

# Understanding In-Context Learning via Synthetic and Controllable Datasets

by

Ziqian Lin

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

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 04/18/2025

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

Kangwook Lee, Assistant Professor, Electrical and Computer Engineering

Sharon Li, Assistant Professor, Computer Sciences

Jerry Zhu, Professor, Computer Sciences

Robert D. Nowak, Professor, Electrical and Computer Engineering



*To my family and mentors, whose guidance shaped my curiosity for science and inspired me to pursue knowledge with integrity.*

# Acknowledgements

First and foremost, I want to sincerely thank my advisor, Professor Kangwook Lee, who taught me how to conduct the right research. When we first met, I was good at experiments but had difficulty with writing and giving presentations. He provided me with detailed guidance on how to improve these weaknesses. He explained clearly how to prepare a research project presentation. I should answer these questions in my presentation: (i) What is the project? (ii) What is the current status? (iii) What did we discuss last week? (iv) What did we plan to do last week? (v) What did we achieve last week, and what were the results? (vi) What will we do next week? He also gave me valuable advice on writing, from basic details like distinguishing between “citet” and “citep,” to high-level aspects such as logical flow. I’m grateful to have observed how he revised my papers, which helped me understand both the rationale behind his changes and how to improve the flow. Gradually, I learned how to conduct the right and high-quality research. I greatly value our meetings. I feel lucky and honored to have Kangwook as my advisor.

I am grateful to my committee members, Professor Jerry Zhu, Professor Robert D. Nowak, and Professor Sharon Li. I first met Jerry in the course CS 861, Theory of

Machine Learning, which sparked my interest in theoretical research. Robert organized the SILO seminars<sup>1</sup>, which invited speakers that broadened my perspective on different research areas. Early in my Ph.D., I worked with Sharon, and together we successfully submitted a paper to CVPR, which started my research journey at the University of Wisconsin-Madison.

I am also thankful to the excellent mentors during my internships at Amazon in 2022 and Google in 2024. At Amazon, I was mentored by Hao Ding and collaborated with Nghia Trong Hoang, Branislav Kveton, Anoop Deoras, and Hao Wang. Although our first submission was not accepted, we continued working and eventually published our work at WSDM 2024. At Google, I was mentored by Yaojie Liu and collaborated with Runze Li, Yujia Chen, Vincent Chu, and Sharon Li. I learned valuable industrial experience, such as communication and reporting skills, from them.

This dissertation would not have been possible without the support from my collaborators: Jungtaek Kim, Thomas Zeng, Sreya Dutta Roy, Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, Yicong Chen, Liu Yang, and Shubham Kumar Bharti. I learned a great deal from them. Additionally, I am thankful for my lifelong friends Qingyun Wang, Yin Liu, and Liang Shang, with whom I studied and enjoyed life together. I would also like to acknowledge Angela Thorp, my graduate coordinator, who kindly answered my questions and provided support whenever I consulted her.

Lastly, I am deeply grateful to my parents. They supported me in pursuing my

---

<sup>1</sup><https://silo.wisc.edu/>

studies abroad. The fourth year of my Ph.D. was especially difficult because of health issues caused by poor eating habits and insufficient sleep. My mother came from China to Madison to care for me during this hard time, as she has always supported me throughout my life. I am incredibly lucky to have her as my mom.

*Thank you to everyone I've encountered along the way in my life for your help, guidance,  
and lessons I've received.*

# Contents

|                                                                               |             |
|-------------------------------------------------------------------------------|-------------|
| <b>Contents</b>                                                               | <b>v</b>    |
| <b>List of Tables</b>                                                         | <b>viii</b> |
| <b>List of Figures</b>                                                        | <b>xi</b>   |
| <b>Abstract</b>                                                               | <b>xxii</b> |
| <b>1 Introduction</b>                                                         | <b>1</b>    |
| 1.1 Large Language Models and In-Context Learning . . . . .                   | 1           |
| 1.2 Synthetic and Controllable Datasets for ICL in the Literature . . . . .   | 3           |
| 1.3 Two Missing Pieces in the Literature . . . . .                            | 7           |
| 1.4 Research Questions and Contributions . . . . .                            | 8           |
| <b>2 Explaining Early Ascent: Dual Operating Modes of In-Context Learning</b> | <b>10</b>   |
| 2.1 Overview . . . . .                                                        | 11          |
| 2.2 Pretraining and Data Generative Model . . . . .                           | 16          |
| 2.3 Inference and Dual Operating Modes . . . . .                              | 21          |

|          |                                                                                                              |           |
|----------|--------------------------------------------------------------------------------------------------------------|-----------|
| 2.4      | Early Ascent . . . . .                                                                                       | 25        |
| 2.5      | Bounded Efficacy of Biased-Label ICL . . . . .                                                               | 29        |
| 2.6      | Extension with Preliminary Results: Coded Prompts for Large Lan-<br>guage Models . . . . .                   | 32        |
| 2.7      | Conclusion . . . . .                                                                                         | 48        |
| <b>3</b> | <b>Incorporating Instructions into Synthetic ICL: In-Context Learning with<br/>Hypothesis-Class Guidance</b> | <b>51</b> |
| 3.1      | Overview . . . . .                                                                                           | 52        |
| 3.2      | Meta-Learning for ICL-HCG . . . . .                                                                          | 56        |
| 3.3      | Experiments . . . . .                                                                                        | 63        |
| 3.4      | Discussion . . . . .                                                                                         | 73        |
| 3.5      | Extension with Preliminary Results: Can Transformers Do In-Context<br>Blind Tree Search? . . . . .           | 74        |
| 3.6      | Conclusion . . . . .                                                                                         | 83        |
| <b>4</b> | <b>Contribution and Future Work</b>                                                                          | <b>84</b> |
| 4.1      | Contribution . . . . .                                                                                       | 84        |
| 4.2      | Future Work . . . . .                                                                                        | 86        |
| <b>A</b> | <b>For Chapter 2</b>                                                                                         | <b>89</b> |
| A.1      | Notations . . . . .                                                                                          | 89        |
| A.2      | Prior Examples . . . . .                                                                                     | 94        |
| A.3      | Coarse Upper Bound for ICL Risk . . . . .                                                                    | 97        |
| A.4      | Transformer Performance in Approximating Bayesian Inference . . .                                            | 99        |

|          |                                                                    |            |
|----------|--------------------------------------------------------------------|------------|
| A.5      | Additional Information for Bounded Efficacy in GPT-4 . . . . .     | 100        |
| A.6      | Bounded Efficacy in Zero-shot ICL . . . . .                        | 103        |
| A.7      | The Derivation of Posterior . . . . .                              | 104        |
| A.8      | Detailed Analysis of Component Shifting and Re-weighting . . . . . | 108        |
| A.9      | Additional Experiments for Early Ascent . . . . .                  | 115        |
| A.10     | Mathematical Derivation for Early Ascent . . . . .                 | 117        |
| A.11     | Proof Tools . . . . .                                              | 121        |
| A.12     | ICL to Learn the In-Context Function . . . . .                     | 125        |
| A.13     | ICL with Biased Labels to Retrieve A Task . . . . .                | 129        |
| A.14     | Proof of Lemma 6 . . . . .                                         | 146        |
| A.15     | Toy Example for Component Shifting and Component Re-weighting .    | 151        |
| <b>B</b> | <b>For Chapter 3</b>                                               | <b>167</b> |
| B.1      | Pseudo Algorithm for ICL-HCG . . . . .                             | 167        |
| B.2      | Implementation Detail of Hypothesis Prefix and Context Query . . . | 167        |
| B.3      | Additional Details of Experiments . . . . .                        | 170        |
| B.4      | Experimental Setup . . . . .                                       | 171        |
|          | <b>Bibliography</b>                                                | <b>178</b> |

# List of Tables

|     |                                                                                                                                                                                                                                                                                                                                                                                                         |    |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | <b>Bounded efficacy in GPT-4.</b> Error rate measured with respect to “addition (+)” and “biased +”. The bounded efficacy phenomenon: the error rate goes down to $k = 2$ , but it increases afterward. Experiment details in Appendix A.5.1. . . . .                                                                                                                                                   | 31 |
| 2.2 | <b>The illustration of coded and uncoded prompts with a real example.</b> A coded prompt predicts multiple samples in a single inference while an uncoded prompt predicts one sample in one inference. Uncoded prompt 1 is in the same format as the coded prompt, while uncoded prompt 2 is a more natural question format for prompting a single example. ✓/✗ = correct/incorrect prediction. . . . . | 44 |
| 2.3 | <b>Coded prompt vs. uncoded prompt.</b> F1-score comparison between coded prompt and uncoded prompt under different prime ranges and different sample batch sizes. F1-score is measured by averaging 400 batches. . . . .                                                                                                                                                                               | 45 |
| 2.4 | <b>Comparison of MSE and rate across different methods.</b> . . . . .                                                                                                                                                                                                                                                                                                                                   | 48 |

- A.1 Prior settings for early ascent.** The pretraining task prior comprises two components for one dimension and three for two or more dimensions. ICL aims to predict following the in-context function  $w^*$ , equivalent to prior center 2's function  $w_2$  ( $w^* = w_2$ ). The in-context task is characterized by having a closer  $x$  distribution to the task of prior center 1 but a closer  $x \rightarrow y$  mapping to the prior center 2. The parameters for all cases are set to  $\sigma_\mu = \sigma_w = 0.05$ ,  $\sigma_x = \tau_x = 1$ , and  $\sigma_y = 2$ . Refer to Fig. A.2b for visualization of the prior centers under dimension  $d \in \{1, 2, 3\}$ . . . . . 98
- A.2 Setup of bounded efficacy.** Experiment setting to reveal the bounded efficacy phenomenon of biased-label ICL in GPT-4. . . . . 101
- A.3 Zero in-context example ( $k = 0$ ).** Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt. . . . . 102
- A.4 Two in-context examples ( $k = 2$ ).** Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt. . . . . 102
- A.5 Eight in-context examples ( $k = 8$ ).** Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt. . . . . 103

- B.1 Hyperparameter search spaces for different model architectures.** The optimal hyperparameters are bolded if multiple possibilities are provided. 172
- B.2 Experimental setups of different generalizations.** The expression  $\min\{512, \#\text{possible}\}$  indicates that when the number of possible hypothesis classes is fewer than 512, we evaluate all possible hypothesis classes for testing; otherwise, we limit the selection to at most 512 hypothesis classes. For example, if  $|\mathcal{H}^{\text{OOD}}| = 16$  and  $|\mathcal{H}| = 2$ , the total number of possible hypothesis classes is given by:  $\binom{|\mathcal{H}^{\text{OOD}}|}{|\mathcal{H}|} = \binom{16}{2} = \frac{16 \times 15}{2} = 120$ . Since  $120 < 512$ , we evaluate all 120 hypothesis classes for testing in this scenario. . . . . 172
- B.3 Additional setups.** Numbers that differ from those in Table B.2 are highlighted in bold for clarity. . . . . 173

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | <b>A summary of our contributions.</b> We first propose a probabilistic model for pretraining data and in-context examples. By analyzing our model, we obtain a quantitative understanding of the dual operating modes of ICL, and explain two real-world phenomena observed with LLMs. . . .                                                                                                                                                                                                   | 13 |
| 2.2 | <b>Different pretraining data models.</b> Raventos et al. and ours. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                     | 18 |
| 2.3 | <b>Numerical experiments.</b> (Left) An illustration of the pretraining priors. (Right) The numerical computational results. . . . .                                                                                                                                                                                                                                                                                                                                                            | 20 |
| 2.4 | <b>Distance to the closest prior vs ICL risk.</b> We compute ICL risks of three target tasks colored red (farthest), green, and blue (closest), under the tetrahedron setting, illustrated in the left-most figure. The red target task has the longest distance to the closest prior center, and the blue target task has the shortest distance to the closest prior center. We can observe that the target task is easier to learn when the distance to the closest prior is smaller. . . . . | 27 |

- 2.5 **The early ascent phenomenon.** Fig. 2.5a and Fig. 2.5b show that the task retrieval mode is dominant up to  $k = 32$ , and component 1's mixture weight increases ( $\mathbb{E}[\tilde{w}]$  approaches  $w_1$ ). Since this component is farther than the other one, the risk starts increasing. At larger  $k$  values, the risk starts decreasing ( $\mathbb{E}[\tilde{w}]$  approaches  $w_2$ ) via task learning. See Appendix A.2.3 for setting details. We further examine the early ascent phenomenon under linear regression with varied levels of label noises in Appendix A.9.1, and under non-linear regression and discrete token prediction in Appendix A.9.2. . . . . 28
- 2.6 **Bounded efficacy.** The error rates of ICL with random labels start increasing at large  $k$ . See Appendix A.6 for more experimental results. . . . . 31
- 2.7 **Illustration of the analogy between information bit transmission in a noisy communication channel and LLM inference.** The communication channel transmits bit  $B$  with a probability  $P_e(B)$  of error occurrence, while LLM infers a sample with true label  $X$  and has a probability  $P_e(X; p)$  of making incorrect predictions. The notation  $W$  is for the noise introduced by channel or LLMs. . . . . 38
- 2.8 **Illustration of the analogy between *encoded* bit transmission in a noisy communication channel and *coded* LLM inference.** The communication channel transmits an encoded bit  $B_1 \oplus B_2$ , while LLM inference predicts multiple samples (two samples in this example) with true label  $X_1, X_2$ . The notation  $W$  is for the noise introduced by the channel or LLMs. . . . 38

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.9 | <b>Uncoded prompts vs. uncoded+coded+(dec2) prompts.</b> Each MSE pair in the scatter diagram represents one experiment, with a total of 100 experiments. The red line represents $y = x$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 49 |
| 3.1 | <b>Common ICL framework vs. ours.</b> Conventional frameworks with synthetic datasets often construct sequences by concatenating multiple $(x, y)$ pairs, overlooking the importance of instructions. In contrast, our approach explicitly incorporates instructions through a <i>hypothesis prefix</i> . Specifically, we transform the hypothesis class $\mathcal{H}$ into a sequence that is prepended to the sequence of $(x, y)$ pairs and then fed into a Transformer. We refer to this method as <i>in-context learning with hypothesis-class guidance (ICL-HCG)</i> . (Real-world examples are demonstrated using the GPT-4 Legacy model.) . . . . . | 53 |
| 3.2 | <b>Four types of generalization.</b> An illustration of the four types of generalization. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 60 |
| 3.3 | <b>Learning ICL-HCG via Transformer.</b> We begin by sampling a subset from the hypothesis universe as the hypothesis class $\mathcal{H}$ . Next, we encode the hypothesis class $\mathcal{H}$ and concatenate it with the context query into a unified sequence of tokens. This sequence is fed into a Transformer model for training with next-token prediction, and testing for evaluating the accuracy on $y$ 's and hypothesis identification. (This figure is a simplified illustration. Please refer to Appendix B.2 and Fig. B.1 for the full details.) . . . . .                                                                                    | 62 |

- 3.4 **The generation of training and testing hypothesis classes.** The hypothesis universe is divided into two pools: one for generating training and ID testing hypothesis classes, and another for generating OOD testing hypothesis classes. . . . . 63
- 3.5 **Multiple runs on ID and OOD hypothesis class generalizations.** (Different runs imply training and testing with different random seeds.) Transformer successfully learns ICL-HCG and generalizes to new hypothesis classes and hypotheses. Generalization on ID hypotheses is easier than on OOD hypotheses. Refer to Appendix B.3.1, Fig. B.2 for more curves of loss, training, and testing accuracy. . . . . 65
- 3.6 **Multiple runs on ID and OOD hypothesis class size generalizations.** (Different runs imply training and testing with different random seeds.) Transformers trained on hypothesis classes with sizes  $|\mathcal{H}| \in \{7, 8, 9\}$  successfully generalize to hypothesis classes with sizes  $|\mathcal{H}| \in \{2, 3, \dots, 13, 14\}$  under ID hypothesis class size generalization. In contrast, the same trained Transformer exhibits poorer performance on OOD hypothesis class size generalization. In the figure, *IS* stands for “in-size,” indicating the hypothesis class sizes included in the training, while *OOS* stands for “out-of-size,” indicating the sizes that are **not** included in the training. Refer to Appendix B.3.1, Fig. B.3 for training accuracy curves. . . . . 66

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>3.7 Various models on ID and OOD hypothesis class generalizations.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
| Transformer and Mamba succeed on both ID and OOD hypothesis class generalization, whereas LSTM and GRU fail. Mamba exhibits slightly higher accuracy than Transformer on OOD generalization. Refer to Appendix B.3.2 and Fig. B.4 for training curves. . . . .                                                                                                                                                                                                                                                                                                                      | 67 |
| <b>3.8 Various models on ID and OOD hypothesis class size generalizations.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |
| In both settings, Transformers and Mamba exhibit strong generalization, whereas LSTM and GRU fail to do so. For hypothesis class sizes $ \mathcal{H}  \in \{7, 8, 9\}$ , Mamba achieves accuracy comparable to Transformer on ID hypothesis class generalization, and surpasses Transformer on OOD hypothesis class generalization. However, Transformers show similar or higher accuracy than Mamba on ID hypothesis class size generalization, suggesting a potential advantage in length generalization. Refer to Appendix B.3.2, Fig. B.5 for training accuracy curves. . . . . | 68 |
| <b>3.9 Effect of training hypothesis class count.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
| Transformer and Mamba trained on ICL-HCG tasks generalize to new hypothesis classes with only 4 to 16 training hypothesis classes. Refer to Appendix B.3.3, Fig. B.6 for training accuracy and more details. . . . .                                                                                                                                                                                                                                                                                                                                                                | 69 |
| <b>3.10 The effect of sample imbalance.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |    |
| Sample imbalance leads to lower convergence speed. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 70 |

|      |                                                                                                                                                                                                                                                                                                                         |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.11 | <b>The effect of instruction.</b> Under ID hypothesis class generalization, providing an instruction (hypothesis prefix) significantly boosts ICL performance, especially when the $y$ token appears early (indicating only a few demonstration examples precede it). . . . .                                           | 71 |
| 3.12 | <b>The effect of pretraining hypothesis diversity.</b> Under hypothesis generalization, increasing the diversity of pretraining hypotheses significantly boosts the performance of ICL when instructions are provided. However, without instructions, this effect is limited. . . . .                                   | 73 |
| 3.13 | <b>A maze example with size <math>6 \times 6</math>.</b> . . . . .                                                                                                                                                                                                                                                      | 80 |
| 3.14 | <b>Comparison on MCTS with varied <math>C</math> values.</b> . . . . .                                                                                                                                                                                                                                                  | 81 |
| 3.15 | <b>Comparison on different <math>C</math> values.</b> We train Transformers on trajectories generated by MCTS with different UCT exploration constant $C$ values. During the training process, Transformers learn from the MCTS to perform search, and $C = 0.1$ achieves the best performance similar to MCTS. . . . . | 82 |
| 4.1  | <b>Works in this thesis and future directions.</b> . . . . .                                                                                                                                                                                                                                                            | 85 |

|                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>A.1 Visualization of the tetrahedron setting.</b> | The figure shows the pretraining prior centers and the in-context task. For $\beta \in \{1, 2, 3, 4\}$ , $(\boldsymbol{\mu}_\beta, \boldsymbol{w}_\beta)$ is a mixture component center in the prior. $(\boldsymbol{\mu}_\alpha, \boldsymbol{w}_\alpha)$ for $\alpha = 1$ (numbers are noted in the center of circles) is the center of the target task for ICL with biased labels, while $(\boldsymbol{\mu}^*, \boldsymbol{w}^*)$ is the in-context task. The dotted purple lines highlight the distance of 1 from the origin $(0, 0, 0)$ to any point denoted by $\boldsymbol{\mu}$ or $\boldsymbol{w}$ . . . . .   | 94  |
| <b>A.14 Illustration of the function</b>             | $\exp(k + 2)/(1 + \exp(k + 2))$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 120 |
| <b>A.2 The early ascent phenomenon.</b>              | Fig. A.2a displays the trends of expected losses, upper bounds, and mixture weights, while Fig. A.2b presents the trend of the expectation of $\tilde{\boldsymbol{w}}$ . We can see that the task retrieval mode is dominant up to $k = 32$ , and component 1's mixture weight increases ( $\mathbb{E}[\tilde{\boldsymbol{w}}]$ approaches $\boldsymbol{w}_1$ ). Since this misleading component 1 is far from the target component 2, the risk starts increasing. At larger $k$ values, the risk starts decreasing ( $\mathbb{E}[\tilde{\boldsymbol{w}}]$ approaches $\boldsymbol{w}_2$ ) via task learning. . . . . | 158 |

- A.3 In-context learning vs ridge regression.**  $\mathcal{R}^*$  indicates the prediction by ridge regression,  $\mathcal{F}^*$  indicates the prediction by ICL with a Bayes-optimal next-token predictor, and  $y_{k+1}^* = \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle$ . Let the  $k$  samples draw from a task  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ , which is drawn from the pretraining prior distribution. The dimension  $d$  of  $\mathbf{x}$  equals 6. We observe that ICL performs better than ridge regression when  $k$  is small, and ridge regression performs better than ICL when  $k \geq d$ . Especially, when the task prior distribution has high task variance (big  $\delta_\mu$  and  $\delta_w$  values), ICL and ridge regression have very similar performance. . . . . 159
- A.4 Prior task noises.** The figure shows the experiment results under varied noise levels.  $\delta_\mu$  and  $\delta_w$  indicate the noise levels of the pretraining task prior.  $\mathcal{F}^*$  indicates the prediction of Bayesian inference while  $\hat{\mathcal{F}}$  indicates the prediction of the trained Transformer network. The results show that the trained Transformer network's performance can approach the performance of Bayesian inference. . . . . 159
- A.5 Number of components.** The figure shows the experiment results under varied component densities.  $M$  indicates the number of mixture components corresponding to different 3D regular polyhedrons described in Appendix A.2.1, and  $\delta_\mu = \delta_w = \frac{1}{16}$ .  $\mathcal{F}^*$  indicates the prediction of Bayesian inference while  $\hat{\mathcal{F}}$  indicates the prediction of the trained Transformer network. The higher the component density is, the harder it is for the Transformer network to approach Bayesian inference. . . . . 160

|                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |     |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>A.6 Experiments on varying feature dimensions.</b>                     | The figure shows the experiment results under varied dimensions. $d$ indicates the dimension and the number of mixture components (see Appendix A.2.2 for setting details), and $\delta_\mu = \delta_w = \frac{1}{16}$ . $\mathcal{F}^*$ indicates the prediction of Bayesian inference while $\hat{\mathcal{F}}$ indicates the prediction of the trained Transformer network. The higher the feature dimension is, the harder it is for the Transformer network to approach Bayesian inference. . . . . | 161 |
| <b>A.7 Ablations on varying numbers of examples in the demonstrations</b> | $(k)$ . Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used. . . . .                                                                                                                                                                                                                                                                                                                                                                      | 161 |
| <b>A.8 Bounded efficacy phenomenon of real-world LLMs.</b>                | As $k$ increases, the classification error curve of ICL with random labels exhibits the bounded efficacy phenomenon. The curve with true labels further confirms that this phenomenon is not due to models tending to perform worse on long sequences. . . . .                                                                                                                                                                                                                                           | 162 |
| <b>A.9 Numerical analysis on component re-weighting.</b>                  | The trends of $\Psi_\mu$ , $\Psi_w$ , and $\pi_m$ for CR with increasing $k$ under varying task noise parameters. .                                                                                                                                                                                                                                                                                                                                                                                      | 162 |
| <b>A.10 The trend of Component Shifting.</b>                              | Numerical computations of $\ \tilde{\mu}_m - \mu^*\ $ , $\ \tilde{w}_m - w^*\ $ for Component Shifting (CS). . . . .                                                                                                                                                                                                                                                                                                                                                                                     | 163 |
| <b>A.11 Early ascent under varied label noises.</b>                       | Results show that the early ascent phenomenon maintains for noise level $\tau_y \in [0, 1.0]$ . Label noise level $\sigma_y = 1.0$ is used for pretraining. . . . .                                                                                                                                                                                                                                                                                                                                      | 163 |

- A.12 Bounded efficacy under varied label noises.** Results show that the bounded efficacy phenomenon maintains for noise level  $\tau_y \in [0, 0.1]$ . Label noise level  $\sigma_y = 1.0$  is used for pretraining. . . . . 164
- A.13 Early ascent on non-linear regression and discrete token prediction.**  $\hat{\mathcal{F}}$  indicates the prediction by a pretrained Transformer model and  $\mathcal{F}^*$  indicates the prediction by numerical computation following a Bayes optimal predictor. While we cannot derive the optimal predictor under non-linear regression, we can derive the optimal predictor under discrete token prediction. . . . . 164
- A.15 The numerical computation of the task learning.** The second and third rows show the eigenvalues of the matrices  $\delta_w \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k}$  and  $\mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top$ . The fourth row shows the distance between the predicted  $\tilde{\mathbf{w}}$  and  $\mathbf{w}^*$  has a reciprocal decreasing rate with respect to  $k$ . The fifth and sixth rows indicate the expected squared loss follows a quadratic decreasing rate with respect to  $k$ . . . . . 165
- A.16 Proof roadmap of ICL with biased labels, Theorem. 5.** . . . . . 166
- A.17 Visualization of prior, posterior, and observations.** The left part of the figure indicates the pretrained next-token predictor is pretrained on the task prior distribution according to Assumption 10, and the prediction is based on the prior without in-context examples. The right part of the figure indicates that with in-context samples, the prediction is based on the posterior, regarding the in-context examples as observed samples. . . 166

|     |                                                                                                                                                                                                                                         |     |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| B.1 | <b>The framework.</b> We convert hypothesis class $\mathcal{H}$ and ICL sequence $S_K$ into sequences of tokens, concatenate them and input to Transformer. Then we examine whether Transformer can predict correct $y$ and $z$ values. | 169 |
| B.2 | <b>Multiple runs for ID and OOD hypothesis class generalizations.</b>                                                                                                                                                                   | 170 |
| B.3 | <b>Multiple runs for ID and OOD hypothesis class size generalizations.</b>                                                                                                                                                              | 174 |
| B.4 | <b>Various models on ID and OOD hypothesis class generalizations.</b>                                                                                                                                                                   | 175 |
| B.5 | <b>Various models on ID and OOD hypothesis class generalizations.</b>                                                                                                                                                                   | 176 |
| B.6 | <b>Effect of training hypothesis class count on ID and OOD hypothesis class generalization.</b>                                                                                                                                         | 177 |

# Abstract

Large language models (LLMs) have become a central component in modern natural language processing, exhibiting strong capabilities across a wide range of tasks. A key feature of this success is in-context learning (ICL), where LLMs improve their prediction performance on new tasks by conditioning on a set of input-output examples provided in the prompt, without requiring any parameter updates. Despite its empirical effectiveness in applications such as text classification, question answering, and translation, the underlying mechanism of ICL remains not fully understood from a theoretical perspective.

To address this challenge, this thesis leverages **synthetic and controllable datasets** as a fundamental tool for understanding ICL. While existing works have extensively leveraged synthetic and controllable datasets such as those [31, 2, 127] based on linear regression tasks to study ICL, we identify two critical missing pieces in the current literature. The first missing piece is that an empirically observed phenomenon, named “early ascent” in this thesis, remains theoretically unexplained: although we expect the prediction error to decrease as the number of in-context samples increases, we empirically found that the error increases and decreases

for some experiments [11]. The second missing piece is that existing synthetic datasets overlook the instructions used in ICL: while in real-world use cases of ICL, in-context samples are coupled with task instructions that guide the model’s behavior, the current synthetic and controllable datasets in the literature overlook the role of instructions. This thesis fills these two missing pieces by constructing and studying synthetic and controllable datasets.

To fill the first missing piece, we propose a novel probabilistic model for generating pretraining tasks and then pretraining sequences from those tasks, where the pretraining tasks have a latent clustered structure. We analyze such a synthetic and controllable data model and reveal that ICL operates in two distinct modes: task retrieval and task learning. Then we provide the first mathematical explanation for the early ascent phenomenon. Furthermore, we predict the phenomenon of “bounded efficacy” named in this thesis, where the prediction error of ICL with randomly labeled examples initially decreases but eventually increases as the number of in-context examples increases. To fill the second missing piece, we propose a new synthetic and controllable data framework, namely *In-Context Learning with Hypothesis-Class Guidance (ICL-HCG)*, which explicitly incorporates task instructions into synthetic data generation. Through extensive experiments, we demonstrate that the instruction significantly improves the prediction accuracy and generalization ability of ICL compared to ICL without instruction.

Additionally, this thesis explores future directions with preliminary results. Inspired by the dual operating modes of ICL, we propose a new prompt design method named *coded prompts* to enhance LLM performance when multiple test

samples are available at inference time. We also study a new problem setting, *in-context blind tree search*, extending ICL to sequential decision-making scenarios involving action-reward loops.

In summary, this thesis develops synthetic and controllable datasets to enable mathematical analysis, provides new theoretical and empirical insights to our understanding of ICL, and improves LLMs' performance in practice.

# Chapter 1

## Introduction

### 1.1 Large Language Models and In-Context Learning

Large language models (LLMs) have demonstrated great capabilities in solving various complex tasks via simple prompts [131]. Among their capabilities, in-context learning (ICL) [11] is really impressive, because ICL enables LLMs to adapt to new tasks by conditioning on a sequence of input-output examples provided in the prompt, without requiring any explicit parameter updates. In the standard ICL setting, the model is given a set of  $k$  samples  $\{(x_i, y_i)\}_{i=1}^k$  corresponding to a particular task that serve as in-context samples, followed by a new input  $x_{\text{query}}$  for which the model is expected to predict the output  $\hat{y}_{\text{query}}$ . The entire sequence is converted into a single prompt and fed into an LLM, such as a pretrained Transformer model:

$$\text{LLM}(x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_{\text{query}}) \Rightarrow \hat{y}_{\text{query}}.$$

This process allows the model to infer the underlying task from the provided examples and make the prediction without the need for any fine-tuning and parameter updating. ICL has become a powerful paradigm for enabling LLMs to perform many different tasks by leveraging only a few in-context samples, such as text classification [6, 72, 75], question answering [8, 17, 50], and translation [126, 105]. However, a comprehensive mathematical and theoretical understanding of ICL remains open.

To systematically understand the mechanism of ICL, many works have utilized **synthetic and controllable datasets** to facilitate detailed experimental and theoretical studies. An example is the linear regression setup introduced by Garg et al. [31], which is the first work on utilizing the synthetic and controllable dataset to study ICL. In this setting, for each sequence,  $x$ 's are sampled from an isotropic Gaussian, and  $y$ 's are generated by linear mappings  $\langle x, w \rangle$  where  $w$  is a coefficient specific to each sequence. Such a synthetic and controllable dataset allows us to train Transformer models from scratch, explore more specific aspects of ICL, and give precise explanations of how LLMs utilize in-context samples during ICL.

This chapter first reviews existing works using synthetic and controllable datasets to study ICL, then identifies two missing pieces of the literature, and finally introduces the research questions and contributions in this thesis.

## 1.2 Synthetic and Controllable Datasets for ICL in the Literature

Garg et al. [31] first build a synthetic and controllable dataset where the pretraining and testing sequences are generated from noiseless linear regression tasks. Specifically, for each sequence, Garg et al. [31] first sample  $\mathbf{w}$  from a Gaussian distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ , representing the underlying task. Then, a number of vectors  $\{\mathbf{x}\}_{i=1}^K$  are sampled from an isotropic Gaussian distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ , and  $\{y\}_{i=1}^K$  are generated by  $y^{(i)} = \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle$ . Finally, those  $\mathbf{x}$ 's and  $y$ 's are concatenated into a sequence  $(\mathbf{x}^{(1)}, y^{(1)}, \mathbf{x}^{(2)}, y^{(2)}, \dots, \mathbf{x}^K, y^K)$  for pretraining. For inference, the input sequence follows the format  $(\mathbf{x}^{(1)}, y^{(1)}, \mathbf{x}^{(2)}, y^{(2)}, \dots, \mathbf{x}^k, y^k, \mathbf{x}_{\text{query}})$ , where the  $k$  pairs  $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^k$  following a new sampled  $\mathbf{w}$  serve as in-context samples. The model then predicts the output for  $\mathbf{x}_{\text{query}}$  conditioned on  $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^k$ .

**Noiseless Linear Regression** Based on the well-defined synthetic and controllable dataset by Garg et al. [31] using noiseless linear regression tasks, researchers systematically study the mechanisms of ICL and properties of Transformers. For instance, there is an exciting line of research on connecting ICL to gradient descent, firstly hinted by Garg et al. [31]. Akyürek et al. [2] and Von Oswald et al. [112] then show that one attention layer can be exactly constructed to perform gradient descent, and empirically find similarities between ICL and gradient descent. Further, Ahn et al. [1] theoretically show that under certain conditions, Transformers trained on noiseless linear regression tasks minimizing the pretraining loss will implement a gradient descent algorithm. Nevertheless, Fu et al. [27] show that Transformers

are able to approximate the second-order optimization methods, sharing a similar convergence rate as iterative Newton’s method. Besides gradient descent, there are lots of other interesting topics on ICL and Transformers based on this linear regression setting, such as looped Transformer [122, 32], training dynamic [128, 44, 51], generalization [83], etc.

**Noisy Linear Regression** Such a simple noiseless linear regression task is further extended to variants. By extending the linear regression to noisy linear regression  $y = \langle \mathbf{x}, \mathbf{w} \rangle + \epsilon$ , Li et al. [60] analyze the generalization and stability of ICL. Wu et al. [117] and Raventós et al. [91] analyze the effect of task diversity on the risk in ICL. With the tasks of the format  $\mathbf{y} = \mathbf{W}\mathbf{x} + \epsilon$ , where  $\mathbf{W}$  is a matrix rather than a vector, Chen et al. [13] examine the training dynamic of the multi-head attention in ICL.

**More than Linear Regression** Beyond linear regression, researchers are also interested in non-linear regression and classification. The research directions are scattered, and we list them as follows. Bai et al. [4] show that Transformers can perform in-context algorithm selection, *i.e.*, adaptively selecting different ICL algorithms such as gradient descent, least square, or ridge regression. Bhattamishra et al. [7] show that Transformers can learn a variety of Boolean function classes. Cheng et al. [15] provide evidence that Transformers can learn to implement gradient descent to enable them to learn non-linear functions. Guo et al. [37] show that trained Transformer achieves near-optimal ICL performance under  $y = \langle \mathbf{w}, f(\mathbf{x}) \rangle$ , where  $f$  is a shallow neural network. Examining linear and non-linear regression tasks, Fan et al. [25] and Tripuraneni et al. [107] show Transformer can perform

ICL on composited or mixed tasks of pretrained linear or non-linear tasks, and Yadlowsky et al. [121] examine whether trained Transformers can generalize to new tasks beyond pretraining. Park et al. [84] examine whether Mamba can in-context learn a variety of synthetic tasks, such as sparse linear regressions and decision trees. Via examining regression and classification tasks, Kim et al. [52] show that task diversity helps shorten the ICL plateau pretraining. Ramesh et al. [88] assume there are multiple functions composited to connect  $\mathbf{x}$  and  $\mathbf{y}$  pair, *e.g.*,  $\mathbf{y} = f_1 \circ f_2 \circ f_3(\mathbf{x})$  to study the compositional capabilities of Transformer. Li et al. [58] study how the non-linear Transformer learns binary classification.

**Synthetic and Controllable Datasets with Images** Beyond simply leveraging vectors as  $\mathbf{x}$ 's, researchers also consider images as those  $\mathbf{x}$ 's, *i.e.*, the training sequence  $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, \mathbf{x}^{(2)}, \mathbf{y}^{(2)}, \dots)$  has image-label pairs serving as in-context samples. Chan et al. [12] first examine how the pretraining data properties affect ICL under this setting. Singh et al. [102] and Reddy [93] further examine the effect of data on the dynamics of ICL and in-weight learning. Fu et al. [28] study the learning plateaus of ICL with similar pretraining sequences composed of image-label pairs.

**Synthetic and Controllable Datasets with Instruction** To the best of our knowledge, there are two articles on synthetic and controllable datasets with instructions. Huang and Ge [43] append an additional vector  $\boldsymbol{\mu}$  to the sequences with  $(\mathbf{x}, \mathbf{w}^\top \mathbf{x})$  pairs, which leads to the sequence  $(\boldsymbol{\mu}, \mathbf{x}^{(1)}, \mathbf{w}^\top \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{w}^\top \mathbf{x}^{(2)}, \dots)$  where  $\mathbf{x}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ , and show that the trained Transformer can achieve significantly lower loss on ICL when the task descriptor  $\boldsymbol{\mu}$  is provided. Xuanyuan et al. [120]

develop a new synthetic and controllable dataset based on task  $((a \cdot x) \circ (b \cdot y)) \bmod p = r$ , where  $(x, y)$  is the input,  $r$  is the output,  $\circ$  is an operation  $(+, -, /)$ , and each task is defined by the parameters  $(a, b, \circ)$  ( $p$  is a constant). The instruction is constructed as  $(a_l, a_u, b_l, b_u, \circ)$ , where  $a_l$  and  $a_u$  are the lower and upper bounds of  $a$  (similar for  $b$ ), and  $\circ$  is the operation. Therefore, the instruction constrains the possible tasks, providing information on the underlying task of in-context samples. Under such a setting, Xuanyuan et al. [120] study how the information provided by instruction affects the accuracy of ICL.

**Synthetic and Controllable Datasets with Autoregressive Format** The pretraining sequences in this type of synthetic and controllable datasets follow the format  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots)$ . These tokens are generated from a next-token generative model, such as the Hidden Markov Model (HMM) [118], Probabilistic Finite Automata (PFA) [3], or simply a function  $\mathbf{x}^{(i+1)} = f(\mathbf{x}^{(i)}) + \epsilon$  [60, 96]. Xie et al. [118] first proposes to use multiple Hidden Markov Models (HMMs) to represent latent concepts in real-world language and generate pretraining sequences, then explain ICL via a Bayesian perspective. Han et al. [38] leverages the same setting and explains ICL via kernel regression. Sequences under this format are also generated by Akyürek et al. [3] using Probabilistic Finite Automata (PFAs), Edelman et al. [24] using Markov Chain, and Nichani et al. [78] using Markov Chain with causal structure, to study the induction head [79] of LLMs. While the abovementioned datasets only consider single-step dependence between tokens, i.e., the next token only depends on one of the previous tokens, Makkuva et al. [70] further explore higher-order Markov chains.

### 1.3 Two Missing Pieces in the Literature

While existing works have constructed diverse synthetic and controllable datasets and leveraged them to study ICL, we identify two critical missing pieces in the current literature.

**An Unexplained Phenomenon in Real-World ICL: Early Ascent** One missing piece is an unexplained phenomenon observed in real-world ICL named “early ascent” by us [61], initially documented by Brown et al. [11]. It describes a situation where the error rate of ICL initially increases with a small number of in-context samples and then decreases with additional samples. This phenomenon is weird since we expect that more examples always bring benefits to the prediction. Such a phenomenon has not been mathematically explained in the literature.

**A Gap Between Synthetic and Real-World ICL: Instruction** Another missing piece is that the current synthetic and controllable datasets overlook the role of instructions. The instructions are important to ICL since they explicitly provide information about the target task to the model and improve the prediction accuracy of ICL, demonstrated by Brown et al. [11]. However, existing synthetic and controllable datasets overlook instructions, leaving a critical gap between synthetic and real-world ICL. Thus, a synthetic and controllable dataset that explicitly integrates instructions is needed to better mimic the real-world ICL.

## 1.4 Research Questions and Contributions

Motivated by the abovementioned two missing pieces, this thesis aims to fill these missing pieces and enhance our understanding of ICL by designing and studying new synthetic and controllable datasets. Specifically, the thesis investigates research questions and brings corresponding contributions as follows:

1. **Can we design a synthetic and controllable dataset for ICL, with which we can better understand the mechanism of ICL and mathematically explain the early ascent phenomenon? [61]**
  - We propose a novel synthetic and controllable dataset, which has an underlying probabilistic model for generating synthetic pretraining data, modeling the latent clustered structure in practical tasks, and explaining ICL via two modes: task learning and task retrieval.
  - Leveraging our proposed synthetic and controllable dataset, we analyze the Bayesian inference dynamics of ICL, providing the first mathematical explanation for the early ascent phenomenon.
  - Leveraging our proposed synthetic and controllable dataset, we predict the bounded efficacy phenomenon. In this phenomenon, the error rate of ICL with randomly labels first decreases, but eventually increases as the number of in-context samples increases.
2. **Can we design a synthetic and controllable dataset for ICL, with which we are able to incorporate instructions into ICL to bridge the gap between synthetic**

**and real-world ICL in the literature? [62]**

- We design a new synthetic and controllable data framework, namely *In-Context Learning with Hypothesis-Class Guidance (ICL-HCG)*, which explicitly incorporates task instructions into synthetic ICL and enables controlled experiments on ICL with instruction.
- Through various experiments, we show that a model trained with instructions significantly outperforms one trained without instructions. We further show that incorporating instructions for pretraining increases the generalizability of the pretrained model compared to one without instructions.

Additionally, this thesis provides preliminary results on future directions beyond the main studies. Building on insights from the dual operating modes of ICL, we propose a new approach to prompt design, namely *coded prompts*, to enhance the prediction accuracy of a real-world LLM when multiple test samples are available at inference time. Extending the synthetic and controllable dataset construction beyond ICL-HCG, we explore a more complicated problem setting, namely “in-context blind tree search,” which extends ICL to a sequential decision-making process involving action-reward loops. The preliminary results of these two directions are presented after each main chapter, illustrating how our synthetic and controllable datasets inspire future research.

In summary, we construct and study new synthetic and controllable datasets, contribute to both theoretical understanding and real-world application of ICL, and fill the two missing pieces in the literature.

## Chapter 2

# Explaining Early Ascent: Dual Operating Modes of In-Context Learning

ICL exhibits dual operating modes [82]: *task learning*, *i.e.*, acquiring a new skill from in-context samples, and *task retrieval*, *i.e.*, locating and activating a relevant pretrained skill. Recent theoretical work proposes various mathematical models to analyze ICL, but they cannot fully explain the duality. In this work [61], we analyze a generalized probabilistic model for pretraining data, obtaining a quantitative understanding of the two operating modes of ICL. Leveraging our analysis, we provide the first explanation of an unexplained phenomenon observed with real-world LLMs. Under some settings, the ICL risk initially increases and then decreases with more in-context examples. Our analysis offers a plausible explanation for this

“early ascent” phenomenon: a limited number of in-context samples may lead to the retrieval of an incorrect skill, thereby increasing the risk, which will eventually diminish as task learning takes effect with more in-context samples. We also analyze ICL with biased labels, *e.g.*, zero-shot ICL [68], where in-context examples are assigned random labels, and predict the bounded efficacy of such approaches. We corroborate our analysis and predictions with extensive experiments with Transformers and LLMs.

## 2.1 Overview

LLMs exhibit a significant improvement in predictive performance when provided with in-context examples [11]. This emergent ability of LLMs, known as ICL, **operates in two distinct modes: task learning and task retrieval** [82]. Large language models exemplify this duality. They can learn unseen functions from in-context examples, demonstrating the learning mode [11, 92, 31]. Concurrently, LLMs can also retrieve and utilize a *pretrained* skill. A clear evidence of the task retrieval mode is presented by Min et al. [73], where the authors show ICL performance remains largely unaffected even when in-context examples are annotated with random labels. This suggests that LLMs simply retrieve a pretrained skill rather than learn it from in-context examples.

The dual nature of ICL can be explained as follows. LLMs are a next-token predictor that is pretrained on a large pretraining set, consisting of diverse data from diverse domains/tasks. To predict the next token optimally in such a scenario,

the model must first learn the task prior from pretraining data and then implicitly perform Bayesian inference at the test time [118, 90]. Optimal prediction on multi-task pretraining data requires adherence to the learned prior (over the tasks present in the pretraining data) and making predictions based on the posterior. The ability to learn and apply this prior during test-time inference enables task retrieval—if in-context examples align closely with a task encountered during pretraining, the model can swiftly adjust its posterior and predict without learning a new skill. Simultaneously, the model can learn a novel or uncommon skill given sufficient in-context samples and a non-zero prior probability for that skill.

Although the link between pretraining and ICL’s dual modes is conceptually straightforward, formally establishing this connection is an unresolved challenge. Motivated by this, our work seeks to address the following questions:

*How do we rigorously explain the dual operating modes of ICL? Can we define the conditions under which the retrieval mode is a dominant one and vice versa?*

**A New Model for Pretraining Data** To find the answers to these questions, we first propose a new probabilistic model for pretraining data by assuming the pretraining data has a latent clustered structure. In particular, we consider in-context learning of linear functions following the recent work [31, 2, 60, 112, 90, 117]. A next-token prediction model is prompted with (1) a sequence of  $(\mathbf{x}, y)$  pairs, which come from a common linear function, and (2) one test input  $\mathbf{x}_{\text{test}}$ . An ideal model capable of in-context learning linear models should internally fit a linear function (say  $y = \hat{\mathbf{w}}^T \mathbf{x}$ ) using the in-context examples and then generate the predicted label  $y_{\text{test}} = \hat{\mathbf{w}}^T \mathbf{x}_{\text{test}}$

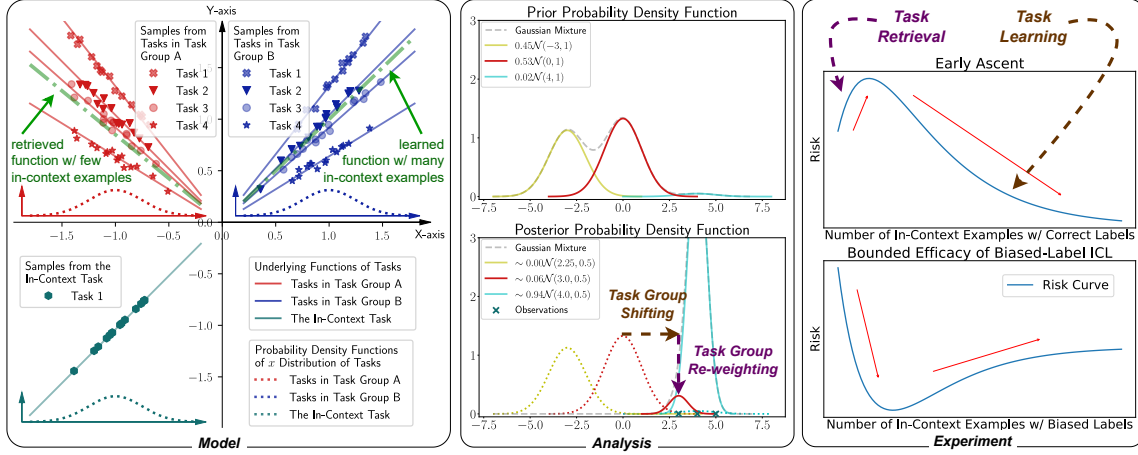


Figure 2.1: **A summary of our contributions.** We first propose a probabilistic model for pretraining data and in-context examples. By analyzing our model, we obtain a quantitative understanding of the dual operating modes of ICL, and explain two real-world phenomena observed with LLMs.

as the next token. The recent work [90, 117] show that such in-context learning is feasible by training a next-token prediction model on a large pretraining dataset, consisting of sequences of labeled samples drawn from diverse linear functions.

We extend the existing model for pretraining data [90] by introducing multiple task groups and task-dependent input distributions. When one generates pretraining data, one must specify a probability distribution of linear functions (equivalently, that of the linear coefficient  $w$ ). While most of the prior work assumes that  $w$  is drawn from a single Gaussian distribution, we will model it as drawn from a Gaussian *mixture* model, where each Gaussian component models a *task group*. This model better reflects real-world data that exhibits a clustered structure [118]. Furthermore, we also allow each mixture component to have its own distribution for input  $x$ . Shown on the left-most panel in Fig. 2.1 is a simple

visualization of our model. The blue task group is modeled as the distribution of linear functions with positive coefficients ( $w \approx 1$ ) with the input distribution centered around  $\mathbb{E}[x] = +1$ . The red lines represent the other task group – linear functions with negative coefficients ( $w \approx -1$ ) with the input distribution centered at  $\mathbb{E}[x] = -1$ . See Sec. 2.2 for more details.

**Analysis** With our new model for pretraining data, we analyze the optimal pre-trained model under the squared loss, *i.e.*, the MMSE estimator of the label given input with in-context examples. Here, the pretraining distribution (of linear functions) is the prior, and in-context examples are the observations. Leveraging the fact that the Gaussian mixture is a conjugate prior to the Gaussian likelihood function, we obtain a closed-form expression of the posterior distribution. By fully quantifying the posterior distribution of  $w$  in the form of a Gaussian mixture, we characterize how in-context examples are used to update each component’s posterior mean and posterior mixture probability. We will call updates of mixture probabilities as *task group (component) re-weighting* and updates of component means as *task group (component) shifting*. See the central panel in Fig. 2.1 for visualization. By analyzing these two effects, we obtain a quantitative understanding of how two different operating modes emerge. In particular, we show that, under some mild assumptions, task group re-weighting is the dominant factor when provided with few in-context samples, rendering the task retrieval mode. With many in-context samples, task group shifting occurs, resulting in the task learning mode.

**Explanation of Two Real-World Phenomena** To demonstrate the practical value of the new insights we have gained from our model, we will leverage our analysis to explain and predict two phenomena observed with LLMs in practice.

- **The *early ascent* phenomenon** refers to the observation that, under certain conditions, the ICL risk initially increases and then decreases when more in-context examples are introduced [11, 118]. See the right-most panel of Fig. 2.1 for visualization. Based on our analysis, we offer a plausible explanation for this early ascent phenomenon—a limited number of in-context samples may lead to the retrieval of an incorrect skill, thereby increasing the risk, which will eventually diminish as task learning takes effect with more in-context samples.
- **Bounded efficacy of biased-label ICL** is predicted by our model. ICL performs well even with in-context examples that are annotated with biased labels [68, 73]. Our model provides a rigorous justification of this approach: If in-context examples with biased labels carry sufficient information for retrieving a correct pretrained task, then this approach would work. At the same time, our analysis suggests that the operating mode of ICL will make a transition from task retrieval to task learning with more in-context examples. When the learning mode starts taking place, the test risks of such methods will start increasing as the pretrained model will start fitting the biased labels. See the right-most panel of Fig. 2.1 for visualization. This bounded efficacy has not been reported in the literature [73, 82]. We found that this was due to the small number of examples tested. With more in-context samples, we observe the predicted bounded efficacy phenomenon with real-world LLMs such as Mistral 7B [47], Mixtral 8×7B [48], Llama 2 [106], and GPT-4 [80].

## 2.2 Pretraining and Data Generative Model

A next-token predictor is a sequential prediction model that predicts the next token given an initial token sequence. Consider pretraining this model on sequences consisting of  $(\mathbf{x}, y)$ <sup>1</sup> pairs in the form of  $(\mathbf{x}_1, y_1, \mathbf{x}_2, y_2, \dots)$ , with the model trained to predict only the  $y$  values, thereby skipping the prediction of  $\mathbf{x}$ . Here, we assume odd-numbered tokens represent  $d$ -dimension real-valued vectors, and even-numbered tokens represent scalars. During inference, the model receives a sequence of  $2k + 1$  tokens. The first  $2k$  tokens are  $k$  labeled samples  $(\mathbf{x}_i, y_i), i \in \{1, \dots, k\} =: [K]$ , and the last token is unlabeled  $\mathbf{x}_{k+1}$ . Ideally, the model should predict  $y_{k+1}$  correctly.

### 2.2.1 Data Generative Model

In the pretraining phase, we assume the next-token predictor is pretrained on diverse tasks, each representing a continuous joint distribution of  $(\mathbf{x}, y)$ . Before we move on to the exact pretraining data generative model proposed in this paper, we first provide a general setting for the data generation process. A task is defined by a joint distribution  $\mathcal{D}_{\mathbf{x}, y}$ , which specifies the likelihood of obtaining a sample  $(\mathbf{x}, y)$  from this task. Each task is sampled from the task prior  $\mathcal{D}^{\text{prior}}$ , meaning  $\mathcal{D}^{\text{prior}}$  represents a distribution over distributions. The pretraining data comprises numerous sequences, each containing  $K$  labeled samples i.i.d. drawn from a distribution  $\mathcal{D}_{\mathbf{x}, y}$ . We formally describe our pretraining data generative model in Assumption 1.

---

<sup>1</sup>It is more rigorous to represent the vector  $\mathbf{x}$  as multiple tokens. However, viewing it as a high-dimensional “token” simplifies our notation while not affecting our analysis. Thus, with a slight abuse of notation, we will treat both  $\mathbf{x}_i$  and  $y_i$  as tokens for simplicity.

**Assumption 1** (Pretraining Data Generative Model). *Given an integer  $K > 0$ , a pretraining task prior  $\mathcal{D}^{\text{prior}}$ , we generate a sequence  $S_K$  as follows:*

- (a) *Sample a task from the task prior:  $\mathcal{D}_{x,y} \sim \mathcal{D}^{\text{prior}}$ ;*
- (b) *Sample  $K$  labeled samples from the chosen task:  $\forall i \in \{1, 2, \dots, K\}, (\mathbf{x}_i, y_i) \sim \mathcal{D}_{x,y}$ ;*
- (c) *Define a sequence  $S_K$ :  $S_K = [\mathbf{x}_1, y_1, \dots, \mathbf{x}_K, y_K]$ .*

In the sequence, the first  $2k$  elements of  $S_K$  is denoted as  $S_k$ , and the first  $2k + 1$  elements will be indicated by  $S_k \oplus x^{(k+1)}$ , e.g.,  $S_0 = [ ]$ , and  $S_1 \oplus \mathbf{x}_2 = [\mathbf{x}_1, y_1, \mathbf{x}_2]$ .

### 2.2.2 Bayes-Optimal Next-Token Predictor

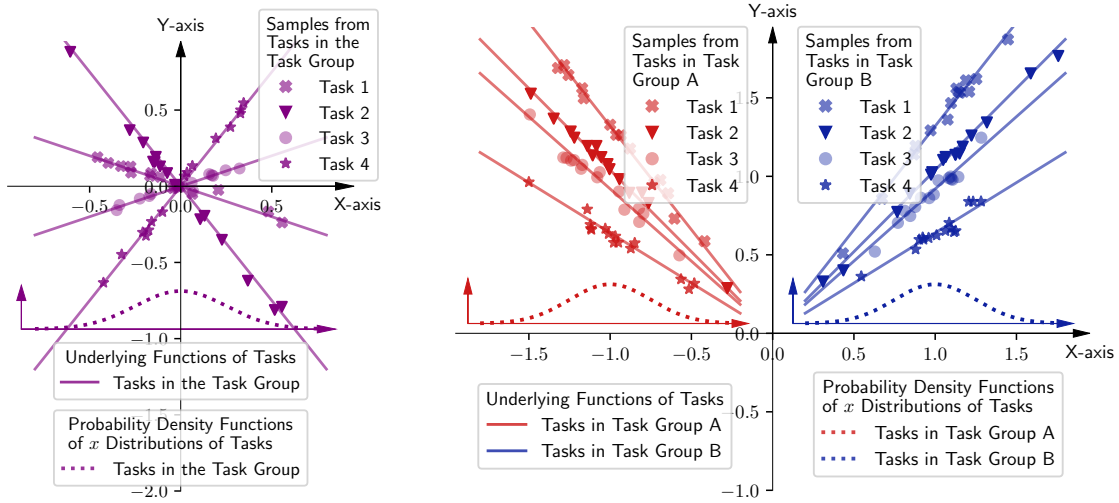
Let  $\mathcal{L}(\mathcal{F}) = \mathbb{E}_{S_K} \left[ \frac{1}{K} \sum_{k=0}^{K-1} (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1})^2 \right]$  as the pretraining objective, where  $\mathcal{F}$  is a next-token predictor and  $S_K$  is generated from  $\mathcal{D}^{\text{prior}}$  following Assumption 1. In other words, for each sequence, we pretrain  $\mathcal{F}$  to predict each label  $y$  based on preceding samples, measuring risk with the squared loss. Due to the linearity of expectation, we have:  $\mathcal{L}(\mathcal{F}) = \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}_{S_K} \left[ (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1})^2 \right]$ . A variable-input-length next-token predictor  $\mathcal{F}$  can be viewed as  $K$  fixed-input-length next-token predictors  $\mathcal{F}_0, \dots, \mathcal{F}_{K-1}$ , where  $\mathcal{F}_k$  takes a sequence of exactly  $2k + 1$  tokens as input. Thus, assuming the sufficient expressiveness of  $\mathcal{F}$ , the optimization problem  $\mathcal{F}^* = \operatorname{argmin}_{\mathcal{F}} \mathcal{L}(\mathcal{F})$  can be decomposed into  $K$  separate optimization problems for  $k \in \{0, \dots, K - 1\}$ :

$$\mathcal{F}_k^* = \operatorname{argmin}_{\mathcal{F}_k} \mathbb{E}_{S_K} [(\mathcal{F}_k(S_k \oplus x^{(k+1)}) - y_{k+1})^2].$$

The solution denoted  $\mathcal{F}_k^*$  is an MMSE estimator [110, page 63] for each  $k$ . Thus, the prediction  $\mathcal{F}^*(S_k \oplus x^{(k+1)}) = \mathcal{F}_k^*(S_k \oplus x^{(k+1)})$  satisfies:

$$\begin{aligned} \mathcal{F}^*(S_k \oplus x^{(k+1)}) &= \mathbb{E}_{S_K} [y_{k+1} | S_k \oplus x^{(k+1)}] \\ &= \mathbb{E}_{\mathcal{D}_{x,y}} \left[ \mathbb{E}_{y_{k+1}} [y_{k+1} | \mathcal{D}_{x,y}, S_k \oplus x^{(k+1)}] \middle| S_k \oplus x^{(k+1)} \right] \\ &= \mathbb{E}_{\mathcal{D}_{x,y}} \left[ \mathbb{E}_{y_{k+1}} [y_{k+1} | \mathcal{D}_{x,y}, \mathbf{x}_{k+1}] \middle| S_k \oplus x^{(k+1)} \right]. \end{aligned} \quad (2.1)$$

Thus,  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$  is the expectation (over task posterior) of  $\mathbb{E}_{y_{k+1}} [y_{k+1} | \mathcal{D}_{x,y}, \mathbf{x}_{k+1}]$  regarding  $S_k \oplus x^{(k+1)}$  as observation. We show that a pretrained Transformer can empirically approximate Bayesian inference in Appendix A.4.



(a) Pretraining data [90]. (b) Our pretraining data with two task groups.

Figure 2.2: **Different pretraining data models.** Raventos et al. and ours.

## 2.2.3 Gaussian/Linear Assumptions on Pretraining Data

### Generative Model

Let us now elaborate further assumptions on  $\mathcal{D}^{\text{prior}}$  and  $\mathcal{D}_{x,y}$  in the Assumption 1 for a tractable posterior, extending beyond the scope of Raventos et al. [90], who propose the data generative model that each task is a noisy linear regression task, the function  $\mathbf{w}$  for each task is drawn from the same Gaussian distribution, and different tasks share the same  $\mathbf{x}$  distribution. In contrast, our model posits that task functions are derived from a Gaussian mixture distribution, and tasks employ varying  $\mathbf{x}$  distributions, as illustrated in Fig. 2.2. We formally formulate this setting in Assumption 2.

**Assumption 2** (Gaussian/Linear Assumptions for Pretraining Data Generative Model).

(a)  $(\boldsymbol{\mu}, \mathbf{w}) \sim \mathcal{D}^{\text{prior}} : P(\boldsymbol{\mu}, \mathbf{w}) = \sum_{m=1}^M \pi_m P(\boldsymbol{\mu}, \mathbf{w} | T_m)$ , where  $T_m$  is the  $m^{\text{th}}$  mixture component<sup>2</sup> of the Gaussian mixture, i.e.,  $P(\boldsymbol{\mu}, \mathbf{w} | T_m) = \mathcal{N}(\boldsymbol{\mu} | \boldsymbol{\mu}_m, \sigma_\mu^2 \mathbf{I}) \cdot \mathcal{N}(\mathbf{w} | \mathbf{w}_m, \sigma_w^2 \mathbf{I})$ , and  $\pi_m$  is the mixture weight.  $\sum_{m=1}^M \pi_m = 1$ ,  $0 < \pi_m < 1$ ,  $(\boldsymbol{\mu}_m, \mathbf{w}_m)$  is the center of the mixture component  $T_m$ , and all components share the same covariance matrix controlled by  $\sigma_\mu$  and  $\sigma_w$ ;

(b) input:  $\mathbf{x} \sim \mathcal{D}_x(\boldsymbol{\mu})$ ,  $P(\mathbf{x} | \boldsymbol{\mu}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \sigma_x^2 \mathbf{I})$ ;

(c) label:  $y | \mathbf{x} \sim \mathcal{D}_{y|x}(\mathbf{w}) : P(y | \mathbf{x}, \mathbf{w}) = \mathcal{N}(y | \mathbf{w}^\top \mathbf{x}, \sigma_y^2)$ ;

(d)  $\|\boldsymbol{\mu}_m\| = \|\mathbf{w}_m\| = 1, \forall m \in [M]$ ;

(e)  $\exists r > 1$  that  $\forall \alpha, \beta \in [M], \frac{1}{r} \leq \frac{\pi_\alpha}{\pi_\beta} \leq r$ ;

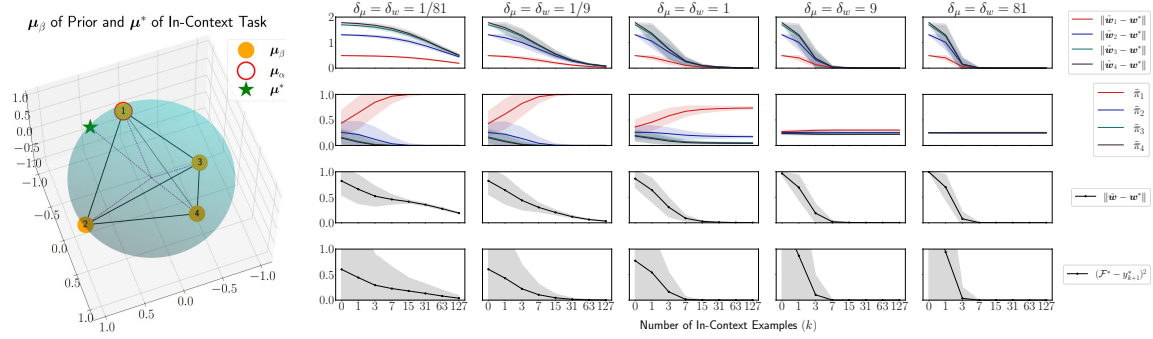
---

<sup>2</sup>The concept ‘‘mixture component’’ is derived from Gaussian mixture models in the statistical literature and is analogous to the term ‘‘Task Group’’ depicted in the left-most panel of Fig. 2.1.

(f)  $\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\mu}_m, \mathbf{w}, \mathbf{w}_m \in \mathbb{R}^d, \mathbf{I} \in \mathbb{R}^{d \times d}$ .

**Remark 1.** Based on Assumptions 2(b) and 2(c), we define the probability of observing a sample  $(\mathbf{x}, y)$  within a task  $(\boldsymbol{\mu}, \mathbf{w})$  as the “noisy linear regression” likelihood.

Assumption 2(a) indicates that the pretraining dataset of an LLM consists of  $M$  different task groups. Assumption 2(b) posits that tasks have varying  $\mathbf{x}$  distribution with varying mean but share the same covariance matrix. Assumption 2(c) assumes tasks as noisy linear regressions with the same noise scale in labels. Assumption 2(e) posits comparable mixture weights  $\pi$  across different task groups.



(a) **The Tetrahedron setting.** An illustration of the in-context task and the prior centers.  $\forall m \in \{1, 2, 3, 4\}$ , we set  $\boldsymbol{\mu}_m = \mathbf{w}_m$ .

(b) **CR, CS, and risks under the Tetrahedron setting.** In the first two rows, we show the effects of CS and CR with an increasing number of in-context examples. In the third row, we show how far the in-context predicted function  $\tilde{\mathbf{w}}$  is from the target function  $\mathbf{w}^*$ . In the fourth row, we show the ICL risk.

Figure 2.3: **Numerical experiments.** (Left) An illustration of the pretraining priors. (Right) The numerical computational results.

## 2.3 Inference and Dual Operating Modes

The previous Sec. 2.2.2 shows that performing ICL with the optimally pretrained next-token predictor is equivalent to computing the posterior mean of the label. In Sec. 2.3.1, we give the generation process of in-context examples. In Sec. 2.3.2, under Assumption 2 and treating  $S_k \oplus x^{(k+1)}$  as observation, we derive a closed-form expression for the task posterior  $\mathcal{D}^{\text{post}}$ , and identify two factors in the transition from prior to posterior: Component Shifting and Component Re-weighting. In Sec. 2.3.3, we derive a closed-form expression of the ICL prediction  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$ . Further, Sec. 2.3.4 presents the results of numerical computation conducted under the tetrahedron setting, as illustrated in Fig. 2.3a. The numerical computation results demonstrate the effects of component shifting and re-weighting. Finally, Sec. 2.3.5 raises the definitions of the dual operating modes.

### 2.3.1 In-Context Task and In-Context Function

We introduce Assumption 3 for the in-context task and the in-context function of in-context examples:

**Assumption 3** (Gaussian/Linear Assumptions for In-Context Examples).

- (a) The input sequence  $S_k \oplus x^{(k+1)}$  of ICL satisfies,  $\forall i, \mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ ,  $y_i = \langle \mathbf{x}_i, \mathbf{w}^* \rangle$ ;
- (b)  $\|\boldsymbol{\mu}^*\| = \|\mathbf{w}^*\| = 1$ .

Assumption 3(a) states that each in-context example  $(\mathbf{x}_i, y_i)$  is drawn from the in-context task  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ , with  $\mathbf{w}^*$  representing the specific in-context function and the labels being free from noise.

### 2.3.2 Closed-Form Expression of Posterior

The following lemma gives the closed-form expression of the posterior  $\mathcal{D}^{\text{post}}$  given any observation  $S_k \oplus x^{(k+1)}$ :

**Lemma 1** (Conjugate Distributions with Noisy Linear Regression Likelihood).

*Under Assumption 2, the posterior probability of task  $(\boldsymbol{\mu}, \mathbf{w})$  given  $S_k \oplus x^{(k+1)}$  is:*

$$\begin{aligned} P(\boldsymbol{\mu}, \mathbf{w} | S_k \oplus x^{(k+1)}) &= \sum_{m=1}^M \tilde{\pi}_m P(\boldsymbol{\mu}, \mathbf{w} | \tilde{T}_m) \\ &= \sum_{m=1}^M \tilde{\pi}_m \cdot \mathcal{N}(\boldsymbol{\mu} | \tilde{\boldsymbol{\mu}}_m, \tilde{\sigma}_\mu^2 \mathbf{I}) \cdot \mathcal{N}(\mathbf{w} | \tilde{\mathbf{w}}_m, \tilde{\sigma}_w^2 \mathbf{I}). \end{aligned}$$

Here, the mixture component  $T_m$  in the prior is mapped to the mixture component  $\tilde{T}_m$  in the posterior with mixture weight  $\tilde{\pi}_m$  and component means  $(\tilde{\boldsymbol{\mu}}_m, \tilde{\mathbf{w}}_m)$ :

$$\begin{aligned} \tilde{\pi}_m &= \pi_m C_1 c_m^\mu c_m^w, \\ c_m^\mu &= \exp \left( -\|\boldsymbol{\mu}_m\|^2 - \|\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}\|_{(\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2 / 2\sigma_\mu^2 \right), \\ c_m^w &= \exp \left( -\|\mathbf{w}_m\|^2 - \|\mathbf{w}_m + k\delta_w \bar{\mathbf{w}}\|_{(\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 / 2\sigma_w^2 \right), \\ \tilde{\boldsymbol{\mu}}_m &= (\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1} (\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}), \\ \tilde{\mathbf{w}}_m &= (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1} (\mathbf{w}_m + k\delta_w \bar{\mathbf{w}}), \\ \tilde{\sigma}_\mu^2 &= \sigma_\mu^2 (\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}, \\ \tilde{\sigma}_w^2 &= \sigma_w^2 (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}, \end{aligned}$$

where  $C_1$  is a normalizing constant, i.e.,  $\sum_m \tilde{\pi}_m = 1$ ,  $\delta_\mu = \frac{\sigma_\mu^2}{\sigma_x^2}$ ,  $\delta_w = \frac{\sigma_w^2}{\sigma_y^2}$ ,  $\bar{\boldsymbol{\Sigma}}_\mu = \mathbf{I}$ ,  $\bar{\boldsymbol{\mu}} = \frac{\sum_{i=1}^{k+1} \mathbf{x}_i}{k+1}$ ,  $\bar{\boldsymbol{\Sigma}}_w = \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k}$ , and  $\bar{\mathbf{w}} = \frac{\sum_{i=1}^k \mathbf{x}_i y_i}{k}$ . See Appendix A.7 for the proof.

**Remark 2.** Gaussian mixture is known to be a conjugate prior to the Gaussian likelihood. The outlined conjugate distributions in this lemma extend the Gaussian mixture conjugate

distributions by substituting the Gaussian likelihood with the “noisy linear regression” likelihood in Remark 1.

Lemma 1 states that the task posterior remains a Gaussian mixture, with its mixture components shifted and re-weighted from the task prior. Therefore, understanding the impact of in-context examples on the posterior requires understanding how in-context examples affect the two factors:

- **Component Shifting (CS):** The component center is shifted from  $(\mu_m, w_m)$  to  $(\tilde{\mu}_m, \tilde{w}_m)$ .
- **Component Re-weighting (CR):** The component weight is re-weighted from  $\pi$  to  $\tilde{\pi}$ .

**Remark 3.** The term “component” comes from the literature on Gaussian mixtures. It serves as an alternative to “Task Group” as shown in Fig. 2.2. The terminology “Component Shifting” and “Component Re-weighting” can be viewed as “Task Group Shifting” and “Task Group Re-weighting.” We will abbreviate “mixture component center” to simply “center” when there is no ambiguity.

Leveraging Assumption 3, we collected mathematical analyses of CS and CR in Appendix A.8. The analysis explores the impacts of pretraining task noises and the number of in-context examples on  $\tilde{\mu}_m$ ,  $\tilde{w}_m$ , and  $\tilde{\pi}_m$ , and examines the convergence of  $\tilde{\mu}_m$ ,  $\tilde{w}_m$ , and  $\tilde{\pi}_m$ , as  $k$  approaches infinity.

### 2.3.3 Closed-Form Expression of ICL Prediction

With Assumption 2 and Lemma 1, we have the following corollary for the prediction  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$ :

**Corollary 2.** *Let  $\tilde{\mathbf{w}} = \sum_{m=1}^M \tilde{\pi}_m \tilde{\mathbf{w}}_m$ . With pretraining data generative model 1 and Assumption 2, if the pretrained model  $\mathcal{F}^*$  minimizes the pretraining risk, then the prediction on any sequence  $S_k \oplus x^{(k+1)}$  by  $\mathcal{F}^*$  is as follows:*

$$\mathcal{F}^*(S_k \oplus x^{(k+1)}) = \left\langle \mathbf{x}_{k+1}, \sum_{m=1}^M \tilde{\pi}_m \tilde{\mathbf{w}}_m \right\rangle = \langle \mathbf{x}_{k+1}, \tilde{\mathbf{w}} \rangle.$$

*Proof.* Apply Assumption 1 to Eq. 2.1,  $\mathcal{F}^*(S_k \oplus x^{(k+1)}) = \mathbb{E}_{(\boldsymbol{\mu}, \mathbf{w}) \sim \mathcal{D}^{\text{prior}}} [\langle \mathbf{x}_{k+1}, \mathbf{w} \rangle | S_k \oplus x^{(k+1)}]$ . Using Lemma 1, this reduces to  $\sum_{m=1}^M \tilde{\pi}_m \mathbb{E}_{(\boldsymbol{\mu}, \mathbf{w}) \sim \tilde{T}_m} [\langle \mathbf{x}_{k+1}, \mathbf{w} \rangle]$ . Due to the linearity of expectation and inner product, the prediction can be simplified as  $\langle \mathbf{x}_{k+1}, \sum_{m=1}^M \tilde{\pi}_m \tilde{\mathbf{w}}_m \rangle = \langle \mathbf{x}_{k+1}, \tilde{\mathbf{w}} \rangle$ .  $\square$

Thus, the prediction is a convex combination of predictions by the centers of those shifted and re-weighted mixture components in the posterior. We are interested in how  $\pi_m$  and  $\mathbf{w}_m$  change to  $\tilde{\pi}_m$  and  $\tilde{\mathbf{w}}_m$  with increasing  $k$  and how the pretraining prior distribution properties affect these changes.

### 2.3.4 Prior Task Noises, CS, CR, and ICL Prediction

We numerically compute how  $\tilde{\pi}_m$ ,  $\tilde{\mathbf{w}}_m$ , and the prediction  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$  evolve as  $k$  increases under different prior task noise conditions. The numerical computation is based on the tetrahedron setting with four prior mixture components as illustrated in Fig. 2.3a. See Appendix A.2.1 for details of the tetrahedron setting. Fig. 2.3b

shows the computational results. The first row shows the CS effect, demonstrating the impact of increasing  $k$  on  $\tilde{w}_m$ . The second row shows the CR effect, illustrating the impact of increasing  $k$  on  $\tilde{\pi}_m$ . The third and fourth rows depict how increasing  $k$  influences the risk of learning the function  $w^*$ . We observe that with low task noises and a small  $k$  value, the CR effect initially prevails, significantly boosting the mixture weight of component 1 over others. Then, as  $k$  increases further, the CS effect aligns all component centers with  $(\mu^*, w^*)$ .

### 2.3.5 Dual Operating Modes

The “*task retrieval*” mode describes a scenario where the impact of component re-weighting surpasses that of component shifting, leading to the prediction that is primarily influenced by the interplay between pretraining priors and in-context examples. An illustration of this is shown in the first column of Fig. 2.3b, where the re-weighting of  $\tilde{\pi}_m$  is more pronounced than the shifting of  $\tilde{w}_m$ , indicating that CR plays a pivotal role in altering the prediction. In contrast, the “*task learning*” mode refers to situations where component shifting dominates over component re-weighting, resulting in the prediction almost depending on in-context examples and neglecting the pretraining priors.

## 2.4 Early Ascent

We now explain the early ascent phenomenon by analyzing a finegrained risk bound of ICL. (See Appendix A.3 Theorem 7 for the coarser bound.)

### 2.4.1 Finegrained Upper Bound

The finegrained upper bound for ICL risk is shown below:

**Theorem 3** (Finegrained Upper Bound for ICL Risk). *Consider a next-token predictor attaining the optimal pretraining risk. As  $k \rightarrow \infty$ , ICL risk is upper bounded by:*

$$\mathbb{E}[\mathcal{L}_k^*] < \sum_{m=1}^M \|\mathbf{w}_m - \mathbf{w}^*\|^2 \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_m \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2],$$

where  $\mathcal{L}_k^* = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^*)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle)^2$ ,  $\|\mathbf{w}_m - \mathbf{w}^*\|$  is the distance between the in-context function  $\mathbf{w}^*$  and the function  $\mathbf{w}_m$  of center  $m$ ,  $\tilde{\pi}_m$  is the posterior mixture weight, and  $\mathbf{A} = (\mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top)^{-1}$ . See Appendix A.12 and Eq. A.14 for proof details. In Appendix A.12.1, we further refine the bound for cases when in-context  $\mathbf{x}_i$  only spans in a subspace of  $\mathbb{R}^d$ , resulting in  $\lambda_1(\mathbf{A}) = 1$  constantly.

In-context examples affect the upper bound by affecting the two factors  $\tilde{\pi}_\beta$  and  $\lambda_1(\mathbf{A})$ , corresponding to CR and CS introduced in Sec. 2.3.2. When ignoring the CR effect and only considering CS, the finegrained upper bound degrades to the general coarse bound in Appendix A.3 Theorem 7.

### 2.4.2 The Effect of Dual Operating Modes on ICL Risk

We numerically compute ICL risk under varied settings to explore the effect of the dual operating modes on the risk in Fig. 2.4. When pretraining task noises are low, *i.e.*,  $\delta_\mu$  and  $\delta_w$  are small, the task retrieval mode happens with a small number of in-context examples, and the upper bound is affected by how  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$  is close to a prior center. Specifically, the task prior boosts the learning process of ICL if

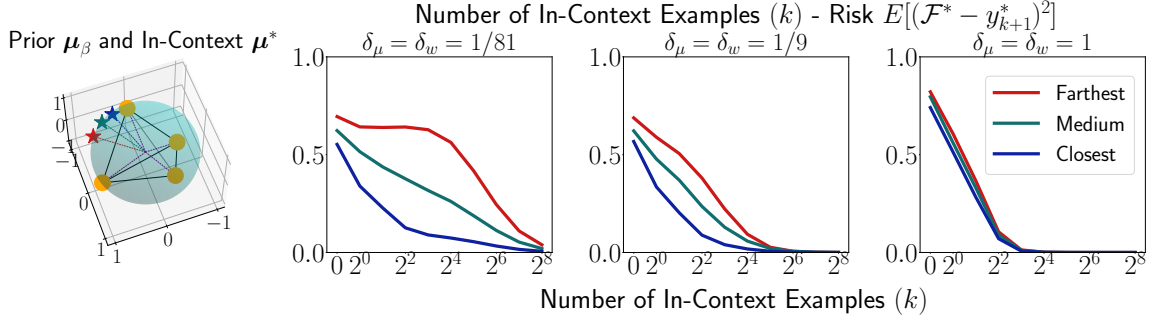
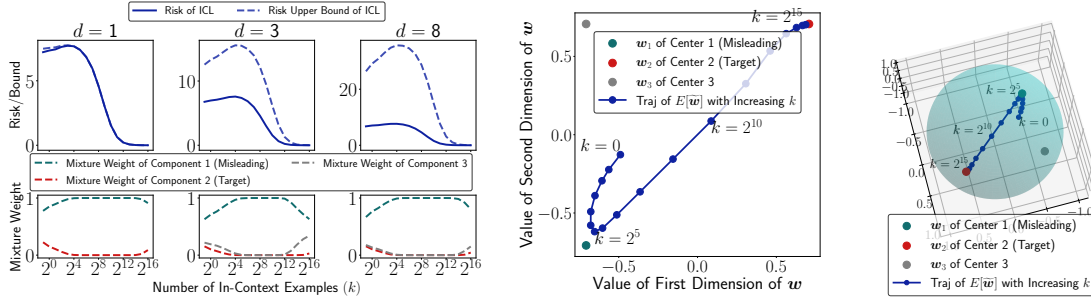


Figure 2.4: **Distance to the closest prior vs ICL risk.** We compute ICL risks of three target tasks colored red (farthest), green, and blue (closest), under the tetrahedron setting, illustrated in the left-most figure. The red target task has the longest distance to the closest prior center, and the blue target task has the shortest distance to the closest prior center. We can observe that the target task is easier to learn when the distance to the closest prior is smaller.

the in-context task is close to a prior center, due to the task retrieval mode quickly retrieving the task of the nearest prior center.

### 2.4.3 Early Ascent with Biased $x$ Distribution

However, the task retrieval mode may not always benefit ICL. We notice that a weird phenomenon is observed by Brown et al. [11] and Xie et al. [118]. As the number of in-context samples increased, the performance of ICL first decreased and then increased. Brown et al. [11] reports that GPT-3 on LAMBADA shows a lower one-shot accuracy (72.5%) than zero-shot accuracy (76.2%), but the few-shot accuracy (86.4%) is higher than the zero-shot accuracy. Xie et al. [118] also replicated this phenomenon with their synthetic dataset. Xie et al. [118] explains this by “the few-shot setting introduces the distracting prompt structure, which can initially lower accuracy.”



(a) Risk and  $\pi_m$  as  $k$  increases under  $d \in \{1, 3, 8\}$ . (b) Expectation of  $\tilde{w}$  as  $k$  increases under  $d \in \{2, 3\}$ .

**Figure 2.5: The early ascent phenomenon.** Fig. 2.5a and Fig. 2.5b show that the task retrieval mode is dominant up to  $k = 32$ , and component 1’s mixture weight increases ( $E[\tilde{w}]$  approaches  $w_1$ ). Since this component is farther than the other one, the risk starts increasing. At larger  $k$  values, the risk starts decreasing ( $E[\tilde{w}]$  approaches  $w_2$ ) via task learning. See Appendix A.2.3 for setting details. We further examine the early ascent phenomenon under linear regression with varied levels of label noises in Appendix A.9.1, and under non-linear regression and discrete token prediction in Appendix A.9.2.

To obtain some insights, we present a simple scenario where  $x$  misleads the prediction by an LLM. Consider the following one-shot prompt for English-to-Korean translation: “What is the color of apple? 사과의 색깔은 무엇인가?<sup>3</sup> What is the color of banana?” The correct answer should be “바나나의 색깔은 무엇인가?”<sup>4</sup> However, GPT-3.5 generates “바나나의 색깔은 노란색 입니다,” which means “The color of bananas is yellow.” This shows that pretrained LLMs could retrieve an incorrect skill (question answering in this example) by observing misleading input ( $x$ ).

Based on our analysis, we further show that the early ascent phenomenon provably occurs under a certain assumption Appendix A.10.1. We also reproduce

<sup>3</sup>“What is the color of apple?” in Korean.

<sup>4</sup>“What is the color of banana?” in Korean.

early ascent in Fig. 2.5a, where the upper bound and the risk initially increase due to the misleading task (of center 1) is retrieved first. Fig. 2.5b further demonstrates the relative locations of the retrieved functions to functions of prior centers. Finally, we give the formal theorem on the early ascent phenomenon:

**Theorem 4** (Early Ascent). *Assume  $\alpha = \arg \min_m \frac{\|\mu_m - \mu^*\|^2}{2\sigma_x^2} + \frac{\|(\mathbf{w}_m - \mathbf{w}^*)^\top \mu^*\|^2 + d\tau_x^2 \|\mathbf{w}_m - \mathbf{w}^*\|^2}{2\sigma_y^2}$  is the most misleading task and the task  $\alpha$  satisfies:*

$$\mathbb{E}_{\mathbf{x}_1} \left[ (\mathcal{F}^*(\mathbf{x}_1) - \langle \mathbf{w}^*, \mathbf{x}_1 \rangle)^2 \right] < \mathbb{E}_{\mathbf{x}_1} \left[ \langle \mathbf{x}_1, \mathbf{w}_\alpha - \mathbf{w}^* \rangle^2 \right].$$

Then, when  $\delta_\mu$  and  $\delta_w$  are small enough,  $\exists k \geq 1$  s.t.:

$$\mathbb{E}_{\mathbf{x}_1} \left[ (\mathcal{F}^*(\mathbf{x}_1) - \langle \mathbf{w}^*, \mathbf{x}_1 \rangle)^2 \right] < \mathbb{E}_{S_k \oplus \mathbf{x}^{(k+1)}} \left[ \left( \mathcal{F}^*(S_k \oplus \mathbf{x}^{(k+1)}) - \langle \mathbf{w}^*, \mathbf{x}_{k+1} \rangle \right)^2 \right],$$

where  $\mathbb{E}_{\mathbf{x}_1} [\langle \mathbf{x}_1, \mathbf{w}_\alpha - \mathbf{w}^* \rangle^2]$  equals to the risk when the prediction fully depends on the misleading task function  $\mathbf{w}_\alpha$  of prior center  $\alpha$ . See Appendix A.10.2 for proof details.

Theorem 4 shows that, if the misleading task  $\alpha$  has a higher risk than the zero-shot risk, then when  $\delta_\mu$  and  $\delta_w$  are small enough, early ascent happens.

## 2.5 Bounded Efficacy of Biased-Label ICL

We further predict the bounded efficacy phenomenon by examining the bound of ICL with biased labels. The assumption for biased-label ICL is described as follows:

**Assumption 4** (ICL with Biased Labels). *The function  $\mathbf{w}^*$  of ICL with biased labels is different from the target function  $\mathbf{w}_\alpha$ , i.e.,  $\mathbf{w}^* \neq \mathbf{w}_\alpha$  where  $\mathbf{w}_\alpha$  is a function of a pretraining task prior center. The in-context task is closer to the prior center  $\alpha$  compared to all the other*

prior centers  $\beta \neq \alpha$ :

$$\forall \beta \neq \alpha, \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 \geq d_\mu^2, \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \geq d_w^2, \text{ and} \\ \tau_x^2 \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - (1 + \tau_x^2) \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \geq \tau_x^2 u_w^2.$$

Assumption 4 depicts that to retrieve  $\mathbf{w}_\alpha$  associated with the prior center  $\alpha$ , the in-context task is selected based on its proximity to center  $\alpha$ , ensuring it is closer to center  $\alpha$ .

### 2.5.1 Upper Bound for ICL Risk with Biased Labels

The following theorem shows an upper bound for ICL risk with biased labels to retrieve a task:

**Theorem 5** (Upper Bound for ICL Risk with Biased Labels). *Consider a next-token predictor attaining the optimal pretraining risk. As  $k \rightarrow \infty$ , ICL risk with biased labels is upper bounded by:*

$$\mathbb{E}_{S_k}[\mathcal{L}_k^\alpha] < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2(1 + d\tau_x^2) + \frac{C_1}{k\delta_w} \exp\left(C_2 k^{\frac{\delta}{2} - \frac{3}{4}}\right) + O(k^{-2}),$$

where  $\mathcal{L}_k^\alpha = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^\alpha)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ . When  $\delta_\mu$  and  $\delta_w$  are sufficiently small, exists a particular interval for  $k$  s.t.:

$$\mathbb{E}_{S_k}[\mathcal{L}_k^\alpha] < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2(1 + d\tau_x^2) \min\{1, 4k^2\delta_w^2(1 + \tau_x^2)^2\} \\ + C_3 \exp\left(-k\left(\frac{d_\mu^2}{8\sigma_x^2} + \frac{u_w^2\tau_x^2}{8\sigma_y^2}\right)\right) + C_4 \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right).$$

As  $k$  increases, the second and third terms dominate and exponential decay when  $k$  is small, and the first term dominates and increases when  $k$  is large.  $C_1, C_2, C_3$ , and  $C_4$  are constants depending on the prior setting,  $\tau_x$ , and  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ . See Appendix A.13 for proof details.

| $k$      | 0      | 1     | 2            | 4            | 8            | 16           |
|----------|--------|-------|--------------|--------------|--------------|--------------|
| +        | 75.0%  | 36.2% | <b>33.9%</b> | <u>49.3%</u> | <u>79.3%</u> | <u>85.1%</u> |
| Biased + | 100.0% | 98.3% | 95.9%        | 60.5%        | 24.4%        | <b>16.8%</b> |

Table 2.1: **Bounded efficacy in GPT-4.** Error rate measured with respect to “addition (+)” and “biased +”. The bounded efficacy phenomenon: the error rate goes down to  $k = 2$ , but it increases afterward. Experiment details in Appendix A.5.1.

### 2.5.2 Bounded Efficacy of Biased-Label ICL in GPT-4

This section further shows that the bounded efficacy phenomenon exists in GPT-4 in Table 2.1. With the task “biased addition (+)” as the in-context task corresponding to  $w^*$ , as the number of in-context examples increases, ICL will first retrieve the skill “addition (+)” corresponding to  $w_\alpha$  which has a strong pretraining prior. Later, it will learn the “biased +” task, leading to the bounded efficacy phenomenon.

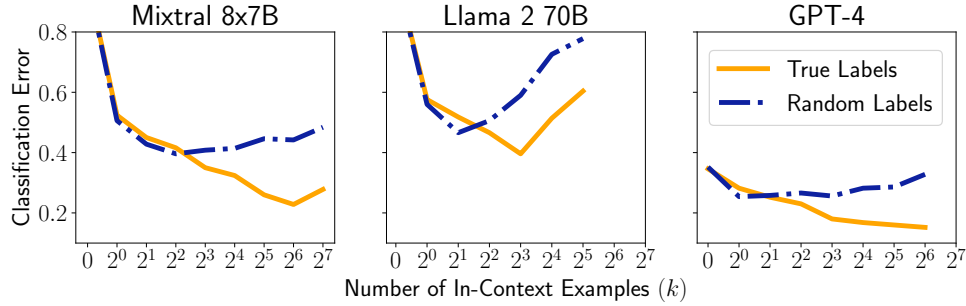


Figure 2.6: **Bounded efficacy.** The error rates of ICL with random labels start increasing at large  $k$ . See Appendix A.6 for more experimental results.

### 2.5.3 Bounded Efficacy for Zero-Shot ICL

We further introduce Lemma 6, a variation of the previous Theorem 5, to explain zero-shot ICL, an ICL algorithm capable of functioning with random labels [68].

**Lemma 6** ((informal) Upper Bound for Zero-Shot ICL). *Assume a next-token predictor attains the optimal pretraining risk, the risk of ICL with random labels (provide no information) will reveal a bounded efficacy phenomenon. See Appendix A.14 for proof.*

Lemma 6 says that as the number of in-context examples increases, the loss curve of zero-shot ICL with random labels will have the bounded efficacy phenomenon, which conflicts with the observation from Min et al. [73] that ICL with random labels has very similar performance as ICL with true labels for the number of in-context examples ranging from 1 to 32. We believe this observation is due to the small number of in-context examples. Thus, we extend the experiment of Min et al. [73] to explore the number of in-context examples beyond 32. Due to LLMs' context lengths constraining the maximum number of in-context examples, we choose different LLMs from Min et al. [73] for a larger context length capacity.

Fig. 2.6 highlights the bounded efficacy phenomenon in the error curve associated with random labels. Compared with true labels, the error rate of ICL with random labels increases at a much smaller  $k$  value, clearly exhibiting the bounded efficacy phenomenon we predicted.

## 2.6 Extension with Preliminary Results: Coded

### Prompts for Large Language Models

As shown in the previous sections, task retrieval can improve the prediction accuracy of ICL even without informative labels. This naturally raises the question: What if we provide multiple testing samples during inference? These test samples could

serve as unlabeled in-context examples for each other, potentially enhancing the prediction accuracy of LLMs. Motivated by this idea, we explore the following extension and present preliminary results.

### 2.6.1 Introduction

LLMs [11, 80] have become a cornerstone of generative AI research, demonstrating remarkable capabilities in various natural language processing tasks. An essential technique to improve the performance of LLMs is prompt engineering. Numerous heuristic strategies [49, 65, 95, 129, 74, 115, 116, 59, 56, 64, 59] have been developed to design better prompts for LLMs. Despite their impressive performance, there is a significant scope for further enhancement, innovation, and optimization.

In response to this opportunity, we propose a novel dimension to prompt design – **coded prompts** for pooled LLM inference (this approach involves inferring an LLM with multiple samples). This innovative approach is inspired by the principles of coding theory [94], a field that focuses on designing coded symbols as functions of multiple information symbols rather than one symbol for reliable communication and storage systems. Drawing on this concept, we design coded prompts for processing multiple inputs simultaneously, thereby enabling pooled inference within the context of LLMs.

In this paper, we review the coding and its potential to improve prompt design in LLMs. We introduce a new framework for coded prompts and provide formal definitions. This framework is the basis for our investigation into the coded prompts’ ability to boost LLM performance. We test this concept with experiments on two

tasks: a classification task of identifying the largest prime number in a range and a regression task of predicting text toxicity. Initial results show that coded prompts can significantly improve task performance, highlighting this approach’s potential.

In summary, our contributions are as follows:

- We introduce the concept of coded prompts, a novel approach to prompt design, inspired by the principles of coding theory. This approach allows for the simultaneous processing of multiple inputs, potentially enhancing the efficiency and performance of LLMs.
- We provide a framework and formal definitions for coded prompts.
- We empirically validate our approach through experiments on two tasks including a classification task of identifying the largest prime number in a range and a regression task of predicting text toxicity. We demonstrate that coded prompts can significantly improve task performance, highlighting the potential of our approach.

### 2.6.2 Related Work

**Prompt Engineering** Prompt engineering has been studied extensively over a long period. Researchers have explored topics including how to ensemble multiple prompts [49, 97, 98, 87, 86, 53], automatically generate good prompts [100, 59, 66, 133], or train a better model for instruction [81, 74, 115]. Further, Wei et al. [116] propose Chain-of-Thoughts (CoT) which explores how to generate a chain of thoughts – a series of intermediate reasoning steps – significantly improves

the ability of large language models. CoT is further improved by varied directions such as ensembling [114, 113, 29], and selecting good steps in multi-step reasoning [108, 123].

**Self-Evaluation for LLMs** Self-evaluation mechanism [69, 85, 101] was introduced that LLMs themselves provide feedback to their own generation candidates. Chen et al. [14] use self-evaluation to improve the accuracy of LLMs to generate code. Xie et al. [119] endow LLMs with self-evaluation to refine multi-step reasoning inference. Yao et al. [123] allow LLMs to perform deliberate decision-making by considering multiple different reasoning paths and self-evaluating choices to decide the next course of action. Zhang et al. [130] employ language models in a cumulative and iterative manner to emulate human thought processes to solve complex problems. Different from these works that predict one sample at each inference, we consider how to leverage multiple inputs together to boost the performance of LLMs.

**Applications of ICL in Real-World LLMs** ICL provides another special angle of prompt design, i.e., leveraging extra samples into the prompt [11] to boost the prediction performance. This method is further explored via improving sample quality, such as calibrating to reduce in-context sample bias [132], choosing better in-context samples [65, 104, 129], training LLMs following in-context instruction [124], or providing samples without true labels [68].

**Coding Theory** Coding theory [19] was adopted in various domains of machine learning. Han et al. [39] applied coding theory to compress neural networks. Dimakis et al. [22] and Rashmi et al. [89] applied coding theory to storage systems. Lee et al. [54] applied coding theory to speed up distributed computing. In this work, we aim to apply coding theory to an LLM which is used as a predictor.

### 2.6.3 Coded Prompts

#### Coding Theory: A Brief Overview

Before introducing our framework for coded prompts, let us briefly overview the key idea of coding theory [94]. Coding theory is concerned with designing efficient and reliable methods for transmitting or storing data. One of the main goals is to develop encoding schemes, that can protect data integrity against errors that might occur during transmission or storage.

To illustrate the key idea, consider the following example concerning the communication of two bits, say  $B_1$  and  $B_2$ . In a naïve approach, one might simply transmit (over a noisy communication channel)  $B_1$  and  $B_2$  as they are. However, this approach is vulnerable to channel errors. If an error occurs during the communication, and if the value of  $B_1$  or  $B_2$  is lost, it will be impossible to recover the lost data. Furthermore, if the values of  $B_1$  or  $B_2$  have altered while being transmitted, it will be impossible even to realize if there was any error.

To protect against this, we can use a simple coding scheme. Instead of just transmitting the original bits  $B_1$  and  $B_2$ , we also transmit the XOR of  $B_1$  and  $B_2$ , denoted as  $B_1 \oplus B_2$ . Here, we call  $B_1 \oplus B_2$  an encoded bit or *coded bit*. Now, even if

one of the two information bits is lost, we can recover it using the remaining one information bit and the encoded bit. For instance, if  $B_1$  is lost, we can recover it by XORing  $B_2$  and  $B_1 \oplus B_2$ , i.e.,  $B_2 \oplus (B_1 \oplus B_2) = B_1$ . Similarly, if  $B_2$  is lost, we can recover it by XORing  $B_1$  and  $B_1 \oplus B_2$ , i.e.,  $B_1 \oplus (B_1 \oplus B_2) = B_2$ .

This simple example illustrates the basic principle of coding theory. In practice, coding theory involves much more complex and sophisticated schemes, but the underlying goal remains the same: to protect data and ensure its integrity during transmission or storage.

### **Analogy Between Noisy Communication and LLM Inference**

To introduce coded prompts, we draw a novel analogy: viewing LLM inference as a noisy communication channel [19]. By drawing inspiration from information and coding theory, we can consider the process of generating predictions from LLMs as analogous to transmitting and receiving information through a noisy channel. With this analogy, the unknown ground truth labels can be considered the “information bits,” while the LLM’s predictions represent the “received bits” after passing through the noisy channel. More specifically, consider a test sample drawn from the data distribution  $(X, Y) \sim D$ . For instance,  $X$  could be a sentence, and  $Y = f(X) \in \{0, 1\}$  could be a binary label denoting if the sentence is toxic (1) or not (0). Here,  $f(\cdot)$  is an unknown deterministic label mapping from  $X$  to  $Y$ . The prediction result of an LLM inference with a particular prompt, say  $p$ , can be modeled as follows:

$$\hat{Y}^p = Y \oplus W_X^p,$$

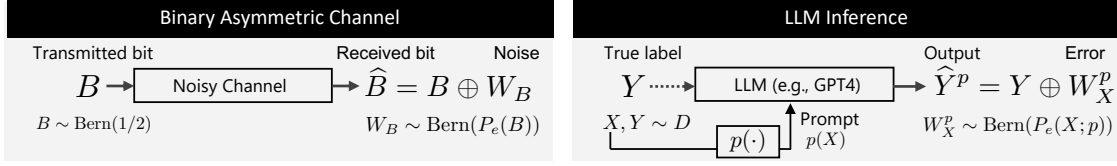


Figure 2.7: **Illustration of the analogy between information bit transmission in a noisy communication channel and LLM inference.** The communication channel transmits bit  $B$  with a probability  $P_e(B)$  of error occurrence, while LLM infers a sample with true label  $X$  and has a probability  $P_e(X; p)$  of making incorrect predictions. The notation  $W$  is for the noise introduced by channel or LLMs.

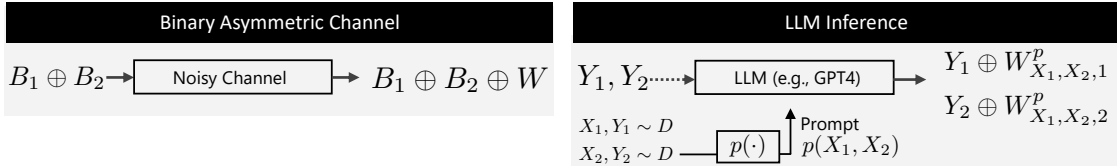


Figure 2.8: **Illustration of the analogy between *encoded* bit transmission in a noisy communication channel and *coded* LLM inference.** The communication channel transmits an encoded bit  $B_1 \oplus B_2$ , while LLM inference predicts multiple samples (two samples in this example) with true label  $X_1, X_2$ . The notation  $W$  is for the noise introduced by the channel or LLMs.

where  $W_X^p$  denotes a binary noise which (1) depends on the input  $X$  and (2) is parameterized by the choice of prompt  $p$ . Note that for a fixed prompt  $p$ , this becomes analogous to the binary asymmetric channel [26, 18, 76, 30], which has been extensively studied in the field of information theory. We illustrate this analogy in Fig. 2.7.

### Coded Prompts

We now present the concept of *coded prompts*, which extends the previously discussed analogy to the transmission of encoded bits. See Figure 2.8 for a visual representation. In the context of communication, as explored in the prior toy exam-

ple, we initially compute the encoded bit ( $B_1 \oplus B_2$ ) and subsequently transmit it over the channel, yielding  $B_1 \oplus B_2 \oplus W$  at the receiver end, where the variable  $W$  represents the channel noise.

How can we implement analogous mechanisms within the framework of LLMs? The equivalent of transmitting a coded bit can be conceptualized as generating a prediction from an LLM using *a coded prompt*. A coded prompt is a specially crafted prompt that accommodates multiple test inputs concurrently, mirroring the coded bit in the communication example.

To illustrate this, let us consider a binary classification task with two inputs,  $X_1$  and  $X_2$ . In a simplistic approach, we could generate predictions from the LLM for each input independently. However, this method is susceptible to noise in the LLM inference process. If the prediction for one input is erroneous due to noise, error detection becomes infeasible.

To safeguard against this, we propose the development of a coding scheme. Instead of merely generating individual predictions, we also generate a prediction using a coded multi-input prompt, which incorporates both  $X_1$  and  $X_2$ . We refer to the prediction derived from this coded prompt as *a coded prediction*. Now, even if one prediction from the two original prompts is inaccurate, we can detect or correct it using the remaining and coded predictions.

One crucial distinction exists here. Given LLMs' capability to generate outputs of arbitrary length, we can produce vector-valued predictions, as depicted in Figure 2.8.

This contrasts with the communication example where only a single bit can be

received when utilizing the communication channel once.

**Remark** While the existing prompt techniques focus on addressing individual test samples, our coded prompting technique processes multiple test samples simultaneously. It is important to note that this is not always feasible – if only one test sample is available, then coding offers no advantage. Indeed, this mirrors the block-length condition necessary for efficient coding – coding techniques are effective when handling a large number of information bits, and their benefits are limited when dealing with one or a small number of information bits [19].

### Formal Definition

For clarity of presentation, we will assume the following simple setting (binary classification) throughout the paper. Our framework can be easily extended to handle more general cases.

For the input feature and label, we define  $(X, Y)$  such that  $(X, Y) \sim D$ , with  $X \in \mathcal{X}$  and  $Y = f(X) \in \mathcal{Y}$ . We denote by LLM the mapping induced by a raw LLM inference followed by the label mapping function (e.g., parser). That is,  $\text{LLM} : \text{text} \rightarrow \cup_{n=1}^{\infty} \mathcal{Y}^n$ . In this context, when using standard prompting, LLM’s output falls within  $\mathcal{Y}$ . However, with a single coded prompt for multiple inputs, LLM can output a sequence of labels, denoted as  $\mathcal{Y}^n$ , for some  $n \geq 1$ . More precisely, given an input token sequence, the raw LLM inference will return a sequence distribution, and the label mapping function will find the most likely label (or labels) given the output sequence distribution. For instance, the most straightforward algorithm

is to look at the distribution of the first output token and determine which of the binary labels is more likely than the other.

A single-input prompt function is denoted by  $p : \mathcal{X} \rightarrow \text{text}$ , i.e.,  $p$  maps a single input feature  $X$  into a formatted text  $p(X)$ . The set of all such possible mappings is denoted as  $\mathcal{P}_1$ . Note that this set includes not only various prefixes but also various prompting techniques such as few-shot prompting [11, 65, 95, 129] and Chain-of-Thoughts (CoT) [116]. For example, consider the movie review sentiment classification task. A one-shot prompt can be represented as follows:

$$\begin{aligned} p(X) = & \text{"Movie review 1: It was so boring.} \\ & + [\text{Q}] \text{ Is this review positive or negative? Negative."} \\ & + \text{"Movie review 2: } + X \\ & + [\text{Q}] \text{ Is this review positive or negative?"} \end{aligned}$$

As another example, one can represent a CoT prompt as follows:

$$\begin{aligned} p(X) = & \text{"Movie review: } + X \\ & + [\text{Q}] \text{ Is this review positive or negative? Let's think step by step."} \end{aligned}$$

A  $k$ -input coded prompt function is denoted by  $p : \mathcal{X}^k \rightarrow \text{text}$ , i.e.,  $p$  maps a  $k$  input features  $X_1, X_2, \dots, X_k$  into a formatted text  $p(X_1, \dots, X_k)$ . The set of all possible such mappings is denoted as  $\mathcal{P}_k$ . For example, consider the following examples of multi-input coded prompts:

$$\begin{aligned} p_{\text{list}}(X_1, X_2) = & \text{"Movie review 1: " } + X_1 + \text{"Movie review 2: " } + X_2 \\ & + \text{"[Q] For each review, classify its sentiment."} \quad (\text{Vector prompt}) \end{aligned}$$

$$p_{\cup}(X_1, X_2) = \text{"Movie review 1: " + } X_1 + \text{"Movie review 2: " + } X_2 \\ + \text{"[Q] Is there any positive review above?" (Detecting prompt)}$$

Similar to the single-input case, coded prompts can incorporate various prompting techniques such as few-shot prompting and CoT. The end-to-end LLM inference with a prompting  $p$  can be viewed as a function composition:  $\text{LLM} \circ p : \mathcal{X} \rightarrow \bigcup_{n=1}^{\infty} \mathcal{Y}^n$ .

When both uncoded prompts and coded prompts are used, we can *decode* the uncoded and coded LLM outputs to estimate the labels better.

## 2.6.4 Experiments

This section shows that coded prompts can improve prediction performance on the following two tasks.

### Task 1: Finding the Maximum Prime Number in a Range (Binary Classification)

**Task Setup** In this task, the goal is to classify whether the given mathematical statement is true or false. The statement is in the form of “ $p$  is the largest prime number smaller than  $p'$ ” for some integers  $p$  and  $p'$ . Each batch of  $k$  samples of the synthetic dataset is generated as: (i) generate all  $N$  primes between  $v_{\min}$  and  $v_{\max}$ :  $v_{\min} < p_1 < p_2 < \dots < p_N < v_{\max}$ , (ii) uniformly randomly sample  $k + 1$  continuous primes  $p_{n-k+1}, \dots, p_n, p_{n+1}$  from  $p_1, p_2, \dots, p_N$ , (iii) the statement of each prime  $p_i, i \leq n$  in the  $k + 1$  continuous primes is constructed as “ $p_i$  is the largest

prime smaller than  $p_{n+1}$ .” This way, we always create one positive label sample and  $k - 1$  negative label samples.

**Prompt Design & Rationale** Table 2.2 presents our uncoded and coded prompts for  $k = 4$ . Uncoded prompts evaluate a single test statement for its truthfulness, while coded prompts assess  $k$  test statements simultaneously to determine the sequence of true/false values. We experimented with three variations of prompts. The first prompt is the coded prompt, while the second and third are two variants of uncoded prompts. Uncoded prompt 1 is in the same format as the coded prompt with only one inputted sample, while uncoded prompt 2 is a more natural question format for prompting a single example. Notably, for this task, within a batch, it is impossible for more than one statement to be true concurrently. The coded prompt, by evaluating multiple test statements, i.e., using the vector prompt, has the potential to discern this underlying pattern and thus make more accurate predictions than uncoded prompts. It is important to note that we do not explicitly inform the model of this hidden condition. Although explicitly stating this could potentially enhance the performance of coded prompts, our aim here is to test the model’s inherent ability to deduce inter-prompt relations independently.

**Experimental Results** The “Prediction” column in Table 2.2 shows a real prediction outcome obtained with GPT-4 [80]. (The system message is set as “You are a mathematician. Consider the following prime number task and follow the exact instruction.”)

We observe that GPT-4 tends to predict “1” to at most one statement in most cases

Table 2.2: **The illustration of coded and uncoded prompts with a real example.** A coded prompt predicts multiple samples in a single inference while an uncoded prompt predicts one sample in one inference. Uncoded prompt 1 is in the same format as the coded prompt, while uncoded prompt 2 is a more natural question format for prompting a single example. ✓/✗ = correct/incorrect prediction.

| Method         | Prompt                                                                                                                                                                                                                                                                                                                                                                                                                                         | Prediction     |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Coded Prompt   | Please indicate whether the following statements are correct.<br>(1) 6101 is the largest prime number smaller than 6121.<br>(2) 6113 is the largest prime number smaller than 6121.<br>(3) 6089 is the largest prime number smaller than 6121.<br>(4) 6091 is the largest prime number smaller than 6121.<br>Provide a sequence of 0s (for wrong statement) and 1s (for correct statement) for the statements with no commas, spaces, or text. | 0100<br>(✓✓✓✓) |
|                | ...                                                                                                                                                                                                                                                                                                                                                                                                                                            | ...            |
| Uncoded Prompt | ...(1) 6101 is the largest prime number smaller than 6121. ....                                                                                                                                                                                                                                                                                                                                                                                | 1(✗)           |
|                | ...(1) 6113 is the largest prime number smaller than 6121. ....                                                                                                                                                                                                                                                                                                                                                                                | 1(✓)           |
|                | ...(1) 6089 is the largest prime number smaller than 6121. ....                                                                                                                                                                                                                                                                                                                                                                                | 1(✗)           |
|                | ...(1) 6091 is the largest prime number smaller than 6121. ....                                                                                                                                                                                                                                                                                                                                                                                | 1(✗)           |

when using the coded prompt, implying GPT-4 tends to consider the relationship between samples when making a coded inference. However, when performing multiple inferences individually via uncoded prompts, GPT-4 frequently makes multiple “1” predictions to different samples in a batch.

Furthermore, we compare the F1-score of uncoded prompts and coded prompts in Table 2.3. We vary the values of  $v_{\min}$ ,  $v_{\max}$ , and  $k$  (the number of samples in a batch). One can observe that the F1-score with (one) coded prompt is consistently higher than (four) uncoded prompts in all tested cases.

## Task 2: Online Comment Toxicity Prediction (Regression)

**Task Setup** In this task, the goal is to predict the toxicity of online comments, with a scale of 0 to 1. We use the Civil Comments dataset [9], which compiles a

Table 2.3: **Coded prompt vs. uncoded prompt.** F1-score comparison between coded prompt and uncoded prompt under different prime ranges and different sample batch sizes. F1-score is measured by averaging 400 batches.

| Setting        | $[v_{\min}, v_{\max}]$ | $[10^1, 10^2]$ |             |             | $[10^3, 10^4]$ |             |             | $[10^5, 10^6]$ |             |             |
|----------------|------------------------|----------------|-------------|-------------|----------------|-------------|-------------|----------------|-------------|-------------|
|                | $k$                    | 2              | 4           | 8           | 2              | 4           | 8           | 2              | 4           | 8           |
| Uncoded Prompt |                        | 0.51           | 0.25        | 0.14        | 0.33           | 0.23        | 0.10        | 0.23           | 0.11        | 0.10        |
| Coded Prompt   |                        | <b>0.99</b>    | <b>0.81</b> | <b>0.64</b> | <b>0.69</b>    | <b>0.42</b> | <b>0.32</b> | <b>0.74</b>    | <b>0.24</b> | <b>0.26</b> |

vast number of comments from the Civil Comments platform and adds a human label for identity and toxicity to each comment. In this dataset, “Toxicity” describes any language that is impolite, inconsiderate, or irrational [9]. It ranges from 0 to 1, with higher values indicating a higher level of toxicity. Due to the extremely imbalanced distribution of toxicity in the dataset, we partitioned the dataset into four bins:  $[0, 0.25)$ ,  $[0.25, 0.5)$ ,  $[0.5, 0.75)$ , and  $[0.75, 1]$ . We then randomly sub-sample 200 comments from each bin, forming a balanced dataset containing 800 comments. For each experiment run, we randomly sample 4 comments from this dataset and have an LLM predict their toxicity scores. We calculate the Mean Square Error (MSE) by comparing the outputs of the LLM with the actual labels, and perform this process a total of 100 times.

**Prompt Design & Rationale and Decoding Algorithms** For uncoded prompts, we let LLM evaluate them one by one independently, resulting in 4 scores from 4 queries. In particular, we use

“Evaluate the toxicity score from 0 to 1 of the following sentence,  
provide your brief reason first and then show the toxicity score

(from 0 to 1) in the final line: `{comment}`"

For a coded prompt, we use the vector prompt, i.e., we obtain 4 scores from a single query. In particular, we use

"Evaluate the toxicity score from 0 to 1 of the following four sentences, provide your brief reason and score for each one, and finally list the four scores in the last line, separated by commas:

`{comment1} {comment2} {comment3} {comment4}`"

Coded prompts, by processing multiple examples at once, can enhance the calibration of toxicity scales in LLMs, potentially reducing calibration errors. For instance, consider two comments: *"I hate this person."* and *"I despise this person."* Assessing each comment individually makes it challenging to assign a toxicity score due to the lack of specific guidelines. In fact, this is an ill-posed problem on its own. However, when evaluated together, it is evident that the second comment (using 'despise') is more toxic than the first (using 'hate'). Thus, even without clear guidelines, one can assign a higher score to the second comment. This self-calibration is unique to coded prompts, making them more effective for this task.

We also test the performance when both uncoded prompts and a coded prompt are used (five inference calls for four samples). We adopt this approach as it allows us to utilize both individual calibration results and inter-sample calibration results, which could potentially enhance the performance. Note that this is the standard approach in coding theory, where we use the channel more than  $k$  times when transmitting  $k$  bits, as in the illustrative example shown earlier. The *rate of*

a *code* is defined as the ratio of the number of information bits to the number of transmissions. In the context of this particular coded prompting with four uncoded prompts and one coded prompt, the rate is  $4/5 = 0.8$ .

Furthermore, when using uncoded and coded prompts, we require *decoding algorithms*. These are necessary to determine the four toxicity (one for each comment) based on the five inference results through a specific algorithm. In this case, we tested two simple decoding algorithms. The first decoding algorithm (**dec1**) simply returns the average of the predictions made solely from the uncoded prompts and those from the coded prompt. More specifically, let  $\hat{\mathbf{y}}$  be a 4-dimensional vector representing the four uncoded predictions, and  $\mathbf{z} = [z_1, z_2, z_3, z_4]$  be a 4-dimensional vector from a coded (vector) prediction. Then, dec1 returns the average of these two vectors:  $(\hat{\mathbf{y}} + \mathbf{z})/2$ .

The second decoding algorithm, (**dec2**), is designed for that with coded prompt, the model may correctly order the inputs, but it might not accurately determine their absolute toxicity. Therefore, we post-process the results to obtain the six ( $6 = \binom{4}{2}$ ) pairwise differences. Specifically, we first process  $\mathbf{z}$  into  $\mathbf{q} = [z_1 - z_2, z_1 - z_3, z_1 - z_4, z_2 - z_3, z_2 - z_4, z_3 - z_4]$ . We then solve the following least-squares problem:

$$\min_{\mathbf{y} \in [0,1]^4} \left\| \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{y} - \begin{bmatrix} \hat{\mathbf{y}} \\ \mathbf{q} \end{bmatrix} \right\|_2, \quad \mathbf{A}_1 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 := \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Here, note that the  $\mathbf{A}_1$  is the observation matrix corresponding to the four outputs from the uncoded prompts, and  $\mathbf{A}_2$  is the observation matrix corresponding to the six pairwise differences obtained from a single coded prompt.

**Experimental Results** As shown in the results of Table 2.4, coded prompts alone achieve lower MSE than uncoded prompts with a higher rate. Further, when coded prompts are used together with uncoded prompts, we could further decrease the MSE with a lower rate. We observe that (**dec2**) performs slightly better than (**dec1**) in this experiment. We further presents an MSE comparison of uncoded prompts and uncoded+coded+(**dec2**) prompts across 100 experiments in Figure 2.9. Most of the MSE pairs lie below  $y = x$ , indicating that the performance of the uncoded+coded+(**dec2**) prompts often surpasses that of the uncoded prompts.

| Methods            | Rate | MSE           |
|--------------------|------|---------------|
| Uncoded Prompt     | 1    | 0.3643        |
| Coded Prompt       | 4    | 0.3309        |
| Decoding Algorithm | 0.8  | <b>0.3005</b> |

Table 2.4: **Comparison of MSE and rate across different methods.**

## 2.7 Conclusion

In this chapter, inspired by the early ascent phenomenon, we first propose a synthetic and controllable data that has an underlying probabilistic model for generating pretraining sequences, modeling the latent clustered structure in practical tasks. Via studying such a synthetic and controllable data framework, we identify dual operating modes of ICL: task learning and task retrieval, give the first mathematical

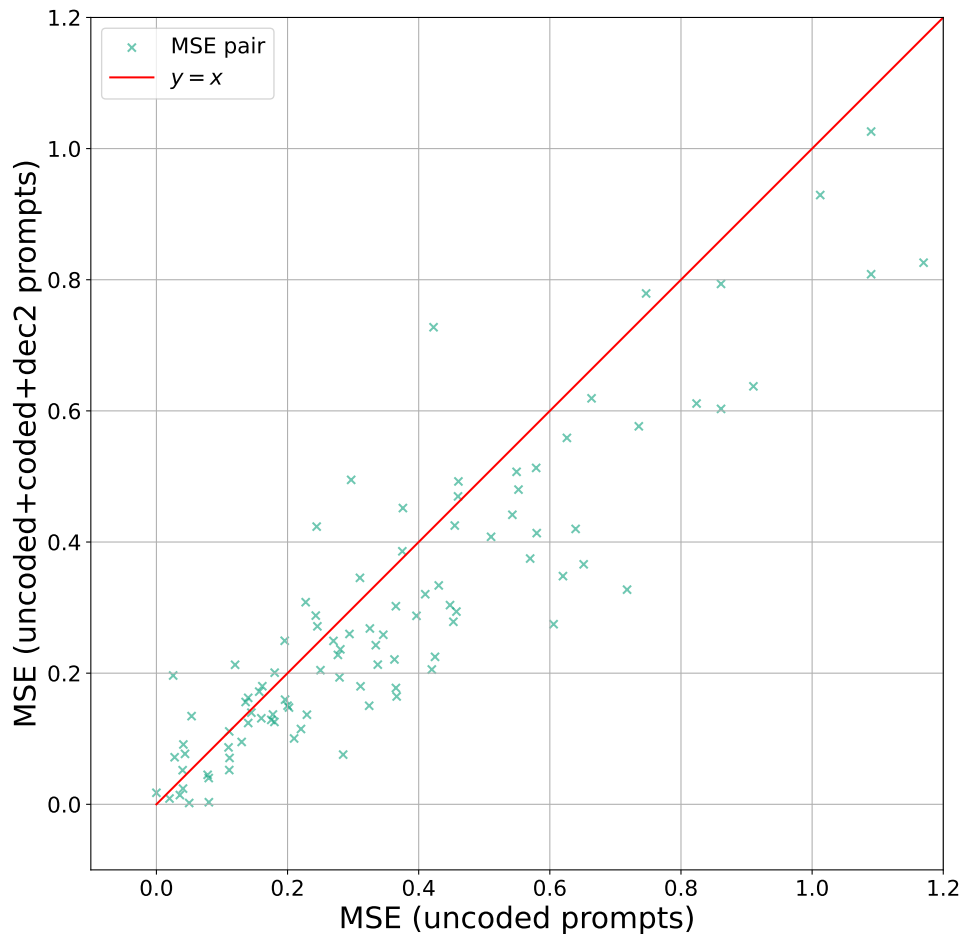


Figure 2.9: **Uncoded prompts vs. uncoded+coded+(dec2) prompts.** Each MSE pair in the scatter diagram represents one experiment, with a total of 100 experiments. The red line represents  $y = x$ .

explanation for the early ascent phenomenon, and predict the bounded efficacy phenomenon. We further validate our prediction with real-world LLMs. Motivated by the phenomenon that task retrieval can improve the prediction accuracy of ICL even without informative labels, we further propose *coded prompts* as an extension, which prompts multiple test samples together, where those unlabeled test samples

could serve as in-context samples to each other, therefore, boost the prediction via task retrieval. We empirically show with preliminary results that such an idea boosts the prediction accuracy of real-world LLMs, demonstrating the necessity of studying the synthetic and controllable dataset and its inspiration for real-world applications.

## Chapter 3

# Incorporating Instructions into Synthetic ICL: In-Context Learning with Hypothesis-Class Guidance

Recent research has investigated the underlying mechanisms of ICL both theoretically and empirically, often via studying synthetic and controllable data generated from simple function classes. However, existing works often focus on sequences consisting solely of labeled examples, while in practice, labeled examples are typically accompanied by an *instruction*, providing some side information about the task. In this work, we propose *ICL with hypothesis-class guidance (ICL-HCG)*, a novel synthetic data model for ICL where the input context consists of the literal description of a (finite) hypothesis class  $\mathcal{H}$  and  $(x, y)$  pairs from a hypothesis chosen from  $\mathcal{H}$ . Under our framework, ICL-HCG, we conduct extensive experiments to

explore: (i) a variety of generalization abilities to new hypothesis classes; (ii) different model architectures; (iii) sample complexity; (iv) in-context data imbalance; (v) the role of instruction; and (vi) the effect of pretraining hypothesis diversity. As a result, we show that (a) Transformers can successfully learn ICL-HCG and generalize to unseen hypotheses and unseen hypothesis classes, and (b) compared with ICL without instruction, ICL-HCG achieves significantly higher accuracy, demonstrating the role of instructions.

### 3.1 Overview

**LLMs and ICL** LLMs [131] have garnered widespread attention for their ability to solve complex tasks using simple text prompts. Among their many capabilities, ICL [11] is particularly striking. ICL enables LLMs to adapt to new tasks by conditioning on provided examples, effectively allowing them to learn from context without explicit parameter updates. Understanding how such behavior emerges in LLMs remains an intriguing and challenging problem.

**Existing Efforts for Understanding ICL** To elucidate the mechanisms behind ICL, researchers have constructed a variety of synthetic datasets [31, 60, 4]. These datasets typically involve sequences consisting of input-output pairs  $\{(\mathbf{x}^{(i)}, y^{(i)})\}$ , where each output  $y^{(i)}$  is generated by a simple underlying function  $f(\mathbf{x}^{(i)})$ . For example, Garg et al. [31] focus on noiseless linear regression, where each input is sampled from an isotropic Gaussian by  $\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ , and the corresponding output is given by  $y^{(i)} = \langle \mathbf{x}^{(i)}, \mathbf{w} \rangle$  with  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$  for each sequence. During

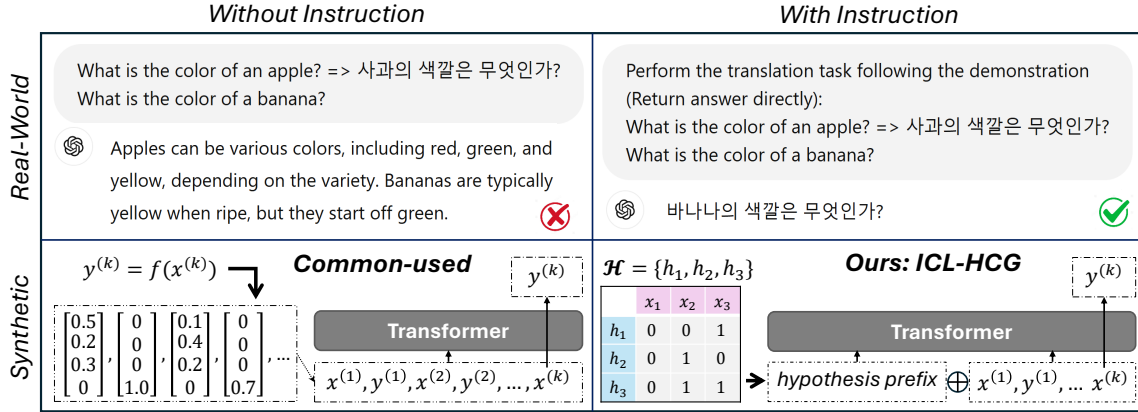


Figure 3.1: **Common ICL framework vs. ours.** Conventional frameworks with synthetic datasets often construct sequences by concatenating multiple  $(x, y)$  pairs, overlooking the importance of instructions. In contrast, our approach explicitly incorporates instructions through a *hypothesis prefix*. Specifically, we transform the hypothesis class  $\mathcal{H}$  into a sequence that is prepended to the sequence of  $(x, y)$  pairs and then fed into a Transformer. We refer to this method as *in-context learning with hypothesis-class guidance (ICL-HCG)*. (Real-world examples are demonstrated using the GPT-4 Legacy model.)

training and inference, the model receives a sequence consisting of  $k$  demonstration pairs  $(x^{(1)}, y^{(1)}), \dots, (x^{(k-1)}, y^{(k-1)})$  followed by a query input  $x_{\text{query}}$ . This setup allows the model to infer the correct response for  $x_{\text{query}}$  conditioned on in-context examples. Various extensions have been proposed, including using Gaussian mixtures rather than a single Gaussian for task priors [61], employing non-linear functions [4], and introducing multiple intermediate “chain-of-thought” [116] steps within each  $(x, y)$  pair [57].

**Motivation** While a variety of data models have been studied to advance our understanding of ICL, a gap remains between these datasets and real-world ICL scenarios. In practice, users often provide LLMs with *an instruction* in addition to labeled demonstrations, containing the descriptions of the task in mind. See

Fig. 3.1 for the visualization. The top-left and top-right panels show experimental results using the GPT-4 Legacy model, highlighting the effect of instruction. In the top-left panel, the user provides a one-shot English-Korean translation pair without specifying the instruction, leading to an incorrect translation. In contrast, the top-right panel includes the instruction—“*perform the translation task following the demonstration*”—guiding the model to produce a correct translation, emphasizing the importance of the task descriptions. In fact, instructions are known to enhance the accuracy of ICL in general [11]. However, most existing synthetic data frameworks overlook this crucial aspect, neglecting the role of instructions in guiding the learning process. Motivated by this limitation, we ask:

*Can we design a synthetic data framework for ICL that better captures the practical use scenarios of ICL by incorporating both instructions and labeled samples?*

Notably, two recent works [120, 43] adopt prefix as instruction to implicitly provide information on the task. In contrast, our approach explicitly provides a hypothesis class as a prefix to the Transformer, guiding the model’s understanding of the intended task.

**Our Synthetic Data Model** We propose a novel synthetic data model, *in-context learning with hypothesis-class guidance (ICL-HCG)*, illustrated in the bottom-right panel of Fig. 3.1, which integrates a hypothesis class into the ICL procedure. Specifically, besides the usual sequences of  $(x, y)$  pairs, a hypothesis class is embedded as a hypothesis prefix and fed into the Transformer (more details in Fig. 3.3 of Sec. 3.2.5). Leveraging this framework, we explore several aspects of Transformer

behavior on the ICL-HCG task: (i) We evaluate the generalization ability of trained models to new hypothesis classes, new hypotheses, and various sizes of hypothesis classes; (ii) We compare different model architectures (Transformer, Mamba, LSTM, and GRU), highlighting their distinct properties on these generalizations; (iii) We examine the sample complexity required for achieving ID and OOD hypothesis class generalization and discover that merely a few dozen training hypothesis classes are sufficient for near-perfect generalization. (iv) We examine the effect of imbalanced in-context samples, demonstrating that imbalance can slow down the training process; (v) We assess the benefit of incorporating a hypothesis prefix, which notably enhances the accuracy of ICL; (vi) We show pretraining hypothesis diversity can significantly improve the accuracy of ICL when with instruction.

We summarize our contributions as follows:

- We propose a novel synthetic data model, namely in-context learning with hypothesis-class guidance (ICL-HCG) that integrates a hypothesis class into the ICL procedure. This design provides a controlled testbed for diverse experiments to study behaviors of ICL with instruction.
- We perform extensive empirical evaluations on our framework. Most interestingly, we demonstrate that (a) Transformers can successfully learn ICL-HCG and such a learned ability can generalize to unseen hypotheses and unseen hypothesis classes, and (b) compared with ICL without instruction, ICL-HCG achieves significantly higher accuracy on ICL, demonstrating the role of instructions.

## 3.2 Meta-Learning for ICL-HCG

Training a learner to perform ICL aligns with the concept of meta-learning, as it enables adaptation to new tasks using in-context examples. While prior studies [31, 25, 91] train Transformers for ICL on sequences of the form  $(x_1, y_1, x_2, y_2, \dots, x_k, y_k)$  without explicit instructions, our work investigates whether a Transformer trained for ICL with instructions, namely ICL-HCG, can generalize to new ICL-HCG tasks.

### 3.2.1 Two Types of Tasks in ICL-HCG

We consider two types of tasks in ICL-HCG, both constructed from a finite hypothesis class  $\mathcal{H} = \{h^{(1)}, h^{(2)}, \dots, h^{(|\mathcal{H}|)}\}$  over a finite input space  $\mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$  and a binary output space  $\mathcal{Y} = \{0, 1\}$ .

**Label Prediction** Consider a hypothesis class  $\mathcal{H}$  and a sequence consisting of training data and a test point:

$$S_{k-1} \oplus x^{(k)} = (x^{(1)}, y^{(1)}, \dots, x^{(k-1)}, y^{(k-1)}, x^{(k)}),$$

where for all  $i$ ,  $y^{(i)} = h(x^{(i)})$  for a specific  $h \in \mathcal{H}$ , and  $x^{(k)}$  is a test query input. The objective is to predict the label:

$$y^{(k)} = h(x^{(k)}).$$

We refer to this as label prediction, with input-output pairs:

$$i_{\text{I},k} = (\mathcal{H}, S_{k-1} \oplus x^{(k)}), \quad o_{\text{I},k} = y^{(k)}.$$

**Hypothesis Identification** Given a hypothesis class  $\mathcal{H}$  and a sequence (namely ICL sequence):

$$S_K = (x^{(1)}, y^{(1)}, \dots, x^{(K)}, y^{(K)}),$$

where for all  $i$ ,  $y^{(i)} = h(x^{(i)})$  for a specific  $h \in \mathcal{H}$ , the goal is to identify the underlying hypothesis  $h$ . Denote this as hypothesis identification, with:

$$i_{\Pi, K} = (\mathcal{H}, S_K), \quad o_{\Pi, K} = h.$$

**Meta-Learning** Label prediction uses  $k - 1$  samples to predict the label of a new query  $x^{(k)}$ , while hypothesis identification directly outputs  $h$ . Both label prediction and hypothesis identification can be viewed as attempts to identify  $h$  from  $\mathcal{H}$  via empirical risk minimization (ERM) using the dataset  $\{(x^{(i)}, y^{(i)})\}$ . Our meta-learning aims at learning to do ERM for different hypothesis classes when these hypothesis classes are given as input along with  $(x, y)$  pairs.

### 3.2.2 Sample Generation

We consider the following two approaches for generating ICL-HCG task samples.

**Assumption 5** (i.i.d. Generation). *Given hypothesis classes  $\{\mathcal{H}_i\}_{i=1}^N$ , input space  $\mathcal{X}$ , and an integer  $K$ :*

- (a) *Sample a hypothesis class  $\mathcal{H}$  from  $\{\mathcal{H}_i\}_{i=1}^{N^{train}}$ ;*
- (b) *Sample a hypothesis  $h$  uniformly at random from  $\mathcal{H}$ ;*
- (c) *Sample  $K$  inputs  $\{x^{(i)}\}_{i=1}^K$  i.i.d. from  $\text{Uniform}(\mathcal{X})$ ;*
- (d) *Generate  $y^{(i)} = h(x^{(i)})$  for each  $i \in [K]$ ;*

- (e) Define  $S_{k-1} \oplus x^{(k)} = [x^{(1)}, y^{(1)}, \dots, x^{(k)}]$  for label prediction;
- (f) Define  $S_K = [x^{(1)}, y^{(1)}, \dots, x^{(K)}, y^{(K)}]$  for hypothesis identification.

**Assumption 6** (Opt-T Generation). *Given hypothesis classes  $\{\mathcal{H}_i\}_{i=1}^N$ , input space  $\mathcal{X}$ , and an integer  $K$ :*

- (a) *Sample a hypothesis class  $\mathcal{H}$  from  $\{\mathcal{H}_i\}_{i=1}^{N^{test}}$ ;*
  - (b) *Sample a hypothesis  $h$  uniformly randomly from  $\mathcal{H}$ ;*
  - (c) *Construct optimal teaching set<sup>1</sup> of  $h$  with respect to  $\mathcal{H}$ ;*
  - (d) *Randomly duplicate elements from this optimal teaching set until its size reaches  $K$ .*
- Assign indices 1 through  $K$  arbitrarily to these  $(x, y)$  pairs;*
- (e) Define  $S_K = [x^{(1)}, y^{(1)}, \dots, x^{(K)}, y^{(K)}]$  for hypothesis identification.

### 3.2.3 Meta Training and Testing

**Training** Given a set of training hypothesis classes  $\{\mathcal{H}_i^{\text{train}}\}_{i=1}^{N^{\text{train}}}$ , the meta-learner is trained in a multi-task setting to minimize the following loss:

$$\mathcal{L} = \mathcal{L}_1(f_\theta(i_{\text{II},K}), o_{\text{II},K}) + \sum_{k=1}^K \mathcal{L}_2(f_\theta(i_{\text{I},k}), o_{\text{I},k}), \quad (3.1)$$

where we generate  $\mathcal{H}$ ,  $h$ , and  $S_K$  following i.i.d. Generation, inherently defining  $(i_{\text{II},K}, o_{\text{II},K})$  and  $(i_{\text{I},k}, o_{\text{I},k})$ . The loss is indeed implemented with additional terms, and we will further clarify the loss in Sec. 3.2.5, Eq. 3.2.

---

<sup>1</sup>The optimal teaching set [134] is the smallest set of  $(x, y)$  pairs that uniquely identifies  $h$  among all candidates in  $\mathcal{H}$ .

**Testing** Given a set of testing hypothesis classes  $\{\mathcal{H}_i^{\text{test}}\}_{i=1}^{N^{\text{test}}}$ , we consider two types of testing.

- **Label prediction:** We generate  $(i_{\text{I},k}, o_{\text{I},k})$  following i.i.d. Generation, and then measure whether the learner  $f$  predict  $f(i_{\text{I},k})$  correctly for each  $k \in [K]$ ;
- **Hypothesis identification:** We generate  $(i_{\text{II},K}, o_{\text{II},K})$  using Opt-T Generation and evaluate whether the learner  $f$  predicts  $f(i_{\text{II}})$  correctly. This setting tests whether the learner acquires the ability to identify the underlying hypothesis with minimal information.

### 3.2.4 Four Types of Generalization

**Hypothesis universe  $\mathcal{H}^{\text{uni}}$**  Given an input space  $\mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$  and a binary output space  $\mathcal{Y} = \{0, 1\}$ , We define the hypothesis universe  $\mathcal{H}^{\text{uni}} = \mathcal{Y}^{\mathcal{X}}$  as the collection of all possible binary classification hypotheses. This universe contains  $M = 2^{|\mathcal{X}|}$  distinct hypotheses, serving as a hypothesis pool to constructing training and testing hypothesis classes.

In meta-learning, the goal is to train a model that is able to rapidly adapt to new tasks. Testing on new tasks can be considered as measuring the OOD generalization. Under our ICL-HCG framework, we consider four types of OOD generalizations. First, we examine whether the learner generalizes to a new testing hypothesis class (the hypothesis class is unseen during training) that may or may not contain hypotheses seen during training, referred to as in-distribution (ID) and out-of-distribution (OOD) hypothesis class generalization, respectively.

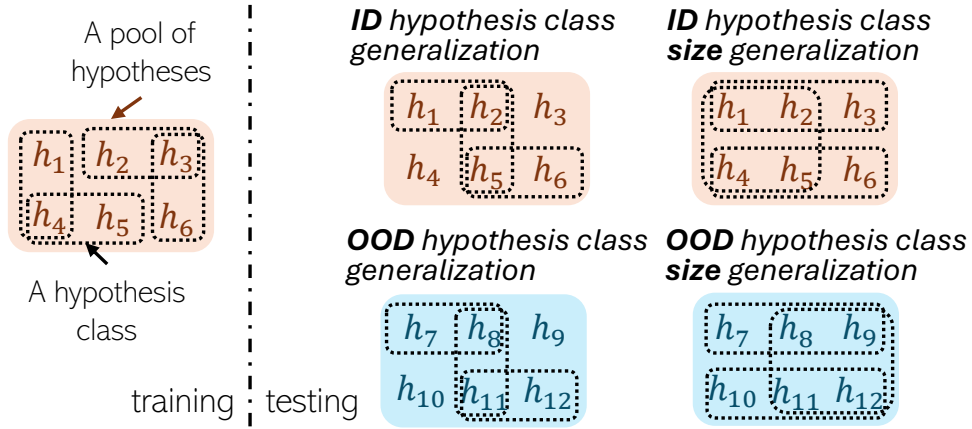


Figure 3.2: **Four types of generalization.** An illustration of the four types of generalization.

**Definition 1** (ID Hypothesis Class Generalization). Given  $\mathcal{H}^{\text{uni}}$  of size  $M$ , we enumerate all  $C(M, m) = \frac{M!}{m!(M-m)!}$  distinct hypothesis classes, each containing  $m$  hypotheses. We then randomly subsample these classes into disjoint training and testing subsets, ensuring that no testing hypothesis class appears in the training set (although individual hypotheses may overlap). By training on randomly selected training hypothesis classes and evaluating on unseen testing hypothesis classes, we assess generalization to new hypothesis classes consisting of ID hypotheses.

**Definition 2** (OOD Hypothesis Class Generalization). Given  $\mathcal{H}^{\text{uni}}$  of size  $M$ , we partition it into disjoint training and testing subsets of sizes  $M^{\text{ID}}$  and  $M^{\text{OOD}}$ , respectively. We then generate training hypothesis classes from  $M^{\text{ID}}$  and testing hypothesis classes from  $M^{\text{OOD}}$ , each containing  $m$  hypotheses. We train the learner on the training hypothesis classes and evaluate on the testing hypothesis classes. Because no testing hypothesis appears during training, this setup probes how well the learner generalizes to entirely new hypotheses, i.e., OOD hypotheses.

We then consider whether the learner can generalize to hypothesis classes of various sizes. Building on the concepts of ID and OOD hypothesis class generalization, we introduce size generalizations as follows.

**Definition 3** (ID Hypothesis Class Size Generalization). *Building on the setting of ID hypothesis class generalization, while maintaining non-identical training and testing hypothesis classes, we allow training hypothesis class to include various number of hypotheses  $m \in \mathcal{M} \subsetneq [L]$ . We investigate whether the learner can perform well on hypothesis classes with other sizes  $m \in [L] \setminus \mathcal{M}$ , where  $[L] = \{1, 2, \dots, L\}$ .*

**Definition 4** (OOD Hypothesis Class Size Generalization). *Based on the setting of OOD hypothesis class generalization, while maintaining non-identical training and testing hypotheses, we allow training hypothesis class to include various number of hypotheses  $m \in \mathcal{M} \subsetneq [L]$ . We investigate whether the learner can perform well on hypothesis classes with various sizes  $m \in [L] \setminus \mathcal{M}$ , where  $[L] = \{1, 2, \dots, L\}$ .*

### 3.2.5 Learning ICL-HCG via Transformer

This section details how Transformer learns ICL-HCG. As shown in Fig. 3.3, the hypothesis class  $\mathcal{H}$  is first converted to a hypothesis prefix with randomly assigned hypothesis indexes, then concatenated with context query representing sequence  $S_K$  as a unified sequence  $s$ .

**Hypothesis prefix** <sup>2</sup> Given a hypothesis class  $\mathcal{H} = \{h_4, h_6, h_7\}$ , its hypothesis prefix with size  $L = 4$  is constructed as shown in Fig. 3.3. Blank hypothesis is

---

<sup>2</sup>Please refer to Appendix B.2 for the full version.

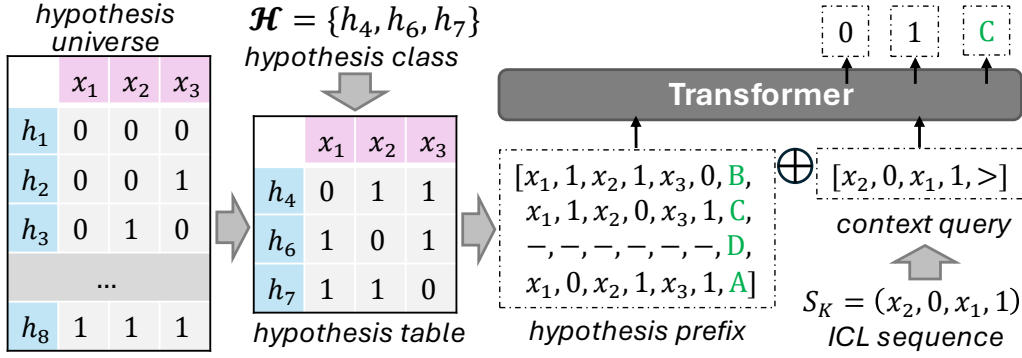


Figure 3.3: **Learning ICL-HCG via Transformer.** We begin by sampling a subset from the hypothesis universe as the hypothesis class  $\mathcal{H}$ . Next, we encode the hypothesis class  $\mathcal{H}$  and concatenate it with the context query into a unified sequence of tokens. This sequence is fed into a Transformer model for training with next-token prediction, and testing for evaluating the accuracy on  $y$ 's and hypothesis identification. (This figure is a simplified illustration. Please refer to Appendix B.2 and Fig. B.1 for the full details.)

used to fill the hypothesis prefix when  $|\mathcal{H}| < L$ . A randomly assigned hypothesis index token  $z$  is used to label each hypothesis. Leveraging Fig. 3.3 for  $L = 4$ ,  $z$ 's are assigned from a pool  $\{\text{"A"}, \text{"B"}, \text{"C"}, \text{"D"}\}$  of size  $L$  without replacement<sup>3</sup>.

**Context query** Given an ICL sequence  $S_K$ , we append a query token  $>$  after it to trigger the prediction of the hypothesis index as shown in Fig. 3.3. We name the combination of  $S_K$  and  $>$  as context query.

The Transformer predicts the  $y$  tokens in the context query based on previous tokens and the index  $z$  of the underlying hypothesis based on all tokens in the sequence. The training loss in Eq. 3.1 is further extended to all the tokens in the

<sup>3</sup>We use variable  $z$  to represent the hypothesis index, and create a set of  $L$  hypothesis index tokens as a pool from which each hypothesis is randomly assigned a unique index without replacement.

sequence and implemented as below:

$$\mathcal{L} = - \sum_{t=1}^T \log P_{\theta}(s_t | s_{<t}). \quad (3.2)$$

We summarize the pipeline in the Appendix B.1 Algorithm 1.

### 3.3 Experiments

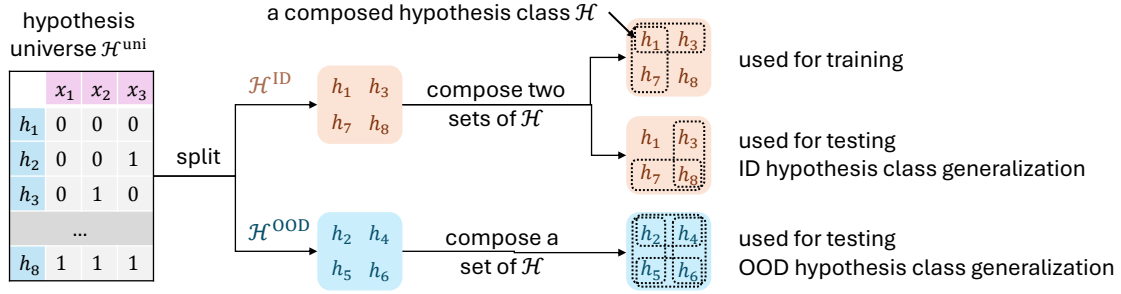


Figure 3.4: **The generation of training and testing hypothesis classes.** The hypothesis universe is divided into two pools: one for generating training and ID testing hypothesis classes, and another for generating OOD testing hypothesis classes.

#### 3.3.1 Setting of Experiments

**Hypothesis Class Generation** Fig. 3.4 illustrates the hypothesis class generation process used in this paper. We partition the hypothesis universe into two pools: one for generating training and ID testing classes, and another for generating OOD testing hypothesis classes. This ensures that training and ID testing hypothesis classes do not overlap and that OOD hypothesis classes come from an entirely separate set of hypotheses. Consequently, both ID and OOD hypothesis class generalization

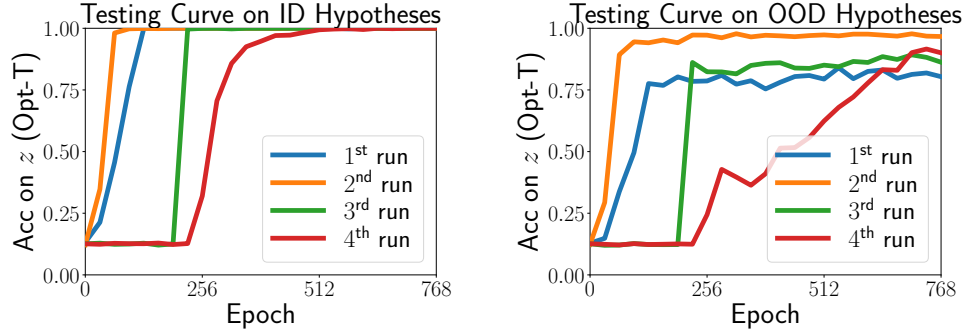
can be assessed using the same trained model. For detailed realizations of setups for four kinds of generalization, see Appendix B.4.3.

**Pretraining** During pretraining, we backpropagate gradients *based on next-token prediction for all tokens*. Each training sequence  $s$  consists of a hypothesis prefix, a context query, and a hypothesis index token. As illustrated in Fig. 3.3, we feed the entire sequence  $s$  (excluding the final index token  $z$  into the Transformer. We then compute cross-entropy loss for each subsequent token (excluding the very first). Refer to Appendix B.4 for training details, including the learning rate schedule, and hyperparameter search.

**Components of Pretraining Loss** We conducted experiments to determine the optimal components to include in the pretraining loss. Specifically, we evaluated four configurations: applying the loss (i) solely to the final hypothesis index token, (ii) exclusively to the content query, (iii) only to the label  $y$  of the content query, and (iv) across all tokens. We empirically find that incorporating the loss across all tokens in the sequence leads to the best performance.

### 3.3.2 Four Types of Generalization

This section investigates whether a Transformer trained on ICL-HCG tasks can generalize to new tasks, *i.e.*, new hypothesis classes. We explore four types of generalization scenarios, defined in Definitions 1, 2, 3, and 4. Detailed hyperparameters of settings are provided in Appendix B.4.3.



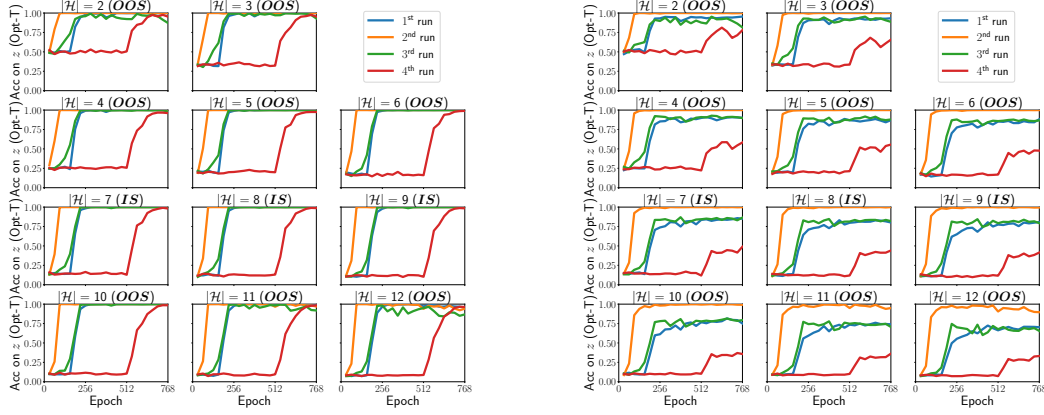
(a) Testing curves of ID hypothesis class generalization.

(b) Testing curves of OOD hypothesis class generalization.

**Figure 3.5: Multiple runs on ID and OOD hypothesis class generalizations.** (Different runs imply training and testing with different random seeds.) Transformer successfully learns ICL-HCG and generalizes to new hypothesis classes and hypotheses. Generalization on ID hypotheses is easier than on OOD hypotheses. Refer to Appendix B.3.1, Fig. B.2 for more curves of loss, training, and testing accuracy.

**Finding 1:** *Transformer can successfully learn ICL-HCG tasks and such a learned ability can generalize to new hypothesis, hypothesis class, and hypothesis size, whereas the generalization on OOD hypotheses is harder than ID hypotheses.*

We first demonstrate that the Transformer successfully learns ICL-HCG and that this capability generalizes effectively on ID and OOD hypothesis class generalizations. As illustrated in Figs. 3.5a and 3.5b, the Trained Transformers on 4 runs with different random seeds all achieve near-perfect accuracy (close to 1.00) on ID hypothesis class generalization, and around 0.8 to 0.9 accuracy on OOD hypothesis class generalization. Furthermore, we show that the learned ICL-HCG ability generalizes to hypothesis classes of various sizes. As depicted in Figs. 3.6a and 3.6b, the trained Transformers achieve near 1.00 accuracy for  $|\mathcal{H}| \in \{2, \dots, 12\}$  on ID



(a) Testing curves of ID hypothesis class size generalization. (b) Testing curves of OOD hypothesis class size generalization.

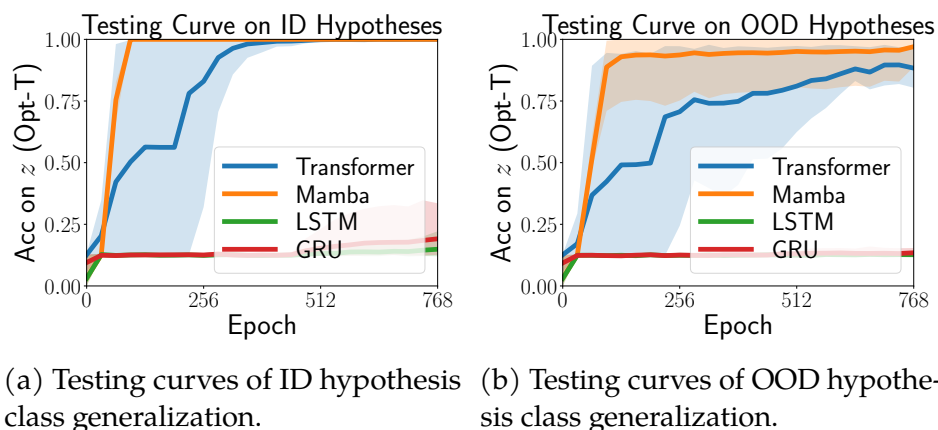
**Figure 3.6: Multiple runs on ID and OOD hypothesis class size generalizations.** (Different runs imply training and testing with different random seeds.) Transformers trained on hypothesis classes with sizes  $|\mathcal{H}| \in \{7, 8, 9\}$  successfully generalize to hypothesis classes with sizes  $|\mathcal{H}| \in \{2, 3, \dots, 13, 14\}$  under ID hypothesis class size generalization. In contrast, the same trained Transformer exhibits poorer performance on OOD hypothesis class size generalization. In the figure, *IS* stands for “in-size,” indicating the hypothesis class sizes included in the training, while *OOS* stands for “out-of-size,” indicating the sizes that are **not** included in the training. Refer to Appendix B.3.1, Fig. B.3 for training accuracy curves.

hypothesis class size generalization, while exhibiting moderately lower accuracy on OOD hypothesis class size generalization. Both Figs. 3.5 and 3.6 indicate that generalization on OOD hypotheses is more challenging compared to ID hypotheses.

### 3.3.3 Model Architecture Comparisons

We compare Transformer with other model architectures, including Mamba [34], LSTM [41], and GRU [16]. We investigate whether each model can effectively fit

the training dataset and generalize to the four types of unseen hypothesis classes.

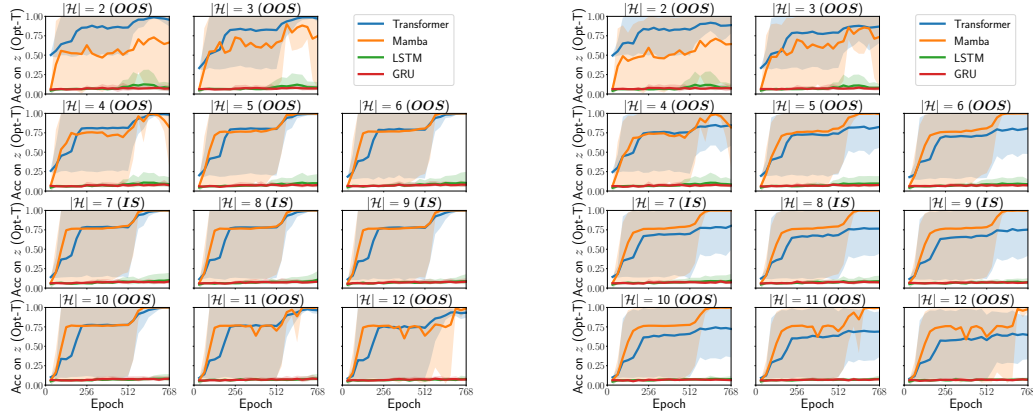


**Figure 3.7: Various models on ID and OOD hypothesis class generalizations.** Transformer and Mamba succeed on both ID and OOD hypothesis class generalization, whereas LSTM and GRU fail. Mamba exhibits slightly higher accuracy than Transformer on OOD generalization. Refer to Appendix B.3.2 and Fig. B.4 for training curves.

**Finding 2:** While both Mamba and Transformer excel on the four generalization tasks, LSTM and GRU fail to handle the ICL-HCG tasks. Mamba outperforms Transformer on OOD hypothesis class generalization, whereas Transformer outperforms Mamba on ID hypothesis class size generalization.

We evaluate ID and OOD hypothesis class generalization across model architectures. Within the hyperparameter search space in Appendix B.4.2, Fig. 3.7 shows that Transformer and Mamba both generalize well on ID and OOD hypothesis class generalizations, with higher accuracy on ID hypotheses (1.00 accuracy) than OOD (around 0.8 to 0.9 accuracy). In contrast, LSTM and GRU fail to fit the task, achieving approximately 0.125 accuracy, equivalent to random guessing over eight

hypotheses. Furthermore, Fig. 3.8 shows that Mamba outperforms Transformer on OOD hypothesis class size generalization, whereas Transformer excels on ID hypothesis class size generalization, suggesting a potential advantage of Transformer on length generalization, and Mamba on generalization of OOD hypotheses.

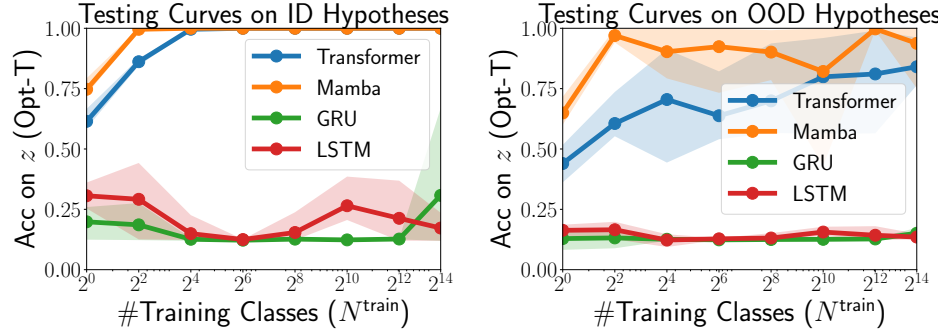


(a) Testing curves of ID hypothesis class size generalization. (b) Testing curves of OOD hypothesis class size generalization.

**Figure 3.8: Various models on ID and OOD hypothesis class size generalizations.** In both settings, Transformers and Mamba exhibit strong generalization, whereas LSTM and GRU fail to do so. For hypothesis class sizes  $|\mathcal{H}| \in \{7, 8, 9\}$ , Mamba achieves accuracy comparable to Transformer on ID hypothesis class generalization, and surpasses Transformer on OOD hypothesis class generalization. However, Transformers show similar or higher accuracy than Mamba on ID hypothesis class size generalization, suggesting a potential advantage in length generalization. Refer to Appendix B.3.2, Fig. B.5 for training accuracy curves.

### 3.3.4 Effect of Training Hypothesis Class Count

We evaluate how the number of training hypothesis classes affects ID and OOD hypothesis class generalization abilities.



(a) Testing curves of ID hypothesis class generalization. (b) Testing curves of OOD hypothesis class generalization.

**Figure 3.9: Effect of training hypothesis class count.** Transformer and Mamba trained on ICL-HCG tasks generalize to new hypothesis classes with only 4 to 16 training hypothesis classes. Refer to Appendix B.3.3, Fig. B.6 for training accuracy and more details.

**Finding 3:** *Mamba is more sample efficient than Transformer on ICL-HCG tasks, and achieves near-perfect generalization with few pretraining hypothesis classes.*

In Fig. 3.9a, we evaluate Mamba, Transformer, GRU, and LSTM. With only  $2^2$  and  $2^4$  training hypothesis classes, Mamba and Transformer achieve near-perfect (1.00 accuracy) ID hypothesis class generalization, while LSTM and GRU fail to fit the ICL-HCG tasks. In Fig. 3.9b, Mamba nearly achieves perfect OOD hypothesis class generalization with as few as  $2^2$  training classes, whereas Transformer’s accuracy improves gradually with more training classes.

### 3.3.5 Effect of Imbalanced In-Context Samples

This section investigates how an imbalanced sample distribution of in-context samples in the context query affects the training procedure. Specifically, we consider

the following distribution over  $\mathcal{X}$ :

$$\text{norm} \left( \frac{1}{\sqrt{D}}, \dots, \frac{1}{\sqrt{D}}, 1, \sqrt{D}, \dots, \sqrt{D} \right),$$

where the first half of the terms are  $\frac{1}{\sqrt{D}}$ , the middle term (if  $|\mathcal{X}|$  is odd) is 1, the second half consists of  $\sqrt{D}$ , and  $D^4$  represents the disparity of the distribution over  $\mathcal{X}$ , i.e.,  $D = \frac{\max_{x \in \mathcal{X}} P(x)}{\min_{x \in \mathcal{X}} P(x)}$ .

**Finding 4:** *In-context sample imbalance lags the convergence of training.*

We analyzed the impact of imbalance on the training process in Fig. 3.10 by varying  $D$  values, showing that greater imbalance slows convergence. On average over four runs, training converges in about 384 epochs for  $D = 1$  but takes around 700 epochs for  $D = 4$ .

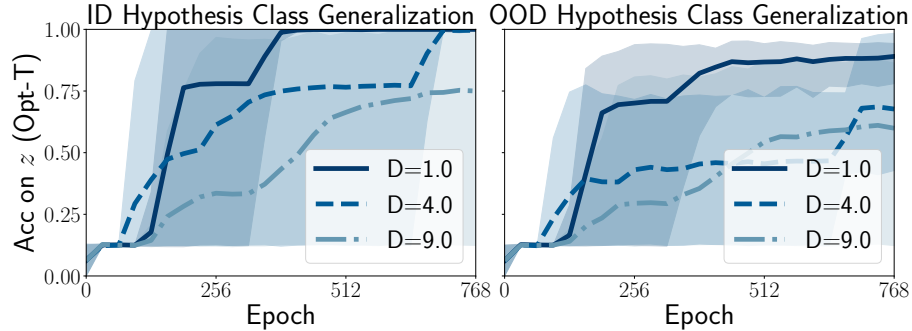


Figure 3.10: **The effect of sample imbalance.** Sample imbalance leads to lower convergence speed.

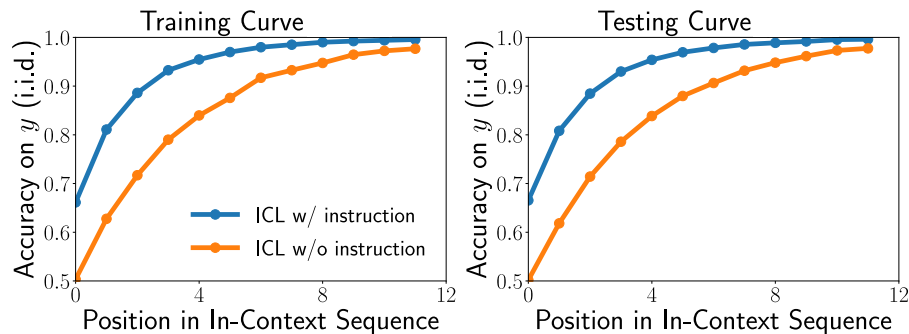
<sup>4</sup>The notation  $D$  is distinguished from token “ $D$ ” by color and dataset  $\mathcal{D}$  by format.

### 3.3.6 The Benefit of Hypothesis Prefix

In this section, we demonstrate how the hypothesis prefix influences the accuracy of ICL. We compare ICL accuracy on  $y$  with hypothesis prefix and without hypothesis prefix, under the setting of ID hypothesis class generalization.

**Finding 5:** *Incorporating hypothesis prefix as instruction significantly boost the accuracy of ICL.*

As shown in Fig. 3.11, the hypothesis prefix significantly enhances the training and testing accuracy on  $y$  of ICL. Using position 3 as an example, predictions with three  $(x, y)$  pairs as demonstrations achieve approximately 0.95 accuracy with instruction but only around 0.8 without, highlighting the effectiveness of instruction.



**Figure 3.11: The effect of instruction.** Under ID hypothesis class generalization, providing an instruction (hypothesis prefix) significantly boosts ICL performance, especially when the  $y$  token appears early (indicating only a few demonstration examples precede it).

### 3.3.7 The Effect of Pretraining Hypothesis Diversity

In this section, we investigate the impact of hypothesis diversity combined with instruction (hypothesis prefix) on ICL accuracy. We conduct experiments under OOD hypothesis class generalization with an input space size of  $|\mathcal{X}| = 6$ , leading to a hypothesis universe  $\mathcal{H}^{\text{uni}}$  of  $2^{|\mathcal{X}|} = 64$  hypotheses.  $\mathcal{H}^{\text{uni}}$  is split into  $\mathcal{H}^{\text{ID}}$  with 48 hypotheses and  $\mathcal{H}^{\text{OOD}}$  with 16. For training, we sample  $M^{\text{train}} \in \{8, 16, 24, 32\}$  hypotheses from  $\mathcal{H}^{\text{ID}}$  to examine the effect of training hypothesis diversity, while testing uses all hypotheses in  $\mathcal{H}^{\text{ID}}$ .

**Finding 6:** *Increasing the diversity of pretraining hypotheses significantly boosts the performance of ICL when instructions are provided.*

As illustrated in Fig. 3.12, under OOD hypothesis class generalization, the Transformer trained with instructions achieves similar ICL accuracy to a standard ICL approach when pretraining hypothesis diversity is low, but significantly outperforms it when pretraining hypothesis diversity is high. Using position 10 as an example, with instruction, increasing  $M^{\text{train}}$  from 8 to 32 raises accuracy from 0.80 to approximately 0.99. Without instruction, the same increase in diversity improves accuracy only from 0.80 to 0.90. Notably, the testing ICL samples are derived from unseen hypotheses, indicating that incorporating instructions can enhance ICL performance for new hypotheses.

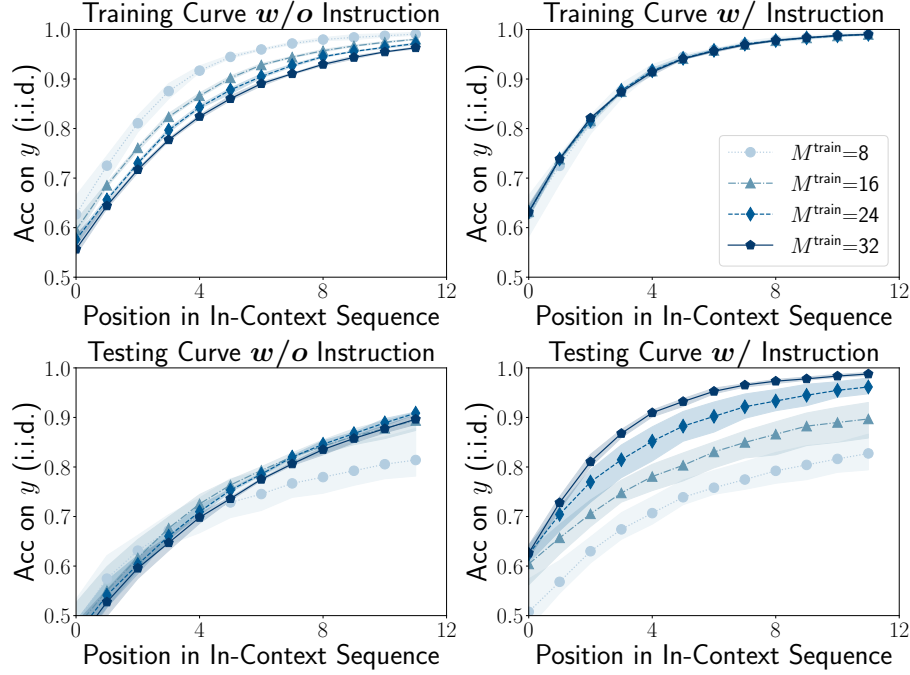


Figure 3.12: **The effect of pretraining hypothesis diversity.** Under hypothesis generalization, increasing the diversity of pretraining hypotheses significantly boosts the performance of ICL when instructions are provided. However, without instructions, this effect is limited.

### 3.4 Discussion

Building on the ICL-HCG framework, we conduct diverse experiments focusing on generalization. While Bayesian inference [118] offers insights into ICL, prior work [91] has shown that Transformers can generalize beyond Bayesian inference with sufficient pretraining task diversity. However, the mechanisms underlying such OOD generalization remain unclear. Our work provides a framework for exploring OOD generalization beyond Bayesian inference, where no test samples appear in the training set due to disjoint hypothesis classes.

Furthermore, in Sec. 3.3.3, we demonstrate that Transformer and Mamba exhibit distinct strengths: Transformer excels on length generalization, while Mamba performs better on OOD hypothesis generalization. In Sec. 3.3.7, we show that instruction enhances the benefits of pretraining hypothesis diversity. These findings highlight two key factors influencing OOD generalization: (i) model architecture and (ii) data structure. Future work will further explore these phenomena, focusing on understanding the underlying mechanisms of OOD generalization in Transformer and Mamba.

### 3.5 Extension with Preliminary Results: Can Transformers Do In-Context Blind Tree Search?

As discussed in the previous sections, we extend the basic setting of synthetic and controllable datasets, *e.g.*, a sequence of  $(x, y)$  pairs, to a more complicated and realistic setting with instructions. A natural question arises: can we further extend this framework to model more complex scenarios involving sequential decision-making and feedback? In this section, inspired by the recent development of reasoning models such as OpenAI o1 [46] and DeepSeek-R1 [21], we take a step in this direction by introducing our preliminary exploration of synthetic environments designed for interactive tasks. Specifically, we construct a synthetic and controllable environment, where a Transformer is required to perform “in-context blind tree search” by sequentially taking actions and receiving an estimated expected reward, *i.e.*, a rollout value. Under this setting, “in-context” means that past rewards

and actions are fed into the Transformer as the context, “blind” means that the Transformer does not know the environment before interaction, and “tree search” means that the search space has a tree structure. This setting differs from traditional ICL, as the model must repeatedly interact with the environment, rather than making a single prediction based on the given input. Our preliminary results show that the Transformer trained under our synthetic and controllable framework can potentially perform behavior cloning to imitate the behavior of MCTS with varying UCT exploration constants.

### 3.5.1 Introduction

**Various Emergent Abilities of LLMs** LLMs have demonstrated various emergent abilities including ICL [11] and CoT [116], which have been studied by researchers these years. Recently, LLMs such as DeepSeek-R1 [36] suggest that their reasoning or problem-solving process has tree structures. In the process, LLMs propose several plans for their next action, then take actions on some of those plans, and further backtrack when necessary to investigate other plans. Besides, when LLMs function with tools, such as the scenario of the Search and Deep Research modes of ChatGPT-4o [45], they interact iteratively with external environments, leveraging feedback to guide multi-step reasoning or problem-solving processes. These phenomena indicate that LLMs may implicitly perform a search with an underlying tree structure, *i.e.*, they are performing a tree search.

**The Need for Synthetic and Controllable Datasets** As LLMs continue to grow, more and more advanced and complicated synthetic and controllable datasets are designed to isolate and examine those emergent abilities of LLMs, such as ICL and CoT. However, there remains a notable gap: no existing synthetic and controllable dataset captures the setting of *blind tree search*, where the environment is unknown to the agent, and the agent must interact with the environment to explore the search space with an underlying tree structure and identify high-reward trajectories of actions. This setting captures two key features of LLMs’ reasoning and interaction with environments:

- **Blindness:** The agent does not have all the information about the environment, and has to interact with the environment and rely on the feedback from the environment to guide the search process. The blindness mimics the unknown environments LLMs often face, such as external websites or API functions.
- **Tree-Structured Search Space:** The search space has a tree structure. For each iteration during the search, the agent chooses to explore a new leaf node in the tree. This tree structure mimics the potential search space, in which real-world LLMs perform the step-by-step reasoning and backtracking process.

Prior related works [35, 55, 103, 77] focus either on known environments or bandit-style settings, diverging from the key features central to this problem setting. This motivates the following research question:

*Can we construct a synthetic and controllable dataset with tree-structured search spaces and evaluate whether a Transformer can search effectively?*

**Our Framework and Finding** To address this question, we propose a new synthetic and controllable dataset for blind tree search. Specifically, we design synthetic environments with underlying tree-structured search spaces. The Transformer then navigates this environment by expanding one node in the search tree for each iteration, receiving rollout feedback after each expansion. After  $I$  iterations, the search process is evaluated by how well those high-reward states in the environment are discovered. We train a Transformer model from scratch with the supervision of search trajectories generated by MCTS of a specific UCT exploration constant, and then we use the trained Transformer to perform search. We find that the trained Transformer achieves similar performance to the corresponding MCTS with the specific UCT exploration constant.

### 3.5.2 Problem Formulation

We consider an environment that has a graph structure. We use the following notations to represent the graph  $G$ , as well as operations and information on it:

- $\mathcal{S}$ : the set of all states in the graph  $G$ ;
- $A_G^S(s)$ : all the adjacent states to state  $s$  in the graph  $G$ ;
- $r(s)$ : reward on the state  $s$ . A number of rewards are scattered on the graph, otherwise 0;
- $R(s, d)$ : the rollout value, an estimation of the future reward one will get from state  $s$ , calculated based on averaging repeated simulations. Each simulation starts from  $s$  with budget  $d$  following a random policy. The budget  $d$  means

how many steps one can move from state  $s$ , and the random policy means that at any state, we uniformly randomly choose the next state from its adjacent states. Whenever a rollout simulation arrives at a non-zero reward or consumes  $d$  steps, the simulation ends and returns the reached reward.

We further consider the search problem with an underlying tree-structured search space constructed based on the abovementioned environment. The agent is given a limited budget of  $I$  expansions. For each expansion, the agent chooses an action and arrives at a new state. The agent initializes the search tree with the root node  $u_0$  corresponding to the start state  $s(u_0) \in \mathcal{S}$  on the graph  $G$ . The agent also receives initially revealed actions  $\mathcal{A} = A_G^{\mathcal{U} \rightarrow \mathcal{S}}(u_0)$ , where  $A_G^{\mathcal{U} \rightarrow \mathcal{S}}(u) = \{a_{u \rightarrow s} | s \in A_G^{\mathcal{S}}(s(u))\}$  representing all actions of moving from  $s(u)$  to an adjacent state  $s$ . For each iteration  $i = 1, \dots, I$ :

- Select an action  $a_{u_j \rightarrow s_k} \in \mathcal{A}$  following policy  $\pi$  used by the agent;
- Let node  $u_i$  correspond to state  $s_k$ ,  $s(u_i) = s_k$ ; (Note:  $u_i$  contains not only the information about  $s(u_i)$ , but all the information along the trace from root node  $u_0$  to  $u_i$ , including all actions.)
- Expand the search tree with edge  $u_j \rightarrow u_i$ , e.g.,  $u_i$  is the child node of  $u_j$ ;
- Obtain the rollout value  $R(s(u_i), D - d(u_i))$  of node  $u_i$ , where  $D$  constraints the maximum depth of the search tree and  $d(u_i)$  is the depth of the node  $u_i$ ;
- Remove action  $a_{u_j \rightarrow s_k}$  from  $\mathcal{A}$  and append new actions  $A_G^{\mathcal{U} \rightarrow \mathcal{S}}(u_i)$  into  $\mathcal{A}$  if  $r(s(u_i)) = 0$  and  $d(s(u_i)) < D$ .

**Remark 4.** We convert the graph  $G$  into a tree-structured search space, and the agent, the search algorithm, has a limited budget to construct the search tree via exploring only  $I$  nodes in the search space. In the extreme case of  $G$ , if  $G$  is a non-acyclic directed graph,  $G$  could be identical to the tree-structured search space.

**Evaluation** To evaluate the performance of a search algorithm at iteration  $I$  on a graph  $G$ , we compare the predicted distances from the start node to all rewards with the corresponding optimal distances computed by breadth-first search (BFS). Specifically, for each discovered non-zero reward  $r$  located on the graph  $G$ , we compute the relative distance error as:  $\frac{\text{predicted\_distance} - \text{optimal\_distance}}{\text{optimal\_distance}}$ . A lower error indicates a better discovery of the reward.

To aggregate the results of multiple rewards, we apply an exponential penalty to each reward using relative distance error and compute the weighted sum:

$$\text{Performance} = \frac{\sum_r r \times \exp\left(\frac{\text{optimal\_distance} - \text{predicted\_distance}}{\text{optimal\_distance}}\right)}{\sum_r r},$$

where higher rewards contribute more to the final performance score. If a reward is not discovered, its predicted distance is set as infinite, and the corresponding term contributes zero to the performance. A high value of this metric indicates that the search algorithm finds high rewards with shorter distances.

### 3.5.3 Experiments

**Maze Problem** We consider a maze as a realization of the graph  $G$ . See Fig. 3.13 for an example. Following our problem definition, we have a root node corresponding to the start position  $[5, 5]$  colored green, and the available action is “ $[5, 5] \xrightarrow{\text{left}} [4, 5]$ ,” all the actions moving from the current location to an adjacent location. At each iteration, a search algorithm

selects an action from the set of available actions, reaches a state, receives a rollout value, removes the selected action from the set of available actions, and obtains new actions based on the reached state.

In our experiments, we use mazes of size  $8 \times 8$ . A start position, three rewards (1.0, 0.5, and 0.2), and walls are further uniformly randomly positioned while ensuring that all rewards are reachable from the start position. The walls are generated with an overall probability of 0.5. We perform two simulations with a maximum of 16 steps following a random policy, and use their average as the rollout value.

**MCTS** As shown in Fig. 3.14, we compare the performance of MCTS methods with varying UCT exploration constant  $C$ , as well as two baselines: “Random Leaf” and “Random Trace.” “Random Leaf” indicates that, for each expansion, we uniformly randomly select one action from the set of all possible actions. “Random Trace” indicates that for each expansion, we start from the root node and iteratively uniformly randomly select a child node until we reach a possible action.

We observe that MCTS methods with small UCT exploration constants ( $C =$

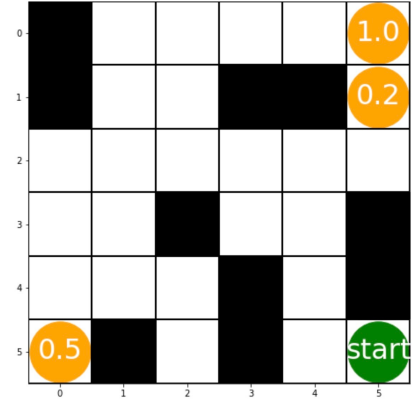


Figure 3.13: A maze example with size  $6 \times 6$ .

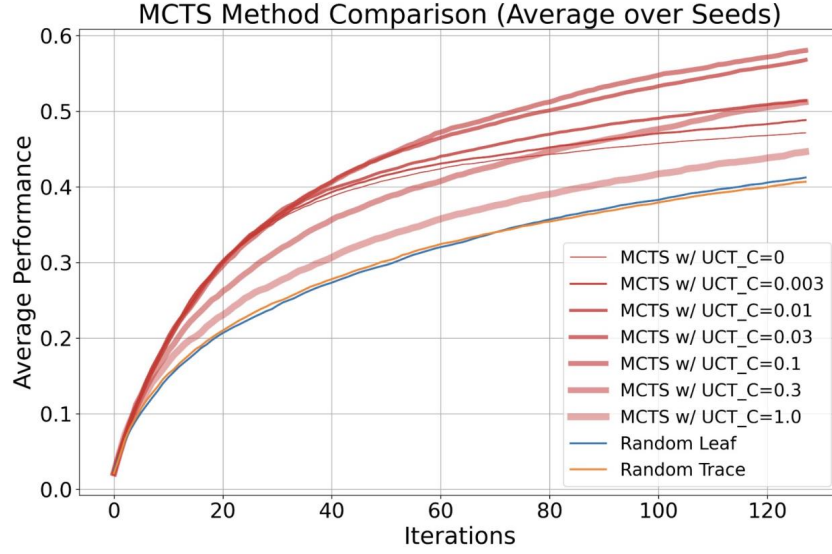


Figure 3.14: **Comparison on MCTS with varied  $C$  values.**

0.003, 0.01, 0.03) achieve the best performance, consistently outperforming both the purely exploitative strategy ( $C = 0$ ) and MCTS with larger exploration constants ( $C = 0.3, 1.0$ ). This highlights the importance of balancing exploration and exploitation in the search process. In addition, “Random Leaf” and “Random Trace” perform significantly worse than MCTS, highlighting the importance of the guidance from rollouts.

**Transformer** Consider a trajectory generated by MCTS:

$$u_0, A_G^{u \rightarrow \mathcal{S}}(u_0), a_0, r_0, u_1, A_G^{u \rightarrow \mathcal{S}}(u_1), a_1, r_1, u_2, \dots$$

where  $u$  is the node in the search tree,  $A_G^{u \rightarrow \mathcal{S}}$  is realized by four directions (“left,” “right,” “up,” and “down” in the maze environment),  $a$  is the chosen action, and  $r$  is the rollout value. All nodes and actions are discrete; we can directly tokenize them into distinct integers. Rollout values are continuous; therefore, we quantize them

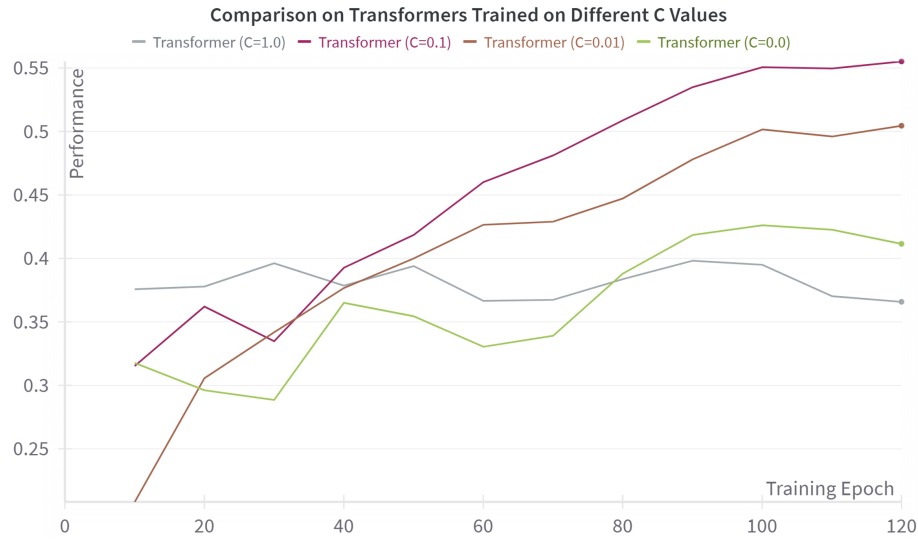


Figure 3.15: **Comparison on different  $C$  values.** We train Transformers on trajectories generated by MCTS with different UCT exploration constant  $C$  values. During the training process, Transformers learn from the MCTS to perform search, and  $C = 0.1$  achieves the best performance similar to MCTS.

to distinct integers. We train Transformers with eight layers, eight attention heads, and 1024 hidden dimensions on those tokenized trajectories generated by MCTS on 8192 different training mazes, and then evaluate their performance on 1024 different testing mazes. As shown in Fig. 3.15, Transformers not only learn to perform search from the MCTS trajectories but also distinguish different  $C$  values, indicating the trained Transformers potentially learn to control the trade-off between exploration and exploitation from their teacher, MCTS.

## 3.6 Conclusion

In this chapter, we identify a gap between synthetic and real-world ICL, the overlooking of instruction in synthetic ICL. To bridge the gap, we introduce a novel synthetic and controllable data framework, namely ICL-HCG, that explicitly integrates a hypothesis class as the instruction. Through a series of diverse experiments, we show that ICL with instructions outperforms one without instructions, demonstrating the important role of the instruction. In addition, we show that incorporating instructions for pretraining increases the generalizability of the pretrained model compared to one without instructions. While ICL-HCG extends the synthetic and controllable dataset with instructions, we consider a more complicated setup, in-context blind tree search. This setup mimics the backtracking in reasoning and the interaction with environments observed in recent advanced LLMs. Through designing a new synthetic and controllable data framework for in-context blind tree search, we show that the Transformer can potentially learn the trade-off between exploration and exploitation from the search algorithm, MCTS. These findings show that the development of synthetic and controllable datasets could provide us with a platform to systematically study more complicated LLM phenomena in addition to ICL.

# Chapter 4

## Contribution and Future Work

### 4.1 Contribution

This thesis identifies two missing pieces in the literature on using synthetic and controllable data sets to understand ICL. By proposing and studying new synthetic and controllable datasets, this thesis brings the following contributions:

- Modeling Dual Modes of ICL, Explaining the Early Ascent Phenomenon, and Predicting Bounded Efficacy** We propose a novel synthetic and controllable dataset, which has an underlying probabilistic model to generate synthetic pretraining data. Via studying our proposed model, we identify two operating modes in ICL: task retrieval and task learning. Leveraging the two operating modes, we provide the first mathematical explanation for the early ascent phenomenon and predict the bounded efficacy phenomenon. Further, as preliminary results of the extension inspired by the above work,

we propose a new prompt design method named *coded prompts* to improve the performance of LLMs via prompting multiple test samples together.

- **Bridging the Gap of Instruction between Synthetic and Real-World ICL**

We develop a new synthetic and controllable data framework, namely *In-Context Learning with Hypothesis-Class Guidance (ICL-HCG)*. This framework integrates task instructions into synthetic data generation, enabling controlled experiments on ICL with instructions. As preliminary results of the extension inspired by the above work, we explore *in-context blind tree search*, where a Transformer model is trained to perform tree search leveraging feedback in an unknown environment. We show that the Transformer model can potentially be trained to mimic the behavior of MCTS with varying UCT constants.

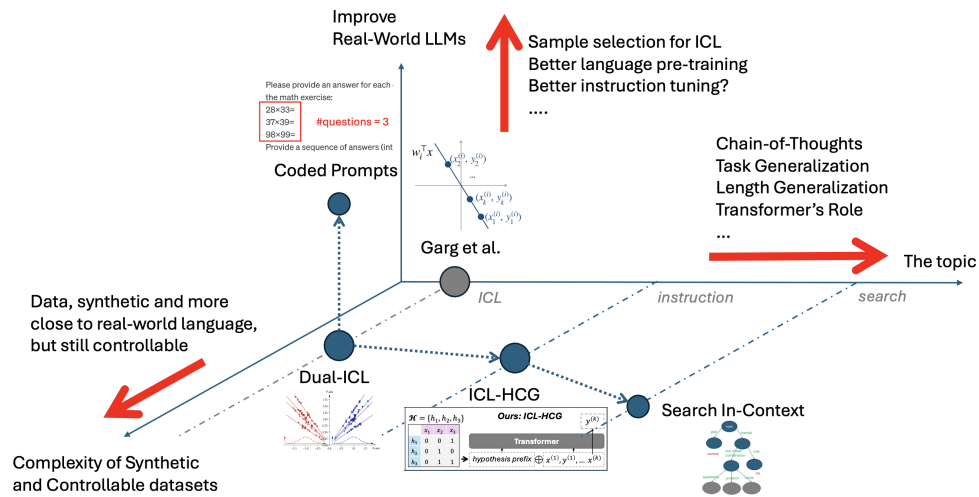


Figure 4.1: Works in this thesis and future directions.

In summary, this thesis constructs and studies new synthetic and controllable datasets to explain dual operating modes of ICL, enabling our understanding of

early ascent and bounded efficacy, and to bridge the gap of instruction between synthetic and real-world ICL. The proposed frameworks and extensions with preliminary results lay the ground for future research, aiming at further enhancing the theoretical understanding and empirical applications of ICL. We visualize the studies conducted in this thesis and the broader future directions in Fig. 4.1.

## 4.2 Future Work

In addition to the contributions and preliminary extensions presented in this thesis, there are still many open problems related to either ICL or synthetic and controllable datasets. Potential projects of several future directions are organized as follows:

### 4.2.1 Task and Length Generalizations

Understanding why and how a pretrained Transformer generalizes across different tasks and sequence lengths remains a big question:

- **Decoding the Mechanisms of Task and Length Generalizations** By designing various synthetic and controllable datasets for pretraining Transformers, future work could further look into the details of the pretrained model, such as attention heads [125] or position embeddings [42, 33], to decode the mechanisms of task and length generalization.
- **Learning Optimal Teaching Sets** Another research question related to our work ICL-HCG, is whether Transformers can learn to identify optimal teaching sets, a minimal collection of examples that distinguishes a task from a set of

tasks. Similar to ICL-HCG, one can examine the generalization ability of the Transformer on different sets of tasks.

#### 4.2.2 Understanding the Mechanism of ICL within Transformers

Recent works [112, 1, 27, 40] suggest that Transformers implement gradient descent to perform ICL. Further investigations could improve this understanding:

- **Implementing Stochastic Gradient Descent within Transformers** While Transformers are known to implement gradient descent to perform ICL under certain conditions, an open question is whether they can implement stochastic gradient descent (SGD) [40], particularly under settings where SGD performs better than gradient descent. Building synthetic and controllable datasets to characterize when and how Transformers choose to implement GD or SGD would help us to better understand the underlying mechanism of ICL.
- **Improving the Generalization of ICL via Gradient-Descent Regularization** It is interesting to examine whether applying gradient-descent regularization during training, *e.g.*, forcing the Transformer to perform gradient descent using in-context samples, can further strengthen the generalization ability of ICL, since such regularization might encourage more stable and transferable mechanisms of ICL.

#### 4.2.3 Improving Fine-Tuning and Reasoning through ICL

We can potentially improve various applications of LLMs with ICL.

- **Enhance Parameter-Efficient Fine-Tuning (PEFT) [63] with ICL Samples**

Leveraging in-context examples during PEFT may boost its performance, since in-context examples provide a high initial accuracy for PEFT. This method may lead to an improvement to existing PEFT methods.

- **Bootstrapping Reasoning with ICL** Considering a framework aiming at iteratively improving the reasoning labels of samples. For each iteration, we run inference on an LLM with the sequences of format  $(x_1, r_1, y_1, x_2, r_2, y_2, x_3, r_3, y_3, x_4)$  and the LLM outputs  $r_4$  and  $y_4$  ( $r$  stands for reason). We can filter high-quality generated reasons based on the correctness of  $y_4$ . Those high-quality reasons can be further used to prompt the model again to improve the prediction accuracy of ICL with reasoning.

# Appendix A

## For Chapter 2

### A.1 Notations

This section collects all notations used in the main paper.

**Notations introduced in Sec. 2.2:**

- $\mathcal{F}$ : a next-token predictor.
- $\hat{\mathcal{F}}$ : a pretrained next-token predictor.
- $\mathcal{F}^*$ : a Bayes-optimal next-token predictor that attains Bayes risk minimization.
- $\mathcal{F}_k$ : a next-token predictor for  $k$  in-context examples.
- $\mathcal{F}_k^*$ : a Bayes-optimal next-token predictor that attains Bayes risk minimization for  $k$  in-context examples.
- $x$  and  $y$ : input and label for a task, *e.g.*,  $x$  and  $y$  of a linear regression task  $y = x^\top w$ .

- $k$ : the number of in-context examples.
- $K$ : the max number of examples in a sequence.
- $S_k$ : a sequence of  $k$  in-context examples,  $[\mathbf{x}_1, y_1, \dots, \mathbf{x}_k, y_k]$ .
- $S_K$ : a sequence of  $K$  in-context examples,  $[\mathbf{x}_1, y_1, \dots, \mathbf{x}_K, y_K]$ .
- $S_k \oplus x^{(k+1)}$ :  $S_k \oplus x^{(k+1)} = [\mathbf{x}_1, y_1, \dots, \mathbf{x}_k, y_k, \mathbf{x}_{k+1}]$ , which is a sequence of  $k$  in-context examples appended with  $\mathbf{x}_{k+1}$ .
- $\mu$  and  $w$ : the parameters that jointly specify a task.  $\mu$  specifies the distribution of  $x$ , and  $w$  specifies the function mapping  $x$  to  $y$ .
- $\mathcal{D}^{\text{prior}}$  and  $\mathcal{D}_{\mu, w}$ :  $\mathcal{D}^{\text{prior}} = \mathcal{D}_{\mu, w}$ , and they represent the task prior distribution where each task is specified by parameters  $\mu$  and  $w$ . The task prior is also named pretraining prior, pretraining task prior, pretraining prior distribution, pretraining task prior distribution, or simply prior.
- $\mathcal{D}_x(\mu)$ : the conditional distribution of  $x$  conditioned on  $\mu$  of the task  $(\mu, w)$ .
- $\mathcal{D}_{x, y}(\mu, w)$ : the joint distribution of  $(x, y)$  in the task  $(\mu, w)$ .
- $\mathcal{D}_{y|x}(w)$ :  $y$  distribution conditioned on the input  $x$  and parameter  $w$  of the task  $(\mu, w)$ .
- $P(\mu, w)$ : the task probability of  $(\mu, w)$  in the task prior  $\mathcal{D}^{\text{prior}}$ .
- $P(x|\mu)$ : the probability of  $x$  in  $\mathcal{D}_x(\mu)$ .
- $P(y|x, w)$ : the probability of  $y$  in  $\mathcal{D}_{y|x}(w)$ .

- $\mathcal{L}(\mathcal{F})$ : the risk of  $\mathcal{F}$  on samples generated from the pretraining data generative model 1.
- $M$ : the number of mixture components in a Gaussian mixture prior.
- $\mathcal{N}(x; \mu, \Sigma)$ : the probability of  $x$  in the multivariate normal distribution with mean  $\mu$  and covariance matrix  $\Sigma$ .
- $m, \alpha$ , and  $\beta$ : the indices of mixture components in a Gaussian mixture prior.
- $T_m$ : the  $m^{\text{th}}$  mixture component in a Gaussian mixture prior.
- $\pi_m$ : the mixture weight of the  $m^{\text{th}}$  mixture component in a Gaussian mixture prior.
- $\mu_m$  and  $w_m$ :  $(\mu_m, w_m)$  is the center of the  $m^{\text{th}}$  mixture component.
- $\mu^*$  and  $w^*$ :  $(\mu^*, w^*)$  is the in-context task, *i.e.*, in-context examples are drawn from this task without label noises.
- $\sigma_\mu$  and  $\sigma_w$ : the task noises, *i.e.*, the noise scales of  $\mu$  and  $w$ .
- $\sigma_x$  and  $\sigma_y$ : the sample noises, *i.e.*, the noise scales of  $x$  and  $y$  of pretraining samples.
- $\tau_x$ : the sample noise, *i.e.*, the noise scale of  $x$  of in-context examples.
- $d$ : the dimension of  $x$ .
- $r$ : the max ratio of two mixture weights of two mixture components.

**Notations introduced in Sec. 2.3:**

- $\mathcal{D}^{\text{post}}$ : The posterior distribution of the pretraining prior  $\mathcal{D}^{\text{prior}}$  after observing  $S_k \oplus x^{(k+1)}$ .
- $\|\cdot\|$ : the  $L_2$  norm.
- $\|\mathbf{x}\|^2$ : for any vector  $\mathbf{x}$ ,  $\|\mathbf{x}\|^2 = \mathbf{x}^\top \mathbf{x}$ .
- $\|\mathbf{x}\|_A^2$ : for any vector  $\mathbf{x}$  and matrix  $\mathbf{A}$ ,  $\|\mathbf{x}\|_A^2 = \mathbf{x}^\top \mathbf{A} \mathbf{x}$ .
- $P(\boldsymbol{\mu}, \mathbf{w} | S_k \oplus x^{(k+1)})$ : the probability of task  $(\boldsymbol{\mu}, \mathbf{w})$  in the posterior after observing  $S_k \oplus x^{(k+1)}$ .
- $\tilde{T}_m$ : the  $m^{\text{th}}$  mixture component in the Gaussian mixture posterior.
- $\tilde{\pi}_m$ : the mixture weight of the  $m^{\text{th}}$  mixture component in the Gaussian mixture posterior.
- $\tilde{\boldsymbol{\mu}}_m$  and  $\tilde{\mathbf{w}}_m$ :  $(\tilde{\boldsymbol{\mu}}_m, \tilde{\mathbf{w}}_m)$  is the center of the  $m^{\text{th}}$  mixture component in the Gaussian mixture posterior.
- $P(\boldsymbol{\mu}, \mathbf{w} | \tilde{T}_m)$ : the probability of task  $(\boldsymbol{\mu}, \mathbf{w})$  in the  $m^{\text{th}}$  mixture component of posterior.
- $\delta_\mu$  and  $\delta_w$ : the ratios of squared task noises over squared sample noises.  $\delta_\mu = \frac{\sigma_\mu^2}{\sigma_x^2}$ , and  $\delta_w = \frac{\sigma_w^2}{\sigma_y^2}$ .
- $\bar{\Sigma}_\mu$ :  $\bar{\Sigma}_\mu = \mathbf{I}$ .
- $\bar{\Sigma}_w$ :  $\bar{\Sigma}_w = \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k}$ .
- $\bar{\boldsymbol{\mu}}$ :  $\bar{\boldsymbol{\mu}} = \frac{\sum_{i=1}^{k+1} \mathbf{x}_i}{k+1}$ .
- $\bar{\mathbf{w}}$ :  $\bar{\mathbf{w}} = \frac{\sum_{i=1}^k \mathbf{x}_i y_i}{k}$ .

- $\tilde{w}$ : the mean of  $w$  in the task posterior, *i.e.*, the predicted function by Bayes-optimal next-token predictor.  $\mathcal{F}^*(S_k \oplus x^{(k+1)}) = \langle x_{k+1}, \tilde{w} \rangle = \left\langle x_{k+1}, \sum_{m=1}^M \tilde{\pi}_m \tilde{w}_m \right\rangle$ .
- $c_m^\mu$  and  $c_m^w$ : parts of the re-weighting coefficient of Component Re-weighting.
- $\Psi_\mu(\alpha, \beta)$  and  $\Psi_w(\alpha, \beta)$ : functions to help analyze the phenomenon of Component Re-weighting.
- $r(\alpha, \beta)$ : the ratio of the mixture weight  $\tilde{\pi}_\alpha$  of  $\tilde{T}_\alpha$  over the mixture weight  $\tilde{\pi}_\beta$  of  $\tilde{T}_\beta$ .
- $\lambda_d(\mathbf{A})$ : the  $d^{\text{th}}$  largest eigenvalue of matrix  $\mathbf{A}$ . In this paper  $\mathbf{A} \in \mathbb{R}^{d \times d}$ , thus  $\lambda_d(\mathbf{A})$  represents the smallest eigenvalue of matrix  $\mathbf{A}$ .
- $\lambda_1(\mathbf{A})$ : the 1<sup>st</sup>, the largest eigenvalue of matrix  $\mathbf{A}$ .
- $y_{k+1}^*$ : the label of learning the function  $w^*$ .  $y_{k+1}^* = \langle x_{k+1}, w^* \rangle$ .

#### Notations introduced in Sec. 2.4:

- The L2 loss of ICL learning to learn the function  $w^*$ .  $\mathcal{L}_k^* = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^*)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle x_{k+1}, w^* \rangle)^2$ .

#### Notations introduced in Sec. 2.5:

- $d_\mu^2$ :  $\forall \beta \neq \alpha, \|\mu_\beta - \mu^*\|^2 - \|\mu_\alpha - \mu^*\|^2 \geq d_\mu^2$ , the  $\mu$ -margin of any other  $\mu_\beta$  over  $\mu_\alpha$ .
- $d_w^2$ :  $\forall \beta \neq \alpha, \|w_\beta - w^*\|^2 - \|w_\alpha - w^*\|^2 \geq d_w^2$ , the  $w$ -margin of any other  $w_\beta$  over  $w_\alpha$ .
- $u_w^2$ :  $\forall \beta \neq \alpha, \tau_x^2 \|w_\beta - w^*\|^2 - (1 + \tau_x^2) \|w_\alpha - w^*\|^2 \geq \tau_x^2 u_w^2$ , the weighted  $w$ -margin of any other  $w_\beta$  over  $w_\alpha$ .

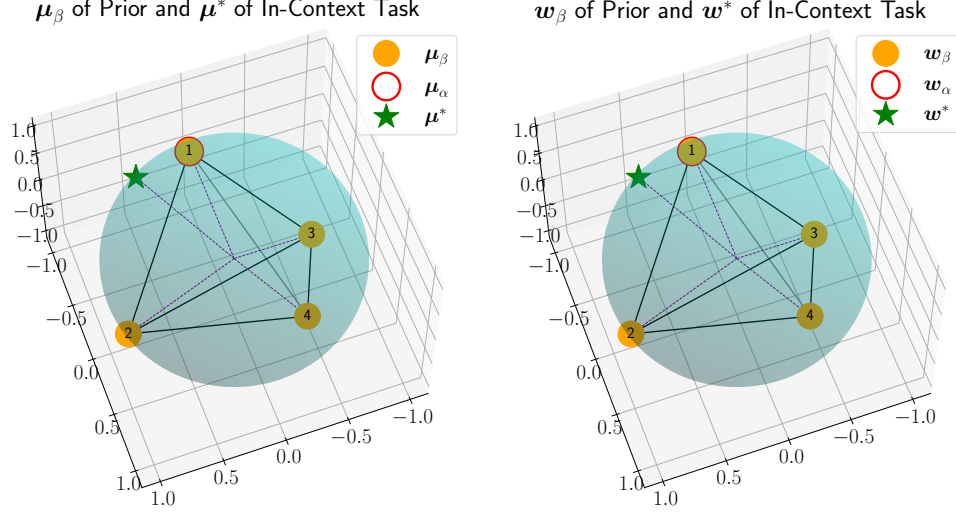


Figure A.1: **Visualization of the tetrahedron setting.** The figure shows the pretraining prior centers and the in-context task. For  $\beta \in \{1, 2, 3, 4\}$ ,  $(\mu_\beta, w_\beta)$  is a mixture component center in the prior.  $(\mu_\alpha, w_\alpha)$  for  $\alpha = 1$  (numbers are noted in the center of circles) is the center of the target task for ICL with biased labels, while  $(\mu^*, w^*)$  is the in-context task. The dotted purple lines highlight the distance of 1 from the origin  $(0, 0, 0)$  to any point denoted by  $\mu$  or  $w$ .

- $y_{k+1}^\alpha$ : the label of retrieving the function  $w_\alpha$ .  $y_{k+1}^\alpha = \langle x_{k+1}, w_\alpha \rangle$ .
- The L2 loss of ICL learning to retrieve the function  $w_\alpha$  of the pretraining prior center  $\alpha$ .  $\mathcal{L}_k^\alpha = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^\alpha)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle x_{k+1}, w_\alpha \rangle)^2$ .

## A.2 Prior Examples

This section outlines our configurations of prior settings in numerical computations and preliminary Transformer experiments, focusing on the geometrical arrangement of the centers in the priors. Specifically, we detail the configurations where

the centers form shapes of 3-dimensional regular polyhedra in Sec. A.2.1, extend to configurations in  $d$ -dimensional spaces in Sec. A.2.2, and discuss a unique setup related to the early ascent phenomenon in Sec. A.2.3.

### A.2.1 Regular Polyhedrons

Taking into account the centers of the mixture components from the pretraining prior, which manifest as distinct points forming the vertices of various shapes, we examine 3-dimensional regular polyhedrons. These include tetrahedron (4 vertices/centers), octahedron (6 vertices/centers), hexahedron (8 vertices/centers), icosahedron (12 vertices/centers), and dodecahedron (20 vertices/centers), listed with increasing density of the centers on a sphere.

The configuration of a regular polyhedron with  $M$  centers is established in accordance with the parameters outlined in Assumption 2, as detailed below:

- Dimension  $d = 3$ , the number of mixture components equals to  $M$ ;
- The centers of mixture components form a regular polyhedron with  $M$  vertices;
- All components' mixture weights are the same,  $\pi_m = 1/M$ , and  $\mu_m = w_m$ , for all  $m \in [M]$ ;
- For noises of  $x$  and  $y$ , we have  $\sigma_x = \sigma_y = 1$ , and  $\tau_x = 1$ ;
- For noises of  $\mu$  and  $w$ , we have  $\sigma_\mu = \sigma_w = 0.25$  if not specified;
- For the in-context task,  $\mu^* = \frac{2\mu_1 + \mu_2}{\|2\mu_1 + \mu_2\|}$  and  $w^* = \frac{2w_1 + w_2}{\|2w_1 + w_2\|}$  if not specified, where  $\mu_2$  is one of the the closest centers to  $\mu_1$ .

We mainly use the **tetrahedron** setting in the paper. Therefore, we further visualize the setting and note down the parameters. The 3D visualization of mixture component centers in the prior and the in-context task are shown in Fig. A.1. The parameters are noted as follows:

- Dimension  $d = 3$ , number of mixture components  $M = 4$ ;
- The centers of topics form a tetrahedron as shown in Fig. A.1.  $\mu_1 = w_1 = [0, 0, -1]^\top$ ,  $\mu_2 = w_2 = [\sqrt{\frac{8}{9}}, 0, \frac{1}{3}]^\top$ ,  $\mu_3 = w_3 = [-\sqrt{\frac{2}{9}}, +\sqrt{\frac{2}{3}}, \frac{1}{3}]^\top$ , and  $\mu_4 = w_4 = [-\sqrt{\frac{2}{9}}, -\sqrt{\frac{2}{3}}, \frac{1}{3}]^\top$ ;
- All components' mixture weights are the same,  $\pi_m = 1/4$ , and  $\mu_m = w_m$ , for all  $m \in \{1, 2, 3, 4\}$ ;
- For noise of  $x$  and  $y$ , we have  $\sigma_x = \sigma_y = 1$ , and  $\tau_x = 1$ ;
- For noises of  $\mu$  and  $w$ , we have  $\sigma_\mu = \sigma_w = 0.25$  if not specified;
- For in-context task, we have  $\mu^* = \frac{2\mu_1 + \mu_2 + 0.2\mu_3}{\|2\mu_1 + \mu_2 + 0.2\mu_3\|}$  and  $w^* = \frac{2w_1 + w_2 + 0.2w_3}{\|2w_1 + w_2 + 0.2w_3\|}$ . We slightly shift the in-context task  $(\mu^*, w^*)$  towards  $(\mu_3, w_3)$  for visualization purposes, to make  $m = 3$  and  $m = 4$  produce slightly different curves.

### A.2.2 $d$ -Dimensional Examples

We consider  $d$ -dimensional examples with  $d$  centers for  $d \in \{2, 4, 8, 16, 32\}$ . A  $d$ -dimensional example with  $d$  vertices is parametered as follows:

- Dimension equals to  $d$ , number of mixture component  $M = d$ ;

- For all  $m \in [M]$ ,  $\boldsymbol{\mu}_m = \mathbf{e}_m$  and  $\boldsymbol{\mu}_{m,i} = \begin{cases} 1 & \text{if } i = m \\ 0 & \text{if } i \neq m \end{cases}$ , i.e.,  $\boldsymbol{\mu}_m$  is the  $m^{\text{th}}$  vector in the standard basis of  $\mathbb{R}^m$ , characterized by having all elements equal to 0 except for the  $m^{\text{th}}$  element, which is 1.
- All components' mixture weights are the same,  $\pi_m = 1/d$ , and  $\boldsymbol{\mu}_m = \mathbf{w}_m$ , for all  $m \in [M]$ ;
- For noise of  $\mathbf{x}$  and  $\mathbf{y}$ , we have  $\sigma_x = \sigma_y = 1$ , and  $\tau_x = 1$ ;
- For noises of  $\boldsymbol{\mu}$  and  $\mathbf{w}$ , we have  $\sigma_\mu = \sigma_w = 0.25$ ;
- For the in-context task, we have  $\boldsymbol{\mu}^* = \frac{2\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{\|2\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2\|}$  and  $\mathbf{w}^* = \frac{2\mathbf{w}_1 + \mathbf{w}_2}{\|2\mathbf{w}_1 + \mathbf{w}_2\|}$ .

### A.2.3 Early Ascent Examples

Table A.1 outlines the prior configuration used to produce the early ascent phenomenon, where the in-context task is designed with a distribution of  $\mathbf{x}$  close to a misleading task. The full results are shown in Fig. A.2.

## A.3 Coarse Upper Bound for ICL Risk

The following theorem shows a coarse upper bound of the ICL risk parallel to Theorem 3:

Table A.1: **Prior settings for early ascent.** The pretraining task prior comprises two components for one dimension and three for two or more dimensions. ICL aims to predict following the in-context function  $\mathbf{w}^*$ , equivalent to prior center 2’s function  $\mathbf{w}_2$  ( $\mathbf{w}^* = \mathbf{w}_2$ ). The in-context task is characterized by having a closer  $\mathbf{x}$  distribution to the task of prior center 1 but a closer  $\mathbf{x} \rightarrow y$  mapping to the prior center 2. The parameters for all cases are set to  $\sigma_\mu = \sigma_w = 0.05$ ,  $\sigma_x = \tau_x = 1$ , and  $\sigma_y = 2$ . Refer to Fig. A.2b for visualization of the prior centers under dimension  $d \in \{1, 2, 3\}$ .

| Case       | Component /Task | Mixture Weight | $\mu$                              | $\mathbf{w}$                              |
|------------|-----------------|----------------|------------------------------------|-------------------------------------------|
| $d = 1$    | Component 1     | $1/2$          | $\mu_1 = [+1]$                     | $\mathbf{w}_1 = [-1]$                     |
|            | Component 2     | $1/2$          | $\mu_2 = [-1]$                     | $\mathbf{w}_2 = [+1]$                     |
|            | Component 3     | /              | /                                  | /                                         |
|            | In-context Task | /              | $\mu^* = [+1]$                     | $\mathbf{w}^* = [+1]$                     |
| $d = 2$    | Component 1     | $1/3$          | $\mu_1 = [+1, +1]$                 | $\mathbf{w}_1 = [-1, -1]$                 |
|            | Component 2     | $1/3$          | $\mu_2 = [-1, -1]$                 | $\mathbf{w}_2 = [+1, +1]$                 |
|            | Component 3     | $1/3$          | $\mu_3 = [+1, -1]$                 | $\mathbf{w}_3 = [-1, +1]$                 |
|            | In-context Task | /              | $\mu^* = [+1, +1]$                 | $\mathbf{w}^* = [+1, +1]$                 |
| $d \geq 2$ | Component 1     | $1/3$          | $\mu_1 = [+1] + [+1] \times (d-1)$ | $\mathbf{w}_1 = [-1] + [-1] \times (d-1)$ |
|            | Component 2     | $1/3$          | $\mu_2 = [-1] + [-1] \times (d-1)$ | $\mathbf{w}_2 = [+1] + [+1] \times (d-1)$ |
|            | Component 3     | $1/3$          | $\mu_3 = [+1] + [-1] \times (d-1)$ | $\mathbf{w}_3 = [-1] + [+1] \times (d-1)$ |
|            | In-context Task | /              | $\mu^* = [+1] \times d$            | $\mathbf{w}^* = [+1] \times d$            |

**Theorem 7** (Coarse Upper Bound for ICL Risk). *Consider a next-token predictor attaining the optimal pretraining risk. As  $k \rightarrow \infty$ , the ICL risk is upper bounded by:*

$$\mathbb{E}_{S_k \oplus x^{(k+1)}}[\mathcal{L}_k^*] < \frac{4(1 + d\tau_x^2)}{\tau_x^4 \delta_w^2 k^2} + O(k^{\delta - \frac{5}{2}}),$$

where  $\mathcal{L}_k^* = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^*)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle)^2$  and  $\delta$  is an arbitrarily small positive constant. See Appendix A.12 for proof details. The upper bound decreases as the square of the inverse of  $k$ . Notice there is no noise for  $y$  labels of in-context examples under our setting, which leads to a faster decay rate than standard  $1/k$  for ridge regression [109].

The notations  $\delta_w$  and  $k$  are colored for easier observation.

We further compare the risk  $\mathbb{E}_{S_k \oplus x^{(k+1)}}[\mathcal{L}_k^*]$  and the risk under ridge regression with L2 regularization parameter equal to  $10^{-6}$ , where the same  $k$  samples without label noises are used as in-context examples for ICL and training samples for ridge regression. Fig. A.3 shows the experiment results. Under certain settings for the task prior  $\mathcal{D}_{\mu, w}$ , when the task prior has low task variances, ICL performs better than ridge regression with a fixed regularization parameter under small  $k$ .

## A.4 Transformer Performance in Approximating Bayesian Inference

We examine if a Transformer network pretrained on samples generated from our pretraining data generative model matches the performance of Bayesian inference. We consider three factors of the task prior in our experiment: *prior task noises*, *number of components*, and *feature dimension*. For scalar  $y$ , we transform it to a  $d$ -dimensional vector  $[y, 0, \dots, 0]$ . Thus,  $S_k \oplus x^{(k+1)}$  forms a  $(2k + 1) \times d$  matrix, comprising  $x_{k+1}$  and  $k$  pairs of  $(x_i, y_i)$ .

**Experiment Setting.** We conduct experiments based on the module GPT2Model from the package Transformers supported by HuggingFace<sup>1</sup>. We use a 10-layer, 8-head Transformer decoder with 1024-dimensional feedforward layers, and the input dimension is set to  $d$ , equal to the dimension of  $x$ . We train the model over three epochs, each consisting of 10,000 batches, with every batch containing 256

---

<sup>1</sup><https://huggingface.co/>

samples. We use AdamW [67] as the optimizer with weight decay as 0.00001 and set the learning rate to 0.00001.

**Experiment Results.** Fig. A.4, A.5, and A.6 show the experimental results, where  $\hat{\mathcal{F}}$  denotes the prediction of the Transformer network,  $\mathcal{F}^*$  denotes the prediction of Bayesian inference, and  $y_{k+1}^* = \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle$  is the label of learning the in-context function. In Fig. A.4, we consider the tetrahedron setting (see Appendix A.2.1 for setting details) under varied task noises ( $\delta_\mu = \delta_w \in \{1/256, 1/64, 1/16, 1/4, 1\}$ ). In Fig. A.5, we consider settings of regular shapes (see Appendix A.2.1 for setting details) with different numbers of vertices/components ( $M \in \{4, 6, 8, 12, 20\}$ ). In Fig. A.6, we consider settings with varied dimensions (see Appendix A.2.2 for setting details,  $d \in \{2, 4, 8, 16, 32\}$ ). We observe that the trained Transformer network can approximate the Bayes-optimal predictor under varied settings, and the larger the number of dimensions and the number of mixture components, the harder it is for the Transformer network to approximate Bayesian prediction.

## A.5 Additional Information for Bounded Efficacy in GPT-4

### A.5.1 Experimental Setting

Table A.2 introduces the experiment setting of GPT-4, including the system message, the prompt, the in-context task, the “biased +” task, and the “addition (+)” task. Designating the “biased +” task as the in-context task, *i.e.*,  $c_i = a_i + b_i + 1$ , we

Table A.2: **Setup of bounded efficacy.** Experiment setting to reveal the bounded efficacy phenomenon of biased-label ICL in GPT-4.

| Setting                                                       | Description                                                                                                                                                                                                                                                                |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LLM                                                           | GPT-4                                                                                                                                                                                                                                                                      |
| System Message                                                | You are a mathematician. Consider the following math problem and follow the exact instruction.                                                                                                                                                                             |
| Prompt                                                        | <p>You are given examples. Each example has two integers as input and one integer as output. Please provide an answer for the last problems in the math exercise:</p> $a_1(?)b_1=c_1$ <p>...</p> $a_k(?)b_k=c_k$ $a_{k+1}(?)b_{k+1}=$ <p>Provide your answer directly.</p> |
| In-Context Task                                               | $a_i$ and $b_i$ are uniformly sampled from $[10, 99]$ , and $c_i = a_i + b_i + 1$ .                                                                                                                                                                                        |
| Goal of Learning the “biased +” Task with True Labels         | Aiming to learn the “biased +” task, $a(?)b=(a+b+1)$ , with in-context examples following the same “biased +” task, $a(?)b=(a+b+1)$ .                                                                                                                                      |
| Goal of Retrieving the “addition (+)” Task with Biased Labels | Aiming to retrieve the “addition (+)” task, $a(?)b=(a+b)$ . However, the in-context examples are provided with a slightly different task “biased +”, $a(?)b=(a+b+1)$ .                                                                                                     |

measure the performances on two goals, including learning the “biased +” task and retrieving the “addition (+)” task.

### A.5.2 Additional Results

This section collects four pairs of prompts and predictions for  $k = 0, 2, 8$  in Tables A.3, A.4, and A.5. The results show that ICL with biased labels will initially retrieve a commonsense pretraining task due to task retrieval, and finally learn the

Table A.3: **Zero in-context example** ( $k = 0$ ). Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt.

|         |                                                                                                                                                                                                         |                                                                                                                               |                                                                                                                                                                                                                   |                        |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Prompt  | ...<br>51(?)36=<br>...                                                                                                                                                                                  | ...<br>27(?)15=<br>...                                                                                                        | ...<br>76(?)82=<br>...                                                                                                                                                                                            | ...<br>55(?)15=<br>... |
| Results | Without knowing the operation or rule that connects the two input integers to the output integer in the examples, it's impossible to provide a correct answer. Please provide the examples or the rule. | Sorry, but your question is not clear. Could you please provide more information about the operation between the two numbers? | Your question seems to be missing some information. Could you please provide the examples you mentioned? They are necessary to understand the relationship between the two input integers and the output integer. | <b>70</b>              |

Table A.4: **Two in-context examples** ( $k = 2$ ). Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt.

|         |                                                     |                                                      |                                                    |                                                      |
|---------|-----------------------------------------------------|------------------------------------------------------|----------------------------------------------------|------------------------------------------------------|
| Prompt  | ...<br>73(?)80=154<br>59(?)22=82<br>54(?)97=<br>... | ...<br>48(?)73=122<br>78(?)80=159<br>21(?)33=<br>... | ...<br>21(?)28=50<br>69(?)29=99<br>47(?)10=<br>... | ...<br>94(?)43=138<br>98(?)70=169<br>96(?)41=<br>... |
| Results | <b>151</b>                                          | <b>54</b>                                            | <b>57</b>                                          | 187                                                  |

in-context task because of task learning.

Table A.5: **Eight in-context examples** ( $k = 8$ ). Prediction is colored **red** if it is correct for task retrieval ( $a(?)b = (a + b)$ ), and colored **blue** if it is correct for task learning ( $a(?)b = (a + b + 1)$ ). “...” denotes the hidden part of the prompt. Please refer to Table A.2 for the whole prompt.

|         |             |             |             |             |
|---------|-------------|-------------|-------------|-------------|
|         | ...         | ...         | ...         | ...         |
|         | 37(?)70=108 | 60(?)76=137 | 66(?)40=107 | 68(?)88=157 |
|         | 41(?)18=60  | 69(?)26=96  | 46(?)81=128 | 34(?)18=53  |
|         | 19(?)12=32  | 72(?)85=158 | 63(?)31=95  | 70(?)70=141 |
|         | 82(?)67=150 | 39(?)10=50  | 41(?)24=66  | 13(?)35=49  |
| Prompt  | 42(?)13=56  | 50(?)47=98  | 70(?)43=114 | 52(?)50=103 |
|         | 26(?)41=68  | 19(?)63=83  | 89(?)84=174 | 72(?)32=105 |
|         | 80(?)39=120 | 45(?)95=141 | 76(?)82=159 | 98(?)82=181 |
|         | 58(?)23=82  | 69(?)41=111 | 46(?)28=75  | 55(?)51=107 |
|         | 40(?)90=    | 81(?)36=    | 49(?)46=    | 50(?)31=    |
|         | ...         | ...         | ...         | ...         |
| Results | <b>130</b>  | <b>118</b>  | <b>96</b>   | <b>82</b>   |

## A.6 Bounded Efficacy in Zero-shot ICL

This section introduces the experiment setting of Fig. 2.6. We start by introducing the experiment results in Fig. A.7 copied and pasted from the work of Min et al. [73]. While our theory shows the bounded efficacy phenomenon for ICL with non-informative labels (Lemma 6), Fig. A.7 seems to imply a conflict phenomenon. Thus, we further extend the number of in-context examples in Fig. A.7 left. The classification task adopts five datasets including (i) glue-mrpc [23], (ii) glue-rte [20], (iii) tweet\_eval-hate [5], (iv) sick [71], and (v) poem-sentiment [99]. We use the GitHub code<sup>2</sup> released by Min et al. [73] to generate the same data and evaluate LLMs with a larger context length capacity aiming at a larger number of

<sup>2</sup><https://github.com/Alrope123/rethinking-demonstrations>

in-context examples. We selected Mistral 7B (32768), Mixtral 8×7B (32768), Llama2 13B (4096), Llama2 70B (4096), and GPT-4 (8192) for our experiments, with the integers in parentheses indicating the maximum context length for each model. We perform inference on large models with 8 H100 with the package vllm<sup>3</sup>.

## A.7 The Derivation of Posterior

This section provides detailed derivations for Lemma 1. We begin by showing the posterior is potentially still a Gaussian mixture in Sec. A.7.1. Then, in Sec. A.7.2, we show how Eq. A.1 is proportion to Eq. A.2, which is precisely a Gaussian mixture.

### A.7.1 Prior to Posterior

We start by showing the posterior is potentially still a Gaussian mixture. For fixed  $S_k \oplus x^{(k+1)}$ :

$$\begin{aligned}
& P(\boldsymbol{\mu}, \boldsymbol{w} | S_k \oplus x^{(k+1)}) \\
& \propto P(\boldsymbol{\mu}, \boldsymbol{w} | S_k \oplus x^{(k+1)}) P(S_k \oplus x^{(k+1)}) \\
& = P(\boldsymbol{\mu}, \boldsymbol{w}, S_k \oplus x^{(k+1)}) \\
& = P(\boldsymbol{\mu}, \boldsymbol{w}) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \\
& = \left( \sum_{m=1}^M \pi_m P(\boldsymbol{\mu}, \boldsymbol{w} | T_m) \right) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \\
& = \sum_{m=1}^M \pi_m P(\boldsymbol{\mu}, \boldsymbol{w} | T_m) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \tag{A.1}
\end{aligned}$$

---

<sup>3</sup><https://docs.vllm.ai/en/latest/>

$$\propto \sum_{m=1}^M \tilde{\pi}_m P(\boldsymbol{\mu}, \mathbf{w} | \tilde{T}_m). \quad (\text{A.2})$$

We give the derivation from Eq. A.1 to Eq. A.2 in the next section.

### A.7.2 Closed-Form Solution from Eq. A.1 to Eq. A.2

We analyze each component (indicated by a specific  $m$ ) in Eq. A.1. Given fixed  $S_k \oplus x^{(k+1)}$ , for all  $m \in [M]$  and all  $(\boldsymbol{\mu}, \mathbf{w})$ , we have:

$$\begin{aligned} & \log(P(\boldsymbol{\mu}, \mathbf{w} | T_m) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \mathbf{w})) \\ &= -\frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}\|^2}{2\sigma_\mu^2} - \frac{\|\mathbf{w}_m - \mathbf{w}\|^2}{2\sigma_w^2} - \frac{\sum_{i=1}^{k+1} \|\boldsymbol{\mu} - \mathbf{x}_i\|^2}{2\sigma_x^2} - \frac{\sum_{i=1}^k \|\mathbf{x}_i^\top \mathbf{w} - y_i\|^2}{2\sigma_y^2} \\ & \quad + \log\left(\frac{(2\pi)^{-d/2}}{\sigma_\mu^d}\right) + \log\left(\frac{(2\pi)^{-d/2}}{\sigma_w^d}\right) + (k+1) \log\left(\frac{(2\pi)^{-d/2}}{\sigma_x^d}\right) + k \log\left(\frac{(2\pi)^{-1/2}}{\sigma_y}\right) \\ & (\text{Let } C_3 = \log\left(\frac{(2\pi)^{-d/2}}{\sigma_\mu^d}\right) + \log\left(\frac{(2\pi)^{-d/2}}{\sigma_w^d}\right) \\ & \quad + (k+1) \log\left(\frac{(2\pi)^{-d/2}}{\sigma_x^d}\right) + k \log\left(\frac{(2\pi)^{-1/2}}{\sigma_y}\right).) \\ &= C_3 - \frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}\|^2}{2\sigma_\mu^2} - \frac{\|\mathbf{w}_m - \mathbf{w}\|^2}{2\sigma_w^2} - \frac{\sum_{i=1}^{k+1} \|\boldsymbol{\mu} - \mathbf{x}_i\|^2}{2\sigma_x^2} - \frac{\sum_{i=1}^k \|\mathbf{x}_i^\top \mathbf{w} - y_i\|^2}{2\sigma_y^2} \\ &= C_3 - \left(\frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}\|^2}{2\sigma_\mu^2} + \frac{\sum_{i=1}^{k+1} \|\boldsymbol{\mu} - \mathbf{x}_i\|^2}{2\sigma_x^2}\right) - \left(\frac{\|\mathbf{w}_m - \mathbf{w}\|^2}{2\sigma_w^2} + \frac{\sum_{i=1}^k \|\mathbf{x}_i^\top \mathbf{w} - y_i\|^2}{2\sigma_y^2}\right) \\ & (\text{Let } \delta_\mu = \frac{\sigma_\mu^2}{\sigma_x^2} \text{ and } \delta_w = \frac{\sigma_w^2}{\sigma_y^2}.) \\ &= C_3 - \frac{1}{2\sigma_\mu^2} \left( (\|\boldsymbol{\mu}_m\|^2 - 2\boldsymbol{\mu}_m^\top \boldsymbol{\mu} + \|\boldsymbol{\mu}\|^2) + \delta_\mu \left( (k+1)\|\boldsymbol{\mu}\|^2 - 2\boldsymbol{\mu}^\top \sum_{i=1}^{k+1} \mathbf{x}_i + \sum_{i=1}^{k+1} \|\mathbf{x}_i\|^2 \right) \right) \\ & \quad - \frac{1}{2\sigma_\mu^2} \left( (\|\mathbf{w}_m\|^2 - 2\mathbf{w}_m^\top \mathbf{w} + \|\mathbf{w}\|^2) + \delta_w \left( \sum_{i=1}^k \mathbf{w}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w} - 2\mathbf{w}^\top \sum_{i=1}^k \mathbf{x}_i y_i + \sum_{i=1}^k y_i^2 \right) \right) \\ &= C_3 - \frac{1}{2\sigma_\mu^2} \left( \|\boldsymbol{\mu}_m\|^2 + (1 + (k+1)\delta_\mu)\|\boldsymbol{\mu}\|^2 - 2\boldsymbol{\mu} \left( \boldsymbol{\mu}_m + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i \right) + \delta_\mu \sum_{i=1}^{k+1} \|\mathbf{x}_i\|^2 \right) \end{aligned}$$

$$\begin{aligned}
& -\frac{1}{2\sigma_w^2} \left( \|\mathbf{w}_m\|^2 + \mathbf{w}^\top \left( \mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w} - 2\mathbf{w} \left( \mathbf{w}_m + \delta_w \sum_{i=1}^k \mathbf{x}_i y_i \right) + \delta_w \sum_{i=1}^k y_i^2 \right) \\
& (\text{Let } C_4 = C_3 - \frac{\delta_\mu}{2\sigma_\mu^2} \sum_{i=1}^{k+1} \|\mathbf{x}_i\|^2 - \frac{\delta_w}{2\sigma_w^2} \sum_{i=1}^k y_i^2.) \\
& = C_4 - \frac{1}{2\sigma_\mu^2} \left( \|\boldsymbol{\mu}_m\|^2 + (1 + (k+1)\delta_\mu) \|\boldsymbol{\mu}\|^2 - 2\boldsymbol{\mu}^\top \left( \boldsymbol{\mu}_m + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i \right) \right) \\
& \quad - \frac{1}{2\sigma_w^2} \left( \|\mathbf{w}_m\|^2 + \mathbf{w}^\top \left( \mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w} - 2\mathbf{w} \left( \mathbf{w}_m + \delta_w \sum_{i=1}^k \mathbf{x}_i y_i \right) \right) \\
& (\text{Let } \bar{\boldsymbol{\Sigma}}_\mu = \mathbf{I} \text{ and } \bar{\boldsymbol{\Sigma}}_w = \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k}.) \\
& = C_4 - \frac{1}{2\sigma_\mu^2} \left( \|\boldsymbol{\mu}_m\|^2 + \|\boldsymbol{\mu}\|_{\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 - 2\boldsymbol{\mu}^\top \left( \boldsymbol{\mu}_m + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i \right) \right) \\
& \quad - \frac{1}{2\sigma_w^2} \left( \|\mathbf{w}_m\|^2 + \|\mathbf{w}\|_{\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w}^2 - 2\mathbf{w}^\top \left( \mathbf{w}_m + \delta_w \sum_{i=1}^k \mathbf{x}_i y_i \right) \right) \\
& (\text{Let } \bar{\boldsymbol{\mu}} = \sum_{i=1}^{k+1} \mathbf{x}_i \text{ and } \bar{w} = \frac{\sum_{i=1}^k \mathbf{x}_i y_i}{k}.) \\
& = C_4 - \frac{1}{2\sigma_\mu^2} (\|\boldsymbol{\mu}_m\|^2 + \|\boldsymbol{\mu}\|_{\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 - 2\boldsymbol{\mu}^\top (\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}})) \\
& \quad - \frac{1}{2\sigma_w^2} (\|\mathbf{w}_m\|^2 + \|\mathbf{w}\|_{\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w}^2 - 2\mathbf{w}^\top (\mathbf{w}_m + k\delta_w \bar{w})) \\
& (\text{Let } \Delta_\mu = (k+1)\delta_\mu \text{ and } \Delta_w = k\delta_w.) \\
& = C_4 - \frac{1}{2\sigma_\mu^2} (\|\boldsymbol{\mu}_m\|^2 + \|\boldsymbol{\mu}\|_{\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 - 2\boldsymbol{\mu}^\top (\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}})) \\
& \quad - \frac{1}{2\sigma_w^2} (\|\mathbf{w}_m\|^2 + \|\mathbf{w}\|_{\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w}^2 - 2\mathbf{w}^\top (\mathbf{w}_m + \Delta_w \bar{w})) \\
& = C_4 - \left( \|\boldsymbol{\mu}_m\|^2 + \left( \|\boldsymbol{\mu}\|_{\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 - 2\boldsymbol{\mu}^\top (\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}}) + \|\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}}\|_{(\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2 \right) - \|\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}}\|_{(\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2 \right) / 2\sigma_\mu^2 \\
& \quad - \left( \|\mathbf{w}_m\|^2 + \left( \|\mathbf{w}\|_{\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w}^2 - 2\mathbf{w}^\top (\mathbf{w}_m + \Delta_w \bar{w}) + \|\mathbf{w}_m + \Delta_w \bar{w}\|_{(\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 \right) - \|\mathbf{w}_m + \Delta_w \bar{w}\|_{(\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 \right) / 2\sigma_w^2 \\
& = C_4 - \frac{1}{2\sigma_\mu^2} \left( \left( \|\boldsymbol{\mu}_m\|^2 - \|\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}}\|_{(\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2 \right) + \|\boldsymbol{\mu} - (\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1} (\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}})\|_{\mathbf{I} + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 \right) \\
& \quad - \frac{1}{2\sigma_w^2} \left( \left( \|\mathbf{w}_m\|^2 - \|\mathbf{w}_m + \Delta_w \bar{w}\|_{(\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 \right) + \|\mathbf{w} - (\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1} (\mathbf{w}_m + \Delta_w \bar{w})\|_{\mathbf{I} + \Delta_w \bar{\boldsymbol{\Sigma}}_w}^2 \right).
\end{aligned}$$

Notice  $C_4$  is independent to  $m$ ,  $\boldsymbol{\mu}$ , and  $\boldsymbol{w}$ , thus we have:

$$\begin{aligned}
& P(\boldsymbol{\mu}, \boldsymbol{w} | T_m) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \\
& \propto \exp \left( -\frac{1}{2\sigma_\mu^2} \left( \left( \|\boldsymbol{\mu}_m\|^2 - \|\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}}\|_{(I + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2 \right) + \|\boldsymbol{\mu} - (I + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}(\boldsymbol{\mu}_m + \Delta_\mu \bar{\boldsymbol{\mu}})\|_{I + \Delta_\mu \bar{\boldsymbol{\Sigma}}_\mu}^2 \right) \right) \\
& \quad \cdot \exp \left( -\frac{1}{2\sigma_w^2} \left( \left( \|\boldsymbol{w}_m\|^2 - \|\boldsymbol{w}_m + \Delta_w \bar{\boldsymbol{w}}\|_{(I + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 \right) + \|\boldsymbol{w} - (I + \Delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\boldsymbol{w}_m + \Delta_w \bar{\boldsymbol{w}})\|_{I + \Delta_w \bar{\boldsymbol{\Sigma}}_w}^2 \right) \right) \\
& \propto \underbrace{\exp \left( -\frac{\|\boldsymbol{\mu}_m\|^2 - \|\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}\|_{(I + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}^2}{2\sigma_\mu^2} \right)}_{c_m^\mu} \underbrace{\exp \left( -\frac{\|\boldsymbol{w}_m\|^2 - \|\boldsymbol{w}_m + k\delta_w \bar{\boldsymbol{w}}\|_{(I + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2}{2\sigma_w^2} \right)}_{c_m^w} \\
& \quad \cdot \mathcal{N}(\boldsymbol{\mu} | (I + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}(\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}), \sigma_\mu^2(I + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}) \\
& \quad \cdot \mathcal{N}(\boldsymbol{w} | (I + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\boldsymbol{w}_m + k\delta_w \bar{\boldsymbol{w}}), \sigma_w^2(I + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}).
\end{aligned}$$

By defining  $P(\boldsymbol{\mu}, \boldsymbol{w} | \tilde{T}) = \mathcal{N}(\boldsymbol{\mu} | (I + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}(\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}), \sigma_\mu^2(I + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}) \cdot \mathcal{N}(\boldsymbol{w} | (I + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\boldsymbol{w}_m + k\delta_w \bar{\boldsymbol{w}}), \sigma_w^2(I + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1})$  and  $\tilde{\pi}_m = \pi_m c_m^\mu c_m^w$ .

We have:

$$\pi_m P(\boldsymbol{\mu}, \boldsymbol{w} | T_m) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \propto \tilde{\pi}_m P(\boldsymbol{\mu}, \boldsymbol{w} | \tilde{T}_m).$$

Therefore,

$$\sum_{m=1}^M \pi_m P(\boldsymbol{\mu}, \boldsymbol{w} | T_m) P(S_k \oplus x^{(k+1)} | \boldsymbol{\mu}, \boldsymbol{w}) \propto \sum_{m=1}^M \tilde{\pi}_m P(\boldsymbol{\mu}, \boldsymbol{w} | \tilde{T}_m).$$

## A.8 Detailed Analysis of Component Shifting and Re-weighting

### A.8.1 Analysis of Component Re-weighting

This section analyzes the CR effect on  $\tilde{\pi}_\beta$  as  $k$  increases. We focus on whether  $\tilde{\pi}_\alpha$  of  $\tilde{T}_\alpha$  surpasses  $\tilde{\pi}_\beta$  of any other  $\tilde{T}_\beta$  with  $\beta \neq \alpha$ , where  $\alpha$  is the index of the closest prior center to the in-context task as described in Assumption 3. We assess this via the ratio  $r(\alpha, \beta)$  of  $\tilde{\pi}_\alpha$  to  $\tilde{\pi}_\beta$ :

$$r(\alpha, \beta) = \frac{\tilde{\pi}_\alpha}{\tilde{\pi}_\beta} = \frac{\pi_\alpha C_0 c_\alpha^\mu c_\alpha^w}{\pi_\beta C_0 c_\beta^\mu c_\beta^w} = \frac{\pi_\alpha}{\pi_\beta} \exp(\Psi_\mu(\alpha, \beta) + \Psi_w(\alpha, \beta)), \quad (\text{A.3})$$

where we define two functions  $\Psi_\mu(\alpha, \beta) = \log(c_\alpha^\mu/c_\beta^\mu)$  and  $\Psi_w(\alpha, \beta) = \log(c_\alpha^w/c_\beta^w)$  to facilitate the analyses of how  $r(\alpha, \beta)$  changes with increasing  $k$ .

**Analysis of  $\Psi_\mu(\alpha, \beta)$ .** We further simplify the function  $\Psi_\mu(\alpha, \beta)$  as follows:

$$\Psi_\mu(\alpha, \beta) = \left( \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 - \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2 \right) / (2\sigma_x^2(1 + (k+1)\delta_\mu)). \quad (\text{A.4})$$

(See Appendix A.8.3 for derivation.) Since  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ , choosing  $\boldsymbol{\mu}^*$  closer to  $\boldsymbol{\mu}_\alpha$  tends to make  $\Psi_\mu(\alpha, \beta)$  positive and increase faster with increasing  $k$ . However, as  $k$  approaches infinity,  $\Psi_\mu(\alpha, \beta)$  stabilizes rather than increasing infinitely, *i.e.*,  $\lim_{k \rightarrow \infty} \Psi_\mu(\alpha, \beta) = (\|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2) / (2\sigma_\mu^2)$ . The leftmost column of Fig. A.9 shows the numerical computation of  $\Psi_\mu(\alpha, \beta)$  with varied task noises under the tetrahedron setting (see Appendix A.2.1 for setting details). The smaller the value of  $\delta_\mu (= \frac{\sigma_\mu^2}{\sigma_x^2})$  is, the easier for  $\Psi_\mu(\alpha, \beta)$  to increase as  $k$  increases.

Meanwhile, we also have:

$$\lim_{\sigma_\mu \rightarrow 0} \Psi_\mu(\alpha, \beta) = \left( \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 - \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2 \right) / (2\sigma_x^2) \quad (\text{A.5})$$

**Analysis of  $\Psi_w(\alpha, \beta)$ .** We further simplify the function  $\Psi_w(\alpha, \beta)$  as follows:

$$\Psi_w(\alpha, \beta) = (\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2) / (2\sigma_w^2). \quad (\text{A.6})$$

(See Appendix A.8.3 for derivation.) Since  $k\delta_w \bar{\boldsymbol{\Sigma}}_w$  ( $= \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top$ , see definition of  $\bar{\boldsymbol{\Sigma}}_w$  in Lemma 1) is semi-positive definite, thus choosing  $\mathbf{w}^*$  closer to  $\mathbf{w}_\alpha$  tends to make  $\Psi_w(\alpha, \beta)$  positive and increase faster as  $k$  increases. However, as  $k$  approaches infinity,  $\lim_{k \rightarrow \infty} k\delta_w \bar{\boldsymbol{\Sigma}}_w = \lim_{k \rightarrow \infty} k\delta_w \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} = k\delta_w (\boldsymbol{\mu}^* \boldsymbol{\mu}^{*\top} + \tau_x^2 \mathbf{I})$ . Thus,  $\lim_{k \rightarrow \infty} \mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1} = \mathbf{I}$  and  $\Psi_w(\alpha, \beta)$  stabilizes rather than increasing infinitely, *i.e.*,  $\lim_{k \rightarrow \infty} \Psi_w(\alpha, \beta) = (\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2) / (2\sigma_w^2)$ . The topmost row of Fig. A.9 shows the numerical computation of  $\Psi_w(\alpha, \beta)$  with varied task noises under the tetrahedron setting (see Appendix A.2.1 for setting details). The smaller the value of  $\delta_w$  ( $= \frac{\sigma_w^2}{\sigma_y^2}$ ) is, the easier for  $\Psi_w(\alpha, \beta)$  to increase as  $k$  increases. However, one should note that  $\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \geq \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2$  does not necessarily imply  $\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 \geq \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2$ .

Meanwhile, we also have:

$$\begin{aligned} \lim_{\sigma_w \rightarrow 0} \Psi_w(\alpha, \beta) &= (\|\mathbf{w}_\beta - \mathbf{w}^*\|_{k\delta_w \bar{\boldsymbol{\Sigma}}_w}^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{k\delta_w \bar{\boldsymbol{\Sigma}}_w}^2) / (2\sigma_w^2) \\ &= (\|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|_{k\bar{\boldsymbol{\Sigma}}_w}^2 - \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|_{k\bar{\boldsymbol{\Sigma}}_w}^2) / (2\sigma_y^2) \\ &= \left( \sum_{i=1}^k \|y_i^\beta - y_i^*\|^2 - \sum_{i=1}^k \|y_i^\alpha - y_i^*\|^2 \right) / (2\sigma_y^2), \end{aligned} \quad (\text{A.7})$$

where  $y_i^\beta = \langle \mathbf{x}_i, \mathbf{w}_\beta \rangle$ ,  $y_i^\alpha = \langle \mathbf{x}_i, \mathbf{w}_\alpha \rangle$ , and  $y_i^* = \langle \mathbf{x}_i, \mathbf{w}^* \rangle$ .

Therefore, combine Eqs. A.5 and A.7 and we have:

$$\begin{aligned}
& \lim_{\sigma_\mu, \sigma_w \rightarrow 0} \Psi_\mu(\alpha, \beta) + \Psi_w(\alpha, \beta) \\
&= \frac{\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 - \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2} + \sum_{i=1}^k \left( \frac{\|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 - \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2} \right. \\
&\quad \left. + \frac{\|y_i^\beta - y_i^*\|^2 - \|y_i^\alpha - y_i^*\|^2}{2\sigma_y^2} \right) \quad (\text{A.8})
\end{aligned}$$

**Numerical Computations of Component Re-weighting.** We have seen how noises  $\sigma_\mu$  and  $\sigma_w$  of the task prior affect the values of  $\Psi_\mu$  and  $\Psi_w$  with increasing  $k$ . We further show the numerical computation of  $\tilde{\pi}_\beta$  in the center of Fig. A.9. The figure shows that the smaller  $\delta_\mu$  and  $\delta_w$  are, the larger  $\Psi_\mu(\alpha, \beta)$  and  $\Psi_w(\alpha, \beta)$  will be with increasing  $k$ , and the easier for the mixture component  $\tilde{T}_\alpha$  to dominates in the posterior with an increasing number of in-context examples.

### A.8.2 Analysis of Component Shifting

The Component Shifting effect in Lemma 1 involves shifting the variables  $\tilde{\boldsymbol{\mu}}_m$  and  $\tilde{\mathbf{w}}_m$ :

$$\tilde{\boldsymbol{\mu}}_m = (\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}(\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}), \quad (\text{A.9})$$

$$\tilde{\mathbf{w}}_m = (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m + k\delta_w \bar{\mathbf{w}}). \quad (\text{A.10})$$

The following analyses examine these two variables with increasing  $k$ .

**Analysis of  $\tilde{\boldsymbol{\mu}}_m$ .** We provide the derivation of  $\tilde{\boldsymbol{\mu}}_m$  in Eq. A.9 (see Appendix A.8.4 for details):

$$\tilde{\boldsymbol{\mu}}_m = (\boldsymbol{\mu}_m + k\delta_\mu \bar{\boldsymbol{\mu}})/(1 + (k+1)\delta_\mu). \quad (\text{A.11})$$

Thus, when  $k$  increases,  $\tilde{\boldsymbol{\mu}}_m$  moves close to the value of  $\frac{\sum_{i=1}^k \mathbf{x}_i}{k}$  and  $\lim_{k \rightarrow \infty} \tilde{\boldsymbol{\mu}}_m = \boldsymbol{\mu}^*$ . We also show the numerical computation of the distance between shifted  $\tilde{\boldsymbol{\mu}}_m$  and  $\boldsymbol{\mu}^*$  in the first row of Fig. A.10.

**Analysis of  $\tilde{\boldsymbol{w}}_m$ .** We provide the derivation of  $\tilde{\boldsymbol{w}}_m$  in Eq. A.10 (see Appendix A.8.4 for details):

$$\tilde{\boldsymbol{w}}_m = (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m - \mathbf{w}^*) + \mathbf{w}^*. \quad (\text{A.12})$$

Notice when  $k \rightarrow \infty$ ,  $k\delta_w \bar{\boldsymbol{\Sigma}}_w = k\delta_w \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \rightarrow k\delta_w (\tau_x^2 \mathbf{I} + \mathbf{w}^* \mathbf{w}^{*\top})$ , thus  $\lambda_d(k\delta_w \bar{\boldsymbol{\Sigma}}_w) \rightarrow \infty$ ,  $\lambda_1((\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}) \rightarrow 0$ ,  $\lim_{k \rightarrow \infty} (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m - \mathbf{w}^*) \leq \lim_{k \rightarrow \infty} \lambda_1((\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}) \cdot \|\mathbf{w}_m - \mathbf{w}^*\| = 0$  and  $\lim_{k \rightarrow \infty} \tilde{\boldsymbol{w}}_m = \mathbf{w}^*$ , where  $\lambda_d(\mathbf{A})$  indicates the minimum eigenvalue of  $\mathbf{A}$ . We also show the numerical computed distance between  $\tilde{\boldsymbol{w}}_m$  and  $\mathbf{w}^*$  in the second row of Fig. A.10.

### A.8.3 Derivation Collection of $\Psi_\mu(\alpha, \beta)$ and $\Psi_w(\alpha, \beta)$

This section collects derivations for  $\Psi_\mu(\alpha, \beta)$  and  $\Psi_w(\alpha, \beta)$ . The derivation of  $\Psi_\mu(\alpha, \beta)$  is collected in Sec A.8.3 and the derivation of  $\Psi_w(\alpha, \beta)$  is collected in Sec A.8.3.

### Derivation of $\Psi_\mu(\alpha, \beta)$

This section collects the derivation of  $\Psi_\mu(\alpha, \beta)$  in Eq. A.4 of Sec. A.8.1:

$$\begin{aligned}
\Psi_\mu(\alpha, \beta) &= \log(c_\alpha^\mu / c_\beta^\mu) \\
&= \log \left( \frac{\exp \left( -\frac{\|\boldsymbol{\mu}_\beta\|^2 - \|\boldsymbol{\mu}_\beta + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}\|^2_{(\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}}{2\sigma_\mu^2} \right)}{\exp \left( -\frac{\|\boldsymbol{\mu}_\alpha\|^2 - \|\boldsymbol{\mu}_\alpha + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}\|^2_{(\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}}}{2\sigma_\mu^2} \right)} \right) \\
&= \frac{(1 + (k+1)\delta_\mu) \|\boldsymbol{\mu}_\beta\|^2 - \|\boldsymbol{\mu}_\beta + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&\quad - \frac{(1 + (k+1)\delta_\mu) \|\boldsymbol{\mu}_\alpha\|^2 - \|\boldsymbol{\mu}_\alpha + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&= \frac{-\|\boldsymbol{\mu}_\beta + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} - \frac{-\|\boldsymbol{\mu}_\alpha + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&= \frac{-\|\boldsymbol{\mu}_\beta\|^2 - 2\boldsymbol{\mu}_\beta^\top (\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i) - \|\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&\quad - \frac{-\|\boldsymbol{\mu}_\alpha\|^2 - 2\boldsymbol{\mu}_\alpha^\top (\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i) - \|\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&= \frac{(k+1)\delta_\mu \|\boldsymbol{\mu}_\beta\|^2 - 2\boldsymbol{\mu}_\beta^\top (\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i) + \delta_\mu \sum_{i=1}^{k+1} \|\mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&\quad - \frac{(k+1)\delta_\mu \|\boldsymbol{\mu}_\alpha\|^2 - 2\boldsymbol{\mu}_\alpha^\top (\delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i) + \delta_\mu \sum_{i=1}^{k+1} \|\mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&= \frac{\sum_{i=1}^{k+1} \delta_\mu \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} - \frac{\sum_{i=1}^{k+1} \delta_\mu \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)} \\
&= \frac{\sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 - \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_\mu^2(1 + (k+1)\delta_\mu)}.
\end{aligned}$$

### Derivation of $\Psi_w(\alpha, \beta)$

This section collects the derivation of  $\Psi_w(\alpha, \beta)$  in Eq. A.6 of Sec. A.8.1:

$$\begin{aligned}
& \Psi_w(\alpha, \beta) \\
&= \log(c_\alpha^w / c_\beta^w) \\
&= \log \left( \frac{\exp \left( -\frac{\|\mathbf{w}_\alpha\|^2 - \|\mathbf{w}_\alpha + k\delta_w \bar{\mathbf{w}}\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \right)}{\exp \left( -\frac{\|\mathbf{w}_\beta\|^2 - \|\mathbf{w}_\beta + k\delta_w \bar{\mathbf{w}}\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \right)} \right) \\
&= \frac{\|\mathbf{w}_\beta\|^2 - \|\mathbf{w}_\beta + k\delta_w \bar{\mathbf{w}}\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} - \frac{\|\mathbf{w}_\alpha\|^2 - \|\mathbf{w}_\alpha + k\delta_w \bar{\mathbf{w}}\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \\
& \text{(Note } k\delta_w \bar{\mathbf{w}} = \delta_w \sum_{i=1}^k \mathbf{x}_i y_i = \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}^* = k\delta_w \bar{\Sigma}_w \mathbf{w}^* \text{.)} \\
&= \frac{\|\mathbf{w}_\beta\|^2 - \|\mathbf{w}_\beta + k\delta_w \bar{\Sigma}_w \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} - \frac{\|\mathbf{w}_\alpha\|^2 - \|\mathbf{w}_\alpha + k\delta_w \bar{\Sigma}_w \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \\
&= \frac{\|\mathbf{w}_\beta\|^2 - \|(\mathbf{w}_\beta - \mathbf{w}^*) + (I + k\delta_w \bar{\Sigma}_w) \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \\
&\quad - \frac{\|\mathbf{w}_\alpha\|^2 - \|(\mathbf{w}_\alpha - \mathbf{w}^*) + (I + k\delta_w \bar{\Sigma}_w) \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \\
&= \frac{\|\mathbf{w}_\beta\|^2 - \|\mathbf{w}_\beta - \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}} - 2(\mathbf{w}_\beta - \mathbf{w}^*)^\top \mathbf{w}^*}{2\sigma_w^2} \\
&\quad - \frac{\|\mathbf{w}_\alpha\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}} - 2(\mathbf{w}_\alpha - \mathbf{w}^*)^\top \mathbf{w}^*}{2\sigma_w^2} \\
&= \frac{\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \|\mathbf{w}_\beta - \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} - \frac{\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2_{(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2} \\
&= \frac{\|\mathbf{w}_\beta - \mathbf{w}^*\|^2_{I-(I+k\delta_w \bar{\Sigma}_w)^{-1}} - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2_{I-(I+k\delta_w \bar{\Sigma}_w)^{-1}}}{2\sigma_w^2}.
\end{aligned}$$

### A.8.4 Derivation Collection of $\tilde{\boldsymbol{\mu}}_m$ and $\tilde{\boldsymbol{w}}_m$

This section collects derivations for  $\tilde{\boldsymbol{\mu}}_m$  and  $\tilde{\boldsymbol{w}}_m$ . The derivation of  $\tilde{\boldsymbol{\mu}}_m$  is collected in Appendix A.8.4, and the derivation of  $\tilde{\boldsymbol{w}}_m$  is collected in Appendix A.8.4.

#### Derivation of $\tilde{\boldsymbol{\mu}}_m$

This section collects the derivation of  $\tilde{\boldsymbol{\mu}}_m$  in Eq. A.11 of Sec. A.8.1:

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_m &= (\mathbf{I} + (k+1)\delta_\mu \bar{\boldsymbol{\Sigma}}_\mu)^{-1}(\boldsymbol{\mu}_m + (k+1)\delta_\mu \bar{\boldsymbol{\mu}}) \\ &= (\mathbf{I} + (k+1)\delta_\mu \mathbf{I})^{-1}(\boldsymbol{\mu}_m + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i) \\ &= \frac{\boldsymbol{\mu}_m + \delta_\mu \sum_{i=1}^{k+1} \mathbf{x}_i}{1 + (k+1)\delta_\mu}.\end{aligned}$$

#### Derivation of $\tilde{\boldsymbol{w}}_m$

This section collects the derivation of  $\tilde{\boldsymbol{w}}_m$  in Eq. A.12 of Sec. A.8.1:

$$\begin{aligned}\tilde{\boldsymbol{w}}_m &= (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m + k\delta_w \bar{\mathbf{w}}) \\ &\quad (\text{Recall } k\delta_w \bar{\mathbf{w}} = \delta_w \sum_{i=1}^k \mathbf{x}_i y_i = \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}^* = k\delta_w \bar{\boldsymbol{\Sigma}}_w \mathbf{w}^*.) \\ &= (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m + k\delta_w \bar{\boldsymbol{\Sigma}}_w \mathbf{w}^*) \\ &= (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m - \mathbf{w}^* + (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w) \mathbf{w}^*) \\ &= (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_m - \mathbf{w}^*) + \mathbf{w}^*.\end{aligned}\tag{A.13}$$

## A.9 Additional Experiments for Early Ascent

### A.9.1 Early Ascent and Bounded Efficacy under Noisy Labels

We further examine phenomena of early ascent and bounded efficacy with noisy labels under varied noise levels. The results show that these two phenomena are robust to label noises to some extent.

### A.9.2 Early Ascent under Non-Linear Regression and Discrete Token Prediction

This section uses Fig. A.13 to show the existence of the early ascent phenomenon on non-linear regression and discrete token prediction with our designed distributions of pretraining and in-context samples. Fig. A.13a shows that the early ascent phenomenon exists when a 2-layer neural network with Tanh Activation function serves as the non-linear function, and Fig. A.13b shows that the early ascent phenomenon exists when the dataset consists of sequences of tokens with discrete values rather than sequences of vectors with continuous values. For the details of experiments including our designed distributions of pretraining and in-context samples, please refer to Sec. A.9.2 for the experiment with non-linear regression and Sec. A.9.2 for the experiment with discrete token prediction.

#### Experiment Design for Non-Linear Regression

The following assumption shows the data generation model to generate a non-linear sequence  $[\mathbf{x}_1, y_1, \dots, \mathbf{x}_K, y_K]$ , where  $\mathbf{x}_i$  is a vector and  $y_i$  is a scalar. The non-linear

function mapping  $\mathbf{x}$  to  $y$  is highlighted in red in the assumption.

**Assumption 7** (Pretraining Data Generative Model for Non-linear Regression).

(a) *sample a task from the task distribution:*

$$(\boldsymbol{\mu}, \mathbf{W}, \mathbf{v}) \sim \mathcal{D}^{prior}, P(\boldsymbol{\mu}, \mathbf{W}, \mathbf{v}) = \sum_{m=1}^M \pi_m P(\boldsymbol{\mu}, \mathbf{W}, \mathbf{v} | T_m),$$

where  $T_m$  represents the  $m^{th}$  mixture component, i.e.,  $P(\boldsymbol{\mu}, \mathbf{W}, \mathbf{v} | T_m) = \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\mu}_m, \sigma_\mu^2 \mathbf{I}) \cdot \frac{1}{\sqrt{(2\pi)^{d^2} \sigma_W^2}} \exp(-\frac{\|\mathbf{W} - \mathbf{W}_m\|_F^2}{2}) \cdot \mathcal{N}(\mathbf{v}; \mathbf{v}_m, \sigma_v^2 \mathbf{I})$ , and  $\pi_m$  is the mixture weight.  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$  denotes the probability of  $\mathbf{x}$  in the multivariate normal distribution with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ ,  $\|\cdot\|_F$  indicates the Frobenius norm,  $\sum_{m=1}^M \pi_m = 1$ ,  $0 < \pi_m < 1$ ,  $(\boldsymbol{\mu}_m, \mathbf{w}_m)$  is the center of the mixture component  $T_m$ , and all components share the same covariance matrix controlled by  $\sigma_\mu$ ,  $\sigma_W$ , and  $\sigma_v$ ;

(b) *input variable distribution: within a sequence,  $\forall i \in [K]$ ,  $\mathbf{x}_i \sim \mathcal{D}_x(\boldsymbol{\mu})$ ,  $P(\mathbf{x} | \boldsymbol{\mu}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \sigma_x^2 \mathbf{I})$ ;*

(c) *label distribution: within a sequence,  $\forall i \in [K]$ ,  $y_i | \mathbf{x}_i \sim \mathcal{D}_{y|\mathbf{x}_i}(\mathbf{W}, \mathbf{v})$ ,  $P(y_i | \mathbf{x}_i, \mathbf{W}, \mathbf{v}) = \mathcal{N}(y_i | \langle \tanh(\mathbf{W} \mathbf{x}_i), \mathbf{v} \rangle, \sigma_y^2)$ , where  $\tanh(\cdot)$  is a Tanh Activation function;*

(d)  *$\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\mu}_m, \mathbf{v}, \mathbf{v}_m \in \mathbb{R}^d$ , and  $\mathbf{W}, \mathbf{W}_m \in \mathbb{R}^{d \times d}$ .*

For experimental setting of Fig. A.13a, we set  $d = 2$ ,  $\sigma_\mu = 1$ ,  $\sigma_W = \sigma_v = 0.5$ ,  $\sigma_x = \sigma_y = 1$ ,  $M = 2$ ,  $\pi_1 = 0.1$ ,  $\pi_2 = 0.9$ ,  $\boldsymbol{\mu}_1 = [1, 0]^\top$ ,  $\boldsymbol{\mu}_2 = [0, 1]^\top$ ,  $\mathbf{W}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ ,  $\mathbf{W}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ , and  $\mathbf{v}_1 = [1, 0]^\top$ ,  $\mathbf{v}_2 = [0, 1]^\top$ . In-context samples follows task  $(\boldsymbol{\mu}^*, \mathbf{W}^*, \mathbf{v}^*)$ , where  $\boldsymbol{\mu}^* = \boldsymbol{\mu}_1$ ,  $\mathbf{W}^* = \mathbf{W}_2$ ,  $\mathbf{v}^* = \mathbf{v}_2$ , and  $\sigma_y = 1$ . Notice that although we add label noise to in-context samples, when evaluating the prediction, we calculate error/loss based on the clean label.

## Experiment Design for Discrete Token Prediction

The following assumption shows the data generation model to generate a non-linear sequence  $[x_1, y_1, \dots, x_K, y_K]$ , where  $x_i$  and  $y_i$  are both integers (discrete tokens).

**Assumption 8** (Pretraining Data Generative Model for Discrete Token Prediction).

- (a) *sample a task from the task distribution:  $(\mu, w) \sim \mathcal{D}^{prior}, \mu \in [M], w \in [M], P(\mu, w) = \sum_{m=1}^M \pi_m P(\mu, w | T_m)$ , where  $T_m$  represents the  $m^{\text{th}}$  mixture component, i.e.,  $P(\mu, w | T_m) = 1_{[w=w_m]}((1 - (M-1)\sigma_\mu)1_{[\mu=\mu_m]} + \sigma_\mu 1_{[\mu \neq \mu_m]})$ , and  $\pi_m$  is the mixture weight.*
- (b) *input variable distribution: within a sequence,  $\forall i \in [K], x_i \sim \mathcal{D}_x(\mu), P(x_i | \mu) = (1 - (M-1)\sigma_x)1_{[x=\mu]} + \sigma_x 1_{[x \neq \mu]}$ .*
- (c) *label distribution: within a sequence,  $\forall i \in [K], y_i | x_i \sim \mathcal{D}_{y|x_i}(w), P(y_i | x_i, w) = (1 - (M-1)\sigma_y)1_{[y_i=x_i+w \bmod M]} + \sigma_y 1_{[y_i \neq x_i+w \bmod M]}.$*

For experimental setting of Fig. A.13b, we set  $M = 6, \pi_1 = 0.04, \pi_3 = 0.481, \pi_5 = 0.479, \pi_2 = \pi_4 = \pi_6 = 0, \sigma_\mu = 0.05, \sigma_x = 0.04, \sigma_y = 0.13, \mu_1 = w_1 = 1, \mu_3 = w_3 = 3, \mu_5 = w_5 = 5$ . In-context samples follows task  $(\mu^*, w^*)$ , where  $\mu^* = \mu_1, w^* = w_3$ , and  $\sigma_y = 0.13$ . Notice that although we add label noise to in-context samples, when evaluating the prediction, we calculate error/loss based on the clean label.

## A.10 Mathematical Derivation for Early Ascent

We show that the early ascent phenomenon occurs under a specific setting in Sec. A.10.1. Then, we give formal theory with proof to show when early ascent happens in Sec. A.10.2.

### A.10.1 A Specific Setting of Early Ascent

To have a cleaner mathematical understanding of this phenomenon, this section uses the setting of  $d = 1$ , the first row, in Table A.1 to show the mathematical logic. (Some parameter settings are described in Table A.1's caption.) Following Theorem 3, the upper bound of ICL risk is as follows:

$$\begin{aligned}
& \mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^*] \\
& < \sum_{\beta=1}^2 \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2] \\
& = \|\mathbf{w}_1 - \mathbf{w}^*\|^2 \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_1 \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2] \\
& \quad + \|\mathbf{w}_2 - \mathbf{w}^*\|^2 \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_2 \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2] \\
& \text{(Notice } \mathbf{w}_2 = \mathbf{w}^*, \|\mathbf{w}_1 - \mathbf{w}^*\|^2 = 2^2 = 4.) \\
& = 4 \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_1 \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2] \\
& \text{(Notice } \tilde{\pi}_1 + \tilde{\pi}_2 = 1.) \\
& = 4 \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{\tilde{\pi}_1}{\tilde{\pi}_1 + \tilde{\pi}_2} \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right] \\
& \text{(Recall } \frac{\tilde{\pi}_1}{\tilde{\pi}_2} = r(1, 2) \text{ as Eq. A.3.)} \\
& = 4 \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{r(1, 2)}{1 + r(1, 2)} \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right].
\end{aligned}$$

Noticing  $\delta_\mu = \frac{0.05^2}{1^2}$  and  $\delta_w = \frac{0.05^2}{2^2}$  are very small, when  $k$  is small, we have  $k\delta_w \approx 0$  and  $\lambda_1(\mathbf{A}) = (\mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top)^{-1} \approx \mathbf{I}$ , thus  $\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{r(1,2)}{1+r(1,2)} \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right] \approx \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{r(1,2)}{1+r(1,2)} \|\mathbf{x}_{k+1}\|^2 \right]$  and a larger  $r(1, 2)$  means a larger upper bound. In the following, we will examine whether the increase of  $k$  leads to the increase of  $r(1, 2)$ .

Following Eq. A.3:

$$\begin{aligned} r(1, 2) &= \frac{1/2}{1/2} \exp(\Psi_{\mu}(1, 2) + \Psi_w(1, 2)) \\ &= \exp(\Psi_{\mu}(1, 2) + \Psi_w(1, 2)). \end{aligned}$$

We first analyze  $\Psi_{\mu}(1, 2)$ , following Eq. A.4:

$$\begin{aligned} \mathbb{E}[\Psi_{\mu}(1, 2)] &= \mathbb{E} \left[ \frac{\sum_{i=1}^{k+1} \|\mu_2 - \mathbf{x}_i\|^2 - \sum_{i=1}^{k+1} \|\mu_1 - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_{\mu})} \right] \\ &\quad (\text{Since } \delta_{\mu} \approx 0, \text{ thus when } k \text{ is small, we have:}) \\ &\approx \mathbb{E} \left[ \frac{\sum_{i=1}^{k+1} \|\mu_2 - \mathbf{x}_i\|^2 - \sum_{i=1}^{k+1} \|\mu_1 - \mathbf{x}_i\|^2}{2\sigma_x^2} \right] \\ &= \frac{k+1}{2\sigma_x^2} \mathbb{E} [\|\mu_2 - \mathbf{x}_1\|^2 - \|\mu_1 - \mathbf{x}_1\|^2] \\ &= \frac{k+1}{2\sigma_x^2} (\mathbb{E}[\|\mu_2 - \mathbf{x}_1\|^2] - \mathbb{E}[\|\mu_1 - \mathbf{x}_1\|^2]) \\ &= \frac{k+1}{2\sigma_x^2} (\mathbb{E}[\|\mu_2 - \mu^*\|^2] + \tau_x^2) - (\mathbb{E}[\|\mu_1 - \mu^*\|^2] + \tau_x^2) \\ &\quad (\mu^* \text{ is the same as } \mu_1, \text{ but different from } \mu_2.) \\ &= \frac{k+1}{2\sigma_x^2} (\mathbb{E}[\|\mu_2 - \mu^*\|^2] - 0) \\ &= \frac{k+1}{2 \times 1^2} \times 2^2 \\ &= 2(k+1). \end{aligned}$$

We then analyze  $\Psi_w(1, 2)$ , following Eq. A.6:

$$\begin{aligned} \mathbb{E}[\Psi_w(1, 2)] &= \mathbb{E} \left[ -\frac{\|\mathbf{w}_1 - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}}^2}{2\sigma_w^2} \right] \\ &\quad (\text{Since } \delta_w \approx 0, \text{ thus when } k \text{ is small, we have:}) \\ &\approx -\mathbb{E} \left[ \frac{(\mathbf{w}_1 - \mathbf{w}^*)^{\top} k\delta_w \bar{\Sigma}_w (\mathbf{w}_1 - \mathbf{w}^*)}{2\sigma_w^2} \right] \end{aligned}$$

$$\begin{aligned}
& \text{(Notice the feature dimension } d = 1, \bar{\Sigma}_w = \frac{\sum_{i=1}^k \|\mathbf{x}_i\|^2}{k}.) \\
& \approx -\mathbb{E} \left[ \frac{\|\mathbf{w}_1 - \mathbf{w}^*\|^2 k \delta_w \sum_{i=1}^k \|\mathbf{x}_i\|^2}{2\sigma_w^2} \right] \\
& = -\mathbb{E} \left[ \frac{2 \sum_{i=1}^k \|\mathbf{x}_i\|^2}{\sigma_y^2} \right] \\
& = -\frac{2k}{\sigma_y^2} \mathbb{E} [\|\mathbf{x}_1\|^2] \\
& = -\frac{2k}{\sigma_y^2} (\|\boldsymbol{\mu}^*\|^2 + \tau_x^2) \\
& = -\frac{2k}{2^2} \times (1 + 1) = -k.
\end{aligned}$$

Therefore, when  $k$  is small,  $r(1, 2) = \Psi_{\boldsymbol{\mu}}(1, 2) + \Psi_w(1, 2) \approx \exp(k + 2)$ , and the upper bound is approximately equal to:

$$4\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{\exp(k + 2)}{1 + \exp(k + 2)} \|\mathbf{x}_{k+1}\|^2 \right],$$

which increases as the number of in-context examples increases.

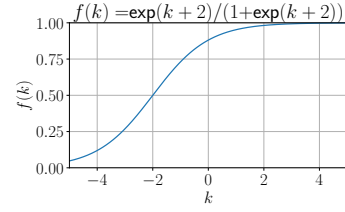


Figure A.14: Illustration of the function  $\exp(k + 2)/(1 + \exp(k + 2))$ .

### A.10.2 Theorem of Early Ascent

**Theorem 4** (Early Ascent). Assume  $\mathbb{E}_{x_1} [(\mathcal{F}^*(x_1) - \langle \mathbf{w}^*, x_1 \rangle)^2] < \mathbb{E}_{x_1} [\langle x_1, \mathbf{w}_\alpha - \mathbf{w}^* \rangle^2]$ , where  $\alpha = \arg \min_m \frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}^*\|^2}{2\sigma_x^2} + \frac{\|(\mathbf{w}_m - \mathbf{w}^*)^\top \boldsymbol{\mu}^*\|^2 + d\tau_x^2 \|\mathbf{w}_m - \mathbf{w}^*\|^2}{2\sigma_y^2}$ . Then, when  $\delta_\mu$  and  $\delta_w$  are small enough, we have the early ascent phenomenon on the risk:

$$\exists k \geq 1 \text{ s.t. } \mathbb{E}_{x_1} [(\mathcal{F}^*(x_1) - \langle \mathbf{w}^*, x_1 \rangle)^2] < \mathbb{E}_{S_k \oplus x^{(k+1)}} [(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{w}^*, \mathbf{x}_{k+1} \rangle)^2].$$

*Proof.* We examine the following case, when  $\sigma_\mu$  and  $\sigma_w$  are small enough, and  $k$  is also big enough to retrieve a task, i.e., making a center dominate:

$$\begin{aligned}
& \lim_{k \rightarrow \infty} \lim_{(\sigma_\mu, \sigma_w) \rightarrow (0,0)} \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{w}^*, \mathbf{x}_{k+1} \rangle \right)^2 \right] \\
&= \lim_{k \rightarrow \infty} \lim_{(\sigma_\mu, \sigma_w) \rightarrow (0,0)} \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left\langle \sum_{m=1}^M \tilde{\pi}_m \mathbf{A}(\mathbf{w}_m - \mathbf{w}^*), \mathbf{x}_{k+1} \right\rangle^2 \right] \\
&= \lim_{k \rightarrow \infty} \lim_{(\sigma_\mu, \sigma_w) \rightarrow (0,0)} \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left\langle \sum_{m=1}^M \tilde{\pi}_m (\mathbf{w}_m - \mathbf{w}^*), \mathbf{x}_{k+1} \right\rangle^2 \right] \\
&= \lim_{k \rightarrow \infty} \lim_{(\sigma_\mu, \sigma_w) \rightarrow (0,0)} \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left\langle \frac{\sum_{m=1}^M \pi_m \exp(\Psi_\mu(m, 1) + \Psi_w(m, 1)) (\mathbf{w}_m - \mathbf{w}^*)}{\sum_{m=1}^M \pi_m \exp(\Psi_\mu(m, 1) + \Psi_w(m, 1))}, \mathbf{x}_{k+1} \right\rangle^2 \right] \\
& \text{(Following Eq. A.8, we have } \lim_{(\sigma_\mu, \sigma_w) \rightarrow (0,0)} \Psi_\mu(m, 1) + \Psi_w(m, 1) = \\
& \quad \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_{k+1}\|^2 - \|\boldsymbol{\mu}_1 - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2} + \sum_{i=1}^k \left( \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_i\|^2 - \|\boldsymbol{\mu}_1 - \mathbf{x}_i\|^2}{2\sigma_x^2} + \frac{\|y_i^m - y_i^*\|^2 - \|y_i^1 - y_i^*\|^2}{2\sigma_y^2} \right) ) \\
&= \lim_{k \rightarrow \infty} \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left\langle \frac{\sum_{m=1}^M \pi_m \exp \left( \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2} + \sum_{i=1}^k \left( \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_i\|^2}{2\sigma_x^2} + \frac{\|y_i^m - y_i^*\|^2}{2\sigma_y^2} \right) \right) (\mathbf{w}_m - \mathbf{w}^*)}{\sum_{m=1}^M \pi_m \exp \left( \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2} + \sum_{i=1}^k \left( \frac{\|\boldsymbol{\mu}_m - \mathbf{x}_i\|^2}{2\sigma_x^2} + \frac{\|y_i^m - y_i^*\|^2}{2\sigma_y^2} \right) \right)}, \mathbf{x}_{k+1} \right\rangle^2 \right] \\
&= \mathbb{E}_{S_k \oplus x^{(k+1)}} [\langle \mathbf{w}_\alpha - \mathbf{w}^*, \mathbf{x}_{k+1} \rangle^2] \\
&= \mathbb{E}_{\mathbf{x}_1} [\langle \mathbf{w}_\alpha - \mathbf{w}^*, \mathbf{x}_1 \rangle^2],
\end{aligned}$$

where  $\alpha = \arg \min_m \frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}^*\|^2}{2\sigma_x^2} + \frac{\|(\mathbf{w}_m - \mathbf{w}^*)^\top \boldsymbol{\mu}^*\|^2 + d\tau_x^2 \|\mathbf{w}_m - \mathbf{w}^*\|^2}{2\sigma_y^2}$ . □

## A.11 Proof Tools

This section introduces the inequalities used in our proofs for Theorems 3 (fine-grained upper bound for ICL risk), 5 (upper bound for ICL with biased labels), 7 (coarse upper bound for ICL risk) and Lemma 6 ((informal) upper bound for zero-shot ICL):

### A.11.1 Gaussian Tail Bound

If  $Z_i \sim \mathcal{N}(0, 1)$ , then for  $t > 0$  we have:

$$\begin{aligned} P\left(\frac{\sum_{i=1}^k Z_i}{k} > t\right) &\leq \exp\left(-\frac{kt^2}{2}\right), \\ P\left(\frac{\sum_{i=1}^k Z_i}{k} < -t\right) &\leq \exp\left(-\frac{kt^2}{2}\right). \end{aligned}$$

### A.11.2 Chi-squared Tail Bound

If  $X \sim \chi(k)$ , i.e.,  $X = \sum_{i=1}^k Z_i^2$  where  $Z_i \sim \mathcal{N}(0, 1)$  then [10]:

$$\begin{aligned} P\left(\frac{X}{k} - 1 > 2\sqrt{t_1} + 2t_1\right) &\leq \exp(-kt_1^2), \\ P\left(\frac{X}{k} - 1 < -2\sqrt{t_1}\right) &\leq \exp(-kt_1^2). \end{aligned}$$

As a looser but symmetric bound, for any  $t > 0$ , we have:

$$\begin{aligned} P\left(\frac{X}{k} - 1 > t\right) &\leq \exp\left(-\frac{kt^2}{8}\right), \\ P\left(\frac{X}{k} - 1 < -t\right) &\leq \exp\left(-\frac{kt^2}{8}\right). \end{aligned}$$

### A.11.3 Norm Tail Bound

If  $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \tau_x^2 \mathbf{I})$ ,  $\epsilon_i \in \mathbb{R}^d$ ,  $\mathbf{I} \in \mathbb{R}^{d \times d}$ , then for  $t > 0$  we have:

$$P\left(\left\|\frac{\sum_{i=1}^k \epsilon_i}{k}\right\| > \sqrt{\frac{\tau_x^2 d}{k}(1+t)}\right) \leq \exp\left(-\frac{kt^2}{8}\right),$$

where  $\|\cdot\|$  indicates the  $L_2$  norm.

*Proof.*

$$\begin{aligned}
& \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\|^2 \\
&= \sum_{j=1}^d \left( \frac{\sum_{i=1}^k \epsilon_{i,j}}{k} \right)^2 \\
&= \frac{\tau_x^2}{k} \sum_{j=1}^d \left( \frac{\sum_{i=1}^k \epsilon_{i,j}}{\tau_x \sqrt{k}} \right)^2 \\
& \text{(Notice } \epsilon_{i,j} \sim \mathcal{N}(0, \tau_x^2) \text{ and let } Z_j = \frac{\sum_{i=1}^k \epsilon_{i,j}}{\tau_x \sqrt{k}} \sim \mathcal{N}(0, 1).) \\
&= \frac{\tau_x^2 d}{k} \frac{\sum_{i=1}^d Z_i^2}{d}.
\end{aligned}$$

Therefore, by applying Appendix A.11.2 we have:

$$P \left( \frac{\tau_x^2 d}{k} \frac{\sum_{i=1}^d Z_i^2}{d} > \frac{\tau_x^2 d}{k} (1+t) \right) \leq \exp \left( -\frac{kt^2}{8} \right).$$

□

#### A.11.4 Eigenvalue Concentration Bound

**Lemma 8.** *If  $\forall i, \mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_x^2 \mathbf{I})$ ,  $\|\boldsymbol{\mu}\| = 1$ ,  $\mathbf{A} = \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k}$ , and  $\boldsymbol{\epsilon}_i = \mathbf{x}_i - \boldsymbol{\mu}$ , we have  $\forall t > 0$ :*

$$P \left( L \leq \lambda_d(\mathbf{A}) \leq \lambda_1(\mathbf{A}) \leq U \text{ and } \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| < \tau_x \sqrt{\gamma(1+t)} \right) > 1 - 3 \exp \left( -\frac{kt^2}{8} \right),$$

where  $L = \tau_x^2(1 - \frac{t}{2} - \gamma)^2 - 2\tau_x\gamma\sqrt{1+t}$ ,  $U = 1 + \tau_x^2(1 + \frac{t}{2} + \gamma)^2 + 2\tau_x\gamma\sqrt{1+t}$ ,  $\lambda_i(\mathbf{A})$  is the  $i^{\text{th}}$  biggest eigenvalue of the matrix  $\mathbf{A}$  and  $\gamma = \sqrt{\frac{d}{k}}$ .

We begin with decomposing  $\mathbf{A}$  to three components:

$$\mathbf{A} = \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} = \frac{\sum_{i=1}^k (\boldsymbol{\mu} + \boldsymbol{\epsilon}_i)(\boldsymbol{\mu} + \boldsymbol{\epsilon}_i)^\top}{k} = \boldsymbol{\mu} \boldsymbol{\mu}^\top + \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^\top}{k} + \frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k},$$

then consider the eigenvalue bound of each of them.

For the first component  $\boldsymbol{\mu}\boldsymbol{\mu}^\top$ , we have:

$$0 \leq \lambda_d(\boldsymbol{\mu}\boldsymbol{\mu}^\top) < \lambda_1(\boldsymbol{\mu}\boldsymbol{\mu}^\top) \leq 1.$$

Then, we analyze the second component  $\frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^\top}{k}$ . Following Vershynin [111, Theorem 4.6.1, p. 97], we have for any  $1 - \sqrt{\frac{d}{k}} > s > 0$ :

$$\begin{aligned} P \left( \left( 1 - s - \sqrt{\frac{d}{k}} \right)^2 \leq \frac{1}{\tau_x^2} \lambda_d \left( \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^\top}{k} \right) < \frac{1}{\tau_x^2} \lambda_1 \left( \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^\top}{k} \right) \leq \left( 1 + s + \sqrt{\frac{d}{k}} \right)^2 \right) \\ > 1 - 2 \exp \left( -\frac{ks^2}{2} \right). \end{aligned}$$

Finally, we examine the third component  $\frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k}$ . We have for all  $\|\mathbf{a}\| = 1$ :

$$\left\| \mathbf{a}^\top \frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k} \mathbf{a} \right\| = 2 \left\| \mathbf{a}^\top \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \boldsymbol{\mu}^\top \mathbf{a} \right\| \leq 2 \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\|$$

(Notice by Norm Tail Bound in Appendix A.11.3, we have:

$$\begin{aligned} P \left( \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| > \sqrt{\frac{\tau_x^2 d}{k}} (1+t) \right) &\leq \exp \left( -\frac{kt^2}{8} \right). \\ \Rightarrow P \left( \left\| \mathbf{a}^\top \frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k} \mathbf{a} \right\| \leq 2 \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| \leq 2 \sqrt{\frac{\tau_x^2 d}{k}} (1+t) \right) &> 1 - \exp \left( -\frac{kt^2}{8} \right) \\ \Rightarrow P \left( -2\tau_x \sqrt{\frac{d}{k}} (1+t) \leq \lambda_d \left( \frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k} \right) \leq \lambda_1 \left( \frac{\sum_{i=1}^k (\boldsymbol{\mu} \boldsymbol{\epsilon}_i^\top + \boldsymbol{\epsilon}_i \boldsymbol{\mu}^\top)}{k} \right) \leq 2\tau_x \sqrt{\frac{d}{k}} (1+t) \right) \\ &> 1 - \exp \left( -\frac{kt^2}{8} \right). \end{aligned}$$

Let  $\gamma = \sqrt{\frac{d}{k}}$ ,  $s = t/2$ , and summarize three components by union bound, we have:

$$\begin{aligned} P \left( \tau_x^2 \left( 1 - \frac{t}{2} - \gamma \right)^2 - 2\tau_x \gamma \sqrt{1+t} \leq \lambda_d(\mathbf{A}) \leq \lambda_1(\mathbf{A}) \leq 1 + \tau_x^2 \left( 1 + \frac{t}{2} + \gamma \right)^2 + 2\tau_x \gamma \sqrt{1+t} \right) \\ > 1 - 3 \exp \left( -\frac{kt^2}{8} \right). \end{aligned}$$

As a summary, we have:

$$P\left(\mathbf{L} \leq \lambda_d(\mathbf{A}) \leq \lambda_1(\mathbf{A}) \leq \mathbf{U} \text{ and } \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| < \tau_x \sqrt{\gamma(1+t)}\right) > 1 - 3 \exp\left(-\frac{kt^2}{8}\right),$$

where  $\gamma = \sqrt{\frac{d}{k}}$ ,  $\mathbf{L} = \tau_x^2(1 - \frac{t}{2} - \gamma)^2 - 2\tau_x\gamma\sqrt{1+t}$ ,  $\mathbf{U} = 1 + \tau_x^2\left(1 + \frac{t}{2} + \gamma\right)^2 + 2\tau_x\gamma\sqrt{1+t}$ , and  $\lambda_i(\mathbf{A})$  is the  $i^{\text{th}}$  biggest eigenvalue of the matrix  $\mathbf{A}$ .

## A.12 ICL to Learn the In-Context Function

This section introduces the proof of Theorem 7 (coarse upper bound for ICL risk) and Theorem 3 (finegrained upper bound for ICL risk). The upper bound of Theorem 3 is derived at Eq. A.14.

*Proof.* Assuming we are using in-context examples following Assumption 3, *i.e.*,  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ ,  $y_i = \langle \mathbf{x}_i, \mathbf{w}^* \rangle$ ,  $\|\boldsymbol{\mu}^*\| = \|\mathbf{w}^*\| = 1$ , and we aim to have the prediction of  $S_k \oplus x^{(k+1)}$  to be  $\langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle$ , *i.e.*, to learn the function  $(\mathbf{w}^*)$  of the in-context task  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ . Let  $\mathcal{L}_k^*$  indicate the squared loss  $(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle)^2$ , where  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$  is the prediction of  $S_k \oplus x^{(k+1)}$  by the Bayes-optimal next-token predictor  $\mathcal{F}^*$  under Assumption 2 for pretraining data generation. We derive the upper bound of the expected squared loss as follows:

$$\begin{aligned} & \mathbb{E}_{S_k \oplus x^{(k+1)}}[\mathcal{L}_k^*] \\ &= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{w}^*, \mathbf{x}_{k+1} \rangle \right)^2 \right] \\ & \text{(By Corollary 2.)} \\ &= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \sum_{m=1}^M \tilde{\pi}_m \langle \tilde{\mathbf{w}}_m, \mathbf{x}_{k+1} \rangle - \langle \mathbf{w}^*, \mathbf{x}_{k+1} \rangle \right)^2 \right] \end{aligned}$$

$$= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \left\langle \sum_{m=1}^M \tilde{\pi}_m (\tilde{\mathbf{w}}_m - \mathbf{w}^*), \mathbf{x}_{k+1} \right\rangle \right)^2 \right]$$

(See Eq. A.13 for the derivation of  $\tilde{\mathbf{w}}_m$ .)

$$= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \left\langle \sum_{m=1}^M \tilde{\pi}_m ((\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1} (\mathbf{w}_m - \mathbf{w}^*) + \mathbf{w}^* - \mathbf{w}^*), \mathbf{x}_{k+1} \right\rangle \right)^2 \right]$$

(Let  $\mathbf{A} = (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}$ , and notice  $\mathbf{A}$  is symmetric positive definite.)

$$= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left\langle \sum_{m=1}^M \tilde{\pi}_m \mathbf{A} (\mathbf{w}_m - \mathbf{w}^*), \mathbf{x}_{k+1} \right\rangle^2 \right]$$

(Notice  $\left( \sum_{\beta=1}^M \tilde{\pi}_\beta a_\beta \right)^2 \leq \sum_{\beta=1}^M \tilde{\pi}_\beta a_\beta^2$ , since  $\mathbb{E}[a]^2 \leq \mathbb{E}[a^2]$ .)

$$\begin{aligned} &\leq \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{m=1}^M \tilde{\pi}_m \langle \mathbf{A} (\mathbf{w}_m - \mathbf{w}^*), \mathbf{x}_{k+1} \rangle^2 \right] \\ &= \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m ((\mathbf{w}_m - \mathbf{w}^*)^\top \mathbf{A} \mathbf{x}_{k+1})^2 \right] \\ &\leq \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m \|\mathbf{w}_m - \mathbf{w}^*\|^2 \lambda_1(\mathbf{A})^2 \|\mathbf{x}_{k+1}\|^2 \right] \\ &= \sum_{m=1}^M \|\mathbf{w}_m - \mathbf{w}^*\|^2 \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right] \\ &\leq 4 \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{m=1}^M \tilde{\pi}_m \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right] \\ &= 4 \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \|\mathbf{x}_{k+1}\|^2 \lambda_1(\mathbf{A})^2 \right] \end{aligned} \tag{A.14}$$

(Notice  $\mathbf{A}$  is a random matrix only depends on  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ , but not  $\mathbf{x}_{k+1}$ .)

$$\begin{aligned} &= 4 \mathbb{E}_{\mathbf{x}_{k+1}} \left[ \|\mathbf{x}_{k+1}\|^2 \right] \mathbb{E}_{S_k} \left[ \lambda_1^2(\mathbf{A}) \right] \\ &= 4(1 + d\tau_x^2) \mathbb{E}_{S_k} \left[ \lambda_1^2(\mathbf{A}) \right]. \end{aligned}$$

We further simplify  $\mathbb{E}_{S_k} [\lambda_1^2(\mathbf{A})]$  using Lemma 8:

$$\begin{aligned} &\mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^*] \\ &\leq 4(1 + d\tau_x^2) \mathbb{E}_{S_k} [\lambda_1^2(\mathbf{A})] \end{aligned}$$

$$\begin{aligned}
&\leq 4(1 + d\tau_x^2)\mathbb{E}_{S_k} \left[ \left( \frac{1}{1 + k\delta_w \lambda_d \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right)} \right)^2 \right] \\
&\quad \text{(By applying Lemma 8 to } \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \text{.)} \\
&\leq 4(1 + d\tau_x^2)\mathbb{E}_{S_k} \left[ \left( \frac{1}{1 + k\delta_w \mathbf{L}} \right)^2 \right] \\
&\leq 4(1 + d\tau_x^2) \left( \left( \frac{1}{1 + k\delta_w (\tau_x^2 (1 - \frac{t}{2} - \gamma)^2 - 2\tau_x \gamma \sqrt{1+t})} \right)^2 + 3 \exp \left( -\frac{kt^2}{8} \right) \right).
\end{aligned}$$

Let  $t = k^{\delta - \frac{1}{2}}$ , where  $\frac{1}{2} > \delta > 0$  and  $\delta$  is arbitrary small. We have:

$$\mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^*] < \frac{4(1 + d\tau_x^2)}{\tau_x^4 \delta^2 k^2} + O(k^{\delta - \frac{5}{2}}).$$

□

We further validate our analysis with numerical computations in Fig. A.15, including the trend of  $\tilde{\pi}_m$  for  $m \in [M]$ ,  $\lambda_j \left( \delta_w \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right)$  for  $j \in [d]$ ,  $\lambda_j \left( \mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top \right)$  for  $j \in [d]$ ,  $1/\|\tilde{\mathbf{w}} - \mathbf{w}^*\|$ ,  $1/\mathbb{E}[\mathcal{F}^*(S_k \oplus x^{(k+1)}) - y_{k+1}^*]$ , and  $1/\mathbb{E}[(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - y_{k+1}^*)^2]$  as  $k$  increases.

### A.12.1 Case When In-context Input Variable Spans in Subspace

In this section, we refine Eq. A.14 for the finegrained bound in Theorem 3. Specifically, we refine the following inequality for case when in-context input variable  $\mathbf{x}_i$  only spans in the subspace of  $\mathbb{R}^d$ , resulting in  $\lambda_1(\mathbf{A}) = 1$  constantly as mentioned in Theorem 3:

$$\begin{aligned}
&\sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m ((\mathbf{w}_m - \mathbf{w}^*)^\top \mathbf{A} \mathbf{x}_{k+1})^2 \right] \\
&\leq \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m \|\mathbf{w}_m - \mathbf{w}^*\|^2 \lambda_1(\mathbf{A})^2 \|\mathbf{x}_{k+1}\|^2 \right],
\end{aligned}$$

where  $\mathbf{A} = (\mathbf{I} + \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top)^{-1}$  is derived in Lemma 1. Violating Assumption 3(a), in this section we consider the case that  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\underbrace{1, \dots, 1}_{d'}, 0, \dots, 0))$ , where  $\boldsymbol{\mu} = [p, \underbrace{0, \dots, 0}_{d'-1}, q, 0, \dots, 0]^\top$ . (If  $\boldsymbol{\mu}$  does not follow the format  $[p, \underbrace{0, \dots, 0}_{d'-1}, q, 0, \dots, 0]^\top$ , we can always rotate the coordinates so  $\boldsymbol{\mu}$  has this format.) Therefore, we have matrix  $\mathbf{A}$  (after rotation) with the following format:

$$\mathbf{A} = \begin{cases} \begin{bmatrix} \mathbf{I}_{d' \times d'} + \sum_{i=1}^k \mathbf{x}_{i,1:d'} \mathbf{x}_{i,1:d'}^\top & \mathbf{0}_{d' \times (d-d')} \\ \mathbf{0}_{(d-d') \times d'} & \mathbf{I}_{(d-d') \times (d-d')} \end{bmatrix}^{-1}, & \text{if } q = 0 \\ \begin{bmatrix} \mathbf{I}_{(d'+1) \times (d'+1)} + \sum_{i=1}^k \mathbf{x}_{i,1:(d'+1)} \mathbf{x}_{i,1:(d'+1)}^\top & \mathbf{0}_{(d'+1) \times (d-d'-1)} \\ \mathbf{0}_{(d-d'-1) \times (d'+1)} & \mathbf{I}_{(d-d'-1) \times (d-d'-1)} \end{bmatrix}^{-1}, & \text{if } q > 0 \end{cases}$$

where  $\mathbf{x}_{i,1:d'} = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,d'}]^\top$ ,  $\mathbf{I}_{a \times a}$  indicates an identity matrix with shape  $a$  by  $a$ , and  $\mathbf{0}_{a \times b}$  indicates a zero matrix with shape  $a$  by  $b$ . Finally, we can revise the upper bound for the case when  $\mathbf{x}_i$  only spans in a subspace of  $\mathbb{R}^d$  using the new format of  $\mathbf{A}$  as follows:

When  $q = 0$ , we have:

$$\begin{aligned} & \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m ((\mathbf{w}_m - \mathbf{w}^*)^\top \mathbf{A} \mathbf{x}_{k+1})^2 \right] \\ & \leq \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m ((\mathbf{w}_m - \mathbf{w}^*)_{1:d'}^\top \mathbf{A}_{1:d',1:d'} \mathbf{x}_{k+1,1:d'} \right. \\ & \quad \left. + (\mathbf{w}_m - \mathbf{w}^*)_{(d'+1):d}^\top \mathbf{I}_{(d-d') \times (d-d')} \mathbf{x}_{k+1,(d'+1):d})^2 \right] \\ & \leq \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m (\|(\mathbf{w}_m - \mathbf{w}^*)_{1:d'}\|^2 \lambda_1(\mathbf{A}_{1:d',1:d'})^2 \|\mathbf{x}_{k+1,1:d'}\|^2 \right. \\ & \quad \left. + \|(\mathbf{w}_m - \mathbf{w}^*)_{(d'+1):d}\|^2 \|\mathbf{x}_{k+1,(d'+1):d}\|^2) \right], \end{aligned}$$

(Notice  $\|\mathbf{x}_{k+1,(d'+1):d}\|^2 = 0$ )

$$= \sum_{m=1}^M \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_m \|(\mathbf{w}_m - \mathbf{w}^*)_{1:d'}\|^2 \lambda_1(\mathbf{A}_{1:d',1:d'})^2 \|\mathbf{x}_{k+1,1:d'}\|^2 \right],$$

When  $q > 0$ , we skip the analysis since the analysis for  $q > 0$  is the same as the analysis for  $q = 0$ . The only difference is that  $d'$  for  $q > 0$  is one bigger than  $d'$  for  $q = 0$ .

## A.13 ICL with Biased Labels to Retrieve A Task

This section details the proof of Theorem 5, with Fig.A.16 serving as a visual guide. The non-asymptotic bound for the bounded efficacy phenomenon and the asymptotic bound share the same foundational elements in the proof. However, they are different in handling the components marked in pink. Fig. A.16 is thus provided to offer a clearer understanding of its overall framework and assist readers in navigating through the proof. In the following sections, Sec. A.13.1 introduces the non-asymptotic bound revealing the bounded efficacy phenomenon, and Sec. A.13.2 introduces the asymptotic bound.

### A.13.1 Non-Asymptotic Bound for the Bounded Efficacy Phenomenon

This section proves the non-asymptotic bound in Theorem 5: Consider a next-token predictor attaining the optimal pretraining risk. When  $\delta_\mu$  and  $\delta_w$  are sufficiently small, there exists a particular interval (refer to Sec.A.13.1 for the interval) for  $k$  such that ICL risk with biased labels is upper bounded by:

$$\mathbb{E}_{S_k}[\mathcal{L}_k^\alpha] < C_3 \exp\left(-k\left(\frac{d_\mu^2}{8\sigma_x^2} + \frac{u_w^2 \tau_x^2}{8\sigma_y^2}\right)\right) + 48(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right)$$

$$+ \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2(1 + d\tau_x^2) \min\{1, 4k^2\delta_w^2(1 + \tau_x^2)^2\}.$$

where  $\mathcal{L}_k^\alpha = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^\alpha)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$   $C_3$  is a constant depending on the prior setting,  $\tau_x$ , and  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ . With small  $k$ , the first and second terms dominate and exponential decay. With large  $k$ , the third term dominates and increases. Thus, the upper bound reveals a bounded efficacy phenomenon.

*Proof.* Assuming we are using in-context examples following Assumptions 3 and 4, i.e.,  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ ,  $y_i = \langle \mathbf{x}_i, \mathbf{w}^* \rangle$ ,  $\|\boldsymbol{\mu}^*\| = \|\mathbf{w}^*\| = 1$ , and we aim to retrieve the function  $\mathbf{w}_\alpha$  of the prior center  $(\boldsymbol{\mu}_\alpha, \mathbf{w}_\alpha)$  which is close to the in-context task. Let  $\mathcal{L}_k^\alpha$  indicate the squared risk  $(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ , where  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$  is the prediction of  $S_k \oplus x^{(k+1)}$  by the Bayes-optimal next-token predictor  $\mathcal{F}^*$ . In order to have an upper bound on the risk, we consider  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$  in two cases: (1) **C**:  $L < \lambda_d \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) \leq \lambda_1 \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) < U$  and  $\left\| \frac{\sum_{i=1}^k \epsilon_i}{k} \right\| < \tau_x \sqrt{\gamma(1+t)}$  (see Lemma 8 for  $t, \gamma, L$  and  $U$ ) and (2)  $\neg \mathbf{C}$ : at least one of the previous inequalities does not hold. Following Lemma 8, the probability of  $\neg \mathbf{C}$  is bounded by:  $P(\neg \mathbf{C}) \leq 3 \exp(-\frac{kt^2}{8})$ .

We start our upper bound analysis on the expected squared risk by splitting the risk into three parts:

$$\begin{aligned} & \mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\ &= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ (\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle)^2 \right] \\ & \text{(By Corollary 2.)} \\ &= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta, \mathbf{x}_{k+1} \rangle - \langle \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle \right)^2 \right] \end{aligned}$$

$$\begin{aligned}
& (\text{Notice } \sum_{\beta=1}^M \tilde{\pi}_\beta = 1.) \\
& = \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \left( \sum_{\beta=1}^M \tilde{\pi}_\beta (\langle \tilde{\mathbf{w}}_\beta, \mathbf{x}_{k+1} \rangle - \langle \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle) \right)^2 \right] \\
& (\text{Notice } \left( \sum_{\beta=1}^M \tilde{\pi}_\beta a_\beta \right)^2 \leq \sum_{\beta=1}^M \tilde{\pi}_\beta a_\beta^2, \text{ since } \mathbb{E}[a]^2 \leq \mathbb{E}[a^2].) \\
& \leq \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta (\langle \tilde{\mathbf{w}}_\beta, \mathbf{x}_{k+1} \rangle - \langle \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle)^2 \right] \\
& = \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \right] \\
& = P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \\
& \quad + P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \neg \mathbf{C} \right] \\
& = P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \tag{Part A} \\
& \quad + P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C}] \tag{Part B} \\
& \quad + P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \neg \mathbf{C} \right]. \tag{Part C}
\end{aligned}$$

We will analyze three parts one by one in the following three sections respectively.

□

### Bounded Efficacy - Part A

*Proof.* We firstly analyze the term  $P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C}]$ , Part A:

$$\begin{aligned}
& P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \\
& < P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]
\end{aligned}$$

(See Eq. A.13 for the derivation of  $\tilde{\mathbf{w}}_\beta$ .)

$$= P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|(\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}(\mathbf{w}_\beta - \mathbf{w}^*) + \mathbf{w}^* - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$$

(Let  $\mathbf{A} = (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}$ , and  $\lambda_1(\mathbf{A})$  is the largest eigenvalue of matrix  $\mathbf{A}$ .)

$$= P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\mathbf{A}(\mathbf{w}_\beta - \mathbf{w}^*) + \mathbf{w}^* - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$$

$$\leq P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta (\|\mathbf{A}(\mathbf{w}_\beta - \mathbf{w}^*)\| + \|\mathbf{w}^* - \mathbf{w}_\alpha\|)^2 \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$$

(Notice  $\|\mathbf{w}_\beta - \mathbf{w}^*\| \leq 2$ .)

$$\leq P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 (2\lambda_1(\mathbf{A}) + \|\mathbf{w}^* - \mathbf{w}_\alpha\|)^2 \middle| \mathbf{C} \right]$$

(Notice  $\mathbf{A} = (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}$  and conditioned on  $\mathbf{C}$  we have:

$$\mathbf{L} < \lambda_d(\bar{\Sigma}_w) < \lambda_1(\bar{\Sigma}_w) < \mathbf{U}.)$$

$$\leq P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \left( \frac{2}{1 + k\delta_w \mathbf{L}} + \|\mathbf{w}^* - \mathbf{w}_\alpha\| \right)^2$$

(Notice  $\|\mathbf{w}^* - \mathbf{w}_\alpha\| \leq 2$ .)

$$\leq 16P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right].$$

(By applying Eqs. A.3, A.4, A.6, and Assumption 2(e) on  $\frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha}$  :)

$$< 16P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} r \exp \left( \frac{-\sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^{k+1} \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \right.$$

$$\cdot \exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}}^2}{2\sigma_w^2} \right) \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$$

(In the first exponential term, by splitting  $\sum_{i=1}^{k+1}$  to  $\sum_{i=1}^k$  and  $i = k+1$  :)

$$< 16P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \underbrace{\sum_{\beta \neq \alpha} r \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right)}_{\text{Part A-1}} \right.$$

$$\cdot \underbrace{\exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}}^2}{2\sigma_w^2} \right)}_{\text{Part A-2}} \left. \right]$$

$$\cdot \underbrace{\exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2}_{\text{Part A-3}} \Big| \mathbf{C} \Big]$$

(Note that  $\mathbf{x}_1, \dots, \mathbf{x}_k$  are dependent on  $\mathbf{C}$  but  $\mathbf{x}_{k+1}$  is not.

Thus, we split them for further analysis.)

In the following, we separately analyze the three terms, Part A-1, Part A-2, and Part A-3. The high-level idea is that, as  $k$  increases, due to the concentration of Part A-1 and Part A-2, they can be upper bounded by a function of  $k$ . Then, regarding Part A-1 and Part A-2 as constant values (their upper bounds), the expectation of Part A-3 can be upper bounded.

**Part A-1.** We first deal with Part A-1. When conditioned on case  $\mathbf{C}$ , we have:

$$\begin{aligned} & \frac{\sum_{i=1}^k (-\|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2)}{1 + (k+1)\delta_\mu} \\ & \text{(Let } \mathbf{x}_i = \boldsymbol{\mu}^* + \boldsymbol{\epsilon}_i) \\ & = k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + \frac{\sum_{i=1}^k 2\langle \boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha, \boldsymbol{\epsilon}_i \rangle}{k}}{1 + (k+1)\delta_\mu} \\ & = k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + \left\langle 2(\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha), \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\rangle}{1 + (k+1)\delta_\mu} \\ & \leq k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 2\|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha\| \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\|}{1 + (k+1)\delta_\mu} \end{aligned}$$

(Recall we have  $\forall \beta \in [M], \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha\| \leq 2$ , and in case  $\mathbf{C}$  we have:

$$\begin{aligned} & \left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| < \tau_x \gamma \sqrt{1+t}. \\ & < k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x \gamma \sqrt{1+t}}{1 + (k+1)\delta_\mu}. \end{aligned}$$

Let  $t = k^{-\frac{1}{4}}$ . Recall in Assumption 4, we have  $\forall \beta \neq \alpha, \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 \geq d_\mu^2$ . If  $\delta_\mu \ll 1$  s.t.  $I_\mu = \{k | (k+1)\delta_\mu \leq 1 \text{ and } \frac{d_\mu^2}{2} > 4\tau_x\gamma\sqrt{1+k^{-\frac{1}{4}}}\} \neq \emptyset$ , then when  $k \in I_\mu$  we have:

$$k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x\gamma\sqrt{1+t}}{1 + (k+1)\delta_\mu} < k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + \frac{d_\mu^2}{2}}{2} = -k \frac{d_\mu^2}{4}.$$

**Part A-2.** We then deal with Part A-2. When conditioned on case **C**, we have:

$$\begin{aligned} & -\|\mathbf{w}_\beta - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2 \\ & (\lambda_1(\mathbf{A}) \text{ and } \lambda_d(\mathbf{A}) \text{ indicate the largest and smallest eigenvalues of the matrix } \mathbf{A}.) \\ & < -\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \lambda_d(\mathbf{I} - (\mathbf{I} + k\delta_w\bar{\Sigma}_w)^{-1}) + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \lambda_1(\mathbf{I} - (\mathbf{I} + k\delta_w\bar{\Sigma}_w)^{-1}) \\ & (\text{Recall in case } \mathbf{C} \text{ we have: } L < \lambda_d(\bar{\Sigma}_w) < \lambda_1(\bar{\Sigma}_w) < U.) \\ & < -\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \left(1 - \frac{1}{1 + k\delta_w L}\right) + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \left(1 - \frac{1}{1 + k\delta_w U}\right) \\ & = -\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \frac{k\delta_w L}{1 + k\delta_w L} + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \frac{k\delta_w U}{1 + k\delta_w U} \\ & < -\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \frac{k\delta_w L}{1 + k\delta_w \tau_x^2} + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \frac{k\delta_w U}{1 + k\delta_w \tau_x^2} \end{aligned}$$

Let  $t = k^{-\frac{1}{4}}$ . If  $\delta_w \ll 1$  s.t.  $I_w = \{k | k\delta_w \tau_x^2 \leq 1 \text{ and } L\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - U\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 > \frac{\tau_x^2 u_w^2}{2}\} \neq \emptyset$ , (note  $\lim_{k \rightarrow \infty} L\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - U\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 = \tau_x^2 \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - (1 + \tau_x^2)\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \geq \tau_x^2 u_w^2$ ) then when  $k \in I_w$ , we have:

$$-\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \frac{k\delta_w L}{1 + k\delta_w \tau_x^2} + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \frac{k\delta_w U}{1 + k\delta_w \tau_x^2} < -\frac{\tau_x^2 u_w^2}{2} \frac{k\delta_w}{1 + k\delta_w \tau_x^2} < -k\delta_w \frac{\tau_x^2 u_w^2}{4}.$$

**Part A-3.** We finally deal with Part A-3. Part A-3 is independent to case **C**, and we have:

$$P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$$

$$\begin{aligned}
&< \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \right] \\
&(\text{Let } \mathbf{x}_{k+1} = \boldsymbol{\mu}^* + \boldsymbol{\epsilon}.) \\
&= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^* - \boldsymbol{\epsilon}\|^2 + \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^* - \boldsymbol{\epsilon}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \right] \\
&= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 + \langle 2(\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha), \boldsymbol{\epsilon} \rangle}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \right] \\
&(\text{Let } -\|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 = -D, 2\sigma_x^2(1 + (k+1)\delta_\mu) = E, \mathbf{b} = 2(\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha).) \\
&= \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-D + \mathbf{b}^\top \boldsymbol{\epsilon}}{E} \right) \|\mathbf{x}_{k+1}\|^2 \right] \\
&(\text{Notice } \|\mathbf{x}_{k+1}\|^2 = \|\boldsymbol{\mu}^* + \boldsymbol{\epsilon}\|^2 \leq 2\|\boldsymbol{\mu}^*\|^2 + 2\|\boldsymbol{\epsilon}\|^2.) \\
&\leq \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-D + \mathbf{b}^\top \boldsymbol{\epsilon}}{E} \right) (2\|\boldsymbol{\mu}^*\|^2 + 2\|\boldsymbol{\epsilon}\|^2) \right] \\
&(\text{Notice } \|\boldsymbol{\mu}^* + \boldsymbol{\epsilon}\|^2 = 1.) \\
&= 2 \left( \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-D + \mathbf{b}^\top \boldsymbol{\epsilon}}{E} \right) \right] + \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-D + \mathbf{b}^\top \boldsymbol{\epsilon}}{E} \right) \|\boldsymbol{\epsilon}\|^2 \right] \right) \\
&= 2 \left( \exp \left( \frac{\tau_x^2 \|\mathbf{b}\|^2}{2E^2} - \frac{D}{E} \right) + \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-D + \mathbf{b}^\top \boldsymbol{\epsilon}}{E} \right) \|\boldsymbol{\epsilon}\|^2 \right] \right) \\
&= 2 \left( \exp \left( \frac{\tau_x^2 \|\mathbf{b}\|^2}{2E^2} - \frac{D}{E} \right) + \tau_x^2 \left( 1 + \frac{\tau_x^2 \|\mathbf{b}\|^2}{E^2} \right) \exp \left( \frac{\tau_x^2 \|\mathbf{b}\|^2}{2E^2} - \frac{D}{E} \right) \right. \\
&\quad \left. + (d-1)\tau_x^2 \exp \left( \frac{\tau_x^2 \|\mathbf{b}\|^2}{2E^2} - \frac{D}{E} \right) \right) \\
&= 2 \left( 1 + \tau_x^2 \left( d + \frac{\tau_x^2 \|\mathbf{b}\|^2}{E^2} \right) \right) \exp \left( \frac{\tau_x^2 \|\mathbf{b}\|^2}{2E^2} - \frac{D}{E} \right) \\
&= C_{k=0}.
\end{aligned}$$

**Summary of Part A.** Thus, summarizing Part A-1, Part A-2, and Part A-3, we have:

$$\begin{aligned}
& P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \\
& < 16P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \underbrace{\sum_{\beta \neq \alpha} r \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right)}_{\text{Part A-1}} \right. \\
& \quad \cdot \underbrace{\exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}}^2}{2\sigma_w^2} \right)}_{\text{Part A-2}} \\
& \quad \cdot \underbrace{\exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2}_{\text{Part A-3}} \middle| \mathbf{C} \Big] \\
& < 16r(M-1)C_{k=0} \exp \left( -\frac{d_\mu^2 k}{8\sigma_x^2} \right) \exp \left( -\frac{u_w^2 \tau_x^2 k}{8\sigma_y^2} \right) \\
& = 16r(M-1)C_{k=0} \exp \left( -k \left( \frac{d_\mu^2}{8\sigma_x^2} + \frac{u_w^2 \tau_x^2}{8\sigma_y^2} \right) \right)
\end{aligned}$$

□

### Bounded Efficacy - Part B

*Proof.* We then deal with the second term  $P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}]$ ,

Part B:

$$\begin{aligned}
& P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] \\
& \leq P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \|\tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \\
& \text{(See Eq. A.13 for the derivation of } \tilde{\mathbf{w}}_\alpha \text{.)} \\
& = P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \|(\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1}(\mathbf{w}_\alpha - \mathbf{w}^*) + \mathbf{w}^* - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \\
& = P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \|(\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\boldsymbol{\Sigma}}_w)^{-1})(\mathbf{w}^* - \mathbf{w}_\alpha)\|^2 \|\mathbf{x}_{k+1}\|^2 | \mathbf{C}]
\end{aligned}$$

(Let  $\lambda_1(\mathbf{A})$  be the maximal eigenvalue of the matrix  $\mathbf{A}$ .)

$$\leq \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \lambda_1^2(\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}) \|\mathbf{x}_{k+1}\|^2 | \mathbf{C}]$$

(Recall that conditioned on  $\mathbf{C}$  we have  $L < \lambda_d(\bar{\Sigma}_w) < \lambda_1(\bar{\Sigma}_w) < U$ .)

$$\begin{aligned} &< \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \tilde{\pi}_\alpha \left(1 - \frac{1}{1 + k\delta_w U}\right)^2 \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \\ &= \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \left(1 - \frac{1}{1 + k\delta_w U}\right)^2 \\ &< \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \mathbb{E}_{\mathbf{x}_{k+1}} [\|\mathbf{x}_{k+1}\|^2] \left(1 - \frac{1}{1 + k\delta_w U}\right)^2 \\ &= \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \left(1 - \frac{1}{1 + k\delta_w U}\right)^2 \\ &= \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \left(\frac{k\delta_w U}{1 + k\delta_w U}\right)^2. \end{aligned}$$

Let  $t = k^{-\frac{1}{4}}$ . if  $\delta_w \ll 1$  s.t.  $I_U = \{k | U < 2(1 + \tau_x^2)\} \neq \emptyset$ , then when  $k \in I_U$  we have:

$$\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \left(\frac{k\delta_w U}{1 + k\delta_w U}\right)^2 < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\}.$$

□

### Bounded Efficacy - Part C

*Proof.* Finally, for the third term  $P(\neg \mathbf{C}) \mathbb{E}_{S_K} [\sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \neg \mathbf{C}]$ , Part C:

$$\begin{aligned} &P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \neg \mathbf{C} \right] \\ &\leq P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \|\tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right] \end{aligned}$$

(See Eq. A.13 for the derivation of  $\tilde{\mathbf{w}}_\beta$ .)

$$\begin{aligned} &= P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \|(\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}(\mathbf{w}_\beta - \mathbf{w}^*) + \mathbf{w}^* - \mathbf{w}_\alpha\|^2 \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right] \\ &< P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta (2\|(\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}(\mathbf{w}_\beta - \mathbf{w}^*)\|^2 + 2\|\mathbf{w}^* - \mathbf{w}_\alpha\|^2) \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right] \end{aligned}$$

$$\begin{aligned}
&< P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \left( 2 \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \lambda_1^2 \left( (\mathbf{I} + k \delta_w \bar{\Sigma}_w)^{-1} \right) + 2 \|\mathbf{w}^* - \mathbf{w}_\alpha\|^2 \right) \|\mathbf{x}_{k+1}\|^2 \right] \neg \mathbf{C} \\
&< P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta (2 \cdot 4 \cdot 1 + 2 \cdot 4) \|\mathbf{x}_{k+1}\|^2 \right] \neg \mathbf{C} \\
&= 16P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 \right] \neg \mathbf{C} \\
&< 16P(\neg \mathbf{C}) \mathbb{E}_{\mathbf{x}_{k+1}} [\|\mathbf{x}_{k+1}\|^2] \neg \mathbf{C}
\end{aligned}$$

(Notice  $\mathbf{C}$  is defined on  $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ )

$$< 16P(\neg \mathbf{C}) \mathbb{E}_{\mathbf{x}_{k+1}} [\|\mathbf{x}_{k+1}\|^2]$$

$$< 16(1 + d\tau_x^2)P(\neg \mathbf{C})$$

(Let  $t = k^{-\frac{1}{4}}$ .)

$$< 48(1 + d\tau_x^2) \exp \left( -\frac{k^{\frac{1}{2}}}{8} \right).$$

□

### Bounded Efficacy - Summary

*Proof.* Summarizing Part A, Part B, and Part C, we have:

$$\begin{aligned}
&\mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\
&< 16r(M-1)C_{k=0} \exp \left( -\frac{d_\mu^2 k}{8\sigma_x^2} \right) \exp \left( -\frac{u_w^2 \tau_x^2 k}{8\sigma_y^2} \right) \\
&\quad + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\} + 48(1 + d\tau_x^2) \exp \left( -\frac{k^{\frac{1}{2}}}{8} \right) \\
&= C_3 \exp \left( -k \left( \frac{d_\mu^2}{8\sigma_x^2} + \frac{u_w^2 \tau_x^2}{8\sigma_y^2} \right) \right) + 48(1 + d\tau_x^2) \exp \left( -\frac{k^{\frac{1}{2}}}{8} \right) \\
&\quad + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\}.
\end{aligned}$$

□

### The Particular Interval

The particular interval for the non-asymptotic bound is the union of  $I_\mu$ ,  $I_w$ , and  $I_U$ :

$$\begin{aligned} k &\leq \min\left\{\frac{1}{\delta_\mu} - 1, \frac{1}{\delta_w \tau_x^2}\right\} \\ 4\tau_x \gamma \sqrt{1 + k^{-\frac{1}{4}}} &< \frac{d_\mu^2}{2} \\ \mathbf{L}\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \mathbf{U}\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 &> \tau_x^2 u_w^2 / 2 \\ \mathbf{U} &< 2(1 + \tau_x^2). \end{aligned}$$

### A.13.2 Asymptotic Bound

This section proves the non-asymptotic bound in Theorem 5: Consider a next-token predictor attaining the optimal pretraining risk. As  $k \rightarrow \infty$ , ICL risk with biased labels is upper bounded by:

$$\mathbb{E}_{S_k}[\mathcal{L}_k^\alpha] < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2(1 + d\tau_x^2) + \frac{C_1}{k} \exp\left(C_2 k^{-\frac{1}{2}}\right) + O(k^{-2}),$$

where  $\mathcal{L}_k^\alpha = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^\alpha)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ , and  $C_1$  and  $C_2$  are constants depending on the prior setting,  $\tau_x$ , and  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ .

The proof of the asymptotic bound is heavily overlapped with the proof of the non-asymptotic bound. **We will hide the overlapped derivations with “(…)”.**

*Proof.* Assuming we are using in-context examples following Assumptions 3 and 4, i.e.,  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ ,  $y_i = \langle \mathbf{x}_i, \mathbf{w}^* \rangle$ ,  $\|\boldsymbol{\mu}^*\| = \|\mathbf{w}^*\| = 1$ , and we aim to retrieve the function  $\mathbf{w}_\alpha$  of the prior center  $(\boldsymbol{\mu}_\alpha, \mathbf{w}_\alpha)$  which is close to the in-context task. Let  $\mathcal{L}_k^\alpha$  indicate the squared risk  $(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ , where  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$

is the prediction of  $S_k \oplus x^{(k+1)}$  by the Bayes-optimal next-token predictor  $\mathcal{F}^*$ . In order to have an upper bound on the risk, we consider  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$  in two cases: (1) **C**:  $L < \lambda_d \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) \leq \lambda_1 \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) < U$  and  $\left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| < \tau_x \sqrt{\gamma(1+t)}$  (see Lemma 8 for  $t, \gamma, L$  and  $U$ ) and (2)  $\neg \mathbf{C}$ : at least one of the previous inequalities does not hold. Following Lemma 8, the probability of  $\neg \mathbf{C}$  is bounded by:  $P(\neg \mathbf{C}) \leq 3 \exp(-\frac{kt^2}{8})$ .

We start our upper bound analysis on the expected squared risk by splitting the risk into three parts:

$$\begin{aligned}
& \mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\
& (\dots) \\
& = P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \quad (\text{Part } A') \\
& \quad + P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] \quad (\text{Part } B') \\
& \quad + P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \neg \mathbf{C} \right]. \quad (\text{Part } C')
\end{aligned}$$

We will analyze three parts one by one in the following three sections respectively.

□

### Asymptotic Bound - Part $A'$

*Proof.* We firstly analyze the term  $P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}]$ , Part  $A'$ :

$$\begin{aligned}
& P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \mathbf{C} \right] \\
& (\dots)
\end{aligned}$$

$$\begin{aligned}
&< P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \left( \frac{2}{1 + k\delta_w L} + \|\mathbf{w}^* - \mathbf{w}_\alpha\| \right)^2 \\
&(\text{Notice } \|\mathbf{w}^* - \mathbf{w}_\alpha\| \leq 2.) \\
&\leq P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \left( \frac{4}{(1 + k\delta_w L)^2} + \frac{8}{1 + k\delta_w L} \right) \quad (\text{A.15})
\end{aligned}$$

$$+ P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \tilde{\pi}_\beta \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \|\mathbf{w}^* - \mathbf{w}_\alpha\|^2. \quad (\text{A.16})$$

Line A.16 will be merged with Part  $B'$  and analyzed in Sec. A.13.2. The current section will analyze line A.15. We start by analyzing  $P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right]$ . By Eqs. A.3, A.4, A.6, and Assumption 2(e) on  $\frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha}$ , we have:

$$\begin{aligned}
&P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \\
&(\dots) \\
&< P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \underbrace{\sum_{\beta \neq \alpha} r \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right)}_{\text{Part A'-1}} \right. \\
&\quad \cdot \underbrace{\exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{\mathbf{I} - (\mathbf{I} + k\delta_w \bar{\Sigma}_w)}^2}{2\sigma_w^2} \right)}_{\text{Part A'-2}} \\
&\quad \cdot \underbrace{\exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2}_{\text{Part A'-3}} \middle| \mathbf{C} \left. \right]
\end{aligned}$$

(Note that  $x_1, \dots, x_k$  are dependent on  $\mathbf{C}$  but  $x_{k+1}$  is not. Thus, we split them.)

In the following, we separately analyze the three terms, Part A'-1, Part A'-2, and Part A'-3. The high-level idea is that, as  $k$  increases, due to the concentration of Part A'-1 and Part A'-2, they can be upper bounded by a function of  $k$ . Then, regarding Part A'-1 and Part A'-2 as constant values (their upper bounds), the expectation of

Part A'-3 can be upper bounded.

**Part A'-1.** We first deal with Part A-1. When conditioned on case **C**, we have:

$$\begin{aligned}
& \frac{\sum_{i=1}^k (-\|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2)}{1 + (k+1)\delta_\mu} \\
& (\dots) \\
& < k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x\gamma\sqrt{1+t}}{1 + (k+1)\delta_\mu}.
\end{aligned}$$

With Assumption 4, we have  $d_\mu^2 \leq \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2$ . With Lemma 8, we have  $\gamma = \sqrt{\frac{d}{k}}$ . Let  $t = k^{\delta-\frac{1}{2}}$  and  $0 < \delta < \frac{1}{2}$ , we have:

$$k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x\gamma\sqrt{1+t}}{1 + (k+1)\delta_\mu} = -\frac{d_\mu^2}{\delta_\mu} + \frac{4\tau_x\sqrt{d}}{\delta_\mu} k^{-\frac{1}{2}} + O(k^{-1}).$$

**Part A'-2.** We then deal with Part A'-2. When conditioned on case **C**, we have:

$$\begin{aligned}
& -\|\mathbf{w}_\beta - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2 \\
& (\dots) \\
& < -\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 \left(1 - \frac{1}{1 + k\delta_w\mathbf{L}}\right) + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \left(1 - \frac{1}{1 + k\delta_w\mathbf{U}}\right) \\
& = -(\|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2) + \left(\frac{\|\mathbf{w}_\beta - \mathbf{w}^*\|^2}{1 + k\delta_w\mathbf{L}} - \frac{\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2}{1 + k\delta_w\mathbf{U}}\right).
\end{aligned}$$

With Assumption 4, we have  $d_w^2 \leq \|\mathbf{w}_\beta - \mathbf{w}^*\|^2 - \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2$ . Lemma 8 gives the definitions of **L** and **U**. Let  $t = k^{\delta-\frac{1}{2}}$  and  $0 < \delta < \frac{1}{2}$ , we have:

$$\begin{aligned}
& = -d_w^2 + \left(\frac{\|\mathbf{w}_\beta - \mathbf{w}^*\|^2}{k\delta_w\tau_x^2} - \frac{\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2}{k\delta_w(1 + \tau_x^2)}\right) + O(k^{-2}) \\
& < -d_w^2 + \frac{\|\mathbf{w}_\beta - \mathbf{w}^*\|^2}{k\delta_w\tau_x^2} + O(k^{-2})
\end{aligned}$$

$$< -d_w^2 + \frac{4}{\delta_w \tau_x^2} k^{-1} + O(k^{-2}).$$

**Part A'-3.** We finally deal with Part A'-3. Part A'-3 is independent to case **C**, and we have:

$$\begin{aligned} & P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \\ & (\dots) \\ & = C_{k=0}. \end{aligned}$$

**Summary of Part A'.** Thus, summarizing Part A'-1, Part A'-2, and Part A'-3, we have:

$$\begin{aligned} & P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta \neq \alpha} \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C} \right] \left( \frac{4}{(1 + k\delta_w \mathbf{L})^2} + \frac{8}{1 + k\delta_w \mathbf{L}} \right) \\ & < P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \underbrace{\sum_{\beta \neq \alpha} r \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right)}_{\text{Part A'-1}} \right. \\ & \quad \cdot \underbrace{\exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{I - (I + k\delta_w \bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{I - (I + k\delta_w \bar{\Sigma}_w)^{-1}}^2}{2\sigma_w^2} \right)}_{\text{Part A'-2}} \\ & \quad \cdot \underbrace{\exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \middle| \mathbf{C}}_{\text{Part A'-3}} \\ & \quad \cdot \left( \frac{4}{(1 + k\delta_w \mathbf{L})^2} + \frac{8}{1 + k\delta_w \mathbf{L}} \right) \\ & \text{(Notice } \lim_{k \rightarrow \infty} \mathbf{L} = \lim_{k \rightarrow \infty} \tau_x^2 \left( 1 - \frac{t}{2} - \gamma \right)^2 - 2\tau_x \gamma \sqrt{1 + t} = \tau_x^2.) \\ & < r \sum_{\beta \neq \alpha} \exp \left( \frac{-\frac{d_\mu^2}{\delta_\mu} + \frac{4\tau_x \sqrt{d}}{\delta_\mu} k^{-\frac{1}{2}} + O(k^{-1})}{2\sigma_x^2} \right) \exp \left( \frac{-d_w^2 + \frac{4}{\delta_w \tau_x^2} k^{-1} + O(k^{-2})}{2\sigma_w^2} \right) C_{k=0} \left( \frac{8}{k\delta_w \tau_x^2} + O(k^{-2}) \right) \end{aligned}$$

$$\begin{aligned}
&= r(M-1)C_{k=0} \exp\left(\frac{-d_{\mu}^2 + 4\tau_x \sqrt{d}k^{-\frac{1}{2}} + O(k^{-1})}{2\sigma_{\mu}^2}\right) \exp\left(\frac{-d_w^2 + \frac{4}{\delta_w \tau_x^2}k^{-1} + O(k^{-2})}{2\sigma_w^2}\right) \left(\frac{8}{k\delta_w \tau_x^2} + O(k^{-2})\right) \\
&= \frac{8r(M-1)C_{k=0}}{k\delta_w \tau_x^2} \exp\left(\frac{-d_{\mu}^2 + 4\tau_x \sqrt{d}k^{-\frac{1}{2}} + O(k^{-1})}{2\sigma_{\mu}^2}\right) \exp\left(\frac{-d_w^2 + \frac{4}{\delta_w \tau_x^2}k^{-1} + O(k^{-2})}{2\sigma_w^2}\right) + O(k^{-2}) \\
&= \frac{8r(M-1)C_{k=0}}{k\delta_w \tau_x^2} \exp\left(\frac{-d_{\mu}^2 + 4\tau_x \sqrt{d}k^{-\frac{1}{2}}}{2\sigma_{\mu}^2}\right) \exp\left(\frac{-d_w^2}{2\sigma_w^2}\right) + O(k^{-2})
\end{aligned}$$

□

### Asymptotic Bound - Part B'

*Proof.* We then deal with the second term  $P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}\langle \tilde{\mathbf{w}}_{\beta} - \mathbf{w}_{\alpha}, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}]$ ,

Part B':

$$\begin{aligned}
&P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}\langle \tilde{\mathbf{w}}_{\alpha} - \mathbf{w}_{\alpha}, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] \\
&(\dots) \\
&< \|\mathbf{w}_{\alpha} - \mathbf{w}^*\|^2 P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \left(1 - \frac{1}{1 + k\delta_w \mathbf{U}}\right)^2.
\end{aligned}$$

We add the line A.16 in Sec. A.13.2 back:

$$\begin{aligned}
&P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}(\langle \tilde{\mathbf{w}}_{\alpha} - \mathbf{w}_{\alpha}, \mathbf{x}_{k+1} \rangle)^2 | \mathbf{C}] + \underbrace{P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}\left[\sum_{\beta \neq \alpha} \tilde{\pi}_{\beta}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}\right]}_{\text{line A.16 in Sec. A.13.2}} \|\mathbf{w}^* - \mathbf{w}_{\alpha}\|^2 \\
&< \|\mathbf{w}_{\alpha} - \mathbf{w}^*\|^2 P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \left(1 - \frac{1}{1 + k\delta_w \mathbf{U}}\right)^2 \\
&\quad + P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}\left[\sum_{\beta \neq \alpha} \tilde{\pi}_{\beta}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}\right] \|\mathbf{w}^* - \mathbf{w}_{\alpha}\|^2 \\
&\leq \|\mathbf{w}_{\alpha} - \mathbf{w}^*\|^2 P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_{\alpha}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}] \\
&\quad + \|\mathbf{w}_{\alpha} - \mathbf{w}^*\|^2 P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}\left[\sum_{\beta \neq \alpha} \tilde{\pi}_{\beta}\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}\right] \\
&(\text{Notice } \sum_{\beta=1}^M \tilde{\pi}_{\beta} = 1) \\
&= \|\mathbf{w}_{\alpha} - \mathbf{w}^*\|^2 P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\|\mathbf{x}_{k+1}\|^2 | \mathbf{C}]
\end{aligned}$$

$$\begin{aligned}
&< \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 \mathbb{E}_{\mathbf{x}_{k+1}} \left[ \|\mathbf{x}_{k+1}\|^2 \right] \\
&= \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2)
\end{aligned}$$

□

### Asymptotic Bound - Part $C'$

*Proof.* Finally for the third term  $P(\neg \mathbf{C}) \mathbb{E}_{S_K} [\sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \neg \mathbf{C}]$ , Part  $C'$ :

$$\begin{aligned}
&P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\beta=1}^M \tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 \middle| \neg \mathbf{C} \right] \\
&(\dots) \\
&< 16(1 + d\tau_x^2) P(\neg \mathbf{C}) \\
&(\text{Let } t = k^{\delta - \frac{1}{2}}.) \\
&< 48(1 + d\tau_x^2) \exp\left(-\frac{k^{2\delta}}{8}\right).
\end{aligned}$$

□

### Asymptotic Bound - Summary

*Proof.* Summarizing Part  $A'$ , Part  $B'$ , and Part  $C'$ , we have:

$$\begin{aligned}
&\mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\
&< \frac{8r(M-1)C_{k=0}}{k\delta_w\tau_x^2} \exp\left(\frac{-d_\mu^2 + 4\tau_x\sqrt{d}k^{-\frac{1}{2}}}{2\sigma_\mu^2}\right) \exp\left(\frac{-d_w^2}{2\sigma_w^2}\right) + O(k^{-2}) \\
&\quad + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) + 48(1 + d\tau_x^2) \exp\left(-\frac{k^{2\delta}}{8}\right) \\
&= \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) + \frac{8r(M-1)C_{k=0}}{k\delta_w\tau_x^2} \exp\left(\frac{-d_\mu^2 + 4\tau_x\sqrt{d}k^{-\frac{1}{2}}}{2\sigma_\mu^2}\right) \exp\left(\frac{-d_w^2}{2\sigma_w^2}\right)
\end{aligned}$$

$$\begin{aligned}
& + O(k^{-2}) \\
& = \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2(1 + d\tau_x^2) + \frac{C_1}{k} \exp(C_2 k^{-\frac{1}{2}}) + O(k^{-2})
\end{aligned}$$

□

## A.14 Proof of Lemma 6

In this subsection, we introduce the proof of Lemma 6. We first give the full version of the lemma:

**Lemma 6** (Upper Bound for Zero-Shot ICL). *Assume a next-token predictor attains the optimal pretraining risk, and Assumption 2 has only two components  $\alpha$  and  $\beta$ , with centers  $(\boldsymbol{\mu}_\alpha, \mathbf{w}_\alpha) = (-\boldsymbol{\mu}_\beta, -\mathbf{w}_\beta)$ . When performing ICL with  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^* | \tau_x^2 \mathbf{I})$ , assume  $\|\boldsymbol{\mu}^*\| = 1$ , and  $y_i = 0$ , i.e.,  $y_i$  has the same preference to prior component  $\alpha$  as  $\beta$ . When  $\delta_\mu$  and  $\delta_w$  are sufficiently small, there is a particular interval for  $k$  that ICL risk is upper bounded by:*

$$\mathbb{E}_{S_k}[\mathcal{L}_k^\alpha] < C_4 \exp\left(-\frac{d_\mu^2 k}{8\sigma_x^2}\right) + 12(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right) + (1 + d\tau_x^2) \min\{1, k^2 \delta_w^2 (1 + \tau_x^2)^2\},$$

where  $\mathcal{L}_k^\alpha = (\mathcal{F}(S_k \oplus x^{(k+1)}) - y_{k+1}^\alpha)^2 = (\mathcal{F}(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ ,  $C_4$  is a constant depending on the prior,  $\tau_x$ , and  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ . When  $k$  is small, the first and second terms dominate and exponential decay. When  $k$  is large, the third term dominates and increases.

*Proof.* The proof techniques are similar to the proof techniques used in Sec. A.13.1. Assuming we are using in-context examples following  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$ ,  $\|\boldsymbol{\mu}^*\| = 1$ ,  $y_i = 0$ , i.e.,  $\mathbf{w}^* = \mathbf{0}$ , and we aim to retrieve the function  $\mathbf{w}_\alpha$  of the prior center

$(\boldsymbol{\mu}_\alpha, \mathbf{w}_\alpha)$  which is close to the in-context task. Let  $\mathcal{L}_k^\alpha$  indicate the squared loss  $(\mathcal{F}^*(S_k \oplus x^{(k+1)}) - \langle \mathbf{x}_{k+1}, \mathbf{w}_\alpha \rangle)^2$ , where  $\mathcal{F}^*(S_k \oplus x^{(k+1)})$  is the prediction of  $S_k \oplus x^{(k+1)}$  by the Bayes-optimal next-token predictor  $\mathcal{F}^*$ . In order to have an upper bound on the loss, we consider  $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^*, \tau_x^2 \mathbf{I})$  in two cases: (1) **C**:  $L < \lambda_d \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) \leq \lambda_1 \left( \frac{\sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top}{k} \right) < U$  and  $\left\| \frac{\sum_{i=1}^k \boldsymbol{\epsilon}_i}{k} \right\| < \tau_x \sqrt{\gamma(1+t)}$  (see Lemma 8 for  $t, \gamma, L$  and  $U$ ) and (2)  $\neg \mathbf{C}$ : at least one of the previous inequalities does not hold. Following Lemma 8, the probability of  $\neg \mathbf{C}$  is bounded by:  $P(\neg \mathbf{C}) \leq 3 \exp(-\frac{kt^2}{8})$ .

Similar to Sec. A.13.1, we split the expected squared loss into three parts:

$$\begin{aligned}
& \mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\
& < P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] & (\text{Part } A'') \\
& + P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] & (\text{Part } B'') \\
& + P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa \langle \tilde{\mathbf{w}}_\kappa - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \neg \mathbf{C} \right]. & (\text{Part } C'')
\end{aligned}$$

□

### A.14.1 Proof of Lemma 6: Part A''

*Proof.* We first analyze the term  $P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}]$ , Part A''. Similar to Sec. A.13.1, we have:

$$\begin{aligned}
& P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} [\tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] \\
& < P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \frac{\tilde{\pi}_\beta}{\tilde{\pi}_\alpha} \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C} \right] \cdot \left( \frac{2}{1 + k\delta_w L} + \|\mathbf{w}^* - \mathbf{w}_\alpha\| \right)^2 \\
& < P(\mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ r \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \right]
\end{aligned}$$

$$\begin{aligned} & \cdot \exp \left( \frac{-\|\mathbf{w}_\beta - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2 + \|\mathbf{w}_\alpha - \mathbf{w}^*\|_{I-(I+k\delta_w\bar{\Sigma}_w)^{-1}}^2}{2\sigma_w^2} \right) \\ & \cdot \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \Big| \mathbf{C} \Big] \cdot \left( \frac{2}{1 + k\delta_w L} + \|\mathbf{w}^* - \mathbf{w}_\alpha\| \right)^2 \end{aligned}$$

(Notice  $\mathbf{w}^* = \mathbf{0}$ ,  $\mathbf{w}_\beta = -\mathbf{w}_\alpha$ .)

$$\begin{aligned} & = rP(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \right. \\ & \quad \cdot \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \Big| \mathbf{C} \Big] \cdot 3^2 \\ & = 9rP(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \underbrace{\exp \left( \frac{-\sum_{i=1}^k \|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \sum_{i=1}^k \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right)}_{A''-1} \right. \\ & \quad \cdot \underbrace{\exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \Big| \mathbf{C} \Big]}_{A''-3} \Big]. \end{aligned}$$

Same to Sec. A.13.1, when conditioned on case  $\mathbf{C}$ , for Part  $A''-1$  we have:

$$\frac{\sum_{i=1}^k (-\|\boldsymbol{\mu}_\beta - \mathbf{x}_i\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_i\|^2)}{1 + (k+1)\delta_\mu} < k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x \gamma \sqrt{1+t}}{1 + (k+1)\delta_\mu}.$$

Let  $t = k^{-\frac{1}{4}}$ . Recall in Assumption 4, we have  $\forall \beta \neq \alpha, \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 \geq d_\mu^2$ .

If  $\delta_\mu \ll 1$  s.t.  $I_\mu = \{k | (k+1)\delta_\mu \leq 1 \text{ and } \frac{d_\mu^2}{2} > 4\tau_x \gamma \sqrt{1 + k^{-\frac{1}{4}}}\} \neq \emptyset$ , then when  $k \in I_\mu$

we have:

$$k \frac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}^*\|^2 - \|\boldsymbol{\mu}_\beta - \boldsymbol{\mu}^*\|^2 + 4\tau_x \gamma \sqrt{1+t}}{1 + (k+1)\delta_\mu} < -\frac{d_\mu^2}{4}.$$

Same to Sec. A.13.1, when conditioned on case  $\mathbf{C}$ , for Part  $A''-3$  we have:

$$P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \exp \left( \frac{-\|\boldsymbol{\mu}_\beta - \mathbf{x}_{k+1}\|^2 + \|\boldsymbol{\mu}_\alpha - \mathbf{x}_{k+1}\|^2}{2\sigma_x^2(1 + (k+1)\delta_\mu)} \right) \|\mathbf{x}_{k+1}\|^2 \Big| \mathbf{C} \right] = C_{k=0}.$$

As a summary of the above analysis, we have:

$$P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_\beta \langle \tilde{\mathbf{w}}_\beta - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] < 9rC_{k=0} \exp\left(-\frac{d_\mu^2 k}{8\sigma_x^2}\right).$$

□

### A.14.2 Proof of Lemma 6: Part B''

*Proof.* We then deal with the second term  $P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_\alpha (\langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle)^2 | \mathbf{C}]$ , Part B''. The analysis is exactly the same as Sec. A.13.1, and we have:

$$P(\mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\tilde{\pi}_\alpha \langle \tilde{\mathbf{w}}_\alpha - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \mathbf{C}] < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \left( \frac{k\delta_w \mathbf{U}}{1 + k\delta_w \mathbf{U}} \right)^2.$$

Let  $t = k^{-\frac{1}{4}}$ . if  $\delta_w \ll 1$  s.t.  $I_U = \{k | \mathbf{U} < 2(1 + \tau_x^2)\} \neq \emptyset$ , then when  $k \in I_U$  we have:

$$\|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \left( \frac{k\delta_w \mathbf{U}}{1 + k\delta_w \mathbf{U}} \right)^2 < \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \min\{1, 4k^2\delta_w^2(1 + \tau_x^2)^2\}.$$

□

### A.14.3 Proof of Lemma 6: Part C''

*Proof.* Finally, for the third term  $P(\neg \mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}}[\sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa \langle \tilde{\mathbf{w}}_\kappa - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle^2 | \neg \mathbf{C}]$ , Part C''. Similar to Sec. A.13.1, we have:

$$\begin{aligned} & P(\neg \mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa (\langle \tilde{\mathbf{w}}_\kappa - \mathbf{w}_\alpha, \mathbf{x}_{k+1} \rangle)^2 \middle| \neg \mathbf{C} \right] \\ & < P(\neg \mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa \left( 2\|(\mathbf{I} + k\delta_w \bar{\Sigma}_w)^{-1}(\mathbf{w}_\kappa - \mathbf{w}^*)\|^2 + 2\|\mathbf{w}^* - \mathbf{w}_\alpha\|^2 \right) \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right] \end{aligned}$$

(Recall  $\mathbf{w}^* = \mathbf{0}$ .)

$$< P(\neg \mathbf{C})\mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa (2 \cdot 1 \cdot 1 + 2 \cdot 1) \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right]$$

$$= 4P(\neg \mathbf{C}) \mathbb{E}_{S_k \oplus x^{(k+1)}} \left[ \sum_{\kappa \in \{\alpha, \beta\}} \tilde{\pi}_\kappa \|\mathbf{x}_{k+1}\|^2 \middle| \neg \mathbf{C} \right]$$

$$< 4P(\neg \mathbf{C}) \mathbb{E}_{\mathbf{x}_{k+1}} [\|\mathbf{x}_{k+1}\|^2 | \neg \mathbf{C}]$$

(Notice  $\mathbf{C}$  is defined on  $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ .)

$$< 4P(\neg \mathbf{C}) \mathbb{E}_{\mathbf{x}_{k+1}} [\|\mathbf{x}_{k+1}\|^2]$$

$$< 4(1 + d\tau_x^2)P(\neg \mathbf{C})$$

(Let  $t = k^{-\frac{1}{4}}$ .)

$$< 12(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right).$$

□

#### A.14.4 Proof of Lemma 6: Summary

*Proof.* Summarizing Part  $A''$ , Part  $B''$ , and Part  $C''$ , we have:

$$\begin{aligned} & \mathbb{E}_{S_k \oplus x^{(k+1)}} [\mathcal{L}_k^\alpha] \\ & < 9rC_{k=0} \exp\left(-\frac{d_\mu^2 k}{8\sigma_x^2}\right) + \|\mathbf{w}_\alpha - \mathbf{w}^*\|^2 (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\} \\ & \quad + 12(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right) \\ & = 9rC_{k=0} \exp\left(-\frac{d_\mu^2 k}{8\sigma_x^2}\right) + (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\} + 12(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right) \\ & = C_4 \exp\left(-\frac{d_\mu^2 k}{8\sigma_x^2}\right) + 12(1 + d\tau_x^2) \exp\left(-\frac{k^{\frac{1}{2}}}{8}\right) + (1 + d\tau_x^2) \min\{1, 4k^2 \delta_w^2 (1 + \tau_x^2)^2\}. \end{aligned}$$

□

### A.14.5 The Particular Interval

The particular interval for the risk bound revealing bounded efficacy is the union of  $I_\mu$  and  $I_U$ :

$$\begin{aligned} k &\leq \frac{1}{\delta_\mu} - 1 \\ 4\tau_x\gamma\sqrt{1 + k^{-\frac{1}{4}}} &< \frac{d_\mu^2}{2} \\ U &< 2(1 + \tau_x^2). \end{aligned}$$

## A.15 Toy Example for Component Shifting and Component Re-weighting

We study how in-context examples affect the prediction of ICL by a pretrained Bayes-optimal next-token predictor and how the pretraining distribution affects this phenomenon. Assume the next-token predictor  $f$  is initially pretrained on a dataset distribution to produce the minimum risk minimizer  $f^*$ , and then the pretrained  $f^*$  is used to predict the next token  $y$  of the token  $x$ . Instead of direct inference via  $f^*(x)$ , we consider inference with additional  $k$  in-context examples  $\{x_i\}_{i=1}^k$  via the format  $f^*([x_1, \dots, x_k, x])$ . We aim to theoretically examine the effect of in-context examples  $\{x_i\}_{i=1}^k$  on the prediction  $f^*([x_1, \dots, x_k, x])$ . While the formal problem setting may involve verbose math, this demo section illustrates the basic phenomenon for better delivering our work.

The following demo subsections are organized as follows. We first introduce the problem setting in Sec. A.15.1. We then connect ICL with Bayesian inference in

Sec. A.15.2. Further, we introduce the assumptions for the pretraining dataset in Sec. A.15.3. Finally, we derive a closed-form posterior and introduce two phenomena, “Component Shifting” and “Component Re-weighting” in Sec. A.15.4.

### A.15.1 Toy Example: Pretraing Data Generative Model

ICL involves two important components: the pretraining dataset, and the next-token predictor supporting varied input lengths. We assume the next-token predictor  $f : \cup_{k \in \{0, \dots, K-1\}} \mathcal{R}^{k \times 1} \rightarrow \mathcal{R}^{1 \times 1}$  can fit the pretraining distribution exactly with enough data and expressivity. To generate a training sample, we first sample a task  $\mu$  from underlying task distribution  $\mathcal{D}_\mu$ , and then we generate tokens of the sequence from a distribution  $\mathcal{D}_x(\mu)$  based on the task  $\mu$ . The sample generation process is described as follows:

**Assumption 9** (Demo: Pretraining Data Generative Model). *Given a task prior distribution  $\mathcal{D}_\mu$ , and a conditioned  $x$  sampler  $\mathcal{D}_x(\mu)$  conditioned on task  $\mu$ , the process of generating a sequence  $S_K = [x_1, x_2, \dots, x_K]$  with length  $K$  follows:*

- (a) *Sample a task  $\mu$  from the task prior:  $\mu \sim \mathcal{D}_\mu$ , and the probability of  $\mu$  is indicated by  $P(\mu)$ ;*
- (b) *Sample  $K$  samples, each denoted by  $x_i$ , from the chosen task: For  $i \in \{1, 2, \dots, K\}$ ,  $x_i \sim \mathcal{D}_x(\mu)$ , and the probability of  $x_i = x$  is indicated by  $P(x|\mu)$ ;*
- (c) *Define a Sequence  $S_k$ : For capital  $K$ ,  $S_K = [x_1, \dots, x_K]$ ; and for lowercase  $k$ , the sequence of the first  $k$  demonstrations of  $S_K$  is indicated by  $S_k = [x_1, \dots, x_k]$ , e.g.,  $S_2 = [x_1, x_2]$ .*

The generation process is related to real-world scenarios via two points: (i) For sampling step 9(a), the LM is trained on varied tasks; (ii) For sampling step 9(b), when one person/agent produces texts for one task, the generated text could be noisy. For instance, given a task such as describing a football game, one person has multiple ways to describe it.

### A.15.2 Toy Example: Bayes-Optimal Next-Token Predictor

Now we consider training  $f(\cdot)$  using sample  $S_K$  generated via the above generation process 9:

$$\mathcal{L}(f) = \mathbb{E}_{S_K} \left[ \frac{1}{K} \sum_{k=0}^{K-1} (f(S_k) - x_{k+1})^2 \right] = \mathbb{E}_{\mu \sim \mathcal{D}_\mu} \left[ \mathbb{E}_{\substack{x_i \sim \mathcal{D}(\mu), \\ i \in \{1, \dots, K\}}} \left[ \frac{1}{K} \sum_{k=0}^{K-1} (f(S_k) - x_{k+1})^2 \middle| \mu \right] \right].$$

$f$  can be viewed as  $K$  separate models  $f_0, \dots, f_{K-1}$ , where  $f_k$  takes a sequence of  $k$  tokens as input. Therefore, when the model  $f$  has enough expressivity, the optimization problem  $f^* = \operatorname{argmin}_f \mathcal{L}(f)$  could be regarded as  $K$  different optimization problems:

$$f_k^* = \operatorname{argmin}_{f_k} \mathbb{E}_{S_K} [(f(S_k) - x_{k+1})^2], \forall k \in \{0, \dots, K-1\}.$$

Thus, the solution  $f_k^*$  for each  $k$  is a minimum mean square error (MMSE) estimator [110, page 63], and the prediction of  $f^*(S_k)$  satisfies:

$$f^*(S_k) = \mathbb{E}_{S_K} [x_{k+1} | S_k] = \mathbb{E}_{\mu \sim \mathcal{D}_\mu} \left[ \mathbb{E}_{\substack{x_i \sim \mathcal{D}(\mu), \\ i \in \{1, \dots, K\}}} [x_{k+1} | \mu, S_k] | S_k \right] = \mathbb{E}_{\mu \sim \mathcal{D}_\mu} \left[ \mathbb{E}_{x_{k+1} \sim \mathcal{D}(\mu)} [x_{k+1} | \mu] | S_k \right]. \quad (\text{A.17})$$

The prediction  $f^*(S_k)$  is the expectation of  $\mathbb{E}_{x_{k+1} \sim \mathcal{D}(\mu)} [x_{k+1} | \mu]$  on the task posterior observing  $S_k$ .

### A.15.3 Toy Example: Gaussian Assumptions on Pretraining Data Generative Model

In Sec. A.15.2, we connect ICL with Bayesian inference, and in Eq. A.17, we observe that the prediction  $f^*(S_k)$  depends on the posterior. We are interested in how the in-context examples affect the prediction and the posterior. We make assumptions on the pretraining dataset to have a closed-form expression of the posterior facilitating further analyses:

**Assumption 10** (Demo: Gaussian Assumptions for Generative Model for Pretraining Data).

- (a) *Task distribution:  $\mu \sim \mathcal{D}_\mu$ ,  $P(\mu) = \sum_{m=1}^M \pi_m P(\mu|T_m)$ , where  $T_m$  is the  $m^{\text{th}}$  mixture component of the Gaussian mixture, i.e.,  $P(\mu|T_m) = \mathcal{N}(\mu|\mu_m, \sigma^2)$ , and  $\pi_m$  is the corresponding mixture weight.  $\sum_{m=1}^M \pi_m = 1$ ,  $0 < \pi_m < 1$ ,  $\mu_m$  is the center of the mixture component  $T_m$ , and all components share the same covariance matrix controlled by  $\sigma$ ;*
- (b) *Token distribution:  $x \sim \mathcal{D}_x(\mu)$ ,  $P(x|\mu) = \mathcal{N}(x|\mu_m, \tau^2)$ .*

### A.15.4 Toy Example: Posterior Analysis

With Assumption 10, we derive the closed-form expression of the posterior as follows:

$$P(\mu|S_k) \propto \sum_{m=1}^M \tilde{\pi}_m \mathcal{N}(\mu|\tilde{\mu}_m, \tilde{\sigma}^2). \quad (\text{A.18})$$

$$(\tilde{\pi}_m = \pi_m \exp \left( \frac{k \left( \mu_m - \frac{\sum_{i=1}^k x_i}{k} \right)^2}{2(\tau^2 + k\sigma^2)} \right), \tilde{\mu}_m = \frac{\tau^2 \mu_m + \sigma^2 \sum_{i=1}^k x_i}{\tau^2 + k\sigma^2}, \tilde{\sigma}^2 = \frac{\tau^2 \sigma^2}{\tau^2 + k\sigma^2})$$

See Sec. A.15.5 for proof details. From Eq. A.18, we observe two factors when comparing the posterior with the prior in Assumption 10: (i) Component Shifting: after observing  $S_k = [x_1, x_2, \dots, x_k]$ , the center of each mixture component is shifted to  $\frac{\tau^2 \mu_m + \sigma^2 \sum_{i=1}^k x_i}{\tau^2 + k\sigma^2}$ ; (ii) Component Re-weighting: the mixture weight  $\pi_m$  of each mixture component is re-weighted by multiplying  $\exp\left(-\frac{k\left(\mu_m - \frac{\sum_{i=1}^k x_i}{k}\right)^2}{2(\tau^2 + k\sigma^2)}\right)$  (which needs to be further normalized so that re-weighted mixture weights sum to 1). Fig. A.17 illustrates the phenomena of Component Shifting and Component Re-weighting by observing in-context examples.

### A.15.5 Proof of Posterior Derivation in Toy Example

In this section, we give a detailed derivation of the posterior in Eq. A.18 of Sec. A.15.4:

$$\begin{aligned}
 P(\mu|S_k) &\propto P(\mu, S_k) \\
 &= P(S_k|\mu)P(\mu) \\
 &= (\prod_{i=1}^k P(x_i|\mu))P(\mu) \\
 &= \sum_{m=1}^M \pi_m \mathcal{N}(\mu|\mu_m, \sigma^2) (\prod_{i=1}^k \mathcal{N}(x_i|\mu, \tau^2)).
 \end{aligned}$$

We then show  $\mathcal{N}(\mu|\mu_m, \sigma^2) (\prod_{i=1}^k \mathcal{N}(x_i|\mu, \tau^2))$  is proportional to a Gaussian distribution:

$$\begin{aligned}
 &\log\left(\mathcal{N}(\mu|\mu_m, \sigma^2) \cdot \prod_{i=1}^k \mathcal{N}(x_i|\mu, \tau^2)\right) \\
 &= \left(\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{(\mu - \mu_m)^2}{2\sigma^2}\right) + \sum_{i=1}^k \left(\log\left(\frac{1}{\sqrt{2\pi}\tau}\right) - \frac{(x_i - \mu)^2}{2\tau^2}\right) \\
 &(\text{Let } C_{10} = \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + k \log\left(\frac{1}{\sqrt{2\pi}\tau}\right))
 \end{aligned}$$

$$\begin{aligned}
&= C_{10} - \frac{(\mu - \mu_m)^2}{2\sigma^2} - \sum_{i=1}^k \frac{(x_i - \mu)^2}{2\tau^2} \\
&= C_{10} - \frac{1}{2\tau^2\sigma^2} \left( \tau^2(\mu - \mu_m)^2 + \sigma^2 \sum_{i=1}^k (x_i - \mu)^2 \right) \\
&\text{(Abbreviate } \sum_{i=1}^k \text{ as } \sum \text{ for simplicity.)} \\
&= C_{10} - \frac{1}{2\tau^2\sigma^2} \left( \mu^2(\tau^2 + k\sigma^2) - 2\mu(\tau^2\mu_m + \sigma^2 \sum x_i) + (\tau^2\mu_m^2 + \sigma^2 \sum x_i^2) \right) \\
&= C_{10} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \left( \left( \mu - \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 \right. \\
&\quad \left. + \frac{\tau^2\mu_m^2 + \sigma^2 \sum x_i^2}{\tau^2 + k\sigma^2} - \left( \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 \right) \\
&= C_{10} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \left( \left( \mu - \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 \right. \\
&\quad \left. + \frac{(\tau^2\mu_m^2 + \sigma^2 \sum x_i^2)(\tau^2 + k\sigma^2) - (\tau^2\mu_m + \sigma^2 \sum x_i)^2}{(\tau^2 + k\sigma^2)^2} \right) \\
&= C_{10} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \left( \left( \mu - \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 \right. \\
&\quad \left. + \frac{k\sigma^2\tau^2\mu_m^2 + \sigma^2 \sum x_i^2(\tau^2 + k\sigma^2) - 2\mu_m\tau^2\sigma^2 \sum x_i - (\sigma^2 \sum x_i)^2}{(\tau^2 + k\sigma^2)^2} \right) \\
&\text{(Let } C_{11} = C_{10} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \cdot \frac{\sigma^2 \sum x_i^2(\tau^2 + k\sigma^2) - (\sigma^2 \sum x_i)^2 - \tau^2\sigma^2(\sum x_i)^2/k}{(\tau^2 + k\sigma^2)^2} \text{.)} \\
&= C_{11} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \left( \left( \mu - \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 \right. \\
&\quad \left. + \frac{k\sigma^2\tau^2\mu_m^2 - 2\mu_m\tau^2\sigma^2 \sum x_i + \tau^2\sigma^2(\sum x_i)^2/k}{(\tau^2 + k\sigma^2)^2} \right) \\
&= C_{11} - \frac{\tau^2 + k\sigma^2}{2\tau^2\sigma^2} \left( \left( \mu - \frac{\tau^2\mu_m + \sigma^2 \sum x_i}{\tau^2 + k\sigma^2} \right)^2 + \frac{k\tau^2\sigma^2}{(\tau^2 + k\sigma^2)^2} \cdot \left( \mu_m - \frac{\sum x_i}{k} \right)^2 \right)
\end{aligned}$$

$$= C_{11} - \frac{k \left( \mu_m - \frac{\sum_{i=1}^k x_i}{k} \right)^2}{2(\tau^2 + k\sigma^2)} - \frac{\left( \mu - \frac{\tau^2 \mu_m + \sigma^2 \sum_{i=1}^k x_i}{\tau^2 + k\sigma^2} \right)^2}{2 \cdot \frac{\tau^2 \sigma^2}{\tau^2 + k\sigma^2}}.$$

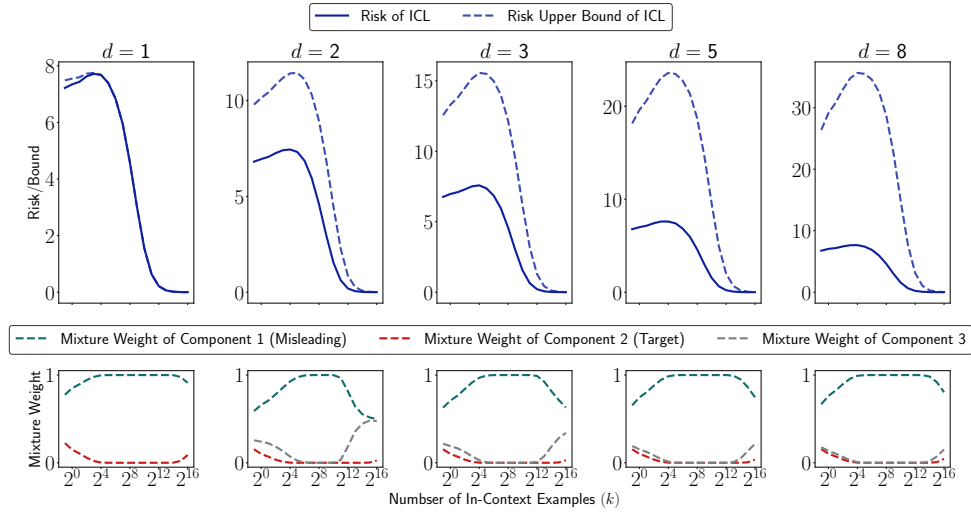
Notice  $C_{11}$  is independent to  $m, \forall m \in [M]$  and  $\mu$ . Therefore, we have:

$$\pi_m \cdot \mathcal{N}(\mu|\mu_m, \sigma^2) \cdot \prod_{i=1}^k \mathcal{N}(x_i|\mu, \tau^2) \propto \tilde{\pi}_m \cdot \mathcal{N}(\mu|\tilde{\mu}_m, \tilde{\sigma}^2),$$

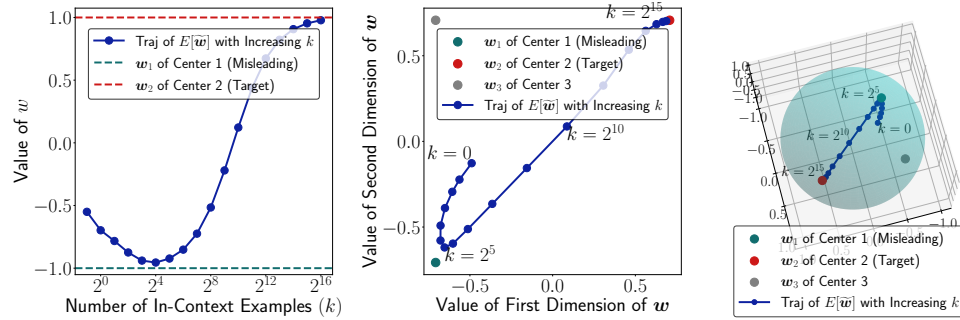
where  $\tilde{\pi}_m = \pi_m \exp \left( -\frac{k \left( \mu_m - \frac{\sum_{i=1}^k x_i}{k} \right)^2}{2(\tau^2 + k\sigma^2)} \right)$ ,  $\tilde{\mu}_m = \frac{\tau^2 \mu_m + \sigma^2 \sum_{i=1}^k x_i}{\tau^2 + k\sigma^2}$ , and  $\tilde{\sigma}^2 = \frac{\tau^2 \sigma^2}{\tau^2 + k\sigma^2}$ .

Thus:

$$\begin{aligned} P(\mu|S_k) &\propto \sum_{m=1}^M \pi_m \mathcal{N}(\mu|\mu_m, \sigma^2) (\prod_{i=1}^k \mathcal{N}(x_i|\mu, \tau^2)) \\ &\propto \tilde{\pi}_m \mathcal{N}(\mu|\tilde{\mu}_m, \tilde{\sigma}^2). \end{aligned}$$



(a) First row: expected L2 loss and upper bound with increasing in-context samples  $k$  under varied dimensions  $d$ . Second row: expected mixture weights with increasing in-context samples  $k$  under varied dimensions  $d$ . We further examine the early ascent phenomenon under linear regression with varied levels of label noises in Appendix A.9.1, and under non-linear regression and discrete token prediction in Appendix A.9.2.



(b) The trajectory of the expectation of  $\tilde{w}$  with increasing  $k$  under  $d$  equal to 1, 2 and 3.

**Figure A.2: The early ascent phenomenon.** Fig. A.2a displays the trends of expected losses, upper bounds, and mixture weights, while Fig. A.2b presents the trend of the expectation of  $\tilde{w}$ . We can see that the task retrieval mode is dominant up to  $k = 32$ , and component 1's mixture weight increases ( $\mathbb{E}[\tilde{w}]$  approaches  $w_1$ ). Since this misleading component 1 is far from the target component 2, the risk starts increasing. At larger  $k$  values, the risk starts decreasing ( $\mathbb{E}[\tilde{w}]$  approaches  $w_2$ ) via task learning.

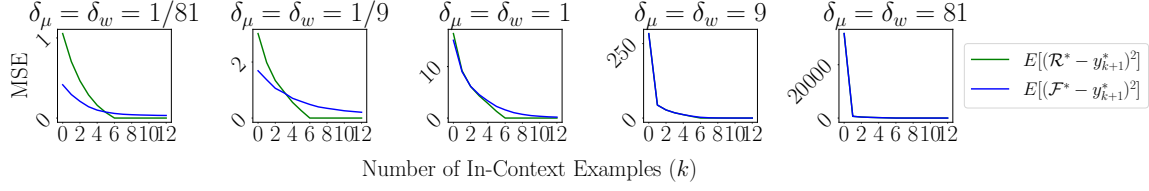


Figure A.3: **In-context learning vs ridge regression.**  $\mathcal{R}^*$  indicates the prediction by ridge regression,  $\mathcal{F}^*$  indicates the prediction by ICL with a Bayes-optimal next-token predictor, and  $y_{k+1}^* = \langle \mathbf{x}_{k+1}, \mathbf{w}^* \rangle$ . Let the  $k$  samples draw from a task  $(\boldsymbol{\mu}^*, \mathbf{w}^*)$ , which is drawn from the pretraining prior distribution. The dimension  $d$  of  $\mathbf{x}$  equals 6. We observe that ICL performs better than ridge regression when  $k$  is small, and ridge regression performs better than ICL when  $k \geq d$ . Especially, when the task prior distribution has high task variance (big  $\delta_\mu$  and  $\delta_w$  values), ICL and ridge regression have very similar performance.

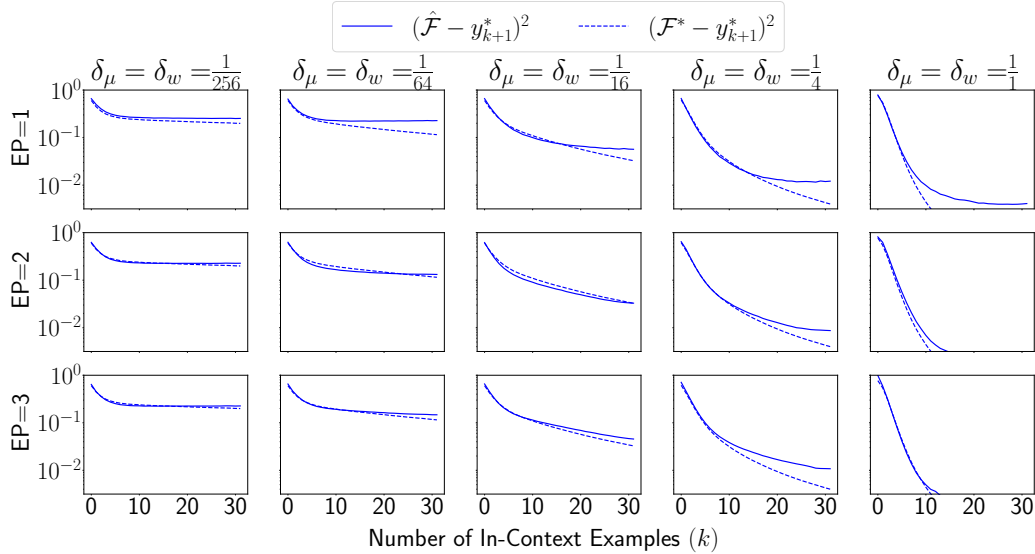


Figure A.4: **Prior task noises.** The figure shows the experiment results under varied noise levels.  $\delta_\mu$  and  $\delta_w$  indicate the noise levels of the pretraining task prior.  $\mathcal{F}^*$  indicates the prediction of Bayesian inference while  $\hat{\mathcal{F}}$  indicates the prediction of the trained Transformer network. The results show that the trained Transformer network's performance can approach the performance of Bayesian inference.

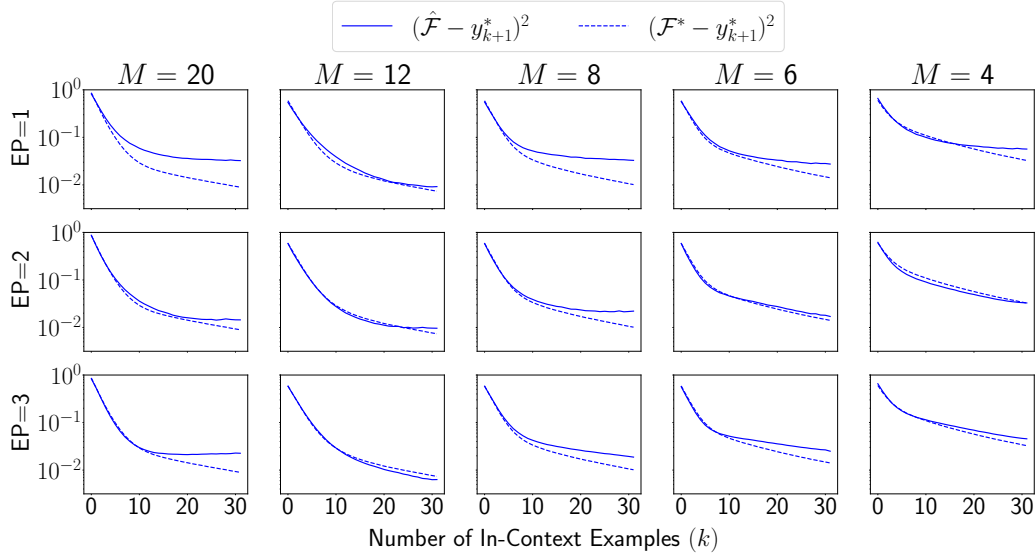


Figure A.5: **Number of components.** The figure shows the experiment results under varied component densities.  $M$  indicates the number of mixture components corresponding to different 3D regular polyhedrons described in Appendix A.2.1, and  $\delta_\mu = \delta_w = \frac{1}{16}$ .  $\mathcal{F}^*$  indicates the prediction of Bayesian inference while  $\hat{\mathcal{F}}$  indicates the prediction of the trained Transformer network. The higher the component density is, the harder it is for the Transformer network to approach Bayesian inference.

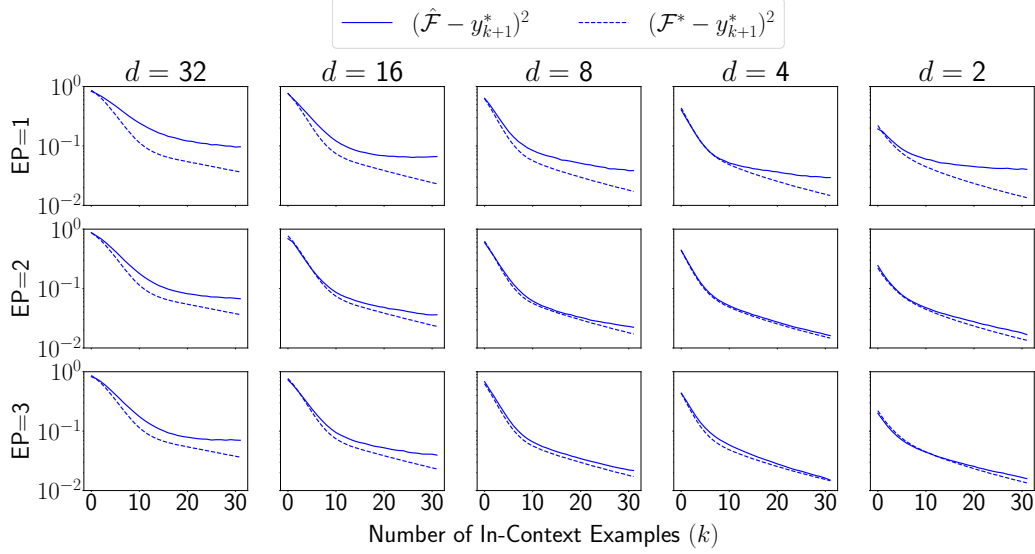


Figure A.6: **Experiments on varying feature dimensions.** The figure shows the experiment results under varied dimensions.  $d$  indicates the dimension and the number of mixture components (see Appendix A.2.2 for setting details), and  $\delta_\mu = \delta_w = \frac{1}{16}$ .  $\mathcal{F}^*$  indicates the prediction of Bayesian inference while  $\hat{\mathcal{F}}$  indicates the prediction of the trained Transformer network. The higher the feature dimension is, the harder it is for the Transformer network to approach Bayesian inference.

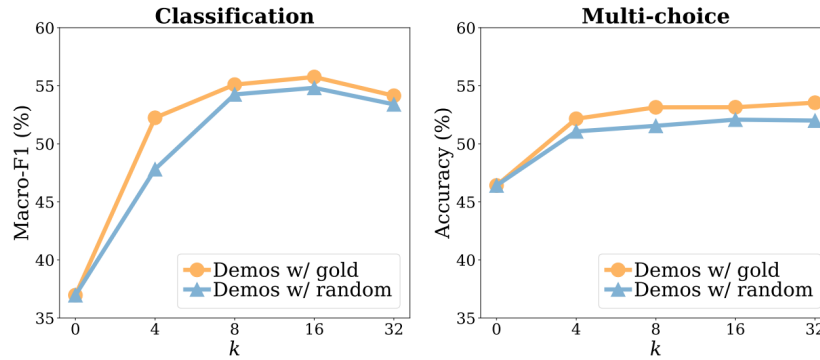


Figure A.7: **Ablations on varying numbers of examples in the demonstrations ( $k$ ).** Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used.

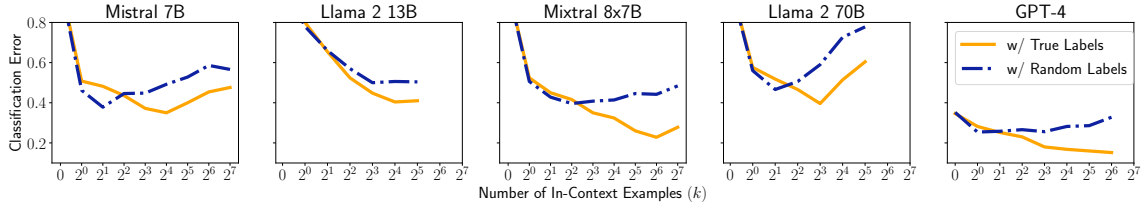


Figure A.8: **Bounded efficacy phenomenon of real-world LLMs.** As  $k$  increases, the classification error curve of ICL with random labels exhibits the bounded efficacy phenomenon. The curve with true labels further confirms that this phenomenon is not due to models tending to perform worse on long sequences.

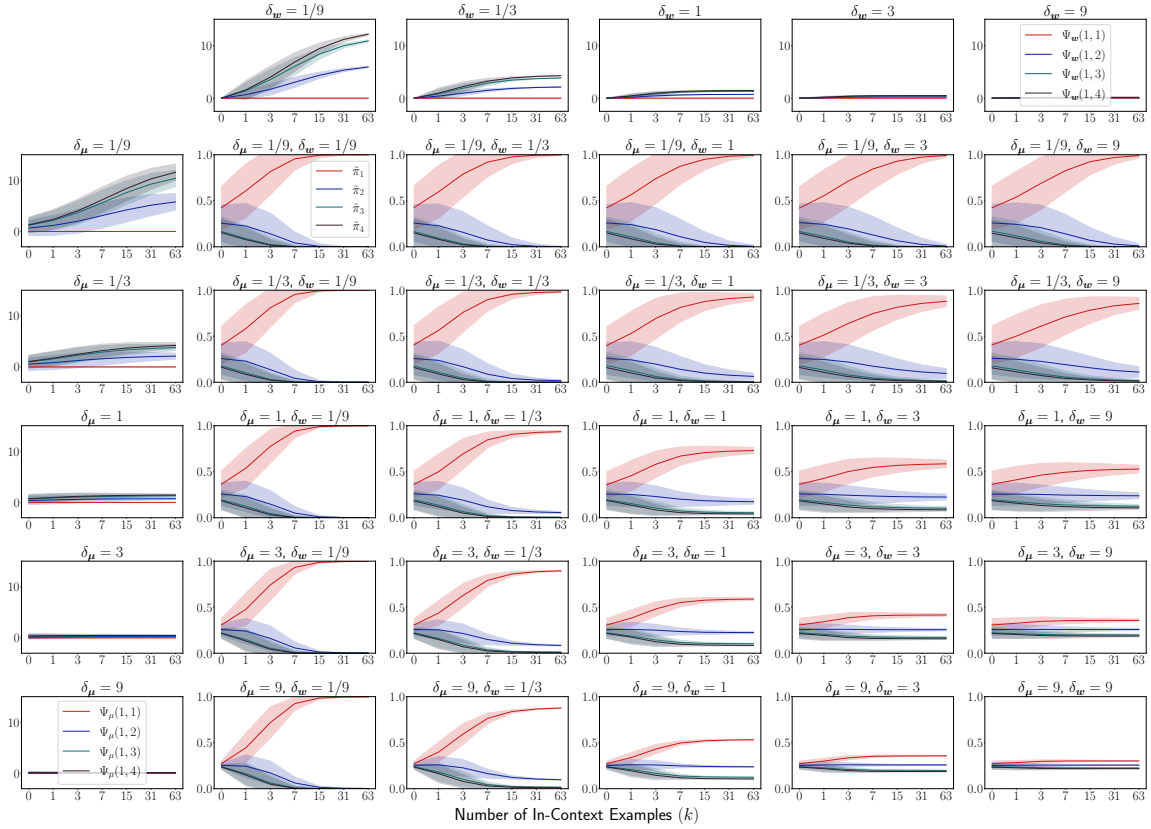


Figure A.9: **Numerical analysis on component re-weighting.** The trends of  $\Psi_\mu$ ,  $\Psi_w$ , and  $\pi_m$  for CR with increasing  $k$  under varying task noise parameters.

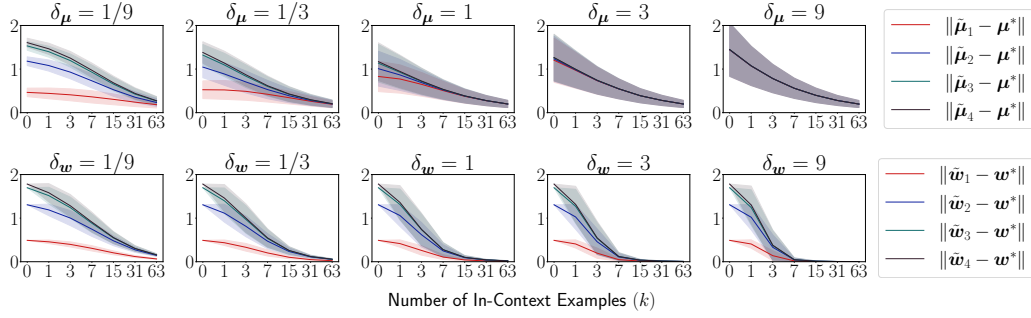


Figure A.10: **The trend of Component Shifting.** Numerical computations of  $\|\tilde{\mu}_m - \mu^*\|$ ,  $\|\tilde{w}_m - w^*\|$  for Component Shifting (CS).

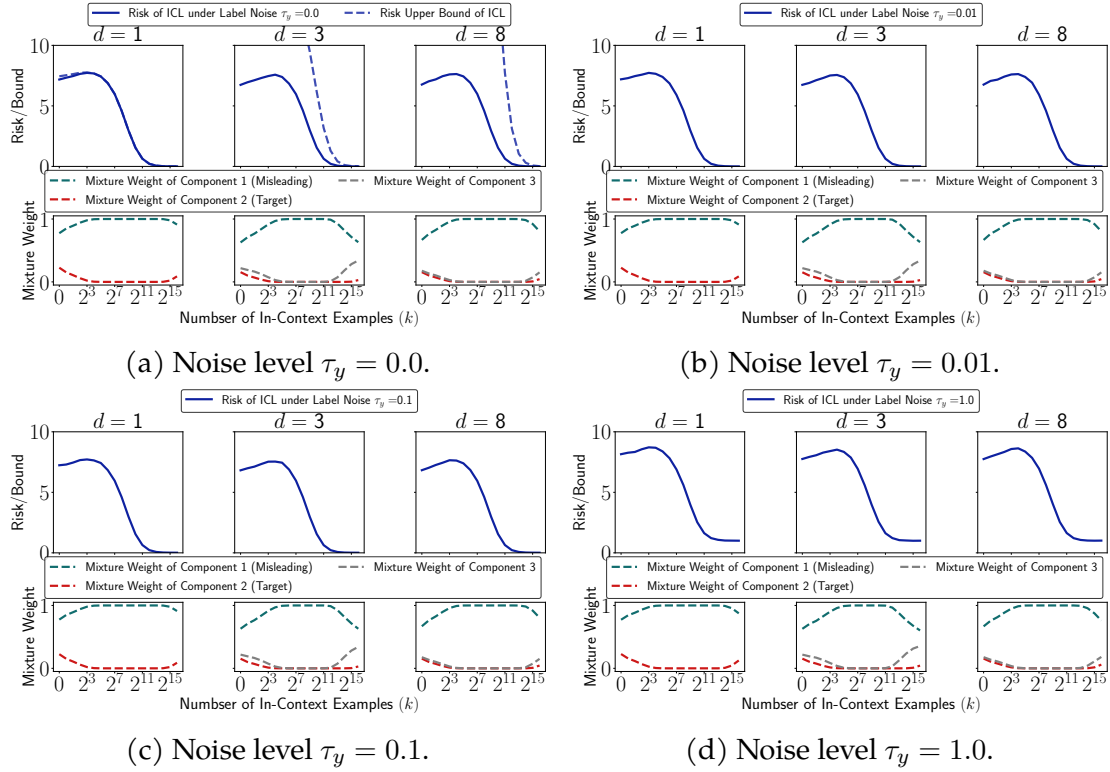


Figure A.11: **Early ascent under varied label noises.** Results show that the early ascent phenomenon maintains for noise level  $\tau_y \in [0, 1.0]$ . Label noise level  $\sigma_y = 1.0$  is used for pretraining.

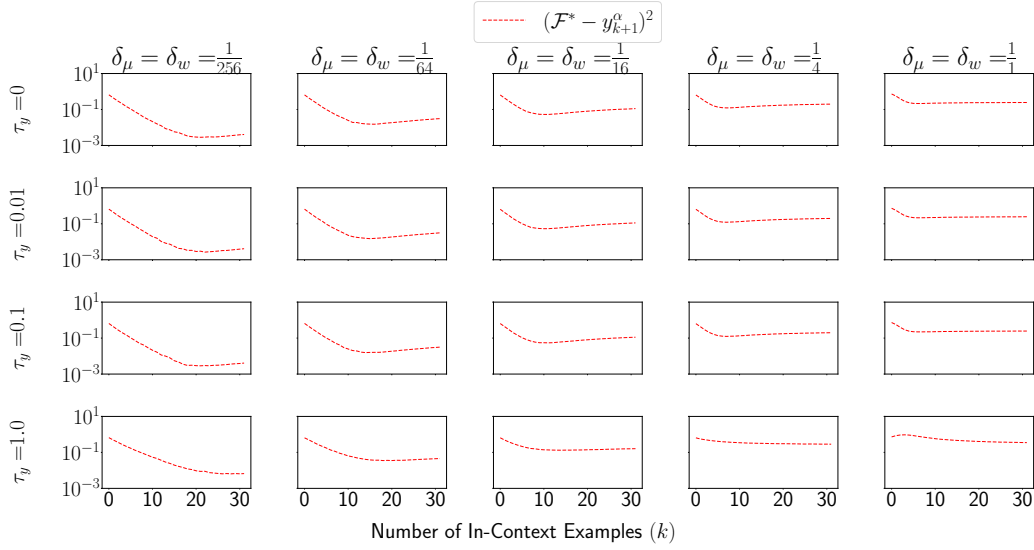
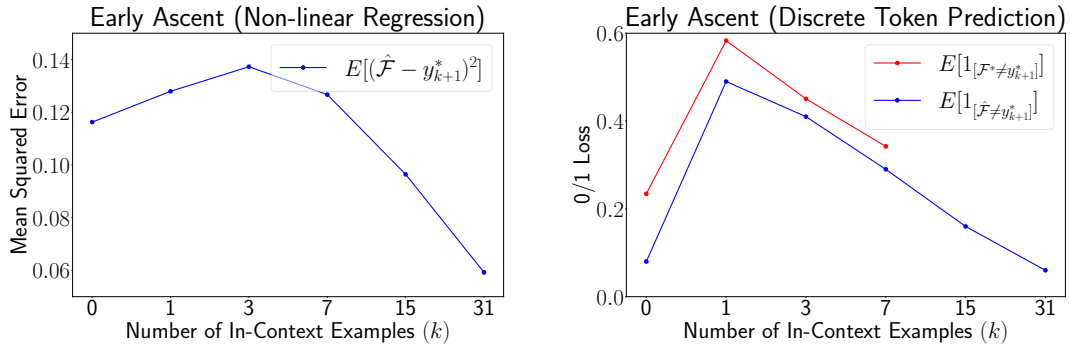


Figure A.12: **Bounded efficacy under varied label noises.** Results show that the bounded efficacy phenomenon maintains for noise level  $\tau_y \in [0, 0.1]$ . Label noise level  $\sigma_y = 1.0$  is used for pretraining.



(a) Experiment under non-linear regression. (b) Experiment under discrete token prediction.

Figure A.13: **Early ascent on non-linear regression and discrete token prediction.**  $\hat{\mathcal{F}}$  indicates the prediction by a pretrained Transformer model and  $\mathcal{F}^*$  indicates the prediction by numerical computation following a Bayes optimal predictor. While we cannot derive the optimal predictor under non-linear regression, we can derive the optimal predictor under discrete token prediction.

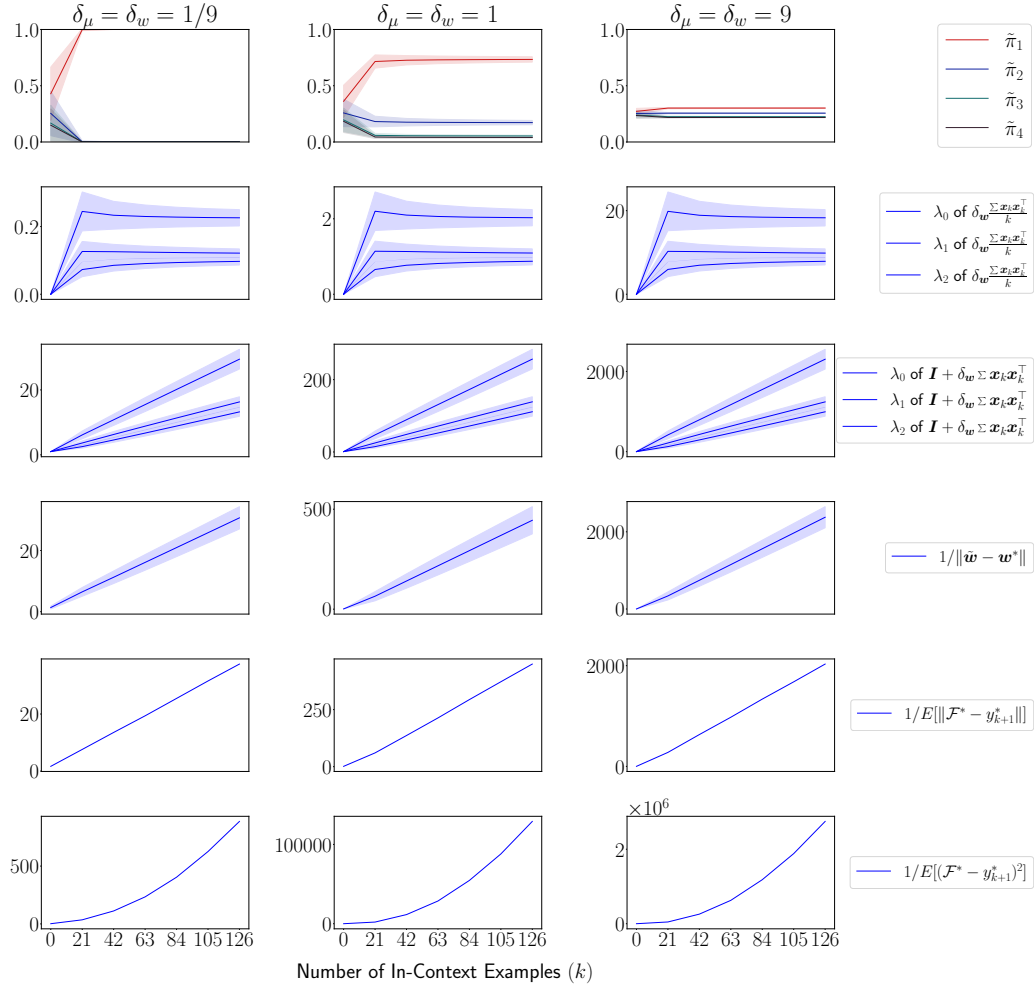


Figure A.15: **The numerical computation of the task learning.** The second and third rows show the eigenvalues of the matrices  $\delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top$  and  $\mathbf{I} + \delta_w \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top$ . The fourth row shows the distance between the predicted  $\tilde{\mathbf{w}}$  and  $\mathbf{w}^*$  has a reciprocal decreasing rate with respect to  $k$ . The fifth and sixth rows indicate the expected squared loss follows a quadratic decreasing rate with respect to  $k$ .

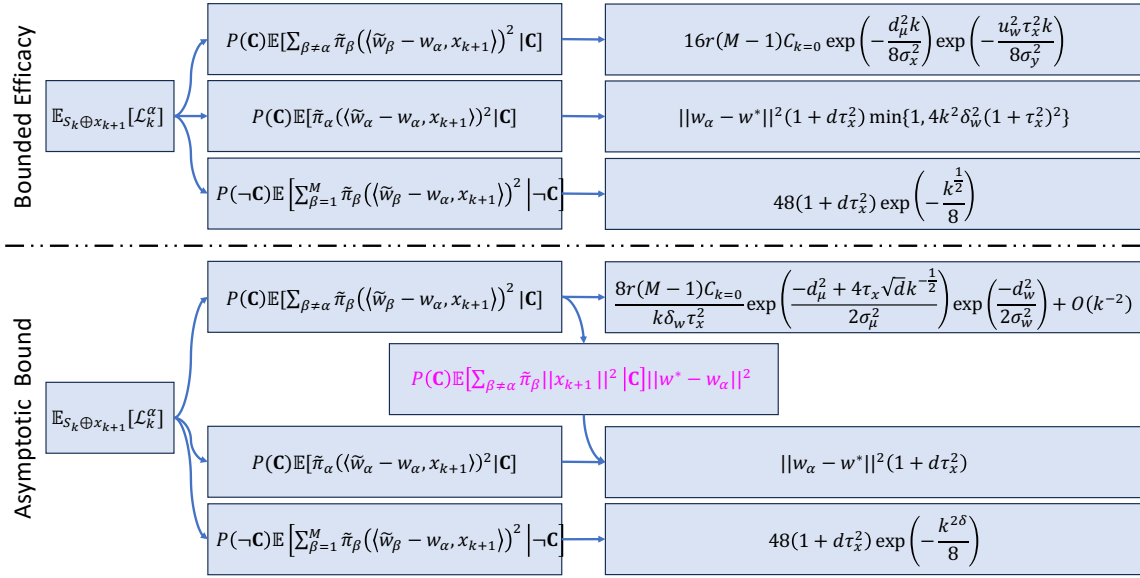


Figure A.16: Proof roadmap of ICL with biased labels, Theorem. 5.

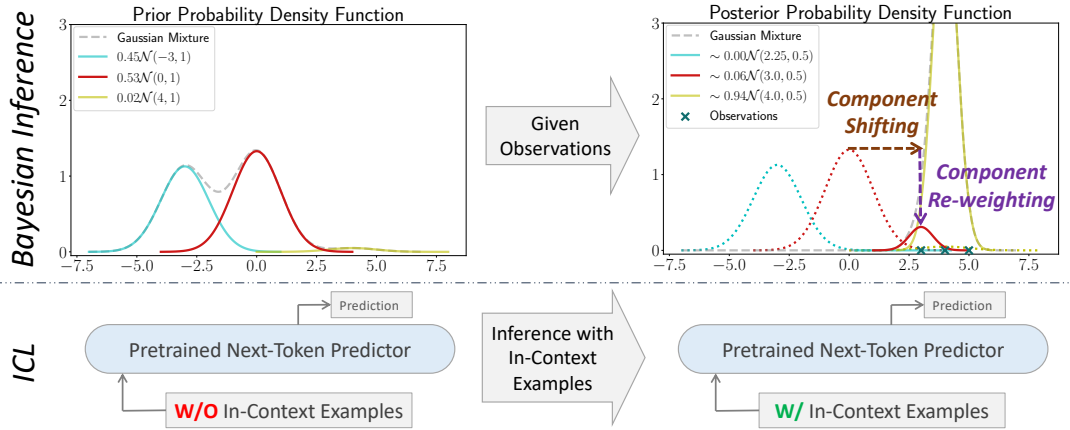


Figure A.17: **Visualization of prior, posterior, and observations.** The left part of the figure indicates the pretrained next-token predictor is pretrained on the task prior distribution according to Assumption 10, and the prediction is based on the prior without in-context examples. The right part of the figure indicates that with in-context samples, the prediction is based on the posterior, regarding the in-context examples as observed samples.

# Appendix B

## For Chapter 3

### B.1 Pseudo Algorithm for ICL-HCG

We summarize our meta framework for ICL-HCG in Algorithm 1.

### B.2 Implementation Detail of Hypothesis Prefix and Context Query

**Hypothesis prefix** Given a hypothesis class  $\mathcal{H}$  and its hypothesis table, the corresponding hypothesis prefix with hypothesis prefix's content length  $L$  is constructed as shown in Fig. B.1. The token “P” serves as the padding token to separate hypotheses, the token “;” serves as the separation token to separate  $(x, y)$  pairs, the token “N” serves as the empty token to fill a blank hypothesis, and the token “>” is used to connect  $(x, y)$  pairs of the hypothesis to a randomly assigned hypothesis

---

**Algorithm 1** Meta-Learning Framework for ICL-HCG
 

---

```

1: Inputs: a set of inputs  $\mathcal{X}$ , a training set of hypothesis classes  $\mathcal{S}^{\text{train}} = \{\mathcal{H}_i^{\text{train}}\}_{i=1}^{N^{\text{train}}}$ ,
   a testing set of hypothesis classes  $\mathcal{S}^{\text{test}} = \{\mathcal{H}_i^{\text{test}}\}_{i=1}^{N^{\text{test}}}$ , batch size  $B$ , hypothesis
   prefix size  $L$ , and context query size  $K$ 
2: for training epoch do
3:   sample  $\{\mathcal{H}_i\}_{i=1}^B \stackrel{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{S}^{\text{train}})$ 
4:   for each hypothesis class  $\mathcal{H} \in \{\mathcal{H}_i\}_{i=1}^B$  do
5:     generate  $h, S_K$  following i.i.d. Generation
6:     // Construct sequence based on  $\mathcal{H}, h$ , and  $S_K$ 
7:     construct hypothesis prefix, context query, and hypothesis index  $z$  based on
        $\mathcal{H}, h, S_K$ 
8:      $s \leftarrow \text{concatenate}(\text{hypothesis prefix, context query, } z)$ 
9:     // Cross-entropy loss for next token prediction
10:     $\mathcal{L} \leftarrow -\sum_{t=2}^{|s|} \log P(s_t \mid s_{<t})$ 
11:   end for
12:   update model parameters using  $\mathcal{L}$  of the batch
13: end for
14: for testing epoch do
15:   sample  $\{\mathcal{H}_i\}_{i=1}^B \stackrel{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{S}^{\text{test}})$ 
16:   for each hypothesis class  $\mathcal{H} \in \{\mathcal{H}_i\}_{i=1}^B$  do
17:     generate  $h, S_K$  via:
18:       either following i.i.d. Generation
19:       or following Opt-T Generation
20:     construct sequence  $s$  based on  $\mathcal{H}, h$ , and  $S_K$ 
21:     evaluate the prediction accuracy on  $y, z$ , etc
22:   end for
23: end for

```

---

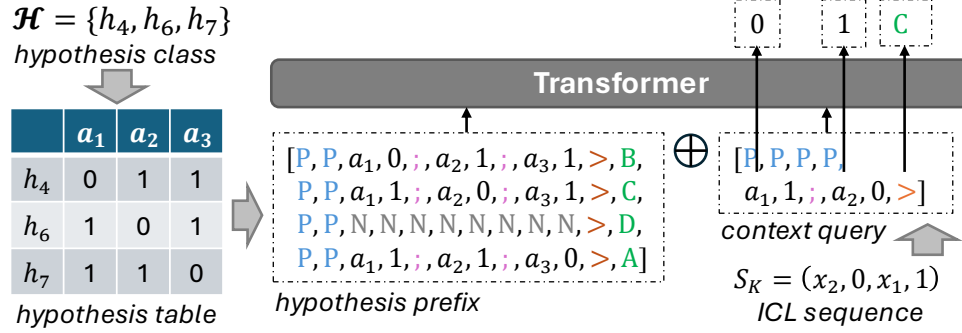


Figure B.1: **The framework.** We convert hypothesis class  $\mathcal{H}$  and ICL sequence  $S_K$  into sequences of tokens, concatenate them and input to Transformer. Then we examine whether Transformer can predict correct  $y$  and  $z$  values.

index  $z^1$ . In the illustrated example in Fig. B.1, the randomly assigned indexes  $z$ 's are sampled from  $M = 4$  hypothesis index tokens  $\{“A”, “B”, “C”, “D”\}$  without replacement<sup>2</sup>.

**Context query** Given an ICL sequence  $S_K$  with  $K$  pairs of  $(x, y)$ , the context query of size  $K$  is constructed to represent the ICL sequence and trigger the prediction of the hypothesis index with padding token “P”, separation token “;”, and query token “>” as shown in Fig. B.1.

<sup>1</sup>We use variable  $z$  to represent the hypothesis index.

<sup>2</sup>A set of  $L$  hypothesis index tokens are created serve as the pool from which the hypothesis indexes are randomly sampled without replacement.

## B.3 Additional Details of Experiments

### B.3.1 Four Types of Generalization

We share more training and testing curves in Fig. B.2 to provide additional results to Fig. 3.5, and in Fig. B.3 to provide additional results to Fig. 3.6.

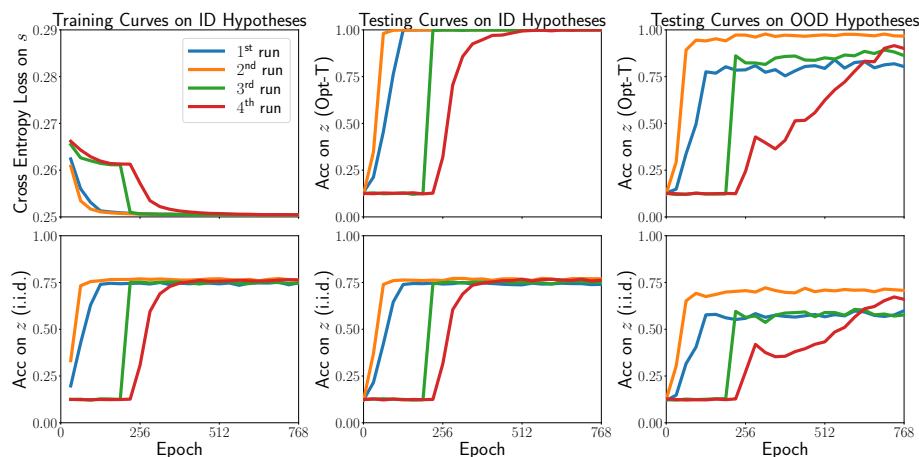


Figure B.2: Multiple runs for ID and OOD hypothesis class generalizations.

### B.3.2 Compare with Other Model Architectures

We share more training and testing curves in Figs. B.4 and B.5 to provide additional results to Figs. 3.7 and 3.8, respectively.

### B.3.3 Effect of Training Class Count

We share more training and testing curves in Fig. B.6 to provide additional results to Fig. 3.9.

## B.4 Experimental Setup

Each experiment is **repeated four times**, with the mean calculated across runs. The shadow region’s boundary is defined by **the minimum and maximum values** observed across the four runs.

### B.4.1 Learning Rate Scheduler

We set the training procedure with 768 total epochs, each epoch containing 1024 batches. The learning rate (lr) is first warmed up linearly from an LR/64 at epoch  $e = 1$  to a peak value LR at epoch  $e = 64$ , following:

$$\text{lr}(e) = \text{LR} \cdot \frac{e}{64}, \quad 1 \leq e \leq 64.$$

After epoch 64, the learning rate undergoes a quadratic decay over the remaining 704 epochs, given by

$$\text{lr}(e) = \text{LR} \cdot \sqrt{\frac{64}{e}}, \quad 64 \leq e \leq 768.$$

### B.4.2 Hyperparameter Search

We list the hyperparameter searching spaces used for Transformer, LSTM, GRU, and Mamba. The best hyperparameter is searched using ID hypothesis class generalization with  $\|\mathcal{X}\| = 5$ ,  $\|\mathcal{H}\| = 8$ , and then used for all other settings.

Table B.1: **Hyperparameter search spaces for different model architectures.** The optimal hyperparameters are bolded if multiple possibilities are provided.

| Model Architecture | #layers    | #hidden dimensions | #learning rate                             | #weight decay | #batch size |
|--------------------|------------|--------------------|--------------------------------------------|---------------|-------------|
| Transformer        | <b>2,8</b> | 128                | 0.00010, <b>0.00020</b> , 0.00050, 0.00100 | 0.0005        | 16          |
| Mamba              | <b>2,8</b> | 128                | 0.00010, 0.00020, <b>0.00050</b> , 0.00100 | 0.0005        | 16          |
| GRU                | <b>2,8</b> | 128                | 0.00020, 0.00050, <b>0.00100</b> , 0.00200 | 0.0005        | 16          |
| LSTM               | <b>2,8</b> | 128                | 0.00020, 0.00050, <b>0.00100</b> , 0.00200 | 0.0005        | 16          |

### B.4.3 Setup for Generating Training and Testing Hypothesis

#### Classes

We list the experimental setup for each experiments in the following Table B.2. When conducting experiments to evaluate accuracy on  $y$ , we modified the experimental setup following Table B.3.

Table B.2: **Experimental setups of different generalizations.** The expression  $\min\{512, \#\text{possible}\}$  indicates that when the number of possible hypothesis classes is fewer than 512, we evaluate all possible hypothesis classes for testing; otherwise, we limit the selection to at most 512 hypothesis classes. For example, if  $|\mathcal{H}^{\text{OOD}}| = 16$  and  $|\mathcal{H}| = 2$ , the total number of possible hypothesis classes is given by:  $\binom{|\mathcal{H}^{\text{OOD}}|}{|\mathcal{H}|} = \binom{16}{2} = \frac{16 \times 15}{2} = 120$ . Since  $120 < 512$ , we evaluate all 120 hypothesis classes for testing in this scenario.

| Generalization Setup                                                 | ID Hypothesis<br>Class Generalization | OOD Hypothesis<br>Class Generalization | ID Hypothesis<br>Class Size Generalization | OOD Hypothesis<br>Class Size Generalization |
|----------------------------------------------------------------------|---------------------------------------|----------------------------------------|--------------------------------------------|---------------------------------------------|
| size of input space ( $ \mathcal{X} $ )                              | 5                                     | 5                                      | 5                                          | 5                                           |
| size of label space ( $ \mathcal{Y} $ )                              | 2                                     | 2                                      | 2                                          | 2                                           |
| size of context query ( $K$ )                                        | 5                                     | 5                                      | 5                                          | 5                                           |
| size of training hypothesis class ( $ \mathcal{H}^{\text{train}} $ ) | 8                                     | 8                                      | 7,8,9                                      | 7,8,9                                       |
| size of testing hypothesis class ( $ \mathcal{H}^{\text{test}} $ )   | 8                                     | 8                                      | 2, ..., 14                                 | 2, ..., 14                                  |
| size of hypothesis prefix ( $L$ )                                    | 8                                     | 8                                      | 16                                         | 16                                          |
| #all hypotheses ( $ \mathcal{H}^{\text{uni}} $ )                     | 32                                    | 32                                     | 32                                         | 32                                          |
| #hypotheses in ID pool ( $ \mathcal{H}^{\text{ID}} $ )               | 16                                    | 16                                     | 16                                         | 16                                          |
| #hypotheses in OOD pool ( $ \mathcal{H}^{\text{OOD}} $ )             | 16                                    | 16                                     | 16                                         | 16                                          |
| #training hypothesis classes                                         | 12358                                 | 12358                                  | 4096                                       | 4096                                        |
| #testing hypothesis classes                                          | 512                                   | 512                                    | $\min\{512, \#\text{possible}\}$           | $\min\{512, \#\text{possible}\}$            |

Table B.3: **Additional setups.** Numbers that differ from those in Table B.2 are highlighted in bold for clarity.

| Section                                                              | Sec. 3.3.6  | Sec. 3.3.7                         |
|----------------------------------------------------------------------|-------------|------------------------------------|
| size of input space ( $ \mathcal{X} $ )                              | <b>4</b>    | <b>6</b>                           |
| size of label space ( $ \mathcal{Y} $ )                              | 2           | 2                                  |
| size of context query ( $K$ )                                        | <b>12</b>   | <b>12</b>                          |
| size of training hypothesis class ( $ \mathcal{H}^{\text{train}} $ ) | <b>4</b>    | 8                                  |
| size of testing hypothesis class ( $ \mathcal{H}^{\text{test}} $ )   | <b>4</b>    | 8                                  |
| size of hypothesis prefix ( $L$ )                                    | <b>4</b>    | 8                                  |
| #all hypotheses ( $ \mathcal{H}^{\text{uni}} $ )                     | <b>16</b>   | <b>64</b>                          |
| #hypotheses in ID pool ( $ \mathcal{H}^{\text{ID}} $ )               | <b>16</b>   | <b>8,16,24,32,48</b>               |
| #hypotheses in OOD pool ( $ \mathcal{H}^{\text{OOD}} $ )             | <b>0</b>    | 16                                 |
| #training hypothesis classes                                         | <b>1308</b> | $\min\{12358, \text{\#possible}\}$ |
| #testing hypothesis classes                                          | 512         | 512                                |

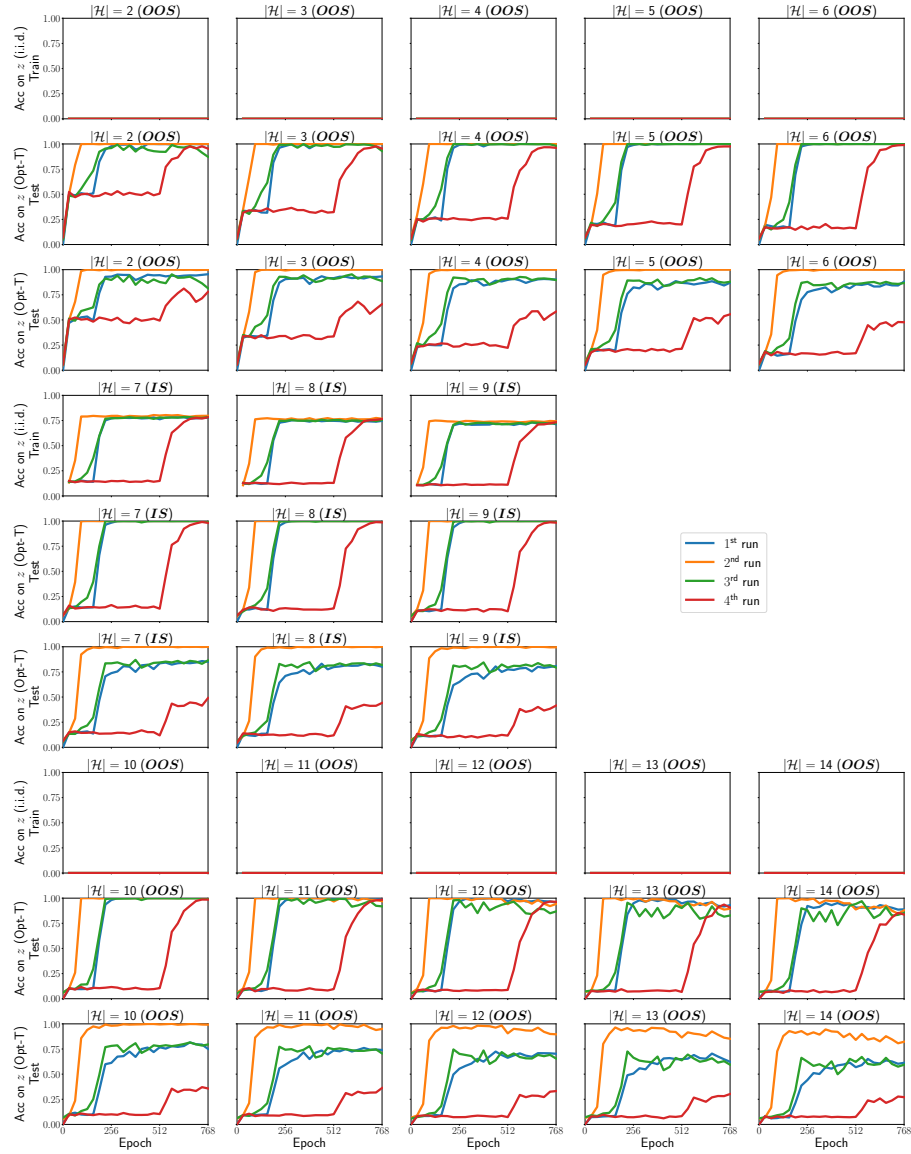


Figure B.3: Multiple runs for ID and OOD hypothesis class size generalizations.

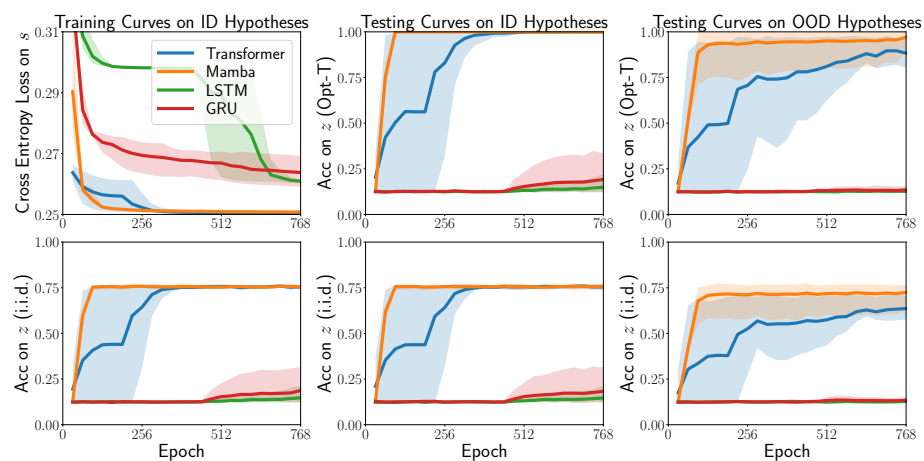


Figure B.4: Various models on ID and OOD hypothesis class generalizations.

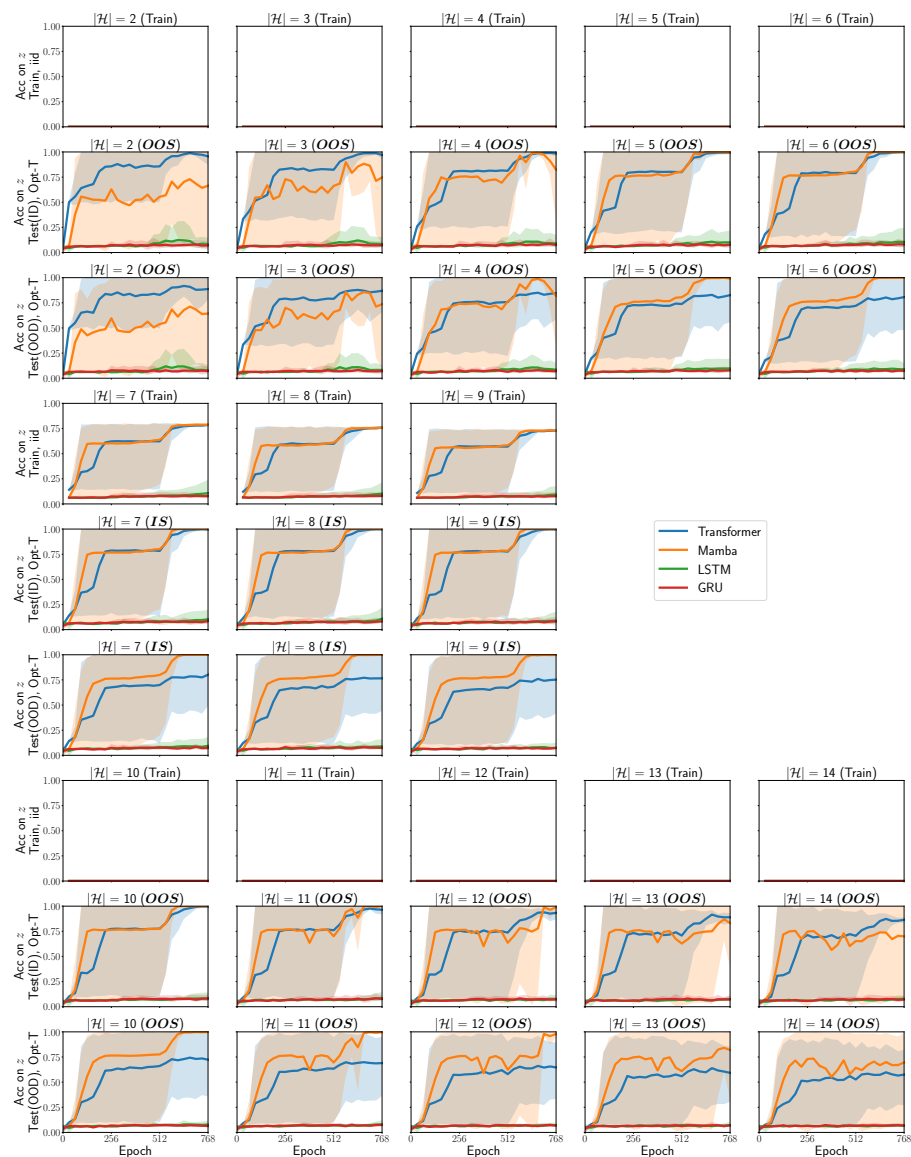


Figure B.5: Various models on ID and OOD hypothesis class generalizations.

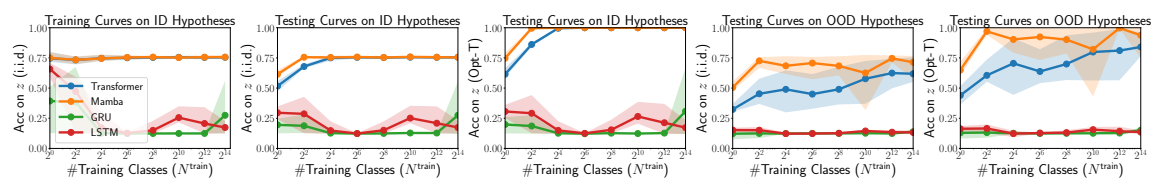


Figure B.6: Effect of training hypothesis class count on ID and OOD hypothesis class generalization.

# Bibliography

- [1] K. Ahn, X. Cheng, H. Daneshmand, and S. Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*, 2023.
- [2] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2023.
- [3] E. Akyürek, B. Wang, Y. Kim, and J. Andreas. In-context language learning: Architectures and algorithms. In *International Conference on Machine Learning*, 2024.
- [4] Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, 2023.
- [5] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020.
- [6] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

- [7] S. Bhattamishra, A. Patel, P. Blunsom, and V. Kanade. Understanding in-context learning in Transformers and LLMs by learning to learn discrete functions. In *International Conference on Learning Representations*, 2024.
- [8] M. Boratko, X. Li, T. O’Gorman, R. Das, D. Le, and A. McCallum. Protoqa: A question answering dataset for prototypical common-sense reasoning. In *Conference on Empirical Methods in Natural Language Processing*, 2020.
- [9] D. Borkan, L. Dixon, J. S. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.
- [10] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.
- [11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.
- [12] S. Chan, A. Santoro, A. Lampinen, J. Wang, A. Singh, P. Richemond, J. McClelland, and F. Hill. Data distributional properties drive emergent in-context learning in Transformers. In *Advances in Neural Information Processing Systems*, 2022.
- [13] S. Chen, H. Sheen, T. Wang, and Z. Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality (extended abstract). In *Annual Conference on Learning Theory*, 2024.
- [14] X. Chen, M. Lin, N. Schärli, and D. Zhou. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*, 2023.

- [15] X. Cheng, Y. Chen, and S. Sra. Transformers implement functional gradient descent to learn non-linear functions in context. In *International Conference on Machine Learning*, 2024.
- [16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [17] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- [18] S. D. Constantin and T. Rao. On the theory of binary asymmetric error correcting codes. *Information and Control*, 1979.
- [19] T. M. Cover and J. A. Thomas. Information theory and statistics. *Elements of information theory*, 1991.
- [20] I. Dagan, O. Glickman, and B. Magnini. The PASCAL recognising textual entailment challenge. In *PASCAL Machine Learning Challenges Workshop*, 2005.
- [21] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [22] A. G. Dimakis, B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory*, 56(9):4539–4551, 2010. doi: 10.1109/TIT.2010.2054295. URL <https://doi.org/10.1109/TIT.2010.2054295>.
- [23] W. B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In *International Workshop on Paraphrasing*, 2005.
- [24] B. L. Edelman, E. Edelman, S. Goel, E. Malach, and N. Tsilivis. The evolution of statistical induction heads: In-context learning Markov chains. *arXiv preprint arXiv:2402.11004*, 2024.

- [25] Y. Fan, S. Yadlowsky, D. Papailiopoulos, and K. Lee. Transformers can learn meta-skills for task generalization in in-context learning. In *NeurIPS Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward*, 2024.
- [26] C. V. Freiman. Optimal error detection codes for completely asymmetric binary channels. *Information and Control*, 1962.
- [27] D. Fu, T.-Q. Chen, R. Jia, and V. Sharan. Transformers learn to achieve second-order convergence rates for in-context linear regression. In *Advances in Neural Information Processing Systems*, 2024.
- [28] J. Fu, T. Yang, Y. Wang, Y. Lu, and N. Zheng. Breaking through the learning plateaus of in-context learning in Transformer. In *International Conference on Machine Learning*, 2024.
- [29] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.
- [30] R. Gabrys and L. Dolecek. Coding for the binary asymmetric channel. In *International Conference on Computing, Networking and Communications*. IEEE, 2012.
- [31] S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can Transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.
- [32] K. Gatmiry, N. Saunshi, S. J. Reddi, S. Jegelka, and S. Kumar. Can looped Transformers learn to implement multi-step gradient descent for in-context learning? In *International Conference on Machine Learning*, 2024.
- [33] O. Golovneva, T. Wang, J. Weston, and S. Sukhbaatar. Contextual position encoding: Learning to count what’s important. *arXiv preprint arXiv:2405.18719*, 2024.
- [34] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

- [35] A. Guez, T. Weber, I. Antonoglou, K. Simonyan, O. Vinyals, D. Wierstra, R. Munos, and D. Silver. Learning to search with MCTSnets. In *International Conference on Machine Learning*, 2018.
- [36] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [37] T. Guo, W. Hu, S. Mei, H. Wang, C. Xiong, S. Savarese, and Y. Bai. How do Transformers learn in-context beyond simple functions? A case study on learning with representations. In *International Conference on Learning Representations*, 2024.
- [38] C. Han, Z. Wang, H. Zhao, and H. Ji. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.
- [39] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
- [40] R. Hataya and M. Imaizumi. Transformers as stochastic optimizers. In *ICML Workshop on In-Context Learning*, 2024.
- [41] S. Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [42] E. Hua, C. Jiang, X. Lv, K. Zhang, N. Ding, Y. Sun, B. Qi, Y. Fan, X. Zhu, and B. Zhou. Fourier position embedding: Enhancing attention’s periodic extension for length generalization. *arXiv preprint arXiv:2412.17739*, 2024.
- [43] R. Huang and R. Ge. Task descriptors help Transformers learn linear models in-context. In *ICML Workshop on In-Context Learning*, 2024.
- [44] Y. Huang, Y. Cheng, and Y. Liang. In-context convergence of Transformers. In *International Conference on Machine Learning*, 2024.

- [45] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. GPT-4o system card. In *arXiv preprint arXiv:2410.21276*, 2024.
- [46] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [47] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.
- [48] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- [49] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 2020.
- [50] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal. Qasc: A dataset for question answering via sentence composition. In *AAAI Conference on Artificial Intelligence*, 2020.
- [51] J. Kim and T. Suzuki. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. In *International Conference on Machine Learning*, 2024.
- [52] J. Kim, S. Kwon, J. Y. Choi, J. Park, J. Cho, J. D. Lee, and E. K. Ryu. Task diversity shortens the ICL plateau. *arXiv preprint arXiv:2410.05448*, 2024.
- [53] L. Lam and S. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568, 1997.

- [54] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 2017.
- [55] L. Lehnert, S. Sukhbaatar, D. Su, Q. Zheng, P. Mcvay, M. Rabbat, and Y. Tian. Beyond A\*: Better planning with transformers via search dynamics bootstrapping. *arXiv preprint arXiv:2402.14083*, 2024.
- [56] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Association for Computational Linguistics*, 2021.
- [57] H. Li, M. Wang, S. Lu, X. Cui, and P.-Y. Chen. How do nonlinear Transformers acquire generalization-guaranteed CoT ability? In *ICML Workshop on High-dimensional Learning Dynamics: The Emergence of Structure and Reasoning*, 2024.
- [58] H. Li, M. Wang, S. Lu, X. Cui, and P.-Y. Chen. How do nonlinear Transformers learn and generalize in in-context learning? In *International Conference on Machine Learning*, 2024.
- [59] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Association for Computational Linguistics*, 2021.
- [60] Y. Li, M. E. Ildiz, D. Papailiopoulos, and S. Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, 2023.
- [61] Z. Lin and K. Lee. Dual operating modes of in-context learning. In *International Conference on Machine Learning*, 2024.
- [62] Z. Lin, S. K. Bharti, and K. Lee. In-context learning with hypothesis-class guidance. *arXiv preprint arXiv:2502.19787*, 2025.
- [63] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

- [64] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, and H. Hajishirzi. Generated knowledge prompting for commonsense reasoning. In *Association for Computational Linguistics*, 2022.
- [65] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for GPT-3? In *Deep Learning Inside Out: The Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022.
- [66] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang. GPT understands, too. *arXiv preprint arxiv.2103.10385*, 2021.
- [67] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [68] X. Lyu, S. Min, I. Beltagy, L. Zettlemoyer, and H. Hajishirzi. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. In *Association for Computational Linguistics*, 2023.
- [69] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegrefe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.
- [70] A. V. Makkuva, M. Bondaschi, A. Girish, A. Nagle, M. Jaggi, H. Kim, and M. Gastpar. Attention with Markov: A framework for principled analysis of Transformers via Markov chains. *arXiv preprint arXiv:2402.04161*, 2024.
- [71] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation*, 2014.
- [72] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation*, 2014.

- [73] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Empirical Methods in Natural Language Processing*, 2022.
- [74] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, 2022.
- [75] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas. ETHOS: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*, 2020.
- [76] S. M. Moser, P.-N. Chen, and H.-Y. Lin. Error probability analysis of binary asymmetric channels. *Dept. El. & Comp. Eng., Nat. Chiao Tung Univ*, 2009.
- [77] S. Mukherjee, J. P. Hanna, Q. Xie, and R. Nowak. Pretraining decision transformers with reward prediction for in-context multi-task structured bandit learning. *arXiv preprint arXiv:2406.05064*, 2024.
- [78] E. Nichani, A. Damian, and J. D. Lee. How Transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.
- [79] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.
- [80] OpenAI. GPT-4 technical report, 2023.
- [81] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [82] J. Pan, T. Gao, H. Chen, and D. Chen. What in-context learning “learns” in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics*, 2023.

- [83] M. Panwar, K. Ahuja, and N. Goyal. In-context learning through the Bayesian prism. In *International Conference on Learning Representations*, 2024.
- [84] J. Park, J. Park, Z. Xiong, N. Lee, J. Cho, S. Oymak, K. Lee, and D. Papailiopoulos. Can Mamba learn how to learn? A comparative study on in-context learning tasks. In *International Conference on Machine Learning*, 2024.
- [85] D. Paul, M. Ismayilzada, M. Peyrard, B. Borges, A. Bosselut, R. West, and B. Faltings. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*, 2023.
- [86] S. Pitis, M. R. Zhang, A. Wang, and J. Ba. Boosted prompt ensembles for large language models. *arXiv preprint arXiv:2304.05970*, 2023.
- [87] G. Qin and J. Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.
- [88] R. Ramesh, E. S. Lubana, M. Khona, R. P. Dick, and H. Tanaka. Compositional capabilities of autoregressive Transformers: A study on synthetic, interpretable tasks. In *International Conference on Machine Learning*, 2024.
- [89] K. V. Rashmi, N. B. Shah, and P. V. Kumar. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Trans. Inf. Theory*, 2011.
- [90] A. Raventos, M. Paul, F. Chen, and S. Ganguli. The effects of pretraining task diversity on in-context learning of ridge regression. In *ICLR Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.
- [91] A. Raventós, M. Paul, F. Chen, and S. Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*, 2024.
- [92] Y. Razeghi, R. L. L. IV, M. Gardner, and S. Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2022.

- [93] G. Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *International Conference on Learning Representations*, 2023.
- [94] T. Richardson and R. Urbanke. *Modern coding theory*. Cambridge university press, 2008.
- [95] O. Rubin, J. Herzig, and J. Berant. Learning to retrieve prompts for in-context learning. In *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.
- [96] M. E. Sander, R. Giryes, T. Suzuki, M. Blondel, and G. Peyré. How do Transformers perform in-context autoregressive learning? In *International Conference on Machine Learning*, 2024.
- [97] T. Schick and H. Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.
- [98] T. Schick and H. Schütze. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*, 2020.
- [99] E. Sheng and D. Uthus. Investigating societal biases in a poetry composition system. In *Workshop on Gender Bias in Natural Language Processing*, 2020.
- [100] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, and S. Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Association for Computational Linguistics*, 2020.
- [101] N. Shinn, B. Labash, and A. Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2023.
- [102] A. Singh, S. Chan, T. Moskovitz, E. Grant, A. Saxe, and F. Hill. The transient nature of emergent in-context learning in Transformers. In *Advances in Neural Information Processing Systems*, 2024.

- [103] D. Su, S. Sukhbaatar, M. Rabbat, Y. Tian, and Q. Zheng. Dualformer: Control-able fast and slow thinking by learning with randomized reasoning traces. In *International Conference on Learning Representations*, 2024.
- [104] H. Su, J. Kasai, C. H. Wu, W. Shi, T. Wang, J. Xin, R. Zhang, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.
- [105] G. Tanzer, M. Suzgun, E. Visser, D. Jurafsky, and L. Melas-Kyriazi. A benchmark for learning to translate a new language from one grammar book. In *International Conference on Learning Representations*, 2024.
- [106] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [107] N. Tripuraneni, L. Doshi, and S. Yadlowsky. Can Transformers in-context learn task mixtures? In *NeurIPS Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.
- [108] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.
- [109] A. Tsigler and P. L. Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 2023.
- [110] H. L. Van Trees. *Detection, estimation, and modulation theory, Part I: Detection, estimation, and linear modulation theory*. John Wiley & Sons, 2004.
- [111] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018.
- [112] J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 2023.

- [113] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, and D. Zhou. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*, 2022.
- [114] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.
- [115] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar, A. Ashok, A. S. Dhanasekaran, A. Naik, D. Stap, et al. Super-natural instructions: Generalization via declarative instructions on 1600+ tasks. In *Empirical Methods in Natural Language Processing*, 2022.
- [116] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.
- [117] J. Wu, D. Zou, Z. Chen, V. Braverman, Q. Gu, and P. L. Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? In *International Conference on Learning Representations*, 2024.
- [118] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.
- [119] Y. Xie, K. Kawaguchi, Y. Zhao, X. Zhao, M.-Y. Kan, J. He, and Q. Xie. Decomposition enhances reasoning via self-evaluation guided decoding. *arXiv preprint arXiv:2305.00633*, 2023.
- [120] M. Xuanyuan, T. Yang, J. Fu, and Y. Wang. On task description of in-context learning: A study from information perspective, 2024.
- [121] S. Yadlowsky, L. Doshi, and N. Tripuraneni. Can Transformer models generalize via in-context learning beyond pretraining data? In *NeurIPS Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2024.

- [122] L. Yang, K. Lee, R. D. Nowak, and D. Papailiopoulos. Looped Transformers are better at learning learning algorithms. In *International Conference on Learning Representations*, 2024.
- [123] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.
- [124] S. Ye, H. Hwang, S. Yang, H. Yun, Y. Kim, and M. Seo. In-context instruction learning. *arXiv preprint arXiv:2302.14691*, 2023.
- [125] K. Yin and J. Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.
- [126] R. Zhang, X. Wang, C. Zhang, Z. He, H. Wu, Z. Li, H. Wang, Y. Chen, and Q. Li. Bstc: A large-scale chinese-english speech translation dataset. *arXiv preprint arXiv:2104.03575*, 2021.
- [127] R. Zhang, S. Frei, and P. L. Bartlett. Trained transformers learn linear models in-context. In *Robustness of Few-shot and Zero-shot Learning in Large Foundation Models (R0-FoMo)*, 2023.
- [128] R. Zhang, S. Frei, and P. L. Bartlett. Trained Transformers learn linear models in-context. *Journal of Machine Learning Research*, 2024.
- [129] Y. Zhang, S. Feng, and C. Tan. Active example selection for in-context learning. In *Association for Computational Linguistics*, 2022.
- [130] Y. Zhang, J. Yang, Y. Yuan, and A. C.-C. Yao. Cumulative reasoning with large language models. *arXiv preprint arXiv:2308.04371*, 2023.
- [131] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

- [132] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021.
- [133] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv.2211.01910*, 2022.
- [134] X. Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI Conference on Artificial Intelligence*, 2015.