prove that

$$E_d(6\kappa - \nu) \cap E_c \cap E_2 \subseteq \{\max_{y=i\neq j} (\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < 0\} \cap E_c \cap E_2.$$

For any $(\mathbf{x}, y) \in E_d(6\kappa - \nu) \cap E_c \cap E_2$ and $y = i \neq j$,

- if $j \in \phi(\mathbf{T}^\top \mathbf{z}, y, z)$ and $j \notin \phi(\bar{\mathbf{K}}^\top \mathbf{z}, y, z)$, we have

$$(\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < 0 - (6\kappa - \nu) \leq 0$$

- if $j \notin \phi(\mathbf{T}^\top \mathbf{z}, y, z)$ and $j \in \phi(\bar{\mathbf{K}}^\top \mathbf{z}, y, z)$, by Lemma F.9,

$$(\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < -(6\kappa - \nu) + 6\kappa - \nu = 0$$

- if $j \notin \phi(\mathbf{T}^\top \mathbf{z}, y, z)$ and $j \notin \phi(\bar{\mathbf{K}}^\top \mathbf{z}, y, z)$, we have

$$(\mathbf{t}_j - \mathbf{t}_i)^\top \mathbf{z} + (\bar{\mathbf{k}}_j - \bar{\mathbf{k}}_i)^\top \mathbf{z} < -(6\kappa - \nu) - 6\kappa - \nu < 0$$

Thus, we have

$$\max_{y=i\neq j}(\mathbf{t}_j-\mathbf{t}_i)^\top\mathbf{z}+(\bar{\mathbf{k}}_j-\bar{\mathbf{k}}_i)^\top\mathbf{z}<0.$$

Therefore, $E_d(6\kappa-\nu)\cap E_c\cap E_2\subseteq\{\max_{y=i\neq j}(\mathbf{t}_j-\mathbf{t}_i)^\top\mathbf{z}+(\bar{\mathbf{k}}_j-\bar{\mathbf{k}}_i)^\top\mathbf{z}<0\}\cap E_c\cap E_2$.

Similarly, by

$$\max_{(\mathbf{x},y)\in E_c,y=i\neq j}(\mathbf{t}_j-\mathbf{t}_i)^\top\mathbf{z}\leq 2\kappa+\tau,$$

we have $E_d(2\kappa+\tau)\cap E_c\cap E_3\subseteq\{\max_{y=i\neq j}(\mathbf{t}_j-\mathbf{t}_i)^\top\mathbf{z}+(\bar{\mathbf{k}}_j-\bar{\mathbf{k}}_i)^\top\mathbf{z}<0\}\cap E_c\cap E_3$.

Thus, as E_2 and E_3 are disjoint and union bound, we have

$$\begin{aligned}\mathrm{ACC}(f)\geq&\Pr(E_4)+\Pr(E_c\cap E_2)\Pr(E_d(6\kappa-\nu)|E_c\cap E_2) &\text{(F.37)}\\ &+\Pr(E_c\cap E_3)\Pr(E_d(2\kappa+\tau)|E_c\cap E_3) &\text{(F.38)}\\ \geq&\Pr(E_4)+(\Pr(E_2)+\Pr(E_3))(1-\rho_d(\max\{6\kappa-\nu,2\kappa+\tau\}))-\rho_c. &\text{(F.39)}\end{aligned}$$

We finish the proof by following $\mathrm{ACC}(f)=1-\mathcal{R}_{0-1}(f)$ and $\Pr(E_1)+\Pr(E_2)+\Pr(E_3)+\Pr(E_4)=1$. □

Remarks: The above theorem characterizes the 0-1 risk upper bound by the modality gap and key properties of retrieved and target distributions. Moreover, logit ensemble utilizes knowledge encoded in different modalities to benefit each other. When $(\Pr(E_2)+\Pr(E_3))(1-\rho_d(\max\{6\kappa-\nu,2\kappa+\tau\}))-\rho_c\geq\max\{\Pr(E_2),\Pr(E_3)\}$, we can see that logit ensemble leads to a lower 0-1 risk (*i.e.,* higher accuracy) compared to the zero-shot model. This happens when the modality gap τ is small and the test data exhibits good clustering properties.

F.4.4 Auxiliary Lemmas

Lemma F.13 (Lipschitz continuity of cross-entropy loss)**.** *When $y \in [C]$, the cross-entropy loss $\ell(\mathbf{v}, y)$ is L-Lipschitz on the hyper-cube, i.e., $\mathbf{v} \in [-1, 1]^C$, where $L = \sqrt{\exp(2) + 1}$.*

Proof of Lemma F.13. Note that since $\ell(\cdot, y) : \mathbb{R}^C \to \mathbb{R}$ is differentiable, it is sufficient to find L such that $\|\nabla \ell(\cdot, y)\|_2 \leq L$. Let $s = \sum_{i \in [C]} \exp(\mathbf{v}_i)$. Applying calculus rules we have that

$$\frac{\partial \ell}{\partial \mathbf{v}_y} = \frac{\exp(\mathbf{v}_y) - s}{s} \quad \text{and} \quad \frac{\partial \ell}{\partial \mathbf{v}_i} = \frac{\exp(\mathbf{v}_y + \mathbf{v}_i)}{s} \ \ \forall i \neq y. \tag{F.40}$$

Thus,

$$\|\nabla \ell(\cdot, y)\|_2^2 = \frac{\left(\sum_{i \neq y} \exp(\mathbf{v}_i)\right)^2 + \exp(2\mathbf{v}_y)\left(\sum_{i \neq y} \exp(2\mathbf{v}_i)\right)}{s^2} \tag{F.41}$$

$$\leq \frac{s^2 + \exp(2\mathbf{v}_y)s^2}{s^2} \tag{F.42}$$

$$\leq \exp(2) + 1. \tag{F.43}$$

Thus, we have $L = \sqrt{\exp(2) + 1}$. □

Lemma F.14 (Bounded logits)**.** *For an input with visual feature $\mathbf{z} \in \mathbb{R}^d$, if $\mathbf{Q}$ is a convex combination among $\{\mathbf{T}, \mathbf{S}, \bar{\mathbf{S}}, \bar{\mathbf{K}}\}$, we have $\mathbf{Q}^\top \mathbf{z} \in [-1, 1]^C$.*

Proof of Lemma F.14. From the definitions of matrices $\mathbf{T}, \mathbf{S}, \bar{\mathbf{S}}, \bar{\mathbf{K}} \in \mathbb{R}^{d \times C}$ defined in Appendix F.4.1 and Appendix F.4.2, we have that the Euclidean norm of each column in $\mathbf{T}, \mathbf{S}, \bar{\mathbf{S}}, \bar{\mathbf{K}}$ and $\mathbf{z}$ is smaller or equal to 1. Thus, their convex combination $\mathbf{Q}$ multiplied by $\mathbf{z}$ satisfies $\mathbf{Q}^\top \mathbf{z} \in [-1, 1]^C$. □

Lemma F.15 (Vector Bernstein inequality. Lemma 18 in Kohler and Lucchi (2017))**.** *Let $\mathbf{v}_1, ..., \mathbf{v}_n \in \mathbb{R}^d$ be independent vector-valued random variables and assume that each one is centered, uniformly bounded with variance bounded*

above:

$$\mathbb{E}[\mathbf{v}_i] = 0 \textit{ and } \|\mathbf{v}_i\|_2 \leq B_2 \textit{ as well as } \mathbb{E}[\|\mathbf{v}_i\|_2^2] \leq \sigma^2. \tag{F.44}$$

Let $\widehat{\mathbf{v}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i$. *Then we have for* $0 < \epsilon < \sigma^2/B_2$,

$$\Pr(\|\widehat{\mathbf{v}}\|_2 \geq \epsilon) \leq \exp\left(-n \cdot \frac{\epsilon^2}{8\sigma^2} + \frac{1}{4}\right). \tag{F.45}$$

F.5 Training-based Adaptation

In Section 7.5, we have shown the average performance of training-based adaptation, where the feature cache is finetuned (based on the RN50 backbone). In this section, we report the performance for each dataset. The results are shown in Figure F.3. The result for each dataset is consistent where I2I retrieval outperforms T2I retrieval and zero-shot CLIP when varying the shot number from 2 to 16.

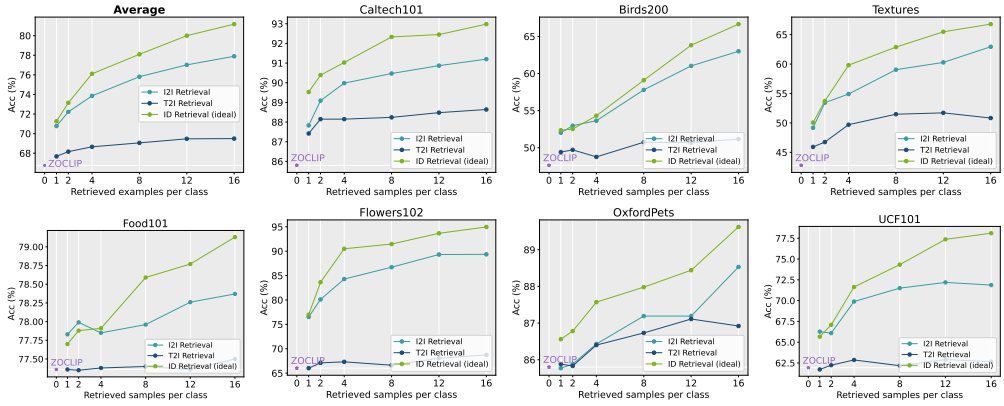


Figure F.3: Comparison of retrieval method on adaptation with finetuned feature. Results are based on RN50. We observe a trend similar to training-free adaptation, where I2I retrieval consistently outperforms T2I retrieval and zero-shot CLIP.

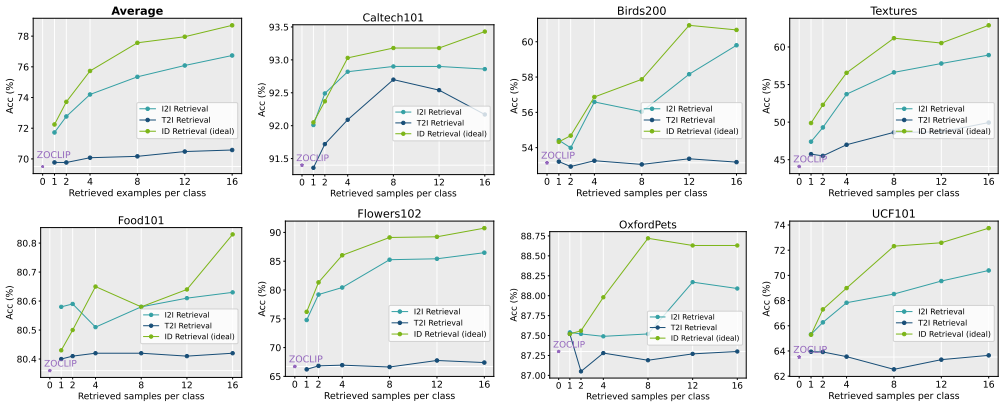


Figure F.4: Impact of model architecture. Results are based on ViT-B/32 (training-free).

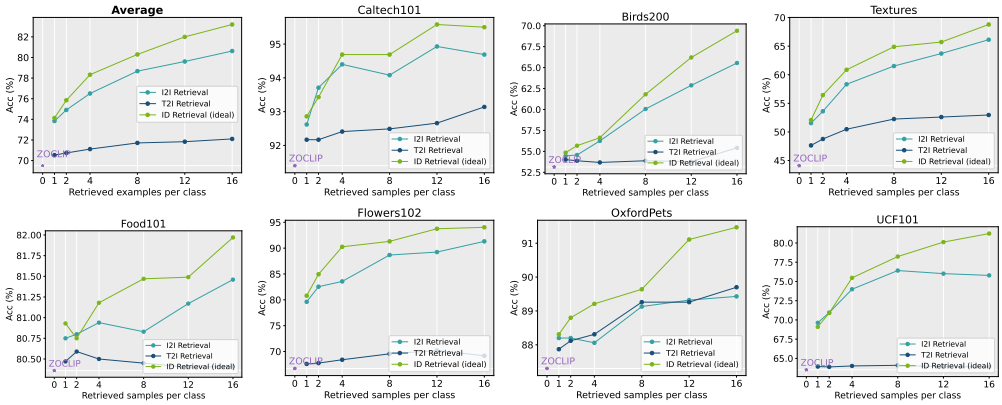


Figure F.5: Impact of model architecture. Results are based on ViT-B/32 (feature cache finetuned).

F.6 Impact of Architecture

In Section 7.5, we show the average performance over all datasets for I2I retrieval and T2I retrieval under different CLIP backbones and observe consistent trends. The results for individual datasets can be seen in Figure F.4 (training-free adaptation based on ViT-B/32), Figure F.5 (training-based adaptation based on ViT-B/32), Figure F.6 (training-free adaptation based on ViT-B/16), Figure F.7 (training-based adaptation based on ViT-

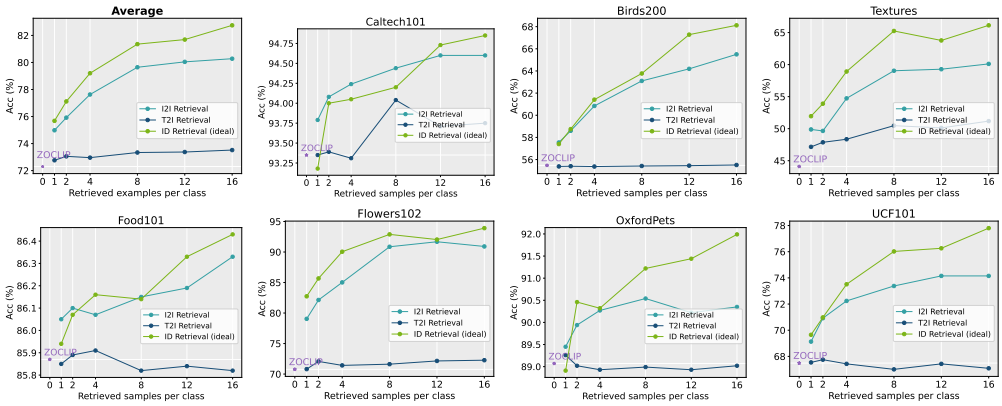


Figure F.6: Impact of model architecture. Results are based on ViT-B/16 (training-free).

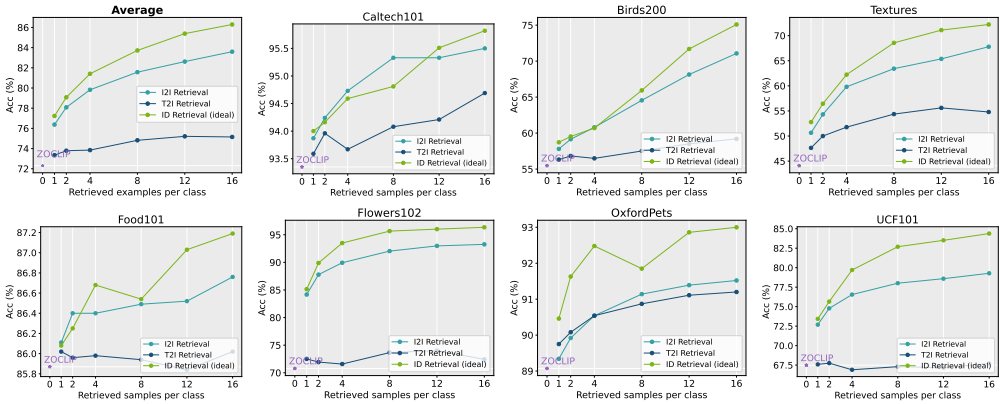


Figure F.7: Impact of model architecture. Results are based on ViT-B/16 (feature cache finetuned).

B/16), Figure F.8 (training-free adaptation based on ViT-L/14), and Figure F.9 (training-based adaptation based on ViT-L/14).

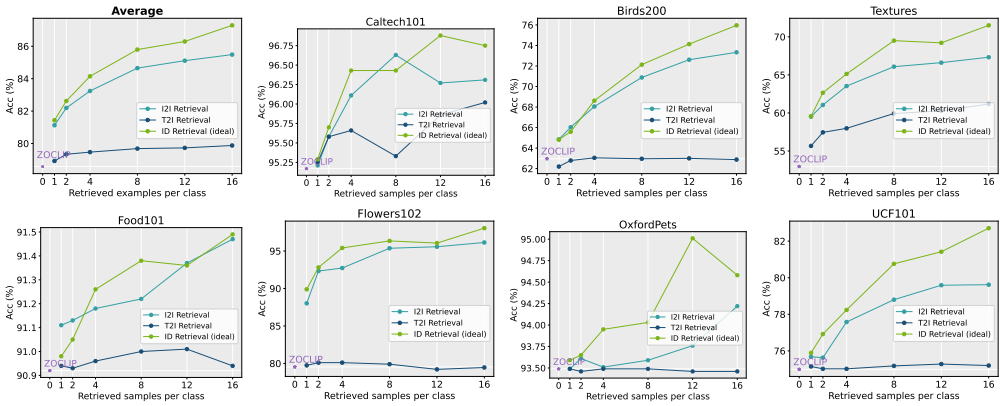


Figure F.8: Impact of model architecture. Results are based on ViT-L/14 (training-free).

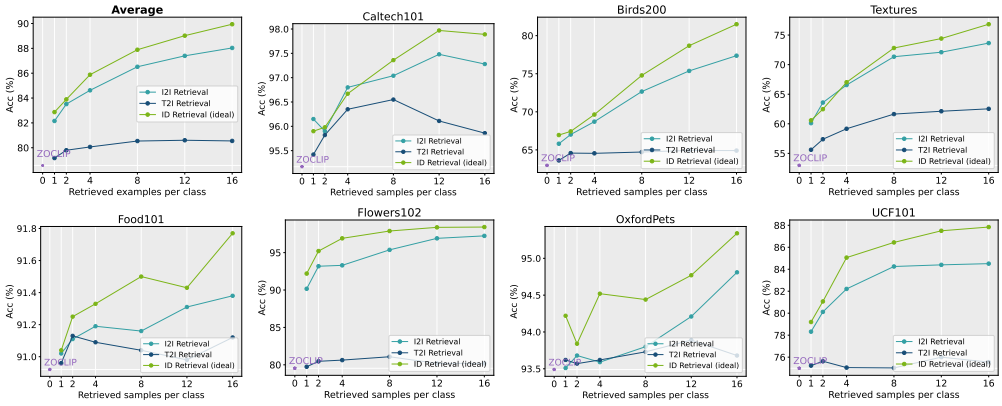


Figure F.9: Impact of model architecture. Results are based on ViT-L/14 (feature cache finetuned).

Appendix G

Appendix for A Critical Analysis of Document Out-of-Distribution Detection

G.1 Dataset and Model Details

G.1.1 Datasets

The full RVL-CDIP dataset consists of 320K/40K/40K training/validation/testing images under 16 categories. We select 12 of them as the ID (In-domain) data. We employ the Google OCR engine[1] to extract the text and layout information, which provides tokens, text blocks and the corresponding bounding boxes.

G.1.2 Quantifying OOD Dataset Construction

The distance between datasets can be measured via Optimal Transport Dataset Distance (OTDD)[2]. We visualize the OTDD distance between ID

[1] https://cloud.google.com/vision/docs/ocr
[2] https://github.com/microsoft/otdd

and the OOD (both in-domain and out-domain) data in Fig. G.1, where we highlight the in-domain OOD data in blue and the out-domain OOD data in green. Specifically, we randomly sample 1000 images from each dataset and calculate the average distance between pairs of datasets. We can see a significant gap between the OTDD of in-domain OOD data and out-domain OOD data. To make the analysis more thorough, we consider two additional in-domain OOD settings: (1) select the classes the model performs well as OOD data; (2) randomly select classes as OOD data. The results are shown in Fig. G.2 and Fig. G.3. We can see that the distance between ID and in-domain OOD is similar to the original scheme (Fig. G.1). This suggests that most in-domain OOD categories are not far from ID data.

While this paper represents an initial endeavor, we hope that our work will serve as a stepping stone towards constructing more comprehensive and diverse OOD benchmarks in the document domain, akin to those available in the NLP and natural image domain.

G.1.3 Models and Training Details

All models reported in Fig. 8.2b, except UDoc, are initialized with pre-trained weights from Huggingface[3] and fine-tuned on the full RVL-CDIP training set. During fine-tuning, we train these models on RVL-CDIP with the cross-entropy loss. The models were optimized with Adam optimizer Kingma and Ba (2014) for 30 epochs with a batch size of 50 and a learning rate of 2×10^{-5} on 8 A100 GPUs. The following are the hyper-parameters of the models used in our paper:

Text-only:

[3] https://huggingface.co/models

Figure G.1: Visualization of optimal transport dataset distance for ID and OOD (in-domain and out-domain) datasets. We highlight the in-domain OOD data in blue and the out-domain OOD data in green. OOD categories are selected based on the worst performance.

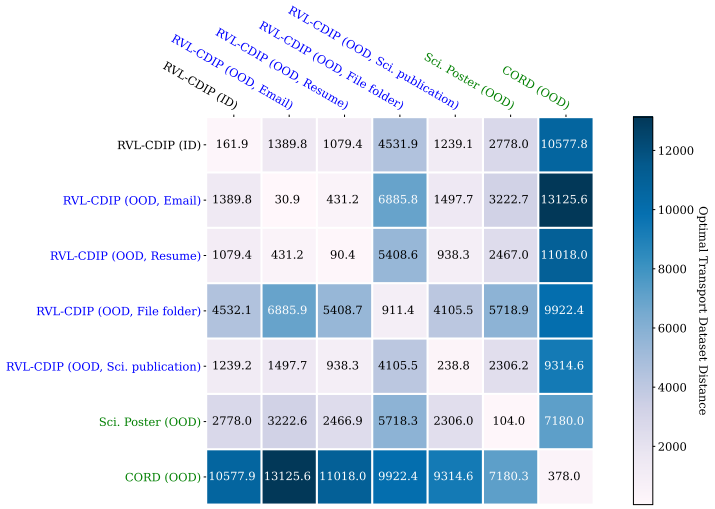


Figure G.2: Visualization of optimal transport dataset distance for ID and OOD datasets. OOD categories are selected based on the best performance.

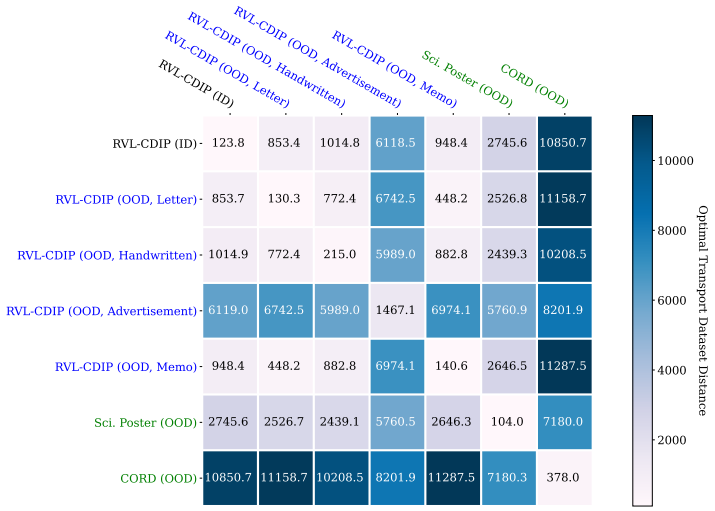
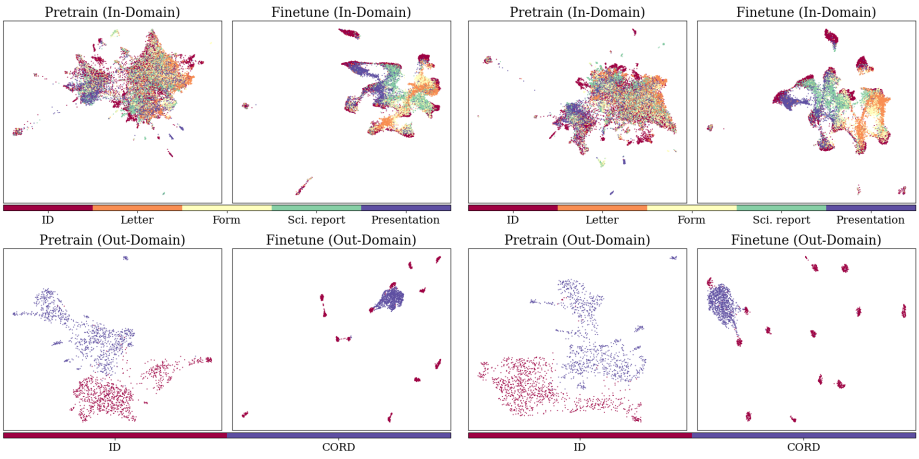


Figure G.3: Visualization of optimal transport dataset distance for ID and OOD datasets. OOD categories are selected randomly.

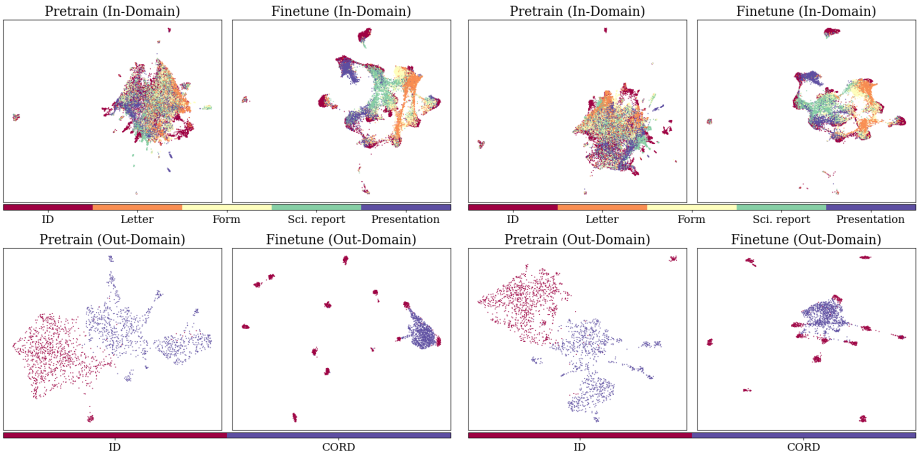
- **BERT** and **RoBERTa**: We adopt $\text{RoBERTa}_{\text{Base}}$ (12 layers) and $\text{BERT}_{\text{Base}}$ (12 layers) as backbones and set the maximum sequence length to 512. For RoBERTa, the classifier consists of two linear layers followed by a tanh activation function.
- **Longformer**$_{\text{Base}}$: We also employ $\text{Longformer}_{\text{Base}}$ (12 layers) as the backbone and set the maximum sequence length to 4,096.

Vision-only:

- **ResNet50**: We adopt ResNet50 pre-trained on ImageNet-1k as the backbone. We fine-tune the model at a resolution of 224×224.
- **ViT**: We consider ViT_{Base} (vit-base-patch16-224, pre-trained on ImageNet-21k) as the backbon and fine-tune at a resolution of 224×224.
- **SwinB**: We also use the Swin Transformer (swin-base-patch4-window7-224-in22k, pre-trained on ImageNet-21k) as the backbone and fine-tune the model at a resolution of 224×224.

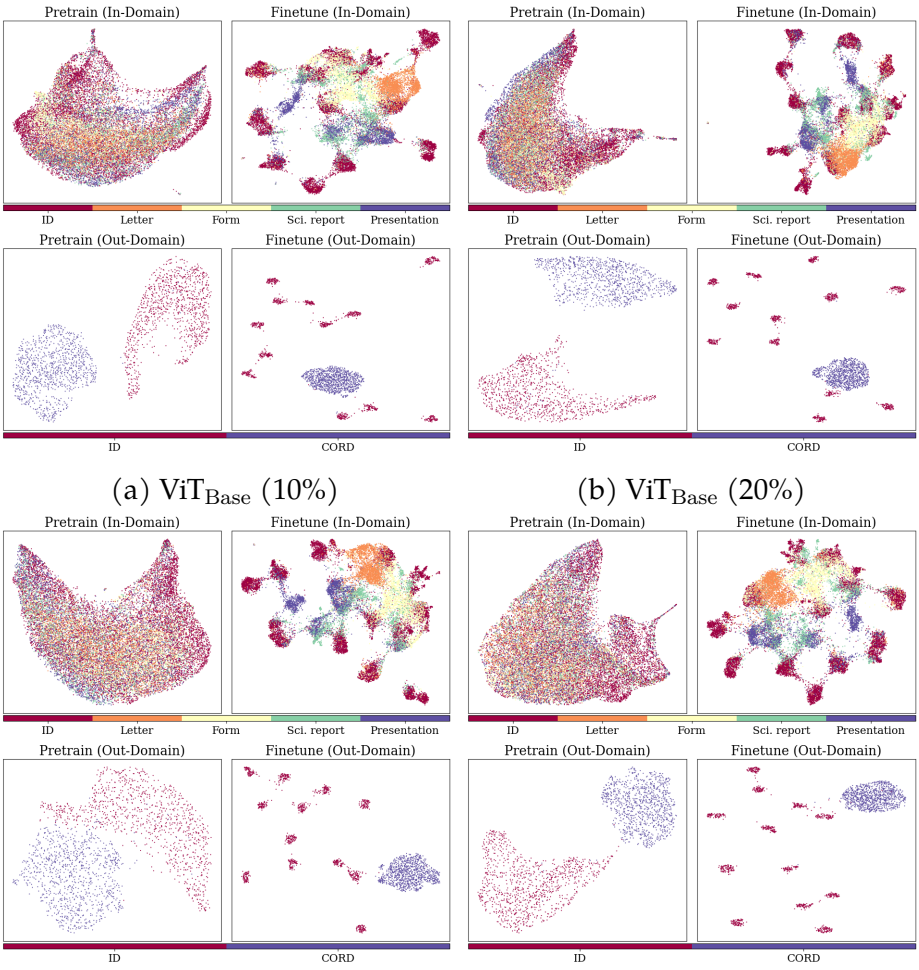


(a) RoBERTa$_{\text{Base}}$ (10%) (b) RoBERTa$_{\text{Base}}$ (20%)



(c) RoBERTa$_{\text{Base}}$ (40%) (d) RoBERTa$_{\text{Base}}$ (100%)

Figure G.4: Feature visualization for pre-trained (with different numbers of pre-training data) and fine-tuned models based on RoBERTa. We show both in-domain (RVL-CDIP) and out-domain (CORD) OOD datasets.



(a) ViT_{Base} (10%) (b) ViT_{Base} (20%)

(c) ViT_{Base} (40%) (d) ViT_{Base} (100%)

Figure G.5: Feature visualization for pre-trained (with different numbers of pre-training data) and fine-tuned models based on ViT. We show both in-domain (RVL-CDIP) and out-domain (CORD) OOD datasets.

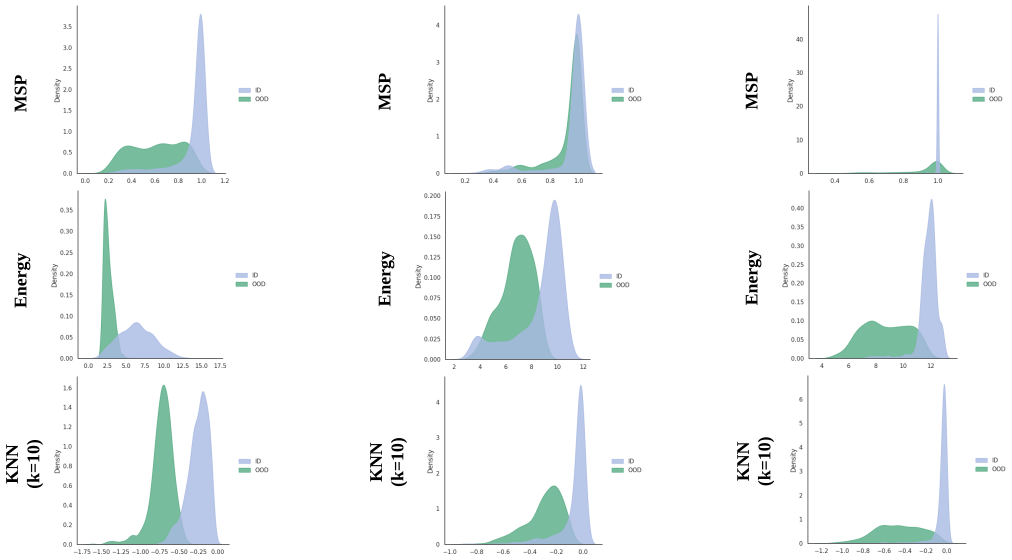


Figure G.6: MSP, Energy, KNN, and Maha score histogram distributions of ID (*blue*) and OOD (*green*) inputs derived from fine-tuned ResNet-50, RoBERTa, and LayoutLMv3. The KNN scores calculated from both vision and language models naturally form smooth distributions. In contrast, MSP and Maha scores for both in- and out-of-distribution data concentrate on high values. Overall our experiments show that using feature space makes the scores more distinguishable between and out-of-distributions and, as a result, enables more effective OOD detection.

Text+Layout:

- **LayoutLMv1**: This model employs the LayoutLM (layoutlm-base-uncased, 12 layers, pre-trained on IIT-CDIP) as the backbone. We set the maximum sequence length to 512.

- **Spatial-RoBERTa$_{\text{Base}}$ (Pre)**: This model combines our spatial-aware adapter to the pre-trained RoBERTa$_{\text{Base}}$ model. The adapter is applied to the word embedding layer. We freeze the pre-trained word embeddings and optimize the spatial-aware adapter and transformers.

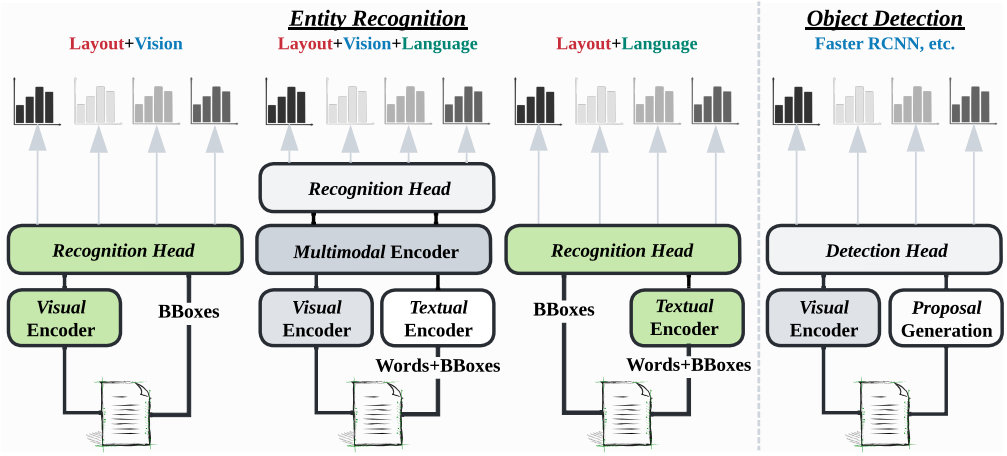


Figure G.7: The network architectures in green blocks are our proposed models. We also show the modality information on top of each architecture.

- **Spatial-RoBERTa$_{\text{Base}}$ (Post)**: Instead of inserting the spatial-aware adapter in the input layer, this model integrates the spatial-aware adapter at the output layer of the transformer.

Vision+Text+Layout:

- **LaytouLMv3**: We use LayoutLMv3 (layoutlmv3-base, 12 layers, pre-trained on IIT-CDIP) as the backbone.
- **UDoc**: We use a slight variant of UDoc with the only difference in the sentence encoder, where we adopt a smaller version of the pre-trained sentence encoder (all-MiniLM-L6-v2, 6 layers) instead of the larger sentence encoder (bert-base-nli-mean-tokens, 12 layers).

G.2 Beyond Document Classification

In the main paper, we mainly focus on document classification to provide a thorough and in-depth analysis. In this section, we go beyond document classification and explore OOD detection for two entity-level tasks in documents: document entity recognition and document object detection. It is natural to detect and recognize basic units in documents such as text, tables, and figures. Document entity recognition aims to predict the label for each semantic entity with given bounding boxes. Document object detection is an object detection task for document images. Specifically, we denote the input as x, the bounding box coordinates associated with object instances in the image as $\boldsymbol{b} \in \mathbb{R}^4$, and use the model with parameters θ to model the bounding box regression $p_\theta(\boldsymbol{b}|x)$ and the label classification $p_\theta(y|x, \boldsymbol{b})$. Given a test input $\hat{x}$, the OOD detection scoring function for entity detection and recognition can be unified as $S(\hat{x}, \hat{\boldsymbol{b}})$, where $\hat{\boldsymbol{b}}$ denotes the object instance predicted by the object detector. In particular, for document entity recognition, since the bounding boxes are provided, the OOD score can be simplified as $S(\hat{x}, \bar{\boldsymbol{b}})$, where $\bar{\boldsymbol{b}}$ is the given object instance.

Document Object Detection. For document object detection, we use PubLayNet as the ID dataset and construct the OOD dataset from IIIT-AR-13K. Unlike PubLayNet, where the documents are scientific articles, IIIT-AR-13K is a dataset for graphical object detection in business documents (*e.g.*, , annual reports), thus there exists an obvious domain gap. We select *natural images* as the OOD entity and filter images that contain the OOD entity. Two object detection models are considered in this paper: (*1*) Vanilla Faster-RCNN with ResNet-50 visual backbone, and (*2*) Faster-RCNN with VOS Du et al. (2022b), a recent unknown-aware learning framework to improve OOD detection performance for natural images. Following the original paper, we use 1,000 samples for each ID class to

estimate the class-conditional Gaussian statistics. The models are trained for 180k iterations with a base learning rate of 0.01 and a batch size of 8 using the Detectron2 framework Wu et al. (2019). The performance of the models is measured using the mean average precision (MAP) @ intersection over union (IOU) [0.50:0.95] of bounding boxes.

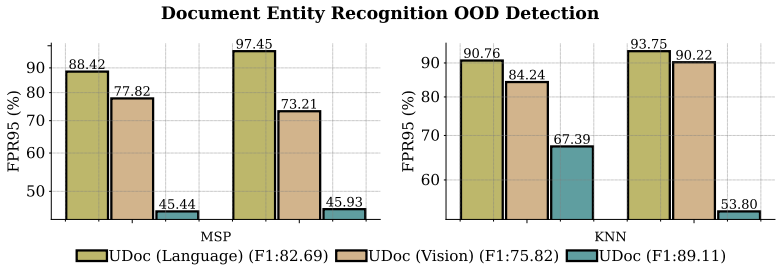
Document Entity Recognition. For entity recognition, we construct ID and OOD datasets from FUNSD. Each semantic entity includes a list of words, a label, and a bounding box. The standard label set for this dataset contains four categories: *question*, *answer*, *header*, and *other*. In this paper, we select entities labeled as *other* or *header* as OOD data, and the entities belonging to the other three categories as ID. Instead of treating entity recognition as a named-entity recognition problem, we follow UDoc and solve this problem at the semantic region level. We replace the sentence encoder in UDoc with a smaller sentence encoder (all-MiniLM-L6-v2[4]) from Huggingface Wolf et al. (2019). We also have the following model variants to verify the effectiveness of the combination of modalities: textual-only, visual-only, textual+spatial, visual+spatial, and visual+textual+spatial.

We provide details on datasets and models as follows.

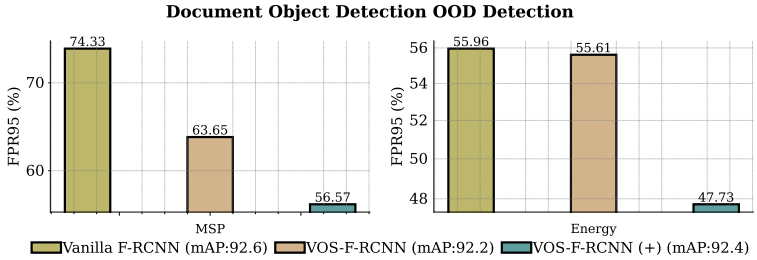
G.2.1 Datasets

The original FUNSD Jaume et al. (2019) dataset contains 149 training and 50 testing images. For document entity recognition, we treat entities with the category *other*/*header* as OOD entities. After the split, if we consider *other* as OOD, we have a total of 8,330 ID and 1,019 OOD entities. Otherwise, if we consider *header* as OOD, we have 8,981 ID and 368 OOD entities in total.

[4] https://huggingface.co/sentence-transformers



(a) Comparison of OOD detection methods on different models on two OOD classes: *other* and *header*.



(b) OOD detection results from different object detection methods and models.

Figure G.8: Ablation on document entity recognition and object detection. Numbers are reported in FPR95.

For document object detection, we consider PubLayNet Zhong et al. (2019), which contains 336K/11K training/validation images with 6 categories (*text*, *title*, *list*, *fig.*, and *table*). The original IIIT-AR-13K Mondal et al. (2020) contains (*table*, *fig.*, *natural image*, *logo*, and *signature*). In this paper, considering the overlap between IIIT-AR-13K and PubLayNet, we select those images containing *natural images* as the OOD test set. After filtering, we obtain 2,880 OOD entities across 1,837 document images.

We consider three ID datasets in this experiment. (*1*) PubLayNet: This is the original PubLayNet dataset. We treat all the entities in training/validation images as ID entities. (*2*) Considering the domain shift between ID data (PubLayNet) and OOD data (IIIT-AR-13K). We combine the PubLayNet training data with the images from IIIT-AR-13K with overlapping annotations (*table* and *figure*) and train the object detection model.

G.2.2 Models

Fig. G.7 illustrates the entity recognition models used in this paper. We consider the entities on regions instead of tokens, as regions provide richer semantic information. As for the pre-trained model, we adopt UDoc (trained on IIT-CDIP) since it models inputs at the regional level. Based on the UDoc framework, we develop the following models.

Vision/Vision+Layout:

- **ResNet-50:** This model is composed of the ResNet-50 from pre-trained UDoc. It adopts the RoI pooling followed by a classifier to extract the entity features.
- **ResNet-50+Position:** This model also adapts UDoc's pre-trained ResNet-50 for further improvement. It makes the RoI features spatially aware by adding position embeddings, which are mapped from the bounding boxes via a linear mapping layer.

Text/Text+Layout:

- **Sentence BERT:** This model adopts the language branch of UDoc and appends the classifier to the output of the sentence encoder.
- **Sentence BERT+Position:** This model is close to the above model but adds position embeddings to the sentence embeddings.

Vision+Text+Layout:

- **ResNet-50+sentence BERT:** This model follows the same framework as UDoc, but replaces the sentence encoder in their original design with a more miniature sentence encoder (all-MiniLM-L6-v2).
- **SwinT+Sentence BERT:** This model replaces the ResNet-50 visual backbone with a pre-trained tiny Swin Transformer (swin-tiny-patch4-window7-224) adopted from the Huggingface.

All the models are fine-tuned with the cross-entropy loss for 100 epochs, using a learning rate of 10^{-5} and a batch size of 8 on an A100 GPU.

G.2.3 Summary of Observations

We provide a summary of observations here and hope to inspire future works on a thorough investigation of OOD detection for entity-level tasks. To identify entity types, models should not only understand the words but also utilize spatial and visual information.

For document entity recognition, the comparison of distance-based and logit-based OOD detection methods with different models are shown in Fig. G.8a. More details are shown in Table G.2. We see that models can better predict the entity type and also achieve better OOD robustness with the help of spatial information. Considering the weak language dependency between entities, it is not surprising that vision-based models achieve better performance than text-based models. In particular, UDoc with ResNet-50 achieves the best performance on two OOD test sets, illustrating that visual information plays a major role in increasing the discrimination of entities with similar semantics. For document object detection, we summarize our findings in Fig. G.8b and describe them in more detail in Table G.1. We can see that the OOD detection performance is further improved by introducing document images from IIIT-AR-13K with the same ID annotations as training data.

To provide more intuitions, in Fig. G.9, we visualize the document entity recognition OOD detection results. In Fig. G.10, we visualize the prediction on sample OOD images, using object detection models trained without VOS (top) and with VOS (bottom), respectively. We can see that vanilla Faster RCNN trained on PubLayNet produces false positives when applied to the OOD document images from IIIT-AR-13K. Table G.1 shows that introducing the unknown-aware learning method optimized for both ID and OOD can reduce the FPR95 while preserving the mAP on the ID

data. This experiment indicates that incorporating uncertainty estimation into the entity detection training procedure can improve the reliability of the document object detection system.

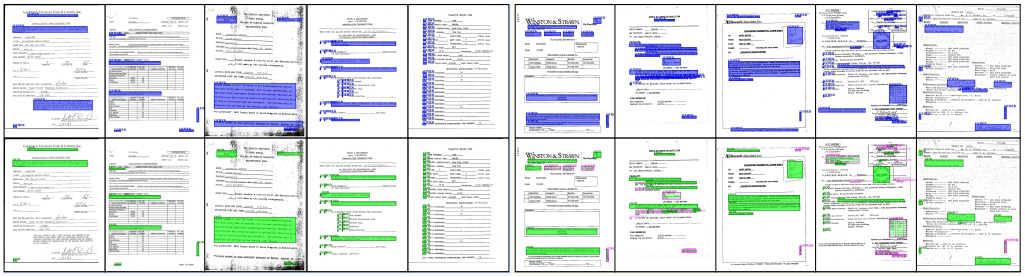
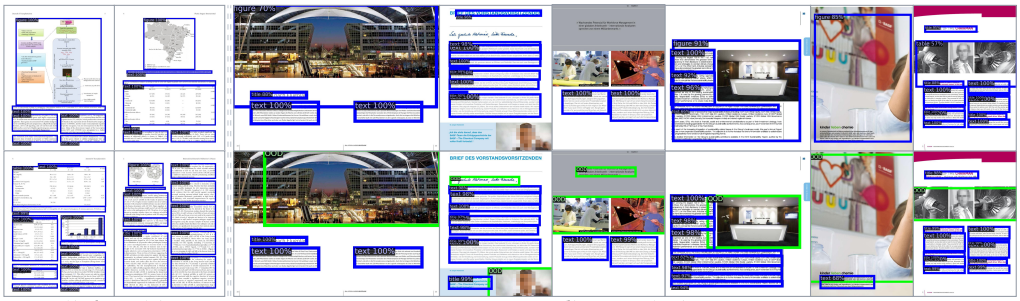


Figure G.9: Visualization of detected OOD entities on the form images. The top part shows the entities in blue are entities annotated as *other*. The bottom part shows the detected OOD entities (green). We also show failure cases on the right part.



(a) PubLayNet (ID) (b) IIIT-AR-13K (OOD)

Figure G.10: Visualization of detected objects on the OOD images (from IIIT-AR-13K) by a vanilla Faster-RCNN (top) and Faster-RCNN with VOS (bottom) is shown. Objects in blue boxes are detected and classified as one of the ID classes. The detected OOD objects (green) reduce false positives among detected objects. We also visualize detected objects on the ID images. There is a clear difference between PubLayNet and IIIT-AR-13K – entities and annotations of *natural images* rarely exist in PubLayNet.

G.3 Detailed Experimental Results

- Table G.2 corresponds to the results shown in Fig. G.9 and Fig. G.8a.

Table G.1: Comparison with different training and detection methods.

Models	ID Dataset	OOD Score	IIIT-AR-13K (*Natural Image* as OOD)			PubLayNet (ID)
			FPR95	AUROC	AUPR	mAP
Vanilla Faster-RCNN	PubLayNet	MSP	74.33	79.12	98.41	92.6
		Energy	55.96	83.55	98.73	
Faster-RCNN with VOS	PubLayNet	MSP	63.65	79.37	98.57	92.2
		Energy	55.61	80.60	98.67	
Faster-RCNN with VOS	PubLayNet+IIIT-AR-13K(ID)	MSP	56.57	82.94	98.59	92.4
		Energy	47.73	84.04	98.67	

Table G.2: Comparison with different models on FUNSD OOD setting. All models are initialized with UDoc pre-trained on IIT-CDIP and fine-tuned on FUNSD data with ID entities. All values are percentages. S-BERT deontes Sentence BERT. A lower FPR95 or a higher AUROC value indicates better performance.

	Test F1	Method	Other (OOD)		ID	Header (OOD)		ID		Test F1	Method	Other (OOD)		ID	Header (OOD)		ID
			FPR95	AUROC	F1	FPR95	AUROC	F1				FPR95	AUROC	F1	FPR95	AUROC	F1
ResNet-50	75.15	KNN_{10}	59.47	79.14	77.65	81.79	63.97	78.04	ResNet-50+Position	75.82	KNN_{10}	73.21	73.19	77.65	90.22	61.42	77.98
		KNN_{20}	69.97	78.15		81.25	63.66				KNN_{20}	72.91	73.44		88.04	61.54	
		KNN_{50}	84.49	77.40		82.61	62.86				KNN_{50}	75.96	74.43		82.88	60.93	
		KNN_{100}	97.94	77.08		84.24	61.62				KNN_{100}	79.69	74.85		83.70	59.39	
		KNN_{200}	97.84	77.15		94.29	59.74				KNN_{200}	86.06	75.14		91.58	57.42	
		KNN_{400}	97.15	76.09		94.84	57.53				KNN_{400}	87.93	74.92		95.92	55.37	
		MSP	50.54	75.80		75.82	76.55				MSP	77.82	67.60		84.24	66.58	
		MaxLogit	52.40	73.70		73.64	76.72				MaxLogit	76.94	67.05		84.24	65.41	
		Energy	52.50	73.70		75.82	76.55				Energy	76.64	66.93		84.51	64.98	
S-BERT	77.15	KNN_{10}	93.72	48.44	82.12	92.66	60.99	82.41	S-BERT+Position	82.69	KNN_{10}	97.45	41.24	87.08	93.75	62.38	87.01
		KNN_{20}	93.92	47.65		92.93	59.00				KNN_{20}	97.55	39.91		93.48	61.51	
		KNN_{50}	93.62	48.94		93.21	57.90				KNN_{50}	97.15	39.56		92.39	61.76	
		KNN_{100}	93.92	48.79		93.21	55.07				KNN_{100}	97.06	41.67		91.85	60.99	
		KNN_{200}	93.92	47.85		93.48	52.86				KNN_{200}	96.57	41.85		89.67	59.08	
		KNN_{400}	94.11	46.21		95.38	49.86				KNN_{400}	97.25	40.83		90.22	54.03	
		MSP	93.62	54.91		94.29	52.14				MSP	88.42	61.11		90.76	59.58	
		MaxLogit	93.72	54.75		94.57	56.51				MaxLogit	89.70	60.19		88.86	60.92	
		Energy	93.23	54.88		93.21	58.22				Energy	90.48	59.61		89.95	61.12	
ResNet-50+S-BERT	89.11	KNN_{10}	45.93	87.85	93.13	53.80	87.97	93.18	SwinT+S-BERT	86.00	KNN_{10}	63.30	83.64	90.82	81.52	64.08	90.40
		KNN_{20}	53.58	86.71		55.71	87.06				KNN_{20}	66.73	82.53		81.52	61.50	
		KNN_{50}	73.21	84.36		62.77	85.49				KNN_{50}	70.17	80.21		82.34	57.77	
		KNN_{100}	89.70	83.01		69.02	83.60				KNN_{100}	83.91	77.71		83.15	54.97	
		KNN_{200}	96.66	81.90		75.54	80.85				KNN_{200}	95.39	75.79		95.38	50.57	
		KNN_{400}	98.82	81.00		91.58	77.42				KNN_{400}	96.76	75.49		99.73	47.45	
		MSP	45.44	87.82		67.39	72.85				MSP	69.28	70.70		80.71	52.02	
		MaxLogit	45.53	90.58		63.04	72.39				MaxLogit	67.12	74.41		81.79	52.77	
		Energy	45.53	90.57		63.86	72.37				Energy	67.22	74.41		81.79	52.77	

- Table G.1 corresponds to the results shown in Fig. G.10 and Fig. G.8b.
- Table G.3 and Table G.7 correspond to the results shown in Fig. 8.5a.
- Table G.4 and Table G.5 correspond to the results shown in Fig. 8.5b.
- Table G.6 corresponds to the results shown in Fig. 8.9 and Fig. 8.10.
- Table G.9 and Table G.8 correspond to the results shown in Fig. 8.7 and Fig. 8.10.

- Table G.10 and Table G.11 correspond to the analysis for Sec. 8.4 and Sec. 8.4.2.
- Table G.12 corresponds to the results shown in Fig. 8.10.

Table G.3: OOD detection performance for document classification with different number of pre-training data from IIT-CDIP. ID (Acc) denotes the ID accuracy obtained by testing on ID test data. We report the KNN-based scores for both pre-trained and fine-tuned models. *Sci. Poster* denotes the document images converted from NJU-Fudan Paper-Poster Dataset. *Receipt* denotes the receipt images collected from the CORD receipt understanding dataset. For in-domain OOD test data, we also report the averaged scores.

	ID	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
	Acc		FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
RoBERTa$_{\text{Base}}$ (10%)	**Pre-train on 10% IIT-CDIP→ fine-tune on RVL-CDIP ID data**															
	90.59	MSP	92.75	69.24	92.21	66.93	94.65	65.40	92.00	70.09	92.90	67.92	96.51	66.93	99.10	52.90
		MaxLogit	98.36	77.85	97.23	78.51	98.76	72.84	98.86	78.08	98.30	76.82	100.00	78.69	100.00	63.74
		Energy	98.60	77.81	97.55	78.49	98.96	72.79	98.94	78.00	98.51	76.77	100.00	78.68	100.00	63.70
		GradNorm	98.04	79.26	97.07	76.85	98.56	72.83	98.62	80.55	98.07	77.37	100.00	85.23	100.00	64.10
		KNN_{10}	63.21	88.18	65.81	88.05	73.02	84.63	67.74	88.92	67.45	87.44	69.77	88.49	90.50	84.44
		KNN_{20}	63.53	88.07	65.89	87.90	72.75	84.48	67.33	88.81	67.38	87.32	68.60	88.13	91.10	84.09
		KNN_{50}	64.17	87.89	66.97	87.77	73.34	84.23	67.21	88.60	67.92	87.12	72.09	87.47	91.60	83.59
		KNN_{100}	64.49	87.64	67.78	87.55	73.46	83.94	67.29	88.37	68.26	86.88	72.09	86.83	91.50	83.21
	Pre-train on 10% IIT-CDIP (no fine-tune)															
	–	KNN_{10}	88.07	66.94	92.13	66.62	94.13	61.90	94.40	54.57	92.18	62.51	67.44	87.04	62.10	84.94
		KNN_{20}	88.59	66.02	92.65	65.25	94.13	60.83	94.72	53.79	92.52	61.47	77.91	85.38	64.60	83.86
		KNN_{50}	89.75	64.40	93.53	63.12	94.37	58.98	95.17	52.33	93.20	59.71	83.72	82.97	69.20	82.29
		KNN_{100}	90.23	62.94	93.85	61.28	94.41	57.45	95.13	51.28	93.40	58.24	83.72	80.91	70.10	81.05
RoBERTa$_{\text{Base}}$ (20%)	**Pre-train on 20% IIT-CDIP→ fine-tune on RVL-CDIP ID data**															
	90.71	MSP	94.28	68.02	94.46	65.98	96.01	62.98	94.81	65.98	94.89	65.74	95.35	63.55	99.10	54.99
		MaxLogit	97.36	77.82	97.19	79.16	98.40	72.64	98.34	77.68	97.82	76.82	100.00	77.36	99.60	66.63
		Energy	98.04	77.80	97.43	79.15	98.76	72.61	98.58	77.64	98.20	76.80	100.00	77.32	99.60	66.61
		GradNorm	97.36	80.68	96.83	76.04	98.44	73.29	97.89	81.37	97.63	77.85	100.00	86.18	99.50	67.49
		KNN_{10}	63.57	88.30	67.06	87.06	73.66	83.92	73.09	87.80	69.34	86.77	69.77	88.01	87.60	83.81
		KNN_{20}	63.85	88.20	67.46	86.90	73.94	83.78	72.93	87.70	69.54	86.64	69.77	87.63	88.30	83.53
		KNN_{50}	63.89	88.02	67.54	86.71	74.38	83.55	72.24	87.46	69.51	86.43	70.93	87.09	88.20	83.12
		KNN_{100}	64.85	87.81	67.62	86.45	74.90	83.25	72.65	87.24	70.00	86.19	72.09	86.65	88.30	82.89
	Pre-train on 20% IIT-CDIP (no fine-tune)															
	–	KNN_{10}	87.15	68.27	90.88	66.89	92.26	62.39	95.01	53.02	91.32	62.64	43.02	92.29	57.00	87.67
		KNN_{20}	87.31	67.35	92.04	65.54	91.54	61.40	94.97	52.33	91.46	61.66	47.67	91.18	62.60	86.61
		KNN_{50}	88.39	65.71	92.69	63.45	92.18	59.57	95.25	50.97	92.13	59.92	56.98	89.64	65.70	85.20
		KNN_{100}	88.83	64.20	93.13	61.61	92.22	57.99	95.45	49.95	92.41	58.44	58.14	88.36	66.90	84.17
RoBERTa$_{\text{Base}}$ (40%)	**Pre-train on 40% IIT-CDIP→ fine-tune on RVL-CDIP ID data**															
	90.76	MSP	92.67	70.09	93.93	65.69	95.05	63.19	95.50	65.54	94.29	66.13	95.35	63.63	95.40	64.97
		MaxLogit	98.08	78.72	97.87	79.85	98.44	71.63	98.30	75.41	98.17	76.40	98.84	78.07	98.90	75.65
		Energy	98.48	78.69	97.91	79.83	98.68	71.61	98.50	75.40	98.39	76.38	100.00	78.04	98.50	75.60
		GradNorm	98.04	81.03	97.47	76.73	98.44	72.77	97.40	79.11	97.84	77.41	100.00	87.47	97.60	77.12
		KNN_{10}	60.57	88.79	68.86	86.36	75.26	83.55	73.90	87.12	69.65	86.46	67.44	89.90	72.70	89.49
		KNN_{20}	61.37	88.72	69.06	86.24	75.46	83.43	73.46	87.00	69.84	86.35	68.60	89.66	73.50	89.25
		KNN_{50}	62.21	88.52	69.18	86.08	75.66	83.21	73.42	86.71	70.12	86.13	70.93	89.20	74.70	88.89
		KNN_{100}	63.77	88.30	69.79	85.84	76.02	82.93	74.19	86.46	70.94	85.88	74.42	88.84	75.30	88.69
	Pre-train on 40% IIT-CDIP (no fine-tune)															
	–	KNN_{10}	85.71	69.08	90.84	68.68	90.46	62.52	94.76	51.76	90.44	63.01	25.58	95.83	57.30	88.60
		KNN_{20}	85.27	68.21	91.64	67.48	89.74	61.32	94.81	51.01	90.36	62.00	29.07	95.22	62.30	87.61
		KNN_{50}	86.19	66.60	92.21	65.54	90.30	59.35	94.93	49.60	90.91	60.27	41.86	94.32	66.80	86.25
		KNN_{100}	87.19	65.04	92.57	63.83	90.50	57.74	95.09	48.44	91.34	58.76	45.35	93.66	68.30	85.14
RoBERTa$_{\text{Base}}$ (100%)	**Pre-train on 100% IIT-CDIP→ fine-tune on RVL-CDIP ID data**															
	91.00	MSP	93.23	68.88	94.54	65.83	96.65	63.11	94.12	68.28	94.64	66.53	98.84	62.52	95.10	71.25
		MaxLogit	97.84	78.86	97.95	80.23	98.48	74.01	98.25	77.59	98.13	77.67	100.00	78.73	98.90	79.36
		Energy	98.20	78.84	97.95	80.22	98.52	74.00	98.78	77.55	98.36	77.65	100.00	78.72	98.70	79.29
		GradNorm	97.88	80.81	97.91	76.37	98.28	75.25	98.25	80.09	98.08	78.13	100.00	86.10	98.30	77.50
		KNN_{10}	62.57	88.26	68.90	86.96	72.39	84.73	70.37	88.23	68.56	87.04	72.09	89.97	65.90	90.51
		KNN_{20}	63.41	88.11	69.59	86.88	73.10	84.56	70.70	88.11	69.20	86.92	74.42	89.58	67.20	90.37
		KNN_{50}	63.85	87.87	69.79	86.79	73.90	84.30	71.14	87.87	69.67	86.71	76.74	88.95	67.90	90.22
		KNN_{100}	65.13	87.61	70.27	86.58	74.86	84.00	71.75	87.65	70.50	86.46	79.07	88.44	68.30	90.19
	Pre-train on 100% IIT-CDIP (no fine-tune)															
	–	KNN_{10}	84.43	70.20	90.20	68.54	90.98	63.18	94.72	52.16	90.08	63.52	27.91	94.10	46.00	91.37
		KNN_{20}	84.51	69.30	91.28	67.35	90.38	61.96	94.72	51.43	90.22	62.51	33.72	93.39	51.50	90.55
		KNN_{50}	85.67	67.75	91.92	65.35	90.82	59.79	94.89	49.77	90.82	60.66	39.53	92.28	56.70	89.32
		KNN_{100}	86.55	66.08	92.97	63.46	91.46	58.00	95.41	48.39	91.60	58.98	44.19	91.29	61.60	88.18

Table G.4: OOD detection performance for document classification with different number of pre-training data from IIT-CDIP$^-$ (remove *pseudo* OOD categories).

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
RoBERTa$_{\text{Base}}$ (10%)	**Pre-train on 10% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	90.62	MSP	90.07	69.00	89.92	68.86	92.58	64.16	91.07	66.78	90.91	67.20	96.51	54.47	96.70	59.63
		MaxLogit	97.76	78.40	97.71	80.58	98.64	71.26	98.70	76.38	98.20	76.66	100.00	73.51	99.80	73.32
		Energy	98.16	78.35	97.75	80.55	98.84	71.20	98.90	76.32	98.41	76.60	100.00	73.46	99.80	73.31
		GradNorm	97.68	79.92	97.27	79.42	98.56	71.31	98.50	79.44	98.00	77.52	100.00	82.62	99.60	75.85
		KNN_{10}	65.85	87.89	66.69	88.12	75.98	82.82	74.55	86.85	70.77	86.42	87.21	85.16	83.90	87.91
		KNN_{20}	66.33	87.80	66.85	88.04	75.94	82.70	73.94	86.75	70.76	86.32	87.21	84.63	83.60	87.71
		KNN_{50}	66.77	87.66	67.30	88.00	76.02	82.49	73.66	86.52	70.94	86.17	88.37	83.73	83.90	87.34
		KNN_{100}	67.25	87.42	67.74	87.84	76.18	82.18	73.99	86.26	71.29	85.92	89.53	82.85	83.90	86.98
	Pre-train on 10% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN_{10}	86.35	65.48	85.74	70.84	92.94	59.55	93.14	56.62	89.54	63.12	29.07	95.42	87.60	83.13
		KNN_{20}	86.87	64.48	87.14	69.68	93.30	58.41	93.30	55.91	90.15	62.12	37.21	94.75	88.00	81.44
		KNN_{50}	87.75	62.73	88.99	67.80	93.50	56.54	93.75	54.52	91.00	60.40	47.67	93.71	90.30	78.97
		KNN_{100}	88.43	61.17	89.59	66.05	93.62	54.91	93.99	53.40	91.41	58.88	48.84	93.09	91.50	77.00
RoBERTa$_{\text{Base}}$ (20%)	**Pre-train on 20% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	90.65	MSP	96.04	67.58	94.90	68.32	96.05	64.92	96.23	68.62	95.80	67.36	100.00	61.49	98.70	56.38
		MaxLogit	97.96	76.92	97.59	80.68	98.48	72.31	98.74	77.72	98.19	76.91	100.00	75.91	99.50	69.21
		Energy	98.16	76.89	98.23	80.65	98.88	72.26	99.07	77.67	98.58	76.87	100.00	75.89	99.50	69.18
		GradNorm	97.84	78.23	97.31	78.57	98.00	71.44	98.46	80.03	97.90	77.07	100.00	85.80	99.00	69.54
		KNN_{10}	66.05	87.60	67.70	87.94	73.42	83.10	73.50	87.96	70.17	86.65	77.91	90.19	90.10	84.32
		KNN_{20}	66.17	87.50	68.38	87.83	73.90	82.93	73.66	87.82	70.53	86.52	77.91	89.84	89.80	84.13
		KNN_{50}	67.21	87.26	68.46	87.73	74.18	82.63	73.66	87.58	70.88	86.30	79.07	89.24	89.60	83.80
		KNN_{100}	68.78	86.98	69.14	87.53	75.50	82.30	74.27	87.36	71.92	86.04	82.56	88.68	89.80	83.59
	Pre-train on 20% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN_{10}	85.63	66.10	85.17	70.34	92.58	60.29	93.43	56.85	89.20	63.40	30.23	95.72	83.20	83.84
		KNN_{20}	86.31	65.17	85.98	69.13	93.30	59.09	93.47	56.05	89.77	62.36	34.88	95.08	84.90	82.16
		KNN_{50}	87.31	63.50	87.63	67.11	93.38	57.17	94.16	54.60	90.62	60.60	44.19	94.07	87.50	79.74
		KNN_{100}	87.83	62.06	88.27	65.31	93.62	55.65	94.32	53.56	91.01	59.14	48.84	93.48	88.80	77.77
RoBERTa$_{\text{Base}}$ (40%)	**Pre-train on 40% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	90.72	MSP	93.84	68.86	93.69	67.62	95.41	63.91	94.20	65.25	94.28	66.41	96.51	63.32	98.90	54.02
		MaxLogit	97.16	78.56	96.87	80.18	98.68	71.84	98.58	74.44	97.82	76.26	100.00	76.72	99.10	65.41
		Energy	97.40	78.53	97.15	80.17	98.68	71.79	98.78	74.39	98.00	76.22	100.00	76.67	99.50	65.39
		GradNorm	97.24	80.59	96.95	78.01	98.52	72.12	98.34	77.16	97.76	76.97	100.00	86.94	99.70	67.46
		KNN_{10}	66.89	87.91	68.58	86.90	77.61	82.31	76.58	85.39	72.41	85.63	75.58	89.45	86.40	84.23
		KNN_{20}	67.57	87.80	68.90	86.79	77.77	82.19	76.30	85.22	72.64	85.50	80.23	89.17	86.80	83.85
		KNN_{50}	67.97	87.58	69.67	86.67	78.01	81.98	76.66	84.85	73.08	85.27	80.23	88.63	87.20	83.21
		KNN_{100}	69.46	87.34	71.23	86.47	79.01	81.72	77.48	84.57	74.30	85.02	82.56	88.19	88.00	82.72
	Pre-train on 40% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN_{10}	88.79	66.14	88.35	68.92	93.50	60.30	95.54	51.09	91.54	61.61	37.21	95.37	55.90	91.90
		KNN_{20}	89.59	65.07	89.80	67.61	93.89	59.10	95.58	50.17	92.21	60.49	46.51	94.41	61.50	91.00
		KNN_{50}	90.59	63.39	91.64	65.68	93.77	57.35	95.66	48.63	92.92	58.76	53.49	93.06	66.40	89.72
		KNN_{100}	91.19	61.79	92.37	63.90	93.66	55.78	95.62	47.42	93.21	57.22	65.12	91.99	68.30	88.72
RoBERTa$_{\text{Base}}$ (100%)	**Pre-train on 100% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	90.74	MSP	94.12	68.24	94.29	66.18	95.93	63.83	95.21	65.66	94.89	65.98	98.84	59.25	96.50	65.42
		MaxLogit	97.24	78.15	97.19	80.27	98.36	72.16	98.38	75.82	97.79	76.60	100.00	73.28	99.30	75.58
		Energy	97.32	78.13	97.51	80.26	98.64	72.12	98.70	75.78	98.04	76.57	100.00	73.27	99.60	75.52
		GradNorm	97.16	80.07	97.39	77.86	98.40	71.83	98.05	79.08	97.75	77.21	100.00	86.32	99.40	73.52
		KNN_{10}	66.81	87.86	69.67	86.91	77.49	82.60	74.59	86.28	72.14	85.91	81.40	87.74	76.90	88.49
		KNN_{20}	66.73	87.75	70.31	86.78	77.89	82.51	75.28	86.13	72.55	85.79	81.40	87.43	77.50	88.39
		KNN_{50}	67.25	87.54	70.59	86.62	77.85	82.32	75.41	85.84	72.78	85.58	83.72	86.85	77.80	88.23
		KNN_{100}	68.13	87.34	71.47	86.39	78.05	82.08	76.14	85.60	73.45	85.35	83.72	86.39	78.50	88.21
	Pre-train on 100% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN_{10}	87.95	66.44	84.49	72.34	95.01	58.47	96.23	49.07	90.92	61.58	31.40	96.19	41.60	94.78
		KNN_{20}	88.91	65.39	85.70	71.25	95.33	57.19	96.59	48.06	91.63	60.47	34.88	95.50	48.40	94.12
		KNN_{50}	90.59	63.69	87.14	69.45	95.53	54.93	97.08	46.26	92.58	58.58	43.02	94.51	55.20	93.05
		KNN_{100}	91.75	62.08	88.55	67.85	95.89	53.05	97.20	44.81	93.35	56.95	50.00	93.60	61.10	92.04

Table G.5: OOD detection performance for document classification with different number of pre-training data from IIT-CDIP$^-$ (remove *pseudo* OOD categories).

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
LayoutLMv1$_{\text{Base}}$ (10%)	**Pre-train on 10% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	95.89	MSP	42.43	76.31	56.05	69.39	54.31	70.25	47.00	73.93	49.95	72.47	43.02	76.55	44.10	75.68
		MaxLogit	41.91	91.27	55.04	89.33	54.19	85.20	44.97	90.93	49.03	89.18	38.37	94.27	41.30	91.38
		Energy	41.83	91.29	54.92	89.35	54.11	85.22	45.01	90.97	48.97	89.21	38.37	94.29	41.10	91.42
		GradNorm	39.15	91.80	54.04	86.93	51.88	86.05	42.49	91.65	46.89	89.11	38.37	91.79	41.40	91.82
		KNN$_{10}$	31.63	94.25	46.52	90.98	46.77	90.49	40.83	92.79	41.44	92.13	24.42	95.95	30.30	95.66
		KNN$_{20}$	32.03	94.11	46.65	90.89	47.01	90.32	41.60	92.63	41.82	91.99	26.74	95.76	31.80	95.44
		KNN$_{50}$	34.39	93.75	49.34	90.46	49.36	89.94	44.52	92.23	44.40	91.60	33.72	95.33	33.20	95.38
		KNN$_{100}$	36.15	93.47	51.27	90.19	51.36	89.65	46.63	91.99	46.35	91.32	33.72	95.10	35.10	95.16
	Pre-train on 10% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN$_{10}$	90.95	72.30	94.66	65.49	90.94	72.38	94.40	67.32	92.74	69.37	48.84	91.56	56.00	75.08
		KNN$_{20}$	91.59	70.54	94.98	63.91	91.66	70.74	94.81	65.95	93.26	67.78	53.49	90.41	57.60	73.51
		KNN$_{50}$	93.07	67.76	95.54	61.24	92.78	68.27	95.25	64.01	94.16	65.32	55.81	88.37	58.50	71.06
		KNN$_{100}$	93.55	65.41	95.90	59.13	93.10	66.19	95.54	62.41	94.52	63.28	67.44	86.44	60.20	69.09
LayoutLMv1$_{\text{Base}}$ (20%)	**Pre-train on 20% IIT-CDIP$^-$ → fine-tune on RVL-CDIP ID data**															
	95.84	MSP	49.20	76.78	61.51	70.13	62.37	69.49	55.52	73.64	57.15	72.51	50.00	77.99	50.70	75.90
		MaxLogit	41.03	91.57	54.00	88.45	56.42	85.70	47.00	90.19	49.61	88.98	38.37	93.62	41.80	90.56
		Energy	40.95	91.60	53.76	88.47	56.19	85.72	46.79	90.22	49.42	89.00	38.37	93.65	41.70	90.59
		GradNorm	37.15	91.89	54.16	84.99	53.03	86.28	43.95	90.94	47.07	88.52	40.70	90.41	42.40	90.91
		KNN$_{10}$	31.63	94.17	47.69	90.29	47.49	90.50	40.54	92.92	41.84	91.97	31.40	95.65	34.50	95.15
		KNN$_{20}$	32.55	94.03	47.89	90.22	48.32	90.34	40.91	92.76	42.42	91.84	33.72	95.45	35.40	94.97
		KNN$_{50}$	35.71	93.67	49.74	89.82	51.04	89.99	44.12	92.39	45.15	91.47	36.05	95.01	36.20	94.92
		KNN$_{100}$	36.75	93.38	50.30	89.60	51.68	89.71	44.97	92.17	45.92	91.22	36.05	94.73	36.50	94.71
	Pre-train on 20% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN$_{10}$	90.39	75.25	79.59	79.43	93.14	72.41	97.12	66.99	90.06	73.52	50.00	91.36	24.70	96.34
		KNN$_{20}$	90.63	73.75	80.47	78.51	93.81	70.58	97.16	65.54	90.52	72.10	55.81	89.91	26.90	95.94
		KNN$_{50}$	91.67	71.19	82.56	76.90	94.45	67.82	97.36	62.98	91.51	69.72	67.44	87.29	29.10	95.31
		KNN$_{100}$	91.95	69.19	83.73	75.55	95.33	65.37	97.36	60.84	92.09	67.74	74.42	84.78	30.30	94.75
LayoutLMv1$_{\text{Base}}$ (40%)	**Pre-train on 40% IIT-CDIP$^-$→ fine-tune on RVL-CDIP ID data**															
	96.01	MSP	51.76	75.76	62.39	69.63	63.37	68.75	54.22	74.03	57.94	72.04	55.81	71.69	42.50	80.56
		MaxLogit	42.03	91.29	54.24	89.47	57.30	84.44	45.66	90.02	49.81	88.80	52.33	93.08	33.00	92.89
		Energy	41.87	91.31	54.20	89.49	57.26	84.47	45.50	90.05	49.71	88.83	52.33	93.13	32.50	92.92
		GradNorm	38.19	91.66	53.64	86.85	55.03	85.66	43.18	91.45	47.51	88.90	52.33	92.39	34.60	92.95
		KNN$_{10}$	31.47	94.43	47.13	90.63	48.20	90.45	38.11	93.30	41.23	92.20	27.91	95.78	24.70	96.09
		KNN$_{20}$	32.59	94.29	47.61	90.55	49.60	90.27	39.25	93.14	42.26	92.06	32.56	95.60	25.50	95.95
		KNN$_{50}$	34.87	93.93	49.50	90.10	52.11	89.87	42.29	92.75	44.69	91.66	38.37	95.16	26.40	95.95
		KNN$_{100}$	36.55	93.65	50.38	89.82	53.55	89.57	43.71	92.51	46.05	91.39	43.02	94.89	27.70	95.77
	Pre-train on 40% IIT-CDIP$^-$(no fine-tune)															
	–	KNN$_{10}$	87.07	80.44	71.76	83.72	86.75	82.31	96.10	76.36	85.42	80.71	75.58	84.96	5.90	98.24
		KNN$_{20}$	88.95	79.03	74.93	82.31	88.99	81.11	96.71	75.01	87.40	79.36	80.23	82.56	7.20	97.93
		KNN$_{50}$	91.47	77.23	80.39	79.90	91.78	79.75	97.40	72.60	90.26	77.37	87.21	78.19	9.00	97.92
		KNN$_{100}$	90.75	75.27	84.77	77.48	91.74	78.31	97.16	70.26	91.10	75.33	89.53	74.11	14.20	97.49
LayoutLMv1$_{\text{Base}}$ (100%)	**Pre-train on 100% IIT-CDIP$^-$→ fine-tune on RVL-CDIP ID data**															
	96.38	MSP	43.43	76.12	57.21	69.16	58.38	68.56	46.14	74.76	51.29	72.15	38.37	78.67	28.30	83.78
		MaxLogit	35.19	91.29	50.22	88.98	53.19	84.54	39.98	90.71	44.64	88.88	24.42	96.39	21.40	95.57
		Energy	35.23	91.32	50.22	89.00	53.19	84.55	39.98	90.73	44.65	88.90	24.42	96.44	21.40	95.58
		GradNorm	30.30	92.54	48.61	88.18	48.96	86.58	36.16	92.63	41.01	89.98	19.77	96.71	19.20	96.35
		KNN$_{10}$	26.50	94.95	43.47	91.69	45.09	90.95	34.09	93.86	37.29	92.86	19.77	97.39	17.80	96.37
		KNN$_{20}$	27.22	94.83	44.07	91.58	45.41	90.79	34.62	93.71	37.83	92.73	19.77	97.22	18.40	96.26
		KNN$_{50}$	29.46	94.49	46.28	91.12	47.69	90.45	37.50	93.33	40.23	92.35	17.44	97.04	18.70	96.80
		KNN$_{100}$	32.15	94.26	48.17	90.85	50.64	90.21	40.38	93.12	42.83	92.11	19.77	96.88	20.70	96.74
	Pre-train on 100% IIT-CDIP$^-$ (no fine-tune)															
	–	KNN$_{10}$	78.74	81.67	74.45	80.86	80.53	83.71	95.01	77.33	82.18	80.89	38.37	94.62	17.70	96.12
		KNN$_{20}$	82.39	80.13	77.86	79.31	83.48	82.75	95.45	75.93	84.80	79.53	44.19	93.42	14.60	96.13
		KNN$_{50}$	86.03	77.65	82.80	76.60	86.91	81.30	96.10	73.07	87.96	77.16	54.65	91.09	9.60	97.21
		KNN$_{100}$	89.11	75.51	88.03	74.08	90.62	79.78	96.71	70.43	91.12	74.95	66.28	88.50	18.00	96.82

Table G.6: OOD detection performance for document classification. Spatial-RoBERTa$_{\text{Base}}$ (Pre) or SR$_{\text{Base}}$ (Pre) denotes applying the spatial-aware adapter in the word embedding layer. Spatial-RoBERTa$_{\text{Base}}$ (Post) or SR$_{\text{Base}}$ (Post) denotes applying the spatial-aware adaptor at the output layer.

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
RoBERTa$_{\text{Base}}$	**Fine-tune on RVL-CDIP (ID)**															
	90.19	MSP	91.19	73.70	90.84	73.49	91.82	71.53	91.03	72.35	91.22	72.77	93.02	80.94	97.60	74.59
		MaxLogit	96.88	79.04	96.87	79.38	98.04	75.85	98.54	77.45	97.58	77.93	100.00	82.76	99.40	79.99
		Energy	97.48	78.96	97.23	79.31	98.40	75.71	99.07	77.25	98.04	77.81	100.00	82.71	99.20	80.06
		KNN$_{10}$	53.20	88.94	58.50	88.62	61.37	86.25	63.72	88.29	59.20	88.02	22.09	96.52	68.60	92.47
		KNN$_{20}$	53.44	88.81	58.90	88.50	61.65	86.07	63.60	88.15	59.40	87.88	27.91	96.38	71.70	92.02
		KNN$_{50}$	53.84	88.52	59.42	88.42	62.01	85.81	64.16	87.80	59.86	87.64	32.56	96.07	74.30	91.37
		KNN$_{100}$	55.56	88.10	60.67	88.20	63.69	85.41	64.77	87.42	61.17	87.28	34.88	95.67	76.50	90.81
	No fine-tune															
	–	KNN$_{10}$	93.11	63.52	88.15	66.34	94.57	66.92	98.42	53.37	93.56	62.54	25.58	95.99	86.00	72.99
		KNN$_{20}$	92.99	63.18	88.39	65.78	94.57	66.08	98.42	52.10	93.59	61.78	26.74	95.71	87.30	70.44
		KNN$_{50}$	92.67	62.41	89.31	64.72	94.17	64.74	98.34	50.07	93.62	60.48	26.74	95.02	90.80	66.04
		KNN$_{100}$	92.67	61.57	89.59	63.57	94.01	63.45	98.17	48.33	93.61	59.23	29.07	94.34	92.80	61.62
SR$_{\text{Base}}$ (Pre)	**Pre-train on IIT-CDIP → fine-tune on RVL-CDIP (ID)**															
	97.11	MSP	46.80	74.52	54.64	70.58	56.26	69.72	54.30	70.74	53.00	71.39	44.19	75.79	57.20	69.23
		MaxLogit	39.43	88.64	46.48	89.92	49.96	85.75	48.30	87.66	46.04	87.99	33.72	93.42	50.60	88.70
		Energy	39.43	88.66	46.48	89.94	50.00	85.76	48.30	87.67	46.05	88.01	33.72	93.45	50.60	88.71
		KNN$_{10}$	31.91	94.41	42.19	92.65	46.65	89.31	42.09	92.65	40.71	92.26	10.47	97.45	52.10	92.93
		KNN$_{20}$	32.31	94.28	42.59	92.64	47.01	89.21	43.43	92.53	41.34	92.16	11.63	97.31	53.30	92.80
		KNN$_{50}$	34.39	93.99	43.83	92.36	49.04	88.93	45.41	92.19	43.17	91.87	12.79	97.01	53.10	92.51
		KNN$_{100}$	35.15	93.76	44.27	92.15	49.48	88.65	46.14	91.97	43.76	91.63	15.12	96.81	49.70	92.44
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN$_{10}$	78.82	78.92	79.99	73.89	77.69	81.32	91.48	76.52	82.00	77.66	10.47	98.08	87.30	80.89
		KNN$_{20}$	79.74	77.95	82.64	72.17	79.81	80.40	92.13	75.11	83.58	76.41	16.28	97.60	92.10	76.94
		KNN$_{50}$	80.42	76.87	85.13	69.62	82.12	78.93	92.98	73.01	85.16	74.61	22.09	96.66	95.20	70.53
		KNN$_{100}$	81.43	75.70	86.90	67.19	83.40	77.12	93.38	71.07	86.28	72.77	27.91	95.86	96.60	64.56
SR$_{\text{Base}}$ (Post)	**Fine-tune on RVL-CDIP (ID)**															
	97.10	MSP	58.05	78.37	76.46	65.44	65.80	75.00	61.81	77.59	65.53	74.10	54.65	81.65	93.50	52.85
		MaxLogit	49.20	89.82	72.36	80.28	57.82	87.28	52.52	90.04	57.98	86.86	34.88	94.88	91.60	73.37
		Energy	47.56	89.87	71.96	80.30	56.58	87.32	51.18	90.10	56.82	86.90	34.88	95.04	91.30	73.39
		KNN$_{10}$	37.43	93.37	64.08	86.83	49.44	89.82	46.92	92.17	49.47	90.55	26.74	96.38	90.10	80.21
		KNN$_{20}$	38.27	93.25	65.33	86.52	50.80	89.66	48.09	91.99	50.62	90.35	26.74	96.23	91.20	79.57
		KNN$_{50}$	40.43	92.98	67.38	86.02	52.83	89.38	50.65	91.58	52.82	89.99	26.74	95.89	92.10	78.48
		KNN$_{100}$	41.99	92.77	67.94	85.62	53.87	89.17	51.22	91.33	53.76	89.72	29.07	95.67	92.60	77.68
SR$_{\text{Large}}$ (Pre)	**Pre-train on IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	97.37	MSP	62.37	67.82	71.27	63.36	72.87	62.54	70.25	63.84	69.19	64.39	76.74	60.61	67.00	65.48
		MaxLogit	33.39	90.15	39.25	89.87	42.30	88.12	37.05	91.66	38.00	89.95	31.40	92.41	27.70	94.23
		Energy	33.39	90.16	39.25	89.88	42.30	88.13	37.05	91.66	38.00	89.96	31.40	92.42	27.70	94.22
		KNN$_{10}$	28.18	94.47	42.43	93.01	37.43	91.74	31.13	94.72	34.79	93.49	25.58	96.24	18.60	96.28
		KNN$_{20}$	28.78	94.32	42.43	92.90	38.07	91.58	32.02	94.55	35.33	93.34	25.58	96.02	18.60	96.33
		KNN$_{50}$	30.22	93.95	43.71	92.69	40.06	91.26	34.54	94.10	37.13	93.00	26.74	95.52	21.40	96.14
		KNN$_{100}$	30.86	93.71	44.11	92.56	40.66	91.05	35.47	93.88	37.78	92.80	26.74	95.22	21.70	96.11
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN$_{10}$	68.49	80.43	88.23	69.83	71.75	83.11	88.11	73.32	79.14	76.67	75.58	84.36	49.80	92.02
		KNN$_{20}$	71.74	78.77	90.24	67.41	75.66	81.38	89.04	71.14	81.67	74.68	81.40	81.55	62.20	90.29
		KNN$_{50}$	75.46	76.49	92.81	63.82	80.17	78.72	90.42	67.84	84.72	71.72	82.56	77.15	78.20	87.49
		KNN$_{100}$	77.62	74.59	94.42	60.94	83.16	76.25	91.80	65.30	86.75	69.27	84.88	73.34	88.20	84.96

Table G.7: OOD detection performance for document classification with the different number of pre-training data from IIT-CDIP.

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
ViT$_{\text{Base}}$ (10%)	**Pre-train on 10% IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	94.89	MSP	55.80	88.37	48.61	91.38	63.93	83.83	55.52	88.55	55.96	88.03	52.05	89.60	34.10	95.04
		MaxLogit	50.36	91.51	37.77	94.30	62.37	87.97	53.69	92.11	51.05	91.47	38.36	94.24	28.60	96.06
		Energy	50.56	91.48	37.08	94.33	63.49	87.89	55.19	92.00	51.58	91.42	38.36	94.29	29.40	95.96
		GradNorm	55.56	79.75	45.96	84.79	66.92	74.07	58.44	81.07	56.72	79.92	47.95	82.04	34.90	91.68
		KNN_{10}	50.40	92.60	43.51	93.92	51.60	90.54	74.47	88.87	55.00	91.48	20.55	97.19	9.20	98.21
		KNN_{20}	49.80	92.70	40.38	94.43	53.39	90.26	74.72	88.77	54.57	91.54	23.29	96.98	10.40	98.05
		KNN_{50}	46.72	92.89	34.27	95.24	56.07	89.92	74.55	88.45	52.90	91.62	27.40	96.56	12.80	97.80
		KNN_{100}	45.48	92.89	29.33	95.67	57.62	89.56	75.04	88.25	51.87	91.59	30.14	96.21	15.00	97.57
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN_{10}	98.92	43.08	97.67	49.00	99.52	54.41	99.35	40.26	98.86	46.69	93.15	92.51	6.90	98.06
		KNN_{20}	98.88	42.47	97.75	48.57	99.52	53.75	99.35	39.56	98.88	46.09	94.52	92.24	8.60	97.91
		KNN_{50}	98.80	41.70	97.83	48.04	99.52	52.91	99.35	38.62	98.88	45.32	95.89	91.80	10.60	97.66
		KNN_{100}	98.76	41.20	97.79	47.70	99.48	52.32	99.35	38.01	98.84	44.81	98.63	91.31	14.50	97.41
ViT$_{\text{Base}}$ (20%)	**Pre-train on 20% IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	94.62	MSP	54.36	89.01	51.63	91.31	64.57	85.23	60.51	88.67	57.77	88.56	60.27	89.34	44.20	93.73
		MaxLogit	44.32	92.16	38.21	94.18	64.92	87.63	58.56	91.33	51.50	91.32	45.21	92.63	39.70	94.36
		Energy	44.36	92.17	37.89	94.24	66.56	87.51	60.39	91.22	52.30	91.28	46.58	92.62	41.50	94.18
		GradNorm	90.51	54.92	92.04	51.67	94.29	45.41	98.13	32.36	93.74	46.09	95.89	40.44	89.70	59.01
		KNN_{10}	52.20	92.58	45.84	93.73	53.79	90.75	77.84	87.02	57.42	91.02	17.81	97.33	16.90	97.40
		KNN_{20}	51.60	92.66	43.55	94.15	55.63	90.46	78.04	86.79	57.20	91.02	19.18	97.06	19.40	97.11
		KNN_{50}	50.12	92.86	39.98	94.82	58.02	90.18	78.77	86.54	56.72	91.10	19.18	96.63	23.10	96.68
		KNN_{100}	48.04	92.91	34.75	95.28	60.38	89.88	78.98	86.42	55.54	91.12	20.55	96.27	26.20	96.35
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN_{10}	98.16	41.13	97.51	47.12	99.48	53.05	99.31	38.79	98.62	45.02	94.52	91.80	8.00	97.41
		KNN_{20}	98.12	40.71	97.51	46.79	99.48	52.52	99.31	38.31	98.60	44.58	94.52	91.48	8.70	97.25
		KNN_{50}	98.04	40.10	97.55	46.31	99.48	51.84	99.39	37.63	98.62	43.97	95.89	91.01	11.50	96.99
		KNN_{100}	98.00	39.74	97.55	45.98	99.48	51.34	99.39	37.26	98.60	43.58	97.26	90.55	14.60	96.70
ViT$_{\text{Base}}$ (40%)	**Pre-train on 40% IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	94.63	MSP	55.48	88.65	52.27	91.54	64.49	85.52	58.08	89.20	57.58	88.73	67.12	84.62	45.80	93.82
		MaxLogit	47.12	91.74	40.06	94.09	61.05	88.68	56.57	92.01	51.20	91.63	69.86	89.81	32.90	95.46
		Energy	47.12	91.73	39.94	94.10	62.33	88.62	58.60	91.88	52.00	91.58	69.86	89.65	32.70	95.44
		GradNorm	47.00	85.76	41.90	89.64	60.69	81.37	53.73	87.06	50.83	85.96	64.38	81.12	34.00	92.93
		KNN_{10}	53.28	92.13	48.33	92.99	46.45	92.20	75.61	88.87	55.92	91.55	34.25	95.53	6.80	98.56
		KNN_{20}	52.76	92.24	45.88	93.57	48.12	91.95	74.84	88.75	55.40	91.63	32.88	95.21	7.80	98.36
		KNN_{50}	51.28	92.52	40.94	94.51	50.52	91.70	75.08	88.46	54.46	91.80	35.62	94.67	10.90	98.04
		KNN_{100}	50.32	92.62	36.16	95.12	53.35	91.36	75.93	88.24	53.94	91.84	39.73	94.25	13.60	97.76
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN_{10}	97.56	40.60	97.03	46.28	99.24	53.76	99.15	39.62	98.24	45.06	82.19	92.02	1.00	99.59
		KNN_{20}	97.56	40.00	96.95	45.86	99.24	53.18	99.15	39.12	98.22	44.54	82.19	91.63	1.00	99.55
		KNN_{50}	97.56	39.24	96.99	45.20	99.24	52.39	99.15	38.49	98.24	43.83	86.30	91.07	1.00	99.50
		KNN_{100}	97.60	38.78	97.03	44.79	99.24	51.76	99.15	38.15	98.26	43.37	90.41	90.67	1.20	99.45
ViT$_{\text{Base}}$ (100%)	**Pre-train on 100% IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	94.79	MSP	54.28	88.80	49.14	91.80	64.60	84.45	58.85	88.78	56.72	88.46	61.64	89.44	41.00	94.27
		MaxLogit	44.96	92.13	38.01	94.52	63.97	87.97	56.49	91.81	50.86	91.61	68.49	90.65	34.60	95.26
		Energy	45.72	92.11	38.01	94.55	65.84	87.86	57.91	91.70	51.87	91.56	72.60	90.41	34.80	95.14
		GradNorm	48.72	84.21	44.36	87.50	63.49	78.07	56.25	84.79	53.20	83.64	60.27	82.96	35.60	91.24
		KNN_{10}	45.16	93.14	39.13	94.62	51.68	90.85	73.58	88.81	52.39	91.86	50.68	93.09	10.40	98.04
		KNN_{20}	44.88	93.14	36.64	95.04	53.35	90.59	74.27	88.67	52.28	91.86	50.68	92.67	12.00	97.81
		KNN_{50}	43.67	93.19	31.18	95.60	56.74	90.29	75.28	88.49	51.72	91.89	57.53	92.23	15.60	97.45
		KNN_{100}	43.63	93.15	27.52	95.94	58.74	90.02	76.18	88.38	51.52	91.87	61.64	92.01	18.90	97.18
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN_{10}	97.04	42.35	93.97	50.17	97.41	52.68	98.01	43.19	96.61	47.10	12.33	97.47	3.10	98.38
		KNN_{20}	97.16	41.99	94.01	49.96	97.81	52.01	98.09	42.73	96.77	46.67	15.07	96.95	3.00	98.31
		KNN_{50}	96.96	41.62	94.34	49.56	98.00	51.20	98.05	42.24	96.84	46.16	21.92	96.08	2.70	98.18
		KNN_{100}	97.00	41.48	94.90	49.31	98.12	50.65	98.13	42.03	97.04	45.87	36.99	95.29	2.30	98.27

Table G.8: OOD detection performance for document classification. Longformer$_{4096}$ denotes the original model adopted from the Huggingface model hub. Longformer$_{4096}$ (+) denotes the additional pre-training on IIT-CDIP.

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
Longformer$_{4096}$	**Fine-tune on RVL-CDIP (ID)**															
	90.71	MSP	95.00	64.32	95.62	62.17	95.89	60.53	93.95	66.89	95.12	63.48	88.37	77.50	98.60	54.72
		MaxLogit	97.12	72.84	97.07	75.22	98.24	70.39	95.82	77.57	97.06	74.00	90.70	86.62	99.60	68.10
		Energy	97.48	72.82	97.35	75.21	98.36	70.37	96.59	77.56	97.44	73.99	91.86	86.63	99.80	68.08
		KNN$_{10}$	58.45	88.21	65.65	86.88	67.80	83.99	56.78	89.53	62.17	87.15	27.91	96.01	82.10	86.31
		KNN$_{20}$	58.97	88.04	65.57	86.60	68.12	83.80	57.35	89.34	62.50	86.94	29.07	95.82	82.60	85.93
		KNN$_{50}$	60.25	87.64	66.57	86.25	68.91	83.41	58.81	88.96	63.64	86.56	30.23	95.46	82.70	85.27
		KNN$_{100}$	61.97	87.19	68.14	85.81	70.15	82.95	60.47	88.60	65.18	86.14	34.88	95.04	82.80	84.75
	No fine-tune															
	–	KNN$_{10}$	98.04	55.45	97.63	59.97	98.76	51.75	98.13	53.16	98.14	55.08	70.93	88.69	100.00	64.97
		KNN$_{20}$	98.12	55.19	97.67	59.64	98.80	51.27	98.17	52.71	98.19	54.70	70.93	88.51	100.00	64.08
		KNN$_{50}$	98.00	54.82	97.63	59.13	98.80	50.57	98.30	52.07	98.18	54.15	73.26	88.29	100.00	62.82
		KNN$_{100}$	97.92	54.48	97.67	58.62	98.84	50.00	98.34	51.62	98.19	53.68	74.42	88.14	100.00	61.70
Longformer$_{4096}$ (+)	**Pre-train on IIT-CDIP→ fine-tune on RVL-CDIP (ID)**															
	91.13	MSP	95.20	64.08	95.62	61.38	96.05	59.47	94.48	63.13	95.34	62.02	90.70	67.26	98.00	55.52
		MaxLogit	96.96	75.41	96.54	76.03	97.89	70.15	96.71	74.56	97.02	74.04	100.00	78.65	99.70	72.88
		Energy	97.28	75.40	96.54	76.03	98.28	70.14	97.16	74.55	97.32	74.03	100.00	78.59	99.70	72.86
		KNN$_{10}$	58.73	89.25	66.21	87.57	72.03	83.76	63.68	88.72	65.16	87.32	48.84	94.78	86.40	87.84
		KNN$_{20}$	58.61	89.18	65.97	87.45	71.67	83.69	63.39	88.61	64.91	87.23	48.84	94.62	85.30	87.70
		KNN$_{50}$	61.17	88.96	66.97	87.29	72.83	83.47	65.83	88.33	66.70	87.01	55.81	94.25	85.20	87.39
		KNN$_{100}$	61.73	88.79	66.93	87.11	73.30	83.24	66.15	88.15	67.03	86.82	55.81	94.00	84.70	87.21
	Pre-train on IIT-CDIP (no fine-tune)															
	–	KNN$_{10}$	95.48	61.40	98.07	53.66	97.73	55.55	98.66	48.70	97.49	54.83	81.40	91.12	97.40	46.27
		KNN$_{20}$	95.56	60.92	97.95	52.95	97.49	54.97	98.50	48.21	97.38	54.26	84.88	90.62	97.50	45.55
		KNN$_{50}$	95.60	59.94	97.95	51.77	97.41	53.97	98.62	47.29	97.40	53.24	87.21	89.95	98.20	44.18
		KNN$_{100}$	95.60	59.04	97.99	50.74	97.21	52.99	98.58	46.51	97.34	52.32	88.37	89.52	98.50	43.09

Table G.9: OOD detection performance for document classification. All models are pre-trained on ImageNet.

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
ResNet-50	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	91.12	MSP	64.49	87.87	55.89	90.94	66.60	87.31	77.88	80.87	66.22	86.75	51.16	92.76	63.10	90.36
		MaxLogit	64.89	88.59	47.97	92.81	65.40	87.52	77.56	81.87	63.96	87.70	41.86	94.62	54.00	93.29
		Energy	67.09	88.30	47.81	92.86	66.68	87.24	78.53	81.75	65.03	87.54	39.53	94.73	48.50	93.68
		KNN_{10}	73.38	86.82	67.98	87.46	71.31	87.84	92.90	77.74	76.39	84.96	6.98	99.12	5.20	98.98
		KNN_{20}	74.90	86.41	66.29	87.79	73.82	87.21	93.95	76.51	77.24	84.48	6.98	98.96	5.50	98.85
		KNN_{50}	76.66	86.04	66.41	88.48	78.29	86.39	95.50	74.76	79.22	83.92	5.81	98.68	5.90	98.70
		KNN_{100}	77.54	85.61	65.41	88.99	82.16	85.43	96.23	73.37	80.33	83.35	6.98	98.34	6.30	98.51
	Pre-train on ImageNet															
	–	KNN_{10}	96.96	51.14	94.62	51.75	98.76	53.84	99.59	37.60	97.48	48.58	83.56	85.00	20.80	97.00
		KNN_{20}	96.96	50.37	94.34	51.54	98.92	52.98	99.59	36.60	97.45	47.87	83.56	84.49	22.70	96.71
		KNN_{50}	96.92	49.29	94.29	51.30	99.00	51.84	99.59	35.15	97.45	46.90	83.56	84.03	26.70	96.21
		KNN_{100}	97.12	48.60	94.54	51.25	99.16	51.11	99.55	34.36	97.59	46.33	82.19	83.31	29.40	95.67
$Swin_{Base}$	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	95.74	MSP	47.64	88.09	49.90	88.11	58.22	83.14	50.28	88.90	51.51	87.06	49.32	91.31	36.50	93.63
		MaxLogit	42.39	93.11	42.47	93.45	58.62	88.79	45.90	93.18	47.34	92.13	50.68	92.50	32.20	95.65
		Energy	43.15	93.05	42.95	93.40	59.02	88.70	46.71	93.07	47.96	92.06	52.05	92.38	33.60	95.49
		KNN_{10}	49.44	92.82	46.73	92.87	42.90	92.57	72.69	88.45	52.94	91.68	16.44	96.73	6.10	98.30
		KNN_{20}	48.84	92.95	43.27	93.51	44.53	92.32	72.28	88.35	52.23	91.78	17.81	96.52	7.40	98.10
		KNN_{50}	46.44	93.26	39.25	94.57	47.41	92.09	73.34	87.87	51.61	91.95	26.03	96.15	8.60	97.80
		KNN_{100}	43.76	93.42	35.03	95.29	50.08	91.72	75.77	87.42	51.16	91.96	28.77	95.94	11.30	97.55
	Pre-train on ImageNet															
	–	KNN_{10}	98.56	52.75	95.06	55.14	99.36	58.85	99.80	41.86	98.20	52.15	65.75	93.26	2.10	99.35
		KNN_{20}	98.44	51.86	95.18	54.72	99.32	57.88	99.80	40.66	98.18	51.28	68.49	92.52	2.60	99.22
		KNN_{50}	98.52	50.69	95.38	54.13	99.16	56.61	99.76	39.01	98.20	50.11	78.08	91.14	3.40	98.99
		KNN_{100}	98.72	49.96	95.66	53.80	99.16	55.84	99.76	38.16	98.32	49.44	79.45	89.89	4.30	98.77
ViT_{Base}	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	94.38	MSP	56.81	89.14	52.19	91.80	67.48	84.26	59.90	88.77	59.10	88.49	47.67	92.98	59.50	91.99
		MaxLogit	50.76	91.37	44.60	93.75	68.04	86.94	55.15	91.81	54.64	90.97	40.70	94.20	52.40	93.16
		Energy	51.16	91.31	44.52	93.75	69.43	86.81	56.09	91.77	55.30	90.91	38.37	94.11	53.20	93.11
		KNN_{10}	62.57	90.12	57.73	90.91	53.67	90.36	84.50	86.19	64.62	89.40	12.79	97.96	13.00	97.92
		KNN_{20}	63.01	90.24	56.01	91.51	55.03	90.02	84.38	86.01	64.61	89.44	15.12	97.76	14.90	97.67
		KNN_{50}	61.97	90.62	53.23	92.62	58.26	89.57	84.25	85.64	64.43	89.61	16.28	97.38	19.80	97.24
		KNN_{100}	60.29	90.85	49.70	93.53	60.38	89.07	84.01	85.43	63.60	89.72	16.28	97.05	23.60	96.82
	Pre-train on ImageNet															
	–	KNN_{10}	98.48	52.15	95.02	56.94	99.48	53.77	99.47	38.90	98.11	50.44	93.15	90.27	20.40	97.13
		KNN_{20}	98.48	51.41	95.06	56.61	99.44	52.92	99.55	37.61	98.13	49.64	94.52	89.44	22.60	96.80
		KNN_{50}	98.32	50.43	94.86	56.21	99.40	51.86	99.59	35.82	98.04	48.58	97.26	88.23	26.60	96.25
		KNN_{100}	98.40	49.76	95.06	55.90	99.44	51.15	99.59	34.59	98.12	47.85	98.63	87.24	31.20	95.76

Table G.10: OOD detection performance for document classification (select OOD categories achieve the best performance across most of the models with different modalities).

	ID	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Email		Resume		File folder		Sci. publication		*Average*		Sci. Poster		Receipt	
	Acc		FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
RoBERTa$_{\text{Base}}$	**Pre-train on pure-text data→ fine-tune on RVL-CDIP (ID)**															
	86.13	MSP	96.22	60.38	90.67	71.72	93.82	59.47	93.86	65.51	93.64	64.27	91.86	70.57	93.00	69.99
		MaxLogit	99.21	66.57	95.80	73.66	95.47	66.81	97.09	65.63	96.89	68.17	94.19	77.17	94.60	74.69
		Energy	99.60	66.53	96.64	73.57	95.14	66.82	97.21	65.35	97.15	68.07	94.19	77.44	95.60	74.90
		KNN_{10}	83.70	82.77	69.02	84.28	88.32	74.06	86.11	74.02	81.79	78.78	43.02	92.74	72.00	88.87
		KNN_{20}	84.50	82.35	69.06	84.21	88.20	73.71	86.72	74.02	82.12	78.57	48.84	92.38	73.80	88.31
		KNN_{50}	84.98	81.57	68.86	84.06	88.08	73.01	87.08	73.94	82.25	78.14	54.65	91.92	75.40	87.44
		KNN_{100}	86.25	80.88	70.26	83.80	88.28	72.40	87.44	73.89	83.06	77.74	58.14	91.50	78.20	86.68
	Pre-train on pure-text data															
	–	KNN_{10}	86.09	75.63	95.12	58.62	97.71	59.75	98.95	50.54	94.47	61.14	10.47	98.46	89.80	63.01
		KNN_{20}	86.29	74.92	95.00	58.14	97.71	58.88	99.03	49.49	94.51	60.36	12.79	98.35	90.80	60.59
		KNN_{50}	87.32	73.55	94.64	57.53	97.83	57.56	99.15	48.11	94.73	59.19	12.79	98.11	93.30	56.61
		KNN_{100}	89.27	72.48	94.28	57.12	97.99	56.52	99.11	47.37	95.16	58.37	11.63	97.89	94.30	52.98
Longformer$_{4096}$	**Pre-train on pure-text data→ fine-tune on RVL-CDIP (ID)**															
	88.34	MSP	96.90	60.55	96.20	59.14	96.31	55.72	97.82	55.12	96.81	57.63	95.35	80.44	99.60	52.82
		MaxLogit	98.97	68.97	97.60	65.64	95.67	63.42	98.63	62.87	97.72	65.23	97.67	88.42	99.70	71.54
		Energy	99.44	68.96	97.92	65.63	95.83	63.42	98.71	62.83	97.98	65.21	97.67	88.46	99.90	71.55
		KNN_{10}	68.28	88.72	69.62	83.36	78.17	85.08	90.88	74.98	76.74	83.04	16.28	96.90	81.60	86.94
		KNN_{20}	68.04	88.61	70.10	83.22	77.53	84.92	90.75	74.95	76.60	82.92	16.28	96.84	81.80	86.49
		KNN_{50}	69.28	88.29	70.98	82.92	78.29	84.46	90.96	74.82	77.38	82.62	19.77	96.59	83.40	85.71
		KNN_{100}	69.28	88.15	71.34	82.69	78.49	84.21	90.43	74.86	77.39	82.48	22.09	96.38	83.90	85.17
	Pre-train on pure-text data															
	–	KNN_{10}	97.42	47.77	95.72	50.09	97.67	46.58	99.52	38.61	97.58	45.76	45.35	93.92	100.00	63.03
		KNN_{20}	97.46	46.91	95.60	49.80	97.71	46.02	99.52	38.21	97.57	45.24	46.51	93.77	100.00	61.92
		KNN_{50}	97.58	45.68	95.56	49.45	97.75	45.19	99.52	37.72	97.60	44.51	50.00	93.60	100.00	60.35
		KNN_{100}	97.66	44.78	95.60	49.17	97.87	44.63	99.56	37.57	97.67	44.04	51.16	93.48	100.00	58.89
ResNet-50	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	85.25	MSP	60.53	87.26	69.53	87.00	27.86	95.13	94.05	75.79	62.99	86.30	91.78	74.40	27.80	95.47
		MaxLogit	59.98	89.27	72.61	88.02	30.04	95.41	93.39	75.38	64.00	87.02	80.82	79.89	30.00	95.29
		Energy	63.71	89.14	75.64	87.55	45.71	94.15	92.77	75.02	69.46	86.46	78.08	81.07	62.20	93.44
		KNN_{10}	72.46	85.68	85.69	85.30	68.62	76.01	96.15	55.35	80.73	75.59	36.99	94.56	2.20	99.37
		KNN_{20}	76.15	84.55	88.65	84.22	66.13	80.67	96.54	56.31	81.87	76.44	38.36	93.81	2.70	99.28
		KNN_{50}	80.37	82.61	92.00	82.49	60.98	86.77	96.93	59.06	82.57	77.73	47.95	92.42	3.80	99.11
		KNN_{100}	84.70	80.54	95.15	80.64	51.29	91.78	97.16	61.19	82.08	78.54	50.68	91.01	4.70	98.91
	Pre-train on ImageNet															
	–	KNN_{10}	99.72	40.94	99.65	21.52	52.47	91.03	98.33	45.40	87.54	49.72	84.93	84.38	20.40	97.12
		KNN_{20}	99.68	41.18	99.65	20.68	50.61	91.63	98.41	44.65	87.09	49.54	86.30	83.94	23.40	96.87
		KNN_{50}	99.64	41.58	99.65	19.48	46.97	92.36	98.37	43.49	86.16	49.23	84.93	83.70	26.90	96.43
		KNN_{100}	99.64	42.19	99.65	18.98	44.91	92.84	98.33	42.86	85.63	49.22	84.93	83.12	29.20	95.98
Swin$_{\text{Base}}$	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	91.25	MSP	70.23	81.87	67.68	85.31	43.97	92.68	83.78	79.40	66.42	84.82	86.30	78.23	54.10	91.62
		MaxLogit	54.73	87.04	46.51	92.30	17.25	96.51	90.86	74.11	52.34	87.49	82.19	83.20	34.40	94.82
		Energy	54.05	87.11	44.38	92.49	16.38	96.63	91.29	73.59	51.53	87.46	84.93	83.07	33.80	94.82
		KNN_{10}	56.08	90.66	48.80	92.84	38.31	93.31	91.02	66.91	58.55	85.93	27.40	96.03	3.30	98.84
		KNN_{20}	54.61	90.95	49.98	92.68	27.58	95.24	91.44	68.54	55.90	86.85	26.03	96.35	4.00	98.76
		KNN_{50}	55.25	90.68	52.15	92.37	15.75	97.28	91.25	71.62	53.60	87.99	28.77	96.10	4.90	98.59
		KNN_{100}	56.20	90.31	54.75	92.17	9.14	98.00	91.13	75.11	52.80	88.90	30.14	95.77	6.50	98.35
	Pre-train on ImageNet															
	–	KNN_{10}	99.84	43.55	99.76	20.64	47.92	93.20	98.91	37.55	86.61	48.74	58.90	93.88	1.60	99.32
		KNN_{20}	99.84	43.78	99.76	19.61	44.76	93.61	98.91	37.01	85.82	48.50	65.75	93.42	2.10	99.20
		KNN_{50}	99.84	44.47	99.80	18.36	41.31	94.14	99.03	36.45	85.00	48.36	72.60	92.69	2.60	99.00
		KNN_{100}	99.88	45.26	99.80	17.92	39.97	94.39	99.03	36.71	84.67	48.57	79.45	91.97	3.70	98.81
ViT$_{\text{Base}}$	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	89.97	MSP	61.25	85.84	66.57	85.04	40.44	93.10	85.84	81.83	63.52	86.45	73.97	80.66	60.30	90.41
		MaxLogit	53.02	90.37	55.77	88.86	19.91	96.25	92.38	79.69	55.27	88.79	76.71	85.16	50.60	93.12
		Energy	51.79	90.49	55.07	89.03	17.53	96.53	92.69	79.20	54.27	88.81	79.45	85.01	50.10	93.20
		KNN_{10}	54.13	91.18	52.86	91.18	58.49	87.46	92.88	65.98	64.59	83.95	42.47	95.07	11.00	97.94
		KNN_{20}	54.21	91.18	53.17	90.99	50.61	89.35	93.04	67.52	62.76	84.76	43.84	94.98	13.10	97.62
		KNN_{50}	54.53	91.05	53.33	90.79	41.95	92.82	93.00	72.06	60.70	86.68	42.47	94.74	17.30	97.12
		KNN_{100}	54.65	90.81	54.12	90.56	30.79	95.78	93.04	75.39	58.15	88.14	45.21	94.24	22.00	96.58
	Pre-train on ImageNet															
	–	KNN_{10}	99.80	46.46	99.68	26.50	58.65	90.61	98.72	46.40	89.21	52.49	87.67	91.39	19.90	97.25
		KNN_{20}	99.80	46.02	99.65	25.69	57.30	91.01	98.72	46.46	88.87	52.30	90.41	90.87	21.70	97.01
		KNN_{50}	99.80	45.48	99.61	24.76	55.16	91.52	98.76	46.69	88.33	52.11	94.52	89.99	24.30	96.62
		KNN_{100}	99.80	45.33	99.65	24.43	54.81	91.90	98.72	47.10	88.24	52.19	95.89	89.31	28.80	96.27

Table G.11: OOD detection performance for document classification (randomly select four categories as OOD).

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Letter		Handwritten		Advertisement		Memo		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
RoBERTa$_{\text{Base}}$	**Pre-train on pure-text data→ fine-tune on RVL-CDIP (ID)**															
	88.86	MSP	70.22	79.21	50.14	87.24	84.64	67.80	91.42	57.99	74.10	73.06	95.35	59.75	94.30	55.12
		MaxLogit	66.04	87.51	39.65	92.53	86.47	77.03	91.67	71.84	70.96	82.23	100.00	77.89	96.80	71.96
		Energy	66.20	87.57	38.19	92.59	87.35	77.03	91.67	71.89	70.85	82.27	100.00	77.92	96.80	71.96
		KNN_{10}	62.62	80.19	60.98	70.90	75.62	80.24	85.84	69.20	71.26	75.13	94.19	81.99	90.40	82.48
		KNN_{20}	63.18	80.10	60.07	71.17	75.90	80.03	85.72	68.88	71.22	75.04	94.19	81.75	91.20	81.89
		KNN_{50}	63.78	80.00	57.30	71.70	76.34	79.67	85.88	68.38	70.82	74.94	94.19	81.45	91.80	81.09
		KNN_{100}	64.77	79.98	54.33	71.94	77.37	79.32	86.08	67.80	70.64	74.76	94.19	81.20	91.90	80.47
	Pre-train on pure-text data															
	–	KNN_{10}	85.53	59.90	98.61	21.79	96.21	56.72	97.69	58.39	94.51	49.20	12.79	98.01	84.50	65.73
		KNN_{20}	85.45	59.27	98.73	21.19	96.21	55.63	97.90	57.05	94.57	48.28	12.79	97.91	86.10	63.57
		KNN_{50}	86.80	57.94	98.77	20.45	96.89	54.12	98.30	55.35	95.19	46.96	13.95	97.60	89.30	59.64
		KNN_{100}	88.47	56.71	98.81	19.97	96.81	52.89	98.18	53.93	95.57	45.88	13.95	97.38	91.10	55.17
Longformer$_{4096}$	**Pre-train on pure-text data→ fine-tune on RVL-CDIP (ID)**															
	92.08	MSP	65.96	69.58	50.38	77.93	81.52	60.89	90.21	54.23	72.02	65.66	82.56	60.14	95.00	50.90
		MaxLogit	62.19	87.35	44.64	89.79	79.97	78.84	88.39	68.08	68.80	81.02	80.23	84.19	94.30	77.36
		Energy	61.27	87.35	43.61	89.81	79.13	78.85	88.15	68.08	68.04	81.02	80.23	84.19	94.30	77.37
		KNN_{10}	58.65	79.54	50.77	71.81	66.56	83.48	80.87	75.19	64.21	77.51	58.14	92.78	90.00	77.76
		KNN_{20}	57.81	79.43	51.40	71.72	67.00	83.35	81.15	74.86	64.34	77.34	58.14	92.57	89.70	77.12
		KNN_{50}	58.77	79.30	51.60	71.67	66.72	83.15	81.31	74.36	64.60	77.12	61.63	92.24	89.80	76.17
		KNN_{100}	61.39	79.16	52.75	71.61	67.84	82.93	81.76	73.91	65.94	76.90	62.79	91.99	89.80	75.29
	Pre-train on pure-text data															
	–	KNN_{10}	99.40	47.83	100.00	27.75	98.28	47.03	93.20	60.40	97.72	45.75	46.51	93.85	100.00	63.64
		KNN_{20}	99.44	47.33	100.00	27.48	98.32	46.49	93.24	60.22	97.75	45.38	48.84	93.70	100.00	62.79
		KNN_{50}	99.44	46.33	100.00	27.23	98.40	45.85	93.41	60.05	97.81	44.86	51.16	93.51	100.00	61.55
		KNN_{100}	99.44	45.67	100.00	27.31	98.44	45.23	93.53	59.90	97.85	44.53	52.33	93.40	100.00	60.31
ResNet-50	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	87.80	MSP	70.58	85.35	55.29	89.88	64.29	86.54	71.15	85.58	65.33	86.84	54.79	91.70	77.20	84.67
		MaxLogit	64.25	87.46	53.59	90.72	49.70	90.60	64.45	88.71	58.00	89.37	36.99	95.13	78.90	86.86
		Energy	62.66	87.65	58.33	90.33	46.00	91.26	63.56	89.05	57.64	89.57	32.88	95.69	83.00	87.05
		KNN_{10}	90.99	79.37	56.36	90.64	72.41	86.20	89.17	81.74	77.23	84.49	2.74	99.32	39.70	93.70
		KNN_{20}	92.17	78.00	47.47	92.61	68.27	88.42	90.85	80.23	74.69	84.82	2.74	99.25	43.80	93.08
		KNN_{50}	94.32	75.96	28.44	94.49	65.65	89.27	92.78	77.91	70.30	84.41	1.37	98.97	49.70	92.09
		KNN_{100}	95.58	74.02	27.21	95.07	60.44	89.78	94.22	75.63	69.36	83.62	2.74	98.67	53.80	91.10
	Pre-train on ImageNet															
	–	KNN_{10}	98.46	42.21	77.29	81.41	27.87	91.16	99.08	43.47	75.68	64.56	80.82	89.98	12.30	98.17
		KNN_{20}	98.66	41.00	76.78	81.70	29.22	92.27	99.08	42.29	75.94	64.32	83.56	89.30	14.10	97.97
		KNN_{50}	98.58	39.53	76.58	81.81	31.01	92.05	99.12	40.80	76.32	63.55	83.56	88.51	16.30	97.61
		KNN_{100}	98.62	38.62	77.13	81.49	32.64	91.84	99.12	39.86	76.88	62.95	83.56	87.80	19.50	97.23
Swin$_{\text{Base}}$	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	92.42	MSP	63.96	87.03	65.21	88.15	73.56	79.72	61.40	88.46	66.03	85.84	84.93	74.34	49.60	92.49
		MaxLogit	56.49	90.22	75.36	87.00	72.64	84.26	44.22	93.01	62.18	88.62	72.60	84.16	29.10	95.70
		Energy	57.43	90.11	77.01	86.60	73.44	84.17	43.78	93.06	62.92	88.48	73.97	84.25	28.00	95.69
		KNN_{10}	60.27	90.12	66.90	90.76	49.66	89.15	47.67	92.67	56.12	90.68	42.47	94.28	7.20	98.56
		KNN_{20}	61.32	90.01	61.37	91.31	48.83	90.33	49.00	92.52	55.13	91.04	30.14	95.56	8.80	98.33
		KNN_{50}	62.22	89.78	56.44	91.56	50.34	89.55	48.52	92.30	54.38	90.80	26.03	95.72	11.80	97.97
		KNN_{100}	62.62	89.60	54.98	91.85	50.70	88.93	47.63	92.18	53.98	90.64	30.14	95.54	13.90	97.66
	Pre-train on ImageNet															
	–	KNN_{10}	99.15	45.57	86.02	79.44	32.45	90.98	99.52	46.20	79.28	65.55	24.66	96.24	0.40	99.78
		KNN_{20}	99.19	44.11	86.89	80.35	33.48	92.19	99.60	44.79	79.79	65.36	27.40	95.62	0.50	99.73
		KNN_{50}	99.23	42.39	87.99	81.66	36.78	91.59	99.60	43.07	80.90	64.68	43.84	94.57	0.80	99.63
		KNN_{100}	99.19	41.46	89.02	82.63	40.60	91.05	99.60	42.14	82.10	64.32	52.05	93.49	1.20	99.53
ViT$_{\text{Base}}$	**Pre-train on ImageNet→ fine-tune on RVL-CDIP (ID)**															
	91.03	MSP	69.68	86.81	69.67	87.88	72.25	80.78	69.38	86.61	70.24	85.52	67.12	85.97	58.50	91.47
		MaxLogit	63.35	89.20	68.40	88.58	69.58	84.38	61.08	89.94	65.60	88.02	57.53	89.41	48.40	93.04
		Energy	62.22	89.21	70.34	88.43	70.26	84.37	60.75	90.03	65.89	88.01	58.90	89.47	49.70	93.03
		KNN_{10}	68.10	88.99	54.90	92.30	53.44	88.05	58.19	91.34	58.66	90.17	38.36	95.02	22.90	96.71
		KNN_{20}	67.61	88.95	49.01	92.85	51.53	89.25	58.59	91.16	56.68	90.55	41.10	94.47	25.40	96.35
		KNN_{50}	67.29	88.91	42.54	93.15	53.96	88.43	58.75	90.88	55.64	90.34	42.47	93.60	29.90	95.78
		KNN_{100}	66.19	88.90	43.80	93.19	55.71	87.73	59.11	90.64	56.20	90.12	45.21	92.86	34.90	95.27
	Pre-train on ImageNet															
	–	KNN_{10}	98.90	41.98	90.96	77.15	34.87	90.69	99.40	41.21	81.03	62.76	54.79	94.27	10.80	98.47
		KNN_{20}	98.94	40.54	91.67	77.20	36.82	91.71	99.44	39.85	81.72	62.32	64.38	93.57	12.70	98.25
		KNN_{50}	99.07	38.75	92.61	76.99	40.00	91.17	99.52	38.14	82.80	61.26	75.34	92.47	15.90	97.87
		KNN_{100}	99.11	37.43	93.25	76.56	43.38	90.68	99.56	36.93	83.82	60.40	82.19	91.52	18.90	97.49

Table G.12: OOD detection performance for document classification. All models are pre-trained on IIT-CDIP. For LayoutLM models, we adopt the checkpoints from the Huggingface model hub. For UDoc, we pre-train the model on our side. All models are fine-tuned on RVL-CDIP ID data.

	ID Acc	Method	OOD Dataset (In-domain)										OOD Dataset (Out-domain)			
			Sci. Report		Presentation		Form		Letter		*Average*		Sci. Poster		Receipt	
			FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
LayoutLMv1$_{\text{Base}}$	97.28	MSP	47.48	74.91	59.74	68.72	66.40	65.36	58.89	69.12	58.13	69.53	43.02	77.15	72.40	62.40
		MaxLogit	27.06	92.38	37.97	91.52	45.65	88.36	35.92	91.22	36.65	90.87	24.42	94.96	57.30	86.70
		Energy	27.06	92.40	37.97	91.54	45.65	88.36	35.92	91.23	36.65	90.88	24.42	94.97	57.30	86.70
		KNN_{10}	20.82	96.09	35.32	93.82	40.06	91.34	28.65	94.80	31.21	94.01	17.44	97.00	49.80	93.92
		KNN_{20}	21.74	95.93	36.20	93.77	41.42	91.12	30.44	94.61	32.45	93.86	17.44	96.82	51.70	93.73
		KNN_{50}	24.34	95.56	38.25	93.41	43.93	90.69	33.64	94.19	35.04	93.46	23.26	96.44	53.80	93.70
		KNN_{100}	25.54	95.30	39.13	93.20	45.17	90.35	34.78	93.99	36.16	93.21	25.58	96.24	54.70	93.45
LayoutLMv3	97.81	MSP	56.16	70.81	63.44	67.17	67.16	65.30	58.60	69.58	61.34	68.22	52.33	72.70	43.60	77.10
		MaxLogit	30.70	89.17	40.42	88.18	42.98	84.09	33.12	88.22	36.80	87.42	19.77	94.50	11.70	97.02
		Energy	30.70	89.18	40.42	88.18	42.98	84.10	33.12	88.23	36.80	87.42	19.77	94.51	11.70	97.03
		KNN_{10}	21.74	95.03	35.68	93.38	32.88	91.86	18.51	96.26	27.20	94.13	11.63	97.58	8.90	97.97
		KNN_{20}	22.74	94.90	36.56	93.20	33.96	91.66	19.64	96.15	28.22	93.98	12.79	97.44	10.00	97.89
		KNN_{50}	24.62	94.62	38.37	92.71	35.83	91.38	21.63	95.93	30.11	93.66	13.95	97.20	10.70	97.72
		KNN_{100}	25.22	94.38	39.29	92.32	36.55	91.09	22.48	95.79	30.88	93.40	16.28	97.04	11.80	97.59
UDoc$_{\text{ResNet50}}$	97.36	MSP	66.13	65.73	69.43	64.09	71.03	63.28	71.06	63.25	69.41	64.09	40.70	78.47	39.80	78.99
		MaxLogit	45.96	82.12	47.21	86.39	49.64	83.16	49.59	83.13	48.10	83.70	2.33	98.57	4.00	98.34
		Energy	45.96	82.12	47.21	86.40	49.64	83.16	49.59	83.13	48.10	83.70	2.33	98.60	4.00	98.36
		KNN_{10}	30.02	94.47	41.22	88.66	41.90	90.99	36.65	93.48	37.45	91.90	1.16	99.13	5.50	98.42
		KNN_{20}	31.10	94.36	41.98	88.44	42.10	90.90	38.03	93.35	38.30	91.76	1.16	99.04	6.90	98.32
		KNN_{50}	33.95	94.07	43.35	87.89	44.01	90.72	40.71	93.06	40.51	91.43	1.16	98.84	7.40	98.26
		KNN_{100}	34.83	93.84	43.75	87.51	45.01	90.61	41.96	92.90	41.39	91.22	1.16	98.72	8.30	98.16

references

Ahuja, Kartik, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. 2020. Invariant risk minimization games. In *International conference on machine learning* (*ICML*), 145–155.

Albuquerque, Isabela, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. 2019. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*.

Alvarez-Melis, David, and Nicolo Fusi. 2020. Geometric dataset distances via optimal transport. In *Conference on neural information processing systems* (*NeurIPS*).

Appalaraju, Srikar, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *The ieee international conference on computer vision* (*ICCV*).

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Arora, Udit, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. In *Conference on empirical methods in natural language processing* (*EMNLP*).

Assran, Mahmoud, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. 2021. Semi-supervised

learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the ieee/cvf international conference on computer vision (ICCV)*, 8443–8452.

Bahng, Hyojin, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. 2020. Learning de-biased representations with biased representations. In *International conference on machine learning (ICML)*, 528–539.

Bahng, Hyojin, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. 2022. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*.

Bai, Haoyue, Rui Sun, Lanqing Hong, Fengwei Zhou, Nanyang Ye, Han-Jia Ye, S-H Gary Chan, and Zhenguo Li. 2021a. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the aaai conference on artificial intelligence (AAAI)*, 6705–6713.

Bai, Haoyue, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. 2021b. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the ieee/cvf international conference on computer vision (ICCV)*, 8320–8329.

Baker, Nicholas, Hongjing Lu, Gennady Erlikhman, and Philip J. Kellman. 2018. Deep convolutional networks do not classify based on global object shape. *PLOS Computational Biology* 14(12):1–43.

Barbu, Andrei, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. 2019. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Conference on neural information processing systems (NeurIPS)*.

Beery, Sara, Grant Van Horn, and Pietro Perona. 2018. Recognition in terra incognita. In *The european conference on computer vision (ECCV)*.

Beltagy, Iz, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79(1-2):151–175.

Bendale, Abhijit, and Terrance E Boult. 2016. Towards open set deep networks. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Bitterwolf, Julian, Alexander Meinke, Maximilian Augustin, and Matthias Hein. 2022. Breaking down out-of-distribution detection: Many methods based on ood training data estimate a combination of the same core quantities. In *International conference on machine learning (ICML)*.

Bitterwolf, Julian, Maximilian Mueller, and Matthias Hein. 2023. In or out? fixing imagenet out-of-distribution detection evaluation. In *International conference on machine learning (ICML)*.

Blanchard, Gilles, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. 2021. Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research* 22(1):46–100.

Blanchard, Gilles, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems (NeurIPS)*, vol. 24.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine

Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *The european conference on computer vision* (*ECCV*), 446–461.

Bucci, Silvia, Francesco Cappio Borlino, Barbara Caputo, and Tatiana Tommasi. 2022. Distance-based hyperspherical classification for multi-source open-set domain adaptation. In *Proceedings of the ieee/cvf winter conference on applications of computer vision* (*WACV*), 1119–1128.

Cai, Mu, and Yixuan Li. 2023. Out-of-distribution detection via frequency-regularized generative models. In *Proceedings of ieee/cvf winter conference on applications of computer vision* (*WACV*).

Cha, Junbum, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. 2021. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems* (*NeurIPS*) 34:22405–22418.

Cha, Junbum, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. 2022. Domain generalization by mutual-information regularization with pre-trained models. In *European conference on computer vision*, 440–457.

Chang, Shiyu, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International conference on machine learning* (*ICML*), 1448–1458.

Chen, Aochuan, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. 2023a. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (*CVPR*), 19133–19143.

Chen, Derek, and Zhou Yu. 2021. Gold: improving out-of-scope detection in dialogues using data augmentation. *arXiv preprint arXiv:2109.03079*.

Chen, Jiefeng, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. 2021. Atom: Robustifying out-of-distribution detection using outlier mining. In *The european conference on machine learning and principles and practice of knowledge discovery in databases*.

Chen, Liang, Yong Zhang, Yibing Song, Ying Shan, and Lingqiao Liu. 2023b. Improved test-time adaptation for domain generalization. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 24172–24182.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Chen, Weihua, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the ieee conference on computer vision and pattern recognition (CVPR)*, 403–412.

Chen, Xinlei, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, Yimeng, Tianyang Hu, Fengwei Zhou, Zhenguo Li, and Zhi-Ming Ma. 2023c. Explore and exploit the diverse knowledge in model zoo for domain generalization. In *International conference on machine learning (ICML)*, 4623–4640. PMLR.

Chen, Yongqiang, Yonggang Zhang, Yatao Bian, Han Yang, MA Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Learning

causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems* (*NeurIPS*) 22131–22148.

Chen, Zixiang, Yihe Deng, Yuanzhi Li, and Quanquan Gu. 2023d. Understanding transferable representation learning and zero-shot transfer in clip. *arXiv preprint arXiv:2310.00927*.

Cimpoi, Mircea, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Cui, Lei, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.

Dai, Rui, Yonggang Zhang, Zhen Fang, Bo Han, and Xinmei Tian. 2023. Moderately distributional exploration for domain generalization. In *International conference on machine learning* (*ICML*).

Daume III, Hal, and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26:101–126.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Deng, Jiankang, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (*CVPR*), 4690–4699.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North american chapter of the association for computational linguistics* (*NAACL*).

DeVries, Terrance, and Graham W Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Du, Xuefeng, Gabriel Gozum, Yifei Ming, and Yixuan Li. 2022a. Siren: Shaping representations for detecting out-of-distribution objects. In *Advances in neural information processing systems (NeurIPS)*.

Du, Xuefeng, Zhaoning Wang, Mu Cai, and Yixuan Li. 2022b. Vos: Learning what you don't know by virtual outlier synthesis. In *Proceedings of the international conference on learning representations (ICLR)*.

Eastwood, Cian, Alexander Robey, Shashank Singh, Julius Von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. 2022. Probable domain generalization via quantile risk minimization. *Advances in Neural Information Processing Systems (NeurIPS)* 35:17340–17358.

El Banani, Mohamed, Karan Desai, and Justin Johnson. 2023. Learning visual representations via language-guided sampling. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 19208–19220.

Esmaeilpour, Sepideh, Bing Liu, Eric Robertson, and Lei Shu. 2022. Zero-shot open set detection by extending clip. In *The aaai conference on artificial intelligence (AAAI)*.

Fang, Zhen, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. 2022. Is out-of-distribution detection learnable? In *Advances in neural information processing system* (*NeurIPS*).

Fei-Fei, Li, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop* (*CVPR-W*).

Fort, Stanislav, Jie Ren, and Balaji Lakshminarayanan. 2021. Exploring the limits of out-of-distribution detection. In *Conference on neural information processing systems* (*NeurIPS*).

Fürst, Andreas, Elisabeth Rumetshofer, Johannes Lehner, Viet Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. 2022. Cloob: Modern hopfield networks with infoloob outperform clip. In *Advances in neural information processing systems* (*NeurIPS*), 20450–20468.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

Gao, Peng, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2023. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision* (*IJCV*) 1–15.

Ge, ZongYuan, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. 2017. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*.

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations* (*ICLR*).

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems* (*NeurIPS*), 2672–2680.

Gretton, Arthur, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems* (*NeurIPS*) 19.

Gu, Jiuxiang, Jason Kuen, Shafiq Joty, Jianfei Cai, Vlad Morariu, Handong Zhao, and Tong Sun. 2020. Self-supervised relationship probing. In *Conference on neural information processing systems* (*NeurIPS*).

Gu, Jiuxiang, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. 2021. Unified pretraining framework for document understanding. In *Advances in neural information processing systems* (*NeurIPS*).

Gu, Jiuxiang, Yifei Ming, Yi Zhou, Jason Kuen, Vlad I Morariu, Handong Zhao, Ruiyi Zhang, Nikolaos Barmpalios, Anqi Liu, Yixuan Li, Tong Sun, and Ani Nenkova. 2023. A critical analysis of out-of-distribution detection for document understanding. In *Findings of conference on empirical methods in natural language processing* (*EMNLP*).

Gu, Zhangxuan, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. 2022. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Gulrajani, Ishaan, and David Lopez-Paz. 2021. In search of lost domain generalization. In *International conference on learning representations (ICLR)*.

Guo, Yaming, Kai Guo, Xiaofeng Cao, Tieru Wu, and Yi Chang. 2023. Out-of-distribution generalization of federated learning via implicit invariant relationships. *International Conference on Machine Learning (ICML)*.

Harley, Adam W, Alex Ufkes, and Konstantinos G Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *International conference on document analysis and recognition(ICDAR)*.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 9726–9735.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Hein, Matthias, Maksym Andriushchenko, and Julian Bitterwolf. 2019. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Hendrycks, Dan, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Xiaodong Song. 2022. Scaling out-of-distribution detection for real-world settings. In *International conference on machine learning (ICML)*.

Hendrycks, Dan, and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International conference on learning representations* (*ICLR*).

Hendrycks, Dan, and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International conference on learning representations* (*ICLR*).

Hendrycks, Dan, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Association for computational linguistics* (*ACL*).

Hendrycks, Dan, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. In *International conference on learning representations* (*ICLR*).

Hendrycks, Dan, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using self-supervised learning can improve model robustness and uncertainty. In *Conference on neural information processing systems* (*NeurIPS*), 15663–15674.

Hendrycks, Dan, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021. Natural adversarial examples. *the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*).

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hong, Teakgyu, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *The aaai conference on artificial intelligence* (*AAAI*).

Hsu, Yen-Chang, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Hu, Yibo, and Latifur Khan. 2021. Uncertainty-aware reliable text classification. In *Sigkdd conference on knowledge discovery and data mining (KDD)*.

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Ieee conference on computer vision and pattern recognition (CVPR)*.

Huang, Rui, Andrew Geng, and Yixuan Li. 2021a. On the importance of gradients for detecting distributional shifts in the wild. In *Conference on neural information processing systems (NeurIPS)*.

Huang, Rui, and Yixuan Li. 2021. Mos: Towards scaling out-of-distribution detection for large semantic space. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Huang, Tony, Jack Chu, and Fangyun Wei. 2022a. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*.

Huang, Yu, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021b. What makes multi-modal learning better than single (provably). *Advances in Neural Information Processing Systems (NeurIPS)* 34:10944–10956.

Huang, Yupan, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022b. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 25th acm international conference on multimedia*.

Huang, Zeyi, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In *European conference on computer vision (ECCV)*, 124–140.

Huang, Zhuo, Miaoxi Zhu, Xiaobo Xia, Li Shen, Jun Yu, Chen Gong, Bo Han, Bo Du, and Tongliang Liu. 2023. Robust generalization against photon-limited corruptions via worst-case sharpness minimization. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 16175–16185.

Iscen, Ahmet, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. 2023. Retrieval-enhanced contrastive vision-text models. *arXiv preprint arXiv:2306.07196*.

Izmailov, Pavel, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *Uncertainty in artificial intelligence*.

Jaume, Guillaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *International conference on document analysis and recognition(ICDAR) workshop*.

Jia, Chao, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning (ICML)*.

Jin, Di, Shuyang Gao, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tur. 2022. Towards textual out-of-domain detection without in-domain labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7(3):535–547.

Joyce, James M. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, 720–722. Springer.

Jupp, P.E., and K.V. Mardia. 2009. *Directional statistics*. Wiley.

Kang, Guoliang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. 2019. Contrastive adaptation network for unsupervised domain adaptation. In *Ieee conference on computer vision and pattern recognition*, 4893–4902.

Khattak, Muhammad Uzair, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 19113–19122.

Khosla, Prannay, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in neural information processing systems (NeurIPS)*, vol. 33, 18661–18673.

Kim, Daehee, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. 2021a. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Ieee international conference on computer vision (ICCV)*, 9619–9628.

Kim, Geewook, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Donut: Document understanding transformer without ocr. In *The european conference on computer vision (ECCV)*.

Kim, Jaeill, Suhyun Kang, Duhun Hwang, Jungwook Shin, and Wonjong Rhee. 2023a. Vne: An effective method for improving deep representation by manipulating eigenvalue distribution. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 3799–3810.

Kim, Sungnyun, Sangmin Bae, and Se-Young Yun. 2023b. Coreset sampling from open-set for fine-grained self-supervised learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 7537–7547.

Kim, Sungyeon, Dongwon Kim, Minsu Cho, and Suha Kwak. 2020. Proxy anchor loss for deep metric learning. In *Ieee/cvf conference on computer vision and pattern recognition (CVPR)*.

Kim, Sungyeon, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. 2019. Deep metric learning beyond binary supervision. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 2288–2297.

Kim, Wonjae, Bokyung Son, and Ildoo Kim. 2021b. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning (ICML)*.

Kingma, Diederik P, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, Durk P, and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems (NeurIPS)*, 10215–10224.

Kirichenko, Polina, Pavel Izmailov, and Andrew G Wilson. 2020. Why normalizing flows fail to detect out-of-distribution data. *Conference on Neural Information Processing Systems (NeurIPS)*.

Koh, Pang Wei, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning (ICML)*.

Kohler, Jonas Moritz, and Aurelien Lucchi. 2017. Sub-sampled cubic regularization for non-convex optimization. In *International conference on machine learning (ICML)*, 1895–1904.

Kolesnikov, Alexander, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big transfer (bit): General visual representation learning. In *The european conference on computer vision (ECCV)*.

Krasin, Ivan, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. 2017. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*.

Krause, Jonathan, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *4th international ieee workshop on 3d representation and recognition (3drr-13)*. Sydney, Australia.

Krizhevsky, Alex, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Krueger, David, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation. In *International conference on machine learning (ICML)*, 5815–5826.

Larson, Stefan, Gordon Lim, Yutong Ai, David Kuang, and Kevin Leach. 2022. Evaluating out-of-distribution performance on document image classifiers. In *Conference on neural information processing systems (NeurIPS)*.

Le, Ya, and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7(7):3.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lee, Kimin, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Conference on neural information processing systems (NeurIPS)*.

Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the empirical methods in natural language processing (EMNLP)*, 3045–3059.

Lewis, D., G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Sigir*.

Li, Chunyuan, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. 2022a. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems (NeurIPS)* 35:9287–9301.

Li, Da, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. 2018a. Learning to generalize: Meta-learning for domain generalization. In *Aaai conference on artificial intelligence (AAAI)*, vol. 32.

Li, Da, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017a. Deeper, broader and artier domain generalization. In *Ieee international conference on computer vision (ICCV)*, 5542–5550.

Li, Gen, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *The aaai conference on artificial intelligence (AAAI)*.

Li, Haoliang, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018b. Domain generalization with adversarial feature learning. In *Ieee conference on computer vision and pattern recognition(CVPR)*, 5400–5409.

Li, Junlong, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022b. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 25th acm international conference on multimedia*.

Li, Junnan, Caiming Xiong, and Steven Hoi. 2020b. Mopro: Webly supervised learning with momentum prototypes. In *International conference on learning representations (ICLR)*.

Li, Junnan, Caiming Xiong, and Steven C.H. Hoi. 2021a. Learning from noisy data with robust representation learning. In *Proceedings of the ieee/cvf international conference on computer vision (ICCV)*, 9485–9494.

Li, Junnan, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. 2021b. Prototypical contrastive learning of unsupervised representations. In *Iclr*.

Li, Liunian Harold, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Li, Peizhao, Jiuxiang Gu, Jason Kuen, Vlad Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. 2021c. Selfdoc: Self-supervised document representation learning. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Li, Wen, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. 2017b. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*.

Li, Xiang Lisa, and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing*, 4582–4597.

Li, Xiaoya, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. 2021d. kfolden: k-fold ensemble for out-of-distribution detection. In *Conference on empirical methods in natural language processing (EMNLP)*.

Li, Ya, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018c. Deep domain generalization via conditional invariant adversarial networks. In *European conference on computer vision*, 624–639.

Li, Yangguang, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. 2022c. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International conference on learning representations (ICLR)*.

Liang, Shiyu, Yixuan Li, and Rayadurgam Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International conference on learning representations (ICLR)*.

Liang, Victor Weixin, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems (NeurIPS)* 35:17612–17625.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *The european conference on computer vision (ECCV)*.

Lin, Ziqian, Sreya Dutta Roy, and Yixuan Li. 2021. Mood: Multi-level out-of-distribution detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*.

Liu, Haotian, Kilho Son, Jianwei Yang, Ce Liu, Jianfeng Gao, Yong Jae Lee, and Chunyuan Li. 2023. Learning customized visual models with retrieval-augmented knowledge. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 15148–15158.

Liu, Weitang, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. In *Conference on neural information processing systems (NeurIPS)*.

Liu, Weiyang, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *The ieee conference on computer vision and pattern recognition (CVPR)*.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Ziquan, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Xiangyang Ji, Antoni B. Chan, and Rong Jin. 2022. Improved fine-tuning by better leveraging pre-training data. In *Advances in neural information processing systems (NeurIPS)*.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of international conference on computer vision (ICCV)*.

Long, Alexander, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hen-

gel. 2022. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 6959–6969.

Loshchilov, Ilya, and Frank Hutter. 2019. Decoupled weight decay regularization. In *International conference on learning representations (ICLR)*.

Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neurips*.

Lu, Yuning, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. 2022. Prompt distribution learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 5206–5215.

Luo, Chuanchen, Chunfeng Song, and Zhaoxiang Zhang. 2020. Generalizing person re-identification by camera-aware invariance learning and cross-domain mixup. In *European conference on computer vision (ECCV)*.

Mahajan, Divyat, Shruti Tople, and Amit Sharma. 2021. Domain generalization using causal matching. In *International conference on machine learning (ICML)*, 7313–7324. PMLR.

Manli, Shu, Nie Weili, Huang De-An, Yu Zhiding, Goldstein Tom, Anandkumar Anima, and Xiao Chaowei. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in neural information processing systems (NeurIPS)*.

Mardia, Kanti V, Peter E Jupp, and KV Mardia. 2000. *Directional statistics*, vol. 2. Wiley Online Library.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software* 3(29):861.

Mettes, Pascal, Elise van der Pol, and Cees Snoek. 2019. Hyperspherical prototype networks. *Advances in neural information processing systems* (*NeurIPS*) 32.

Meyer, Christian M, and Iryna Gurevych. 2012. *Wiktionary: A new rival for expert-built lexicons? exploring the possibilities of collaborative lexicography*. na.

Miller, George A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Min, Seonwoo, Nokyung Park, Siwon Kim, Seunghyun Park, and Jinkyu Kim. 2022. Grounding visual representations with texts for domain generalization. In *European conference on computer vision* (*ECCV*), 37–53. Springer.

Ming, Yifei, Haoyue Bai, Julian Katz-Samuels, and Yixuan Li. 2024. Provable out-of-distribution generalization in hypersphere. In *The twelfth international conference on learning representations* (*ICLR*).

Ming, Yifei, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. 2022a. Delving into out-of-distribution detection with vision-language representations. In *Advances in neural information processing systems* (*NeurIPS*).

Ming, Yifei, Ying Fan, and Yixuan Li. 2022b. Poem: Out-of-distribution detection with posterior sampling. In *International conference on machine learning* (*ICML*).

Ming, Yifei, and Yixuan Li. 2023. How does fine-tuning impact out-of-distribution detection for vision-language models? *International Journal of Computer Vision* (*IJCV*).

Ming, Yifei, Yiyou Sun, Ousmane Dia, and Yixuan Li. 2023. How to exploit hyperspherical embeddings for out-of-distribution detection? In *Proceedings of the international conference on learning representations (ICLR)*.

Ming, Yifei, Hang Yin, and Yixuan Li. 2022c. On the impact of spurious correlation for out-of-distribution detection. In *The aaai conference on artificial intelligence (AAAI)*.

Mokady, Ron, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Mondal, Ajoy, Peter Lipps, and CV Jawahar. 2020. Iiit-ar-13k: a new dataset for graphical object detection in documents. In *International workshop on document analysis systems*.

Morteza, Peyman, and Yixuan Li. 2022. Provable guarantees for understanding out-of-distribution detection. *The AAAI Conference on Artificial Intelligence (AAAI)*.

Movshovitz-Attias, Yair, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies. In *Proceedings of the ieee international conference on computer vision (ICCV)*, 360–368.

Mu, Norman, Alexander Kirillov, David Wagner, and Saining Xie. 2022. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision (ECCV)*, 529–544.

Muandet, Krikamol, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International conference on machine learning (ICML)*, 10–18.

Nakada, Ryumei, Halil Ibrahim Gulluk, Zhun Deng, Wenlong Ji, James Zou, and Linjun Zhang. 2023. Understanding multimodal contrastive

learning and incorporating unpaired data. In *International conference on artificial intelligence and statistics* (*AISTATS*), 4348–4380.

Nalisnick, Eric, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2019. Do deep generative models know what they don't know? In *International conference on learning representations* (*ICLR*).

Nam, Hyeonseob, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. 2021. Reducing domain gap by reducing style bias. In *Ieee conference on computer vision and pattern recognition*, 8690–8699.

Neal, Lawrence, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. 2018. Open set learning with counterfactual images. In *The european conference on computer vision* (*ECCV*).

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning.

Ng, Edwin G., Bo Pang, Piyush Sharma, and Radu Soricut. 2020. Understanding guided image captioning performance across domains. *arXiv preprint arXiv:2012.02339*.

Nguyen, Anh, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*), 427–436.

Nilsback, Maria-Elena, and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Sixth indian conference on computer vision, graphics & image processing*, 722–729.

Oh Song, Hyun, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the ieee conference on computer vision and pattern recognition* (*CVPR*), 4004–4012.

Van den Oord, Aaron, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* arXiv–1807.

van den Oord, Aaron, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. 1807.03748.

Ovadia, Yaniv, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems* (*NeurIPS*) 32:13991–14002.

Oza, Poojan, and Vishal M Patel. 2019. C2ae: Class conditioned auto-encoder for open-set recognition. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Park, Jungwuk, Dong-Jun Han, Soyeong Kim, and Jaekyun Moon. 2023. Test-time style shifting: Handling arbitrary styles in domain generalization. In *International conference on machine learning* (*ICML*).

Park, Seunghyun, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing. In *Conference on neural information processing systems* (*NeurIPS*) *workshop*.

Parkhi, Omkar M., Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. 2012. Cats and dogs. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society* 947–1012.

Podolskiy, Alexander, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *The aaai conference on artificial intelligence (AAAI)*.

Pratt, Sarah, Ian Covert, Rosanne Liu, and Ali Farhadi. 2023. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the ieee/cvf international conference on computer vision (ICCV)*, 15691–15701.

Qiang, Yu-Ting, Yan-Wei Fu, Xiao Yu, Yan-Wen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2019. Learning to generate posters of scientific papers by probabilistic graphical models. *JCST*.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning (ICML)*.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Rame, Alexandre, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. 2023. Model ratatouille: Recycling diverse models for out-of-distribution generalization. *International Conference on Machine Learning (ICML)*.

Ren, Jie, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *Conference on neural information processing systems (NeurIPS)*.

Ren, Jie, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *Iclr*.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Sigkdd conference on knowledge discovery and data mining (KDD)*.

Ridnik, Tal, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*.

Robinson, Joshua David, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International conference on learning representations (ICLR)*.

Rojas-Carulla, Mateo, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant models for causal transfer learning. *The Journal of Machine Learning Research* 19(1):1309–1342.

Rosenfeld, Elan, Pradeep Kumar Ravikumar, and Andrej Risteski. 2021. The risks of invariant risk minimization. In *International conference on learning representations (ICLR)*.

Roy, Abhijit Guha, Jie Ren, Shekoofeh Azizi, Aaron Loh, Vivek Natarajan, Basil Mustafa, Nick Pawlowski, Jan Freyberg, Yuan Liu, Zach Beaver, et al. 2022. Does your dermatology classifier know what it doesn't know? detecting the long-tail of unseen conditions. *Medical Image Analysis* 75: 102274.

Rubner, Yossi, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *International conference on computer vision* (*ICCV*), 59–66.

Sagawa, Shiori, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International conference on learning representations* (*ICLR*).

Sastry, Chandramouli Shama, and Sageev Oore. 2020. Detecting out-of-distribution examples with in-distribution examples and gram matrices. In *Proceedings of the 37th international conference on machine learning* (*ICML*).

Schiappa, Madeline C, Yogesh S Rawat, Shruti Vyas, Vibhav Vineet, and Hamid Palangi. 2022. Multi-modal robustness analysis against language and visual perturbations. In *Conference on neural information processing systems* (*NeurIPS*).

Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* (*NeurIPS*) 35:25278–25294.

Sehwag, Vikash, Mung Chiang, and Prateek Mittal. 2021. Ssd: A unified framework for self-supervised outlier detection. In *International conference on learning representations* (*ICLR*).

Serrà, Joan, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. 2020. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International conference on learning representations* (*ICLR*).

Sharma, Piyush, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (*ACL*), 2556–2565.

Shen, Sheng, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, et al. 2022. K-lite: Learning transferable visual models with external knowledge. *Advances in Neural Information Processing Systems* (*NeurIPS*) 35:15558–15573.

Shen, Yilin, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in slu. *arXiv preprint arXiv:2106.14464*.

Smith, Ray. 2007. An overview of the tesseract ocr engine. In *International conference on document analysis and recognition*(*ICDAR*).

Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Su, Weijie, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. In *Proceedings of the international conference on learning representations* (*iclr*).

Sun, Baochen, and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision* (*ECCV*), 443–450.

Sun, Quan, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Sun, Yiyou, Chuan Guo, and Yixuan Li. 2021. React: Out-of-distribution detection with rectified activations. In *Conference on neural information processing systems (NeurIPS)*.

Sun, Yiyou, and Yixuan Li. 2022. Dice: Leveraging sparsification for out-of-distribution detection. In *Proceedings of european conference on computer vision (ECCV)*.

Sun, Yiyou, Yifei Ming, Xiaojin Zhu, and Yixuan Li. 2022. Out-of-distribution detection with deep nearest neighbors. In *International conference on machine learning (ICML)*.

Sustik, Mátyás A, Joel A Tropp, Inderjit S Dhillon, and Robert W Heath Jr. 2007. On the existence of equiangular tight frames. *Linear Algebra and its applications* 426(2-3):619–635.

Tack, Jihoon, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Conference on neural information processing systems (NeurIPS)*.

Tan, Ming, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. 2019. Out-of-domain detection for low-resource text classification tasks. In *Conference on empirical methods in natural language processing (EMNLP)*.

Tang, Zineng, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Tapaswi, Makarand, Marc T Law, and Sanja Fidler. 2019. Video face clustering with unknown number of clusters. In *Proceedings of the ieee/cvf international conference on computer vision (ICCV)*, 5027–5036.

Teh, Eu Wern, Terrance DeVries, and Graham W Taylor. 2020. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European conference on computer vision* (*ECCV*), 448–464. Springer.

Tong, Peifeng, Wu Su, He Li, Jialin Ding, Zhan Haoxiang, and Song Xi Chen. 2023. Distribution free domain generalization. In *International conference on machine learning* (*ICML*), 34369–34378. PMLR.

Torralba, Antonio. 2003. Contextual priming for object detection. *International journal of computer vision* (*IJCV*) 53(2):169–191.

Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Ieee conference on computer vision and pattern recognition* (*CVPR*), 7167–7176.

Udandarao, Vishaal, Ankush Gupta, and Samuel Albanie. 2023. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the ieee/cvf international conference on computer vision* (*ICCV*), 2725–2736.

Uppal, Shagun, Sarthak Bhagat, Devamanyu Hazarika, Navonil Majumder, Soujanya Poria, Roger Zimmermann, and Amir Zadeh. 2022. Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion* 77:149–171.

Van Horn, Grant, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. 2018. The inaturalist species classification and detection dataset. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Vapnik, Vladimir. 1992. Principles of risk minimization for learning theory. In *Advances in neural information processing systems* (*NeurIPS*), 831–838.

Vapnik, Vladimir N. 1999. An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10(5):988–999.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Conference on Neural Information Processing Systems (NeurIPS)*.

Vaze, Sagar, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Open-set recognition: A good closed-set classifier is all you need. In *International conference on learning representations (ICLR)*.

Velavan, Thirumalaisamy P, and Christian G Meyer. 2020. The covid-19 epidemic. *Tropical medicine & international health* 25(3):278.

Vershynin, Roman. 2018. *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge university press.

Von Kügelgen, Julius, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. 2021. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems (NeurIPS)* 34:16451–16467.

Wah, C., S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology.

Wang, Feng, and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

Wang, Hao, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss

for deep face recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition (CVPR)*, 5265–5274.

Wang, Haobo, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. 2022a. Pico: Contrastive label disambiguation for partial label learning. In *International conference on learning representations (ICLR)*.

Wang, Haoqi, Zhizhong Li, Litong Feng, and Wayne Zhang. 2022b. Vim: Out-of-distribution with virtual-logit matching. In *The ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 4921–4930.

Wang, Haoran, Weitang Liu, Alex Bocchieri, and Yixuan Li. 2021. Can multi-label classification networks know what they don't know? *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, Haotao, Aston Zhang, Yi Zhu, Shuai Zheng, Mu Li, Alex J Smola, and Zhangyang Wang. 2022c. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. In *International conference on machine learning (ICML)*.

Wang, Jindong, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022d. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, Rongguang, Pratik Chaudhari, and Christos Davatzikos. 2022e. Embracing the disharmony in medical imaging: A simple and effective framework for domain adaptation. *Medical Image Analysis* 76:102309.

Wang, Tongzhou, and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning (ICML)*.

Wang, Wenjin, Zhengjie Huang, Bin Luo, Qianglong Chen, Qiming Peng, Yinxu Pan, Weichong Yin, Shikun Feng, Yu Sun, Dianhai Yu, et al. 2022f. mmlayout: Multi-grained multimodal transformer for document understanding. In *Proceedings of the 25th acm international conference on multimedia*.

Wang, Yufei, Haoliang Li, and Alex C Kot. 2020. Heterogeneous domain generalization via domain mixup. In *Ieee international conference on acoustics, speech and signal processing*, 3622–3626.

Wang, Zilong, Jiuxiang Gu, Chris Tensmeyer, Nikolaos Barmpalios, Ani Nenkova, Tong Sun, Jingbo Shang, and Vlad I Morariu. 2022g. Mgdoc: Pre-training with multi-granular hierarchy for document image understanding. In *Conference on empirical methods in natural language processing (EMNLP)*.

Wei, Hongxin, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. 2022. Mitigating neural network overconfidence with logit normalization. In *International conference on machine learning (icml)*.

Winkens, Jim, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, et al. 2020. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Xiao, Jianxiong, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *The ieee / cvf computer vision and pattern recognition conference* (*CVPR*).

Xiao, Kai Yuanqing, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2021. Noise or signal: The role of image backgrounds in object recognition. In *International conference on learning representations* (*ICLR*).

Xiao, Tong, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. 2017. Joint detection and identification feature learning for person search. In *Proceedings of the ieee conference on computer vision and pattern recognition* (*CVPR*), 3415–3424.

Xiao, Zhisheng, Qing Yan, and Yali Amit. 2020. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In *Conference on neural information processing systems* (*NeurIPS*), vol. 33.

Xie, Chen-Wei, Siyang Sun, Xiong Xiong, Yun Zheng, Deli Zhao, and Jingren Zhou. 2023. Ra-clip: Retrieval augmented contrastive language-image pre-training. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (*CVPR*), 19265–19274.

Xing, Chen, Sercan Arik, Zizhao Zhang, and Tomas Pfister. 2020. Distance-based learning from errors for confidence calibration. In *International conference on learning representations* (*ICLR*).

Xu, Hu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2024. Demystifying clip data. In *International conference on learning representations* (*ICLR*).

Xu, Keyang, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021a. Unsupervised out-of-domain detection via pre-trained transformers. In *Association for computational linguistics (ACL)*.

Xu, Pingmei, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. 2015. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*.

Xu, Yang, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2021b. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *"association for computational linguistics (ACL)"*.

Xu, Yiheng, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Sigkdd*.

Yang, Jianwei, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. 2022. Unified contrastive learning in image-text-label space. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 19163–19173.

Yang, Jingkang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. 2021a. Semantically coherent out-of-distribution detection. In *Proceedings of the ieee international conference on computer vision (ICCV)*.

Yang, Jingkang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021b. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Yao, Lewei, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. Filip:

Fine-grained interactive language-image pre-training. *International Conference on Learning Representations (ICLR)*.

Yao, Xufeng, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. 2022. Pcl: Proxy-based contrastive learning for domain generalization. In *Ieee conference on computer vision and pattern recognition (CVPR)*, 7097–7107.

Ye, Haotian, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. 2021. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems (NeurIPS)* 34:23519–23531.

Ye, Nanyang, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. 2022. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Ieee conference on computer vision and pattern recognition*, 7947–7958.

Yu, Fisher, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Yu, Jiahui, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*.

Yu, Tao, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. 2023. Task residual for tuning vision-language models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 10899–10909.

Zadeh, Amir, Paul Pu Liang, and Louis-Philippe Morency. 2020. Foundations of multimodal co-learning. *Information Fusion* 64:188–193.

Zech, John R, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. 2018. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine* 15(11).

Zhai, Xiaohua, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the ieee/cvf international conference on computer vision* (*ICCV*).

Zhai, Xiaohua, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. 2022. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (*CVPR*), 18123–18133.

Zhan, Li-Ming, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. *Association for Computational Linguistics* (*ACL*).

Zhang, Hongyi, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. Mixup: Beyond empirical risk minimization. In *International conference on learning representations* (*ICLR*).

Zhang, Marvin Mengxin, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. 2021a. Adaptive risk minimization: Learning to adapt to domain shift. In *Advances in neural information processing systems* (*NeurIPS*).

Zhang, Michael, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. 2022a. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In *International conference on machine learning* (*ICML*), 26484–26516.

Zhang, Renrui, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021b. Tip-adapter: Training-

free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Zhang, Renrui, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. 2023. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)*, 15211–15222.

Zhang, Renrui, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022b. Tip-adapter: Training-free adaption of clip for few-shot classification. In *17th european conference on computer vision (ECCV)*, 493–510.

Zhang, Yuanhan, Kaiyang Zhou, and Ziwei Liu. 2022c. Neural prompt search. *arXiv preprint arXiv:2206.04673*.

Zheng, Yinhe, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *TASLP*.

Zhong, Xu, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *International conference on document analysis and recognition(ICDAR)*.

Zhou, Bolei, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Zhou, Ce, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Zhou, Kaiyang, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2022a. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, Kaiyang, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Conditional prompt learning for vision-language models. In *The ieee / cvf computer vision and pattern recognition conference (CVPR)*.

———. 2022c. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*.

Zhou, Kaiyang, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Learning to generate novel domains for domain generalization. In *European conference on computer vision (ECCV)*, 561–578.

Zhou, Kaiyang, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021a. Domain generalization with mixstyle. In *International conference on learning representations (ICLR)*.

Zhou, Wenxuan, Fangyu Liu, and Muhao Chen. 2021b. Contrastive out-of-distribution detection for pretrained transformers. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhou, Yunhua, Peiju Liu, and Xipeng Qiu. 2022d. KNN-contrastive learning for out-of-domain intent classification. In *Association for computational linguistics (ACL)*.

Zhou, Zhi, Lan-Zhe Guo, Zhanzhan Cheng, Yu-Feng Li, and Shiliang Pu. 2021c. Step: Out-of-distribution detection in the presence of limited in-distribution labeled data. In *Advances in neural information processing systems (NeurIPS)*.

Zhu, Zhuotun, Lingxi Xie, and Alan Yuille. 2017. Object recognition with and without objects. In *International joint conferences on artificial intelligence (IJCAI)*.