Title

Statistical and Optimization Principles for Understanding and Controlling Foundation Models

ISBN

9798189277023

Author

Zhang, Ruiqi

Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Statistical and Optimization Principles for Understanding and Controlling Foundation Models

by

Ruiqi Zhang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Peter L. Bartlett, Co-chair
Professor Song Mei, Co-chair
Professor Ryan Tibshirani
Professor Nikita Zhivotovskiy

Spring 2026

Statistical and Optimization Principles for Understanding and Controlling Foundation Models

Abstract

Statistical and Optimization Principles for Understanding and Controlling Foundation Models

by

Ruiqi Zhang

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Peter L. Bartlett, Co-chair

Professor Song Mei, Co-chair

Foundation models operate at unprecedented scale and exhibit empirical behaviors that often depart from classical learning and optimization theory, while their deployment introduces new requirements on safety and computational efficiency. This thesis uses statistical and optimization principles to explain representative non-classical phenomena in modern foundation models and design practical algorithms that improve their efficiency and safety under realistic constraints.

On the explanatory side, we first study in-context learning (ICL), the ability to solve new tasks at inference time by conditioning on a small number of demonstrations without parameter updates. In simplified yet representative transformer models trained on in-context linear regression data, we establish global convergence guarantees for gradient flow, characterize the effective function class induced at convergence, and analyze robustness under several forms of distribution shift. We further show that augmenting attention with an MLP component can provably reduce approximation error and achieve nearly Bayes-optimal risk in a Gaussian linear-regression setting, and we formalize an equivalence between such architectures and one-step gradient-based adaptation with learnable initialization.

We then investigate optimization dynamics beyond the descent-lemma regime, motivated by edge-of-stability behavior observed in large-scale deep learning. Using logistic regression with linearly separable data as a canonical testbed, we analyze gradient descent with large and adaptive stepsizes and show that it reaches an arbitrarily small risk immediately after at most $1/\gamma^2$ burn-in iterations, where $\gamma > 0$ is the margin. We also prove a matching lower bound: any batch or online first-order method requires $\Omega(1/\gamma^2)$ iterations to find a separator, establishing minimax optimality and matching the Perceptron algorithm even in constants. Our analysis extends to a broad class of losses and certain two-layer networks.

On the algorithmic-design side, we study machine unlearning for LLMs, motivated by the memorization of sensitive, private, or copyrighted data during pre-training. Traditional unlearning methods, which are based on gradient ascent (GA) on the loss of the undesirable data, either fail to effectively unlearn the target data or suffer from catastrophic collapse—a drastic degradation of the model utility. We propose Negative Preference Optimization (NPO) and theoretically show that its progression toward catastrophic collapse is exponentially slower than GA. Empirically, NPO achieves a better trade-off between forget quality and model utility.

Finally, we study efficient alignment at inference time under resource constraints. We focus on best-of-N (BoN) decoding, a simple and effective alignment method whose practical deployment is limited by its high computational cost. Motivated by hardware and resource utilization considerations, we propose Speculative Rejection, a resource-inspired inference-time alignment algorithm that leverages the correlation between partial and final rewards and dynamically allocates computation by early-stopping low-quality generations based on partial reward signals. This approach achieves reward performance comparable to BoN decoding with much larger N, while reducing required GPU resources by approximately 16 to 32 times.

To My Family.

Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to the many people who supported and guided me throughout my Ph.D. at the University of California, Berkeley. This dissertation would not have been possible without their mentorship, collaboration, and encouragement.

First and foremost, I would like to thank my advisor, Professor Peter L. Bartlett. Over the course of my Ph.D., he provided me with invaluable guidance and exceptional mentorship. We collaborated on multiple research projects, and his detailed comments and constructive suggestions consistently and significantly helped improve every piece of work. Peter gives his students remarkable freedom in choosing research directions and shaping the substance of their work, while at the same time holding an uncompromising standard of rigor. His ability to identify the core structure of complex problems and to offer insightful, well-timed suggestions has had a lasting impact on the way I think about research. I have learned an immense amount from working with him, both intellectually and professionally.

I would also like to thank my co-advisor, Professor Song Mei, for his thoughtful guidance and constant support throughout my Ph.D. journey. I am very grateful for the many inspiring ideas he shared during our collaborations, as well as for the countless discussions in which he helped clarify key questions and refine the direction of my research. Beyond research, he has been a generous mentor, offering thoughtful advice and perspective on career development. I also had the pleasure of working with him as a Graduate Student Instructor for STAT 154/254, an experience from which I learned a great deal about teaching, mentorship, and academic responsibility.

I am deeply grateful to my committee members, Professor Ryan Tibshirani and Professor Nikita Zhivotovskiy, for their clear, direct, and highly constructive feedback during and after my qualifying examination, which was both insightful and extremely helpful. I also sincerely appreciate their understanding, flexibility, and support throughout this process.

I would also like to thank all of the instructors whose courses I took at Berkeley. These courses formed an essential part of my Ph.D. training. This includes courses taught by faculty members I have already mentioned, as well as many others who generously shared their knowledge and perspectives. In particular, I would also like to thank Professor Bin Yu. I attended every lecture of STAT 215A. She was always willing to communicate individually with students, prepared exceptionally detailed course materials and assignments, and conveyed statistical ideas that were not only elegant but also highly practical.

I would like to thank all of my collaborators over the years, without whom much of my research would not have been possible. In particular, I would like to express my sincere gratitude to several collaborators who had a profound impact on my academic journey.

I am deeply thankful to Professor Mengdi Wang, under whose supervision I conducted a summer research project, which was my first exposure to research in machine learning. I am grateful for her careful guidance, for walking me through the process of refining research ideas and papers, and especially for the consistently positive and encouraging feedback she gave me. Beyond research supervision, she has been exceptionally generous with her support and

advocacy at important stages of my early academic development, for which I am sincerely grateful.

I would also like to thank Professor Andrea Zanette, whom I had the great fortune to meet on LinkedIn. I am sincerely grateful for his generous invitation to collaborate on research and for his openness and enthusiasm. Throughout our collaboration, he showed remarkable patience, understanding, and support, along with an incredible level of energy, creativity, and intellectual generosity.

I am also very grateful to Dr. Yu Bai for his exceptionally helpful technical guidance during our collaboration. Much of what I know about writing clean, effective code and debugging complex systems was taught to me directly by him, often through hands-on demonstrations. He has an extraordinary intuition and elegant taste in research, writing papers, and coding. I am also deeply thankful for his generosity in offering valuable opportunities for my career development, as well as for the practical advice and insights he shared about professional life.

I would like to sincerely thank Dr. Spencer Frei for collaborating with me on my first research project at Berkeley and for his guidance throughout the entire research process. He helped me think more clearly about problem formulation and structure arguments more carefully. His writing style and strong commitment to academic rigor left a deep impression on me and shaped how I think about research and communication.

I would also like to thank Dr. Jingfeng Wu for his guidance on research and paper writing. He sets very high standards for academic research, and working with him pushed me to think much more carefully and precisely in every definition and conclusion. This influence, though often subtle, substantially improved the quality of our projects.

I also want to thank La Shana Porlaris, David Apilado Jr., Tanisha Robinson, Keyla Y. Gomez, Ryan Lovett, and all staff in the Department of Statistics at Berkeley. They patiently answered countless questions regarding academic requirements, course registration, reimbursements, and departmental procedures, and consistently provided support across many aspects of the Ph.D. program. Their willingness to assist made it possible for me to focus on research with far fewer distractions, and I am sincerely grateful for their professionalism and support.

I would also like to thank my managers and colleagues in the industry during my internship. In particular, I want to thank Dr. Rudy Chin for his consistently positive and supportive feedback, as well as for his exceptionally careful guidance throughout the project. He was always generous with his time and readily available whenever I had questions. He went out of his way to join my technical talk remotely, even waking up at 3 a.m. to attend. Beyond the project itself, I also greatly appreciate his honesty, generosity, and willingness to share valuable advice and support as I navigated important career decisions.

I would also like to thank Dr. Zhuang Ma for setting a high and rigorous standard during my internship and for consistently emphasizing the importance of being grounded in facts and reasoning from first principles in quantitative finance. I am also truly thankful to Dr. Fang Wu and Dr. Peng Zhao. They are individuals of remarkable character and conviction, and I am deeply grateful for their candid advice on career decisions. They were always willing

to take the time to speak with me one-on-one, patiently answering my many questions. Their openness and encouragement gave me a great deal of confidence and support.

I would also like to thank my colleagues and friends at Citadel Securities who supported and encouraged me during my internship, including Dr. Sheng Gao, Dr. Shuxiao Chen, Dr. Tongzheng Ren, Dr. Xinmeng Huang, Dr. Zhen Zhang, Dr. Florian Kreyssig, Alex Shen, and Dr. Xiangyi Huang. From the interview process and pre-internship travel to the internship itself, they offered tremendous help and guidance.

I would also like to thank Dr. Renjie Zheng, Dr. Guanlin Liu, Dr. Zheng Wu, and all of my colleagues at TikTok for their support and collaboration. I am grateful for the opportunity to work with such a talented and supportive group of people, and I truly appreciated the open discussions, teamwork, and encouragement I received during my time there.

I am very fortunate to have been surrounded by a supportive and talented group of peers and cohorts. Specifically, I would like to thank Tianyu Guo, Licong Lin, Baihe Huang, Yuhang Cai, and Hengyu Fu. I have enjoyed sharing dinners, playing badminton and card games, hiking trails, and discussing research questions with all of you. I would also like to thank Dr. Sizhu Lu, Yaxuan Huang, Yang Chu, Austin Zane, Dr. Saptarshi Chakraborty, Dr. Omer Ronen, Lucas Da Rocha Schwengber, Dr. Druv Pai, Dr. Simon Zhai, Fangzhou Su, Dr. Hanlin Zhu, Karissa Huang, Andy Shen, James Butler, Zhexiao Lin, Arisa Sadeghpour, Van Hovenga, Chengzhong Ye, and Neo Yin.

On a more personal note, I want to thank my partner, Jiani Wang, and my two cats, Coalball and Snowball. I am deeply grateful to her for her unwavering support throughout my undergraduate years and Ph.D. We have known each other for over five years. Over the years, our marriage has been an important source of reflection and personal growth for me, and I am grateful for what I have learned through this experience. I sincerely wish her happiness and fulfillment in whatever paths lie ahead. I also thank my two cats, Coalball and Snowball, for their endlessly energetic and adorable companionship, bringing joy to my life.

I would like to thank my parents, Fang Wu and Jian Zhang. Over more than twenty-five years of my life, they have continuously supported me in ways both visible and unseen. The environments I grew up in—and the values and expectations that shaped those environments—changed many times. I am deeply grateful that, amid so much change, they were able to offer me moments of stability and a sense of peace, and that they made genuine efforts to understand me, even when my path diverged from what might traditionally be expected.

If I may claim even a small measure of accomplishment, it is because of the many individuals who have supported and guided me along the way. Their help sustained me through difficult periods and allowed me to continue moving forward, even when progress was slow. Ultimately, whether in research, in professional work, or in personal growth, I have come to view these experiences with humility and gratitude. I am thankful for the guidance and perspective that carried me through this journey, and for the faith that gave me strength and direction.

Chapter 1

Introduction

In recent years, *foundation models*—large-scale neural networks trained on broad, heterogeneous data—have driven major advances across natural language processing, computer vision, and generative modeling. For instance, Transformer-based language models, including the GPT series (Radford et al., 2018; Radford et al., 2019; Brown et al., 2020; OpenAI, 2023), have demonstrated strong capabilities in language understanding and generation, and now serve as core components in modern AI systems.

The gap between foundation models and classical statistical and machine learning models can be traced to two intertwined shifts. The first is *scale*: modern foundation models have massive parameter counts and are trained with huge datasets and compute budgets. This is empirically associated with qualitative changes in behavior, including scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), emergent capabilities (Wei et al., 2022), and inference-time generalization behaviors such as in-context learning (Brown et al., 2020). The second shift is *non-classical mechanisms*: training dynamics and model behavior can differ substantially from standard textbook assumptions. For example, over-parameterization can lead to benign overfitting and implicit bias, where optimization selects solutions with strong generalization despite the existence of many interpolating solutions (Bartlett et al., 2020; Soudry et al., 2018; Ji and Telgarsky, 2019). Moreover, large-scale training often uses stepsizes that violate classical sufficient conditions for stability, yet still exhibits fast progress and convergence consistent with the edge-of-stability picture (Cohen et al., 2020). Together, these observations indicate that, while foundation models are highly effective, comparably precise theoretical explanations of these phenomena remain limited relative to what is available for simpler models.

In addition to raising new explanatory questions, the scale and generality of foundation models also create new constraints and new algorithmic requirements. First, *resource constraints* become central: both training and deployment are compute- and memory-intensive, and even purely inference-time procedures (e.g., generating and selecting candidates) can incur substantial cost under realistic budgets, making efficiency a first-order algorithmic concern rather than a secondary engineering consideration. Second, *safety and reliability constraints* become more important: large models may memorize and reproduce sensitive or

copyrighted content, which can surface through direct regurgitation or targeted extraction (Carlini et al., 2021; Carlini et al., 2022), and they may produce outputs misaligned with user intent or human values. These concerns are qualitatively different from the small-model era, where models were often narrow and easier to constrain by design. Therefore, when designing algorithms for modern foundation models, we must explicitly account for both safety and resource constraints from the outset, which makes the development of effective methods substantially more challenging.

Since the pioneering work of Valiant (1984), Vapnik (1995), and others, learning theory has provided a powerful set of principles and tools—drawing from statistics, optimization, and information theory—to both explain model behavior and guide the development of better algorithms. In the era of foundation models, a central challenge is to extend this tradition to settings where models are over-parameterized and operate far from classical regimes: *how can we use statistical and optimization principles to (i) explain the empirical phenomena exhibited by modern foundation models, and (ii) design new methods that improve their effectiveness, efficiency, and safety?* This thesis contributes to this goal in two complementary modes: (i) theoretical analyses in simplified but representative settings that isolate key phenomena, and (ii) algorithmic designs that address practical constraints in realistic LLM workflows, including stability and computational efficiency.

We organize the thesis around two overall questions:

- *Explanation: How can we explain key empirical phenomena observed in foundation models?*
- *Design: How can we design theoretically grounded algorithms that improve the training and deployment of foundation models under safety and resource constraints?*

These questions are instantiated in four complementary directions. The first part of the thesis focuses on *explanation*. Chapters 2 and 3 study in-context learning as a canonical example of inference-time generalization in Transformers, and analyze both training dynamics and architectural effects in simplified settings. Chapter 4 studies gradient descent beyond the descent-lemma regime and characterizes how large *adaptive* stepsizes can yield fast, provably optimal convergence in a canonical separable classification problem, shedding light on edge-of-stability training dynamics.

The second part of the thesis focuses on *algorithmic design*. Chapter 5 addresses unlearning as a principled way to remove the influence of designated data from an LLM while preserving utility, and proposes a stable preference-optimization-based objective. Finally, Chapter 6 studies inference-time alignment under hardware constraints and proposes a resource-inspired decoding strategy that accelerates Best-of-N selection by reducing wasted computation.

1.1 In-Context Learning of Linear Transformers

A central capability of modern foundation models is *inference-time adaptation*: the model can change its input–output behavior at test time without any parameter updates. In large

language models (LLMs), this is most prominently seen in *in-context learning* (ICL), where a model conditions on a short prompt containing instructions and/or a few input–output demonstrations, and then produces correct predictions for a query in the same format. This phenomenon, popularized by the GPT series (Radford et al., 2019; Brown et al., 2020), departs from the classical pretraining–fine-tuning paradigm: instead of fine-tuning a pretrained model on a task-specific dataset, a single pretrained model can often infer the latent task from context and execute it immediately. For instance, given a prompt such as

$$\textit{cat: chat, dog: chien, human: ?},$$

a pretrained model may infer the latent task (English-to-French translation) and complete the query accordingly. From a deployment perspective, ICL turns the prompt into a lightweight *control interface* for specifying tasks and behaviors; from a theoretical perspective, it raises a basic question: *what computation is the model performing on the in-context examples, and why does standard training produce such behavior?*

A widely used formalization isolates this phenomenon by considering a distribution over tasks and prompts. Concretely, one studies the problem of predicting the value of an unknown function $h \in \mathcal{H}$ from a small number of in-context examples:

$$x_1, h(x_1);\ x_2, h(x_2);\ \ldots;\ x_T, h(x_T);\ x_{\text{query}}, ? \tag{1.1}$$

where the covariates $\{x_i\}$ are drawn from a data distribution and the task h is drawn from a distribution over a hypothesis class $\mathcal{H}$. In the synthetic *training on in-context examples* setup, one repeatedly samples $(h, x_1, \ldots, x_T, x_{\text{query}})$ and trains a transformer end-to-end to predict the next token (or next label) in sequences of the form (1.1). Although this differs from web-scale language pretraining, it captures a core aspect of ICL: the model must learn a mapping from a small *inference-time dataset* (the context) to a predictor that generalizes to a held-out query. This setup also makes it possible to ask precise questions about (i) the *effective function class* implemented by the trained model at test time, (ii) the role of architecture (attention, MLP blocks, depth), and (iii) robustness to distribution shift between the training and test prompt distributions.

On the empirical side, a growing literature documents both the power and the fragility of ICL. Large pretrained models exhibit strong few-shot performance across many tasks (Brown et al., 2020), but performance can depend sensitively on prompt formatting, demonstration selection, and the distributional match between training and test prompts (Min et al., 2022). In the synthetic training-on-prompts setting, the work of Garg et al. (2022) demonstrates that transformers trained on sequences (1.1) can recover classical estimators across several hypothesis classes: near-OLS behavior for linear regression, near-LASSO behavior for sparse regression, and strong performance on decision-tree tasks that can surpass standard hand-designed baselines. Subsequent studies further explore structured task distributions where Bayes-optimal predictors are tractable, providing evidence that trained transformers can approach Bayes-optimal performance in certain regimes (Panwar et al., 2023b), and systematically probe distribution shift phenomena at test time (Ahuja and Lopez-Paz, 2023). Together,

these results suggest that ICL is not merely superficial pattern matching; rather, training can induce nontrivial inductive biases that exploit shared structure across tasks—while still leaving open when such behavior is stable under shifts in the prompt distribution.

On the theory side, much of the progress has been driven by an *algorithm-learning* viewpoint: the transformer forward pass implicitly implements a learning algorithm that "fits" the in-context demonstrations and then applies the fitted rule to the query. This perspective is supported by two complementary lines of work. First, mechanistic interpretability studies identify internal circuits (e.g., induction-head-like mechanisms) that enable copying or pattern completion and may underlie certain forms of ICL (Olsson et al., 2022). Second, a series of constructive approximation-theoretic results shows that transformers can represent recognizable learning algorithms within the forward computation: for instance, one-step (or multi-step) gradient descent for linear regression (Akyürek et al., 2022; Von Oswald et al., 2023), preconditioned variants (Ahn et al., 2023c), second-order methods (Fu et al., 2023a), and linear classification procedures such as SVM-like algorithms (Li et al., 2023b; Tarzanagh et al., 2023). These works provide compelling *existence* results—there exist transformer parameters that implement certain learning rules—but they leave open a complementary and practically central question: *does standard training (e.g., gradient descent/gradient flow on in-context examples) actually find such solutions, and if so, what solutions does it select among many possibilities?* Relatedly, robustness questions remain: how does the learned in-context procedure behave when the task distribution or the covariate distribution in prompts changes, and what architectural features mitigate brittleness?

In Chapter 2, we address these gaps by shifting focus from representational constructions to *training dynamics*. We analyze a simplified but representative model: a transformer with a single linear self-attention layer trained by gradient flow on linear regression in-context examples. We show that despite non-convexity, gradient flow with a suitable random initialization finds a global minimum of the objective function. At this global minimum, when given a test prompt of labeled examples from a new prediction task, the transformer achieves prediction error competitive with the best linear predictor over the test prompt distribution. We additionally characterize the robustness of the trained transformer to a variety of distribution shifts and show that although a number of shifts are tolerated, shifts in the covariate distribution of the prompts are not. Motivated by this, we consider a generalized ICL setting where the covariate distributions can vary across prompts. We show that although gradient flow succeeds at finding a global minimum in this setting, the trained transformer is still brittle under mild covariate shifts. We complement this finding with experiments on large, nonlinear transformer architectures which we show are more robust under covariate shifts. This chapter is adapted from my joint work with Doctor Spencer Frei and Prof. Peter L. Bartlett. This paper was published in the Journal of Machine Learning Research (JMLR) (Zhang et al., 2024c).

In Chapter 3, we further investigate how architectural components beyond attention shape in-context adaptation. We move beyond attention-only mechanisms by studying a *Linear Transformer Block* (LTB) that combines a linear attention component with a linear multi-layer perceptron (MLP) component. For ICL of linear regression with a Gaussian prior

and a non-zero mean, we show that LTB can achieve nearly Bayes optimal ICL risk. In contrast, using only linear attention must incur an irreducible additive approximation error. Furthermore, we establish a correspondence between LTB and one-step gradient descent estimators with learnable initialization (GD-$\boldsymbol{\beta}$), in the sense that every GD-$\boldsymbol{\beta}$ estimator can be implemented by an LTB estimator and every optimal LTB estimator that minimizes the in-class ICL risk is effectively a GD-$\boldsymbol{\beta}$ estimator. Finally, we show that GD-$\boldsymbol{\beta}$ estimators can be efficiently optimized with gradient flow, despite a non-convex training objective. Our results reveal that LTB achieves ICL by implementing GD-$\boldsymbol{\beta}$, and they highlight the role of MLP layers in reducing approximation error. This chapter is adapted from my joint work with Doctor Jingfeng Wu and Prof. Peter L. Bartlett. This paper was published at the 38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024) (Zhang et al., 2024e).

1.2 Large Adaptive Stepsizes for Logistic Regression

Chapter 4 highlights an important gap between classical optimization theory and modern deep learning practice concerning the choice of stepsize. Classical analyses of gradient descent (GD) typically assume a *small* (often constant) stepsize so that the objective decreases monotonically, a property commonly referred to as the *descent lemma*. Concretely, in this classical regime, GD is analyzed under a constant stepsize

$$w_{t+1} = w_t - \eta \nabla L(w_t), \qquad t \geq 0, \tag{1.2}$$

where $\eta > 0$ is fixed. If $L(\cdot)$ is β-smooth, the descent lemma implies that for $\eta \in (0, 2/\beta)$ each step guarantees a decrease of the objective, and standard complexity bounds follow (e.g., $O(1/t)$ rates for smooth convex objectives and linear rates for smooth strongly convex objectives); see, e.g., (Nesterov, 2013; Nesterov, 2018). This monotone-descent paradigm is central to textbook theory and underlies the conventional wisdom that "stable" training requires conservative stepsizes.

Empirically, however, large-scale deep learning is often trained outside this monotone regime. A striking illustration is the *Edge of Stability* (EoS) phenomenon identified by Cohen et al. (2020): when training neural networks with GD using stepsizes that violate the descent lemma, the training loss can oscillate substantially over short timescales, yet it continues to decrease over long timescales. Moreover, Cohen et al. (2020) report that along the training trajectory, the largest Hessian eigenvalue (a notion of "sharpness") tends to self-adjust so that it hovers near the critical threshold $2/\eta$, i.e., near the boundary between local stability and instability. These observations suggest that operating close to instability is not merely tolerable but can be beneficial in practice, motivating a theoretical understanding of GD beyond the descent-lemma regime. A canonical setting where such behavior can already be observed—and where sharp theoretical statements are possible—is logistic regression with linearly separable data, a standard testbed for studying the implicit bias of GD (Soudry et al., 2018; Ji and Telgarsky, 2018). Recent work further shows that even *constant* large

stepsizes can yield provable acceleration in this setting and its regularized variants (Wu et al., 2024a; Wu et al., 2025a), reinforcing the relevance of EoS-type dynamics in simple models.

Chapter 4 focuses on the EoS regime under *adaptive* stepsizes. We consider the general GD recursion

$$w_{t+1} \;=\; w_t \;-\; \eta_t \nabla L(w_t), \qquad t \geq 0, \tag{1.3}$$

where η_t is the stepsize at iteration t. Classical analyses typically operate in a descent-lemma regime, in which η_t is small enough to guarantee a monotone decrease of the objective. Concretely, let $\mathcal{I}_t$ denote the line segment connecting w_t and w_{t+1}, and define a local smoothness constant

$$\Lambda_t \;:=\; \sup_{u \in \mathcal{I}_t} \lambda_{\max}\Big(\nabla^2 L(u)\Big),$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a symmetric matrix. A standard second-order Taylor argument implies that if $\eta_t < 2/\Lambda_t$, then GD satisfies the descent lemma and the loss is guaranteed to decrease, for example in the form

$$L(w_{t+1}) \;\leq\; L(w_t) \;-\; \eta_t\left(1 - \frac{1}{2}\eta_t \Lambda_t\right) \|\nabla L(w_t)\|^2,$$

and in particular $L(w_{t+1}) \leq L(w_t)$. Under this monotone regime, a large body of theory establishes convergence guarantees for GD and related first-order methods (Lan, 2020). In separable logistic regression, it has also been shown that when stepsizes are chosen to preserve monotone descent, *adaptive* stepsizes can accelerate the convergence of both the risk and the margin: Nacson et al. (2019) and Ji and Telgarsky (2021) establish risk convergence rates on the order of $\exp(-\Omega(\sqrt{t}))$ and $\exp(-\Omega(t))$, respectively. Importantly, these analyses rely crucially on monotone decrease of the risk (or a closely related potential), which is enforced by keeping stepsizes within the descent-lemma regime.

For linearly separable logistic regression, the minimizer lies at infinity, and as the risk decreases, the effective curvature along the trajectory shrinks. This motivates stepsize rules that *compensate* for shrinking curvature by increasing as the risk decreases. Accordingly, we focus on stepsizes that adapt to the current risk, scaled by a constant hyperparameter $\eta > 0$. More specifically, we study stepsizes of the form

$$\eta_t \;:=\; \eta \cdot \Big(-(\ell^{-1})'\Big)(L(w_t)) \;\approx\; \frac{\eta}{L(w_t)}, \tag{1.4}$$

where ℓ is the underlying pointwise classification loss (e.g., exponential or logistic loss) and ℓ^{-1} denotes its inverse. The approximation on the right captures the prototypical behavior that the stepsize grows roughly like $1/L(w_t)$ as the risk decreases. This brings us to the central question of this chapter: *can we obtain the benefits of large stepsizes while using a principled adaptive stepsize rule?*

Building on the adaptive rule (1.4), we show that GD with *large and adaptive* stepsizes can achieve an arbitrarily small risk immediately after at most $1/\gamma^2$ burn-in iterations, where $\gamma > 0$ is the margin of the dataset, even though the risk evolution during training may be

non-monotone. More precisely, after this burn-in phase, the *average iterate* of GD attains a risk bounded by $\exp(-\Theta(\eta))$; consequently, by taking η sufficiently large, GD can drive the risk to an arbitrarily small value as soon as the burn-in phase ends. To isolate the role of the EoS phase, we also prove a lower bound showing that this acceleration cannot be achieved if the iterates never enter an EoS regime and the loss decreases monotonically throughout. Furthermore, we establish minimax lower bounds by constructing hard margin-γ instances on which any batch (or online) first-order method requires $\Omega(1/\gamma^2)$ iterations to identify a linear separator. This implies that GD with large adaptive stepsizes is minimax optimal (up to constants) among first-order batch methods. Notably, the classical Perceptron algorithm (Novikoff, 1962), a first-order online method, also achieves a $1/\gamma^2$ step complexity for finding a separator, matching our GD guarantees even in constants. Finally, we extend the analysis beyond logistic regression by providing conditions on the loss under which the same phenomenon persists, and by proving analogous guarantees for a class of two-layer networks. This chapter is adapted from Zhang et al. (2025a). A shorter version of this paper was published at the International Conference on Machine Learning (ICML).

1.3 Effectively Unlearning Large Language Models

A central challenge in deploying foundation models is not only to *adapt* them to downstream needs, but also to *control* what they retain and what they should not reproduce. Large language models (LLMs) trained on massive web-scale corpora can memorize portions of their training data, and such memorization can surface at inference time via verbatim regurgitation or targeted extraction attacks (Carlini et al., 2021; Carlini et al., 2022). This raises concrete concerns when the pretraining data contains sensitive personal information, proprietary content, or copyrighted material. From the standpoint of reliable deployment, these issues motivate the problem of *machine unlearning*: *given a trained model, can we efficiently remove the influence of a designated subset of training examples, while preserving the model's utility on the remaining data distribution?*

A standard formalization partitions the original training dataset into a *forget set* D_{FG} and a *retain set* D_{RT}, and considers an initial model π_{ref} trained on $D_{\mathrm{FG}} \cup D_{\mathrm{RT}}$. The goal is to produce an updated model π whose behavior is close to that of the *retrained* model π_{retrain} that would have been obtained by training from scratch on D_{RT} alone (Cao and Yang, 2015). In principle, retraining provides the gold standard, but for modern LLMs, it is typically intractable. Consequently, practical unlearning must operate in a regime where (i) the procedure is far cheaper than full retraining, and (ii) the output model achieves a favorable *tradeoff* between *forget quality* (the extent to which information in D_{FG} is removed) and *model utility* (performance on D_{RT} and general tasks). This tradeoff is intrinsic: aggressive unlearning can erase targeted behaviors but may also destroy useful generalization, whereas conservative updates preserve utility but fail to forget.

Classical work on machine unlearning has developed formal metrics and algorithms in settings where one can maintain additional training-time state or compute second-order

information. For example, a line of work proposes deletion guarantees and efficient removal mechanisms by leveraging Hessian-based updates, influence-function approximations, or differential-privacy-inspired stability measures (Ginart et al., 2019; Guo et al., 2020; Sekhari et al., 2021). However, such methods often rely on computational primitives (e.g., Hessian computations or strong convexity assumptions) that are not directly compatible with large-scale deep networks, let alone modern foundation models. To bridge this gap, several systems- and training-protocol-based approaches have been proposed for deep models, such as sharded or incremental training strategies that facilitate faster removal at the cost of additional training-time structure (Bourtoule et al., 2021; Golatkar et al., 2020). More recently, the emergence of LLMs has triggered renewed interest in unlearning methods tailored to next-token prediction objectives and large-scale fine-tuning pipelines.

Prior to our work, most practical approaches to *LLM unlearning* were variations on a simple and intuitive idea: perform *gradient ascent* on the log-likelihood (or equivalently, gradient descent on the negative loss) of the forget set (Jang et al., 2022; Yao et al., 2024b). The rationale is that since π_{ref} was trained to *increase* likelihood on D_{FG}, one can attempt to "reverse" this effect by locally maximizing the forget-set loss. To mitigate utility degradation, GA is commonly combined with regularizers or retain-set objectives, including cross-entropy loss on D_{RT} or KL penalties that constrain deviation from π_{ref} on either retain or forget distributions (Yao et al., 2024b; Chen and Yang, 2023; Maini et al., 2024). In parallel, complementary directions include: task-specific constructions of positive samples to replace sensitive knowledge (e.g., "forgetting Harry Potter") (Eldan and Russinovich, 2023) and alternatives such as knowledge negation or representation-control-based removal (Liu et al., 2024; Li et al., 2024b). These works reflect a broader theme: for LLMs, unlearning is fundamentally a *control* problem over the model's output distribution, and effective solutions must jointly manage targeted forgetting, distributional drift, and optimization stability.

A key limitation of GA-based unlearning is an instability phenomenon we refer to as *catastrophic collapse* (Zhang et al., 2024d). Empirically, on challenging unlearning tasks, pure GA and GA augmented with regularizers can drive the model rapidly away from π_{ref}, causing model utility to deteriorate sharply; the model may produce low-diversity or nonsensical outputs, while forget quality improves only transiently and then also degrades, as the unlearned model also deviates far from the gold standard (the retrained model). This failure mode is consistent with the fact that GA explicitly maximizes the standard likelihood objective on D_{FG}, a direction that can be highly destabilizing in parameter space and can induce fast divergence from the reference model. In practice, GA-based methods often rely heavily on early stopping, but the appropriate stopping time can be instance-dependent and hard to diagnose. These observations suggest that effective unlearning requires an objective that remains *anchored* to the reference policy while still pushing down the likelihood of the forget set—that is, a more principled control objective than raw GA on log-likelihood.

In Chapter 5, we propose *Negative Preference Optimization* (NPO), a novel unlearning algorithm that casts unlearning as a variant of preference optimization that uses only negative samples (Zhang et al., 2024d). The starting point is the preference-optimization framework developed for RLHF-style alignment, in which one trains a policy relative to a reference

model using comparisons between preferred and dispreferred responses (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022b; Rafailov et al., 2024b). In the unlearning setting, each forget-set pair $(x, y) \in D_{\mathrm{FG}}$ can be viewed as providing only a dispreferred response y for prompt x, with no corresponding "preferred" completion. This leads to the NPO loss

$$L_{\mathrm{NPO},\beta}(\theta) \;=\; \frac{2}{\beta}\,\mathbb{E}_{(x,y)\sim D_{\mathrm{FG}}}\left[\log\left(1+\left(\frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right)^{\beta}\right)\right], \tag{1.5}$$

where $\beta > 0$ is an inverse-temperature parameter and π_{ref} is the fixed reference model. Minimizing (1.5) discourages assigning high probability to forget-set completions *relative* to the reference model, thereby directly aligning with the unlearning goal. We show that NPO is a strict generalization of gradient ascent: in the high-temperature limit $\beta \to 0$, the gradient of $L_{\mathrm{NPO},\beta}$ converges to that of the GA objective, while for moderate β the objective imposes a smoother, reference-anchored update.

Our contributions are both theoretical and empirical. On the theory side, we identify a mechanism underlying catastrophic collapse in GA-based unlearning and show that, in a simple but informative setting, the progression toward collapse under NPO is *exponentially slower* than under GA. This provides a principled explanation for why NPO yields more stable unlearning dynamics even when GA becomes unstable. On the empirical side, we evaluate NPO-based methods on both synthetic tasks and the TOFU benchmark (Maini et al., 2024). We find that NPO achieves a consistently better Pareto frontier between forget quality and model utility than GA-based baselines, and it substantially improves training stability and output readability (avoiding the low-diversity or "gibberish" degeneracy often observed under GA). Notably, on TOFU, NPO-based methods are the first to obtain non-trivial unlearning behavior when the forget set constitutes 50% (and more) of the data, a regime where prior methods already struggle at 10%. This chapter is mainly adapted from my joint work with Licong Lin, Doctor Yu Bai, and Prof. Song Mei (Zhang et al., 2024d), and was published at the first Conference on Language Models (CoLM 2024).

1.4 Fast Inference-Time Alignment

Beyond training-time objectives, a second axis along which one can *control* foundation-model behavior is the *decoding strategy* used at inference time. Modern post-training pipelines—including supervised fine-tuning, Reinforcement Learning from Human Feedback (RLHF), and preference-based objectives such as Direct Preference Optimization (DPO)—aim to fine-tune an LLM toward human intentions and safety constraints by directly modifying model parameters (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022b; Rafailov et al., 2024b). While effective, these post-training procedures introduce substantial complexity. They require curated data, reward modeling or preference modeling, and engineering-heavy pipelines that include repeated rounds of training and evaluation. Moreover, the optimization itself (especially for RL-based post-training) can be sensitive to hyperparameters and prone

to instabilities. This motivates *inference-time alignment*: rather than changing the model weights, one seeks to improve aligned behavior by changing how responses are generated and selected at deployment time (Nakano et al., 2021; Mudgal et al., 2024; Wang et al., 2024). Inference-time methods can be especially attractive when one wishes to (i) deploy a base or instruction-tuned model without heavy fine-tuning via an additional RLHF stage, or (ii) trade compute for quality dynamically under resource constraints.

A particularly simple and widely used inference-time alignment primitive is *Best-of-N* (BoN, also known as reward-model-based rejection sampling or reranking). Given a prompt x, Best-of-N samples N candidate completions $Y_1, \ldots, Y_N$ from a base policy $\pi(\cdot|x)$ and selects the candidate with highest score under a reward model $r(x, Y)$:

$$Y_{\mathrm{BoN}}(x) \in \arg\max_{1 \le i \le N} r(x, Y_i).$$

Best-of-N was popularized in early RLHF systems and has remained a strong baseline in aligned generation (Nakano et al., 2021; Ouyang et al., 2022). It enjoys several practical advantages: it is conceptually simple, essentially hyperparameter-free aside from N, and the parameter N can be tuned at inference time without retraining. Moreover, Best-of-N has become a building block in broader alignment pipelines: it is commonly used to generate high-quality synthetic data for later supervised fine-tuning (often termed Expert Iteration or Iterative fine-tuning), and played an important role in the development of widely used instruction-tuned models (Touvron et al., 2023b). At a higher level, Best-of-N can be viewed as a way to convert a reward model into a controlled sampling policy, often yielding competitive quality relative to more complex post-training approaches when sufficient compute is available.

The main drawback of Best-of-N is computational cost. Even when candidate generation is parallelized so that wall-clock latency is not strongly affected by N, the *total* compute scales roughly linearly with N. In practice, large values of N can quickly exceed the maximum batch size that fits on a single accelerator, forcing multi-GPU execution or multiple sequential batches, each of which increases cost and/or latency. As a result, practical deployments typically use modest N (e.g., single- to double-digit candidates), whereas substantially larger N may be needed to match the best attainable reward levels in some settings (Nakano et al., 2021; Dubois et al., 2024b; Gao et al., 2023). This leads to a trade-off: Best-of-N is easy to use at inference time, but increasing N directly is computationally wasteful since most candidates are never selected.

Several lines of work aim to reduce inference cost or to incorporate reward signals into decoding more directly. On the alignment side, controlled decoding and inference-time guidance methods attempt to steer generation using auxiliary models or constrained objectives, sometimes avoiding the need to sample very large candidate sets (Mudgal et al., 2024; Wang et al., 2024). On the systems side, significant progress has been made in high-throughput LLM serving and memory management, improving the feasible batch size under KV-cache constraints (Kwon et al., 2023). There is also a large literature on speculative decoding and related techniques that accelerate autoregressive generation by using a fast draft model

and verifying with a stronger model (Leviathan et al., 2023). However, these approaches primarily aim to accelerate a *single* generation (or to reduce the per-token cost), rather than treating Best-of-N as an end-to-end procedure to be accelerated. The main inefficiency of Best-of-N is different: it intentionally generates many candidates and ultimately discards most low-quality responses. Therefore, a method tailored to accelerating Best-of-N must reduce the *total number of generated tokens that are later discarded*, while preserving the quality gains obtained by selecting the highest-scoring completion.

In Chapter 6, we study this inefficiency through a resource-oriented lens and propose *Speculative Rejection*, a resource-inspired inference-time alignment algorithm that aims to retain the quality benefits of Best-of-N while dramatically reducing wasted computation. Our key observation is that reward models often provide meaningful information about response quality *before* a completion is finished: for many prompts, the relative ranking induced by reward scores on partial prefixes is already predictive of the final ranking. This suggests a natural control strategy for compute allocation: begin with a large pool of candidates to explore diverse generations, but *early-stop* and discard clearly low-quality candidates based on partial reward signals, and devote the remaining compute budget to finishing only the most promising trajectories.

Concretely, Speculative Rejection generates a large initial batch of responses until approaching the GPU memory limit, then enters a sequence of *rejection rounds*. In each round, it scores each partially generated response with a reward model, computes a prompt-dependent cutoff (e.g., an α-quantile threshold of partial rewards), and discards the bottom fraction of candidates. The algorithm then continues generation only for the surviving set, repeating this procedure multiple times as memory pressure builds, until all remaining candidates complete and the final output is selected by the full reward score. By design, Speculative Rejection explicitly couples the decoding policy with hardware constraints: it uses early rejection both to prevent memory exhaustion and to reallocate compute toward trajectories that are most likely to win under the reward model.

Our results show that this resource-inspired control of compute can yield large practical gains. Across a range of model and reward-model pairs, Speculative Rejection achieves reward performance comparable to that of Best-of-N with much larger N, while requiring substantially fewer GPU resources—typically improving compute efficiency by factors on the order of $16\times$ to $32\times$ relative to naive Best-of-N baselines at matched reward levels. This chapter is adapted from my work jointly with Hanshi Sun, Momin Haider, Huitao Yang, Jiahao Qiu, Doctor Ming Yin, Prof. Mengdi Wang, Prof. Peter Bartlett, and Prof. Andrea Zanette. Our work was published at the 38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024) (Sun et al., 2024b).

1.5 What is Not in the Thesis?

For clarity and coherence, some collaborative work involving the author is not included in this thesis. This includes works on the theory of sequential decision making and reinforcement

learning (Ni et al., 2022; Zhang et al., 2022; Zhang and Zanette, 2023; Zhang et al., 2024f), unlearning large language models (Fan et al., 2024), large language model explainability (Guo et al., 2025), and process-supervised reward models (Chen et al., 2024b).

Chapter 2

In-Context Learning of a Linear Self-Attention

2.1 Introduction

This chapter initiates the explanatory part of the thesis by studying in-context learning in a tractable Transformer model. Transformer-based neural networks have quickly become the default machine learning model for problems in natural language processing, forming the basis of chatbots like ChatGPT (OpenAI, 2023), and are increasingly popular in computer vision (Dosovitskiy et al., 2021). These models can take as input sequences of tokens and return relevant next-token predictions. When trained on sufficiently large and diverse datasets, these models are often able to perform *in-context learning* (ICL): when given a short sequence of input-output pairs (called a *prompt*) from a particular task as input, the model can formulate predictions on test examples without having to make any updates to the parameters in the model.

Recently, Garg et al. (2022) initiated the investigation of ICL from the perspective of learning particular function classes. At a high level, this refers to when the model has access to instances of prompts of the form $(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_N, h(\mathbf{x}_N), \mathbf{x}_{\mathsf{query}})$ where $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}}$ are sampled i.i.d. from a distribution $\mathcal{D}_{\mathbf{x}}$ and h is sampled independently from a distribution over functions in a function class $\mathcal{H}$. The transformer succeeds at in-context learning if when given a new prompt $(\mathbf{x}'_1, h'(\mathbf{x}'_1), \ldots, \mathbf{x}'_N, h'(\mathbf{x}'_N), \mathbf{x}'_{\mathsf{query}})$ corresponding to an independently sampled h' it is able to formulate a prediction for $\mathbf{x}'_{\mathsf{query}}$ that is close to $h'(\mathbf{x}'_{\mathsf{query}})$ given a sufficiently large number of examples N. The authors showed that when transformer models are trained on prompts corresponding to instances of training data from a particular function class (e.g., linear models, neural networks, or decision trees), they succeed at in-context learning, and moreover the behavior of the trained transformers can mimic those of familiar learning algorithms like ordinary least squares.

Following this, a number of follow-up works provided constructions of transformer-based neural network architectures which are capable of achieving small prediction error for

query examples when the prompt takes the form $(\mathbf{x}_1, \langle \boldsymbol{w}, \mathbf{x}_1 \rangle, \ldots, \mathbf{x}_N, \langle \boldsymbol{w}, \mathbf{x}_N \rangle, \mathbf{x}_{\mathsf{query}})$ where $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}}, \boldsymbol{w} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ (Von Oswald et al., 2023; Akyürek et al., 2022). However, this leaves open the question of how it is that *gradient-based optimization algorithms* over transformer architectures produce models which are capable of in-context learning.[1]

In this chapter, we investigate the learning dynamics of gradient flow in a simplified transformer architecture when the training prompts consist of random instances of linear regression datasets. Our main contributions are as follows.

- We establish that for a class of transformers with a single layer and with a linear self-attention module (LSAs), gradient flow on the population loss with a suitable random initialization converges to a global minimum of the population objective, despite the non-convexity of the underlying objective function.
- We characterize the learning algorithm that is encoded by the transformer at convergence, as well as the prediction error achieved when the model is given a test prompt corresponding to a new (and possibly nonlinear) prediction task.
- We use this to conclude that transformers trained by gradient flow indeed in-context learn the class of linear models. Moreover, we characterize the robustness of the trained transformer to a variety of distribution shifts. We show that although a number of shifts can be tolerated, shifts in the covariate distribution of the features $\mathbf{x}_i$ cannot.
- Motivated by this failure under covariate shift, we consider a generalized setting of in-context learning where the covariate distribution can vary across prompts. We provide global convergence guarantees for LSAs trained by gradient flow in this setting and show that even when trained on a variety of covariate distributions, LSAs still fail under covariate shift.
- We then empirically investigate the behavior of large, nonlinear transformers when trained on linear regression prompts. We find that these more complex models are able to generalize better under covariate shift, especially when trained on prompts with varying covariate distributions.

2.2 Related Works

The literature on transformers and non-convex optimization in machine learning is vast. In this section, we will focus on those works most closely related to theoretical understanding of in-context learning of function classes.

As mentioned previously, Garg et al. (2022) empirically investigated the ability for transformer architectures to in-context learn a variety of function classes. They showed

[1] We note a concurrent work also explores the optimization question we consider here (Ahn et al., 2023c); we shall provide a more detailed comparison to this work in Section 2.2.

that when trained on random instances of linear regression, the models' predictions are very similar to those of ordinary least squares. Additionally, they showed that transformers can in-context learn two-layer ReLU networks and decision trees, showing that by training on differently-structured data, the transformers learn to implement distinct learning algorithms. A number of works further investigated the types of algorithms implemented by transformers trained on in-context examples of linear models (Panwar et al., 2023a; Ahuja and Lopez-Paz, 2023).

Akyürek et al. (2022) and Von Oswald et al. (2023) examined the behavior of transformers when trained on random instances of linear regression, as we do in this chapter. They considered the setting of isotropic Gaussian data with isotropic Gaussian weight vectors, and showed that the trained transformer's predictions mimic those of a single step of gradient descent. They also provided a construction of transformers which implement this single step of gradient descent. By contrast, we explicitly show that gradient flow provably converges to transformers which learn linear models in-context. Moreover, our analysis holds when the covariates are anisotropic Gaussians, for which a single step of vanilla gradient descent is unable to achieve small prediction error.[2]

Let us briefly mention a number of other works on understanding in-context learning in transformers and other sequence-based models. Han et al. (2023) suggests that Bayesian inference on prompts can be asymptotically interpreted as kernel regression. Dai et al. (2023) interprets ICL as implicit fine-tuning, viewing large language models as meta-optimizers performing gradient-based optimization. Xie et al. (2022) regards ICL as implicit Bayesian inference, with transformers learning a shared latent concept between prompts and test data, and they prove the ICL property when the training distribution is a mixture of HMMs. Similarly, Wang et al. (2023b) perceives ICL as a Bayesian selection process, implicitly inferring information pertinent to the designated tasks. Li et al. (2024c) explores the functional resemblance between a single layer of self-attention and gradient descent on a softmax regression problem, offering upper bounds on their difference. Min et al. (2022) notes that the alteration of label parts in prompts does not drastically impair the ICL ability. They contend that ICL is invoked when prompts reveal information about the label space, input distribution, and sequence structure.

Another collection of works have sought to understand transformers from an approximation theoretic perspective. Yun et al. (2020a) and Yun et al. (2020b) established that transformers can universally approximate any sequence-to-sequence function under some assumptions. Investigations by Edelman et al. (2022) and Likhosherstov et al. (2023) indicate that a single-layer self-attention can learn sparse functions of the input sequence, where sample complexity and hidden size are only logarithmic relative to the sequence length. Further studies by Pérez et al. (2019), Dehghani et al. (2019), and Bhattamishra et al. (2020) indicate that the vanilla transformer and its variants exhibit Turing completeness. Liu et al. (2023a) showed that

[2]To see this, suppose $(\mathbf{x}_i, y_i)$ are i.i.d. with $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ and $y = \langle \boldsymbol{w}, \mathbf{x} \rangle$. A single step of gradient descent under the squared loss from a zero initialization yields the predictor $\mathbf{x} \mapsto \mathbf{x}^\top \left(\frac{1}{n} \sum_{i=1}^n y_i \mathbf{x}_i \right) = \mathbf{x}^\top \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \right) \boldsymbol{w} \approx \mathbf{x}^\top \mathbf{\Lambda} \boldsymbol{w}$. Clearly, this is not close to $\mathbf{x}^\top \boldsymbol{w}$ when $\mathbf{\Lambda} \neq \mathbf{I}_d$.

transformers can approximate finite-state automata with few layers. Bai et al. (2023) showed that transformers can implement a variety of statistical machine learning algorithms as well as model selection procedures. Abernethy et al. (2024) showed that a pretrained transformer can be used to define a transformer that segments a prompt into examples and labels and learns to solve a sparse retrieval task. Zhang et al. (2025b) interpreted in-context learning via a Bayesian model averaging process.

A handful of recent works have developed provable guarantees for transformers trained with gradient-based optimization. Jelassi et al. (2022) analyzed the dynamics of gradient descent in vision transformers for data with spatial structure. Li et al. (2023c) demonstrated that a single-layer transformer trained by a gradient method could learn a topic model, treating learning semantic structure as detecting co-occurrence between words and theoretically analyzing the two-stage dynamics during the training process.

Finally, we note a concurrent work by Ahn et al. (2023c) on the optimization landscape of single-layer transformers with linear self-attention layers as we do in this chapter. They show that there exist global minima of the population objective of the transformer that can achieve small prediction error with anisotropic Gaussian data, and they characterize some critical points of deep linear self-attention networks. In this chapter, we show that despite nonconvexity, gradient flow with a suitable random initialization converges to a global minimum that achieves small prediction error for anisotropic Gaussian data. We also characterize the prediction error when test prompts come from a new (possibly nonlinear) task, when there is distribution shift, and when transformers are trained on prompts with possibly different covariate distributions across prompts.

2.3 Background

Notation

We first describe the notation we use in the chapter. We write $[n] = \{1, 2, ..., n\}$. We use $\otimes$ to denote the Kronecker product, and Vec the vectorization operator in column-wise order. For example, $\text{Vec}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = (1, 3, 2, 4)^\top$. We write the inner product of two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ as $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}\mathbf{B}^\top)$. We use $\mathbf{0}_n$ and $\mathbf{0}_{m\times n}$ to denote the zero vector and zero matrix of size n and $m \times n$, respectively. For a general matrix $\mathbf{A}$, $\mathbf{A}_{k:}$ and $\mathbf{A}_{:k}$ denote the k-th row and k-th column, respectively. We denote the matrix operator norm and Frobenius norm as $\|\cdot\|_{op}$ and $\|\cdot\|_F$. We use $\mathbf{I}_d$ to denote the d-dimensional identity matrix, and sometimes we also use $\mathbf{I}$ when the dimension is clear from the context. For a positive semi-definite matrix $\mathbf{A}$, we write $\|\mathbf{x}\|_{\mathbf{A}}^2 := \mathbf{x}^\top \mathbf{A}\mathbf{x}$. Unless otherwise defined, we use lower case letters for scalars and vectors, and use upper case letters for matrices.

A Statistical Framework for In-Context Learning (ICL)

We begin by describing a framework for in-context learning of function classes, as initiated by Garg et al. (2022). In-context learning refers to the behavior of models that operate on sequences, called *prompts*, of input-output pairs $(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N, \mathbf{x}_{\mathsf{query}})$, where $y_i = h(\mathbf{x}_i)$ for some (unknown) function h and examples $\mathbf{x}_i$ and query $\mathbf{x}_{\mathsf{query}}$. The goal for an in-context learner is to use the prompt to form a prediction $\hat{y}(\mathbf{x}_{\mathsf{query}})$ for the query such that $\hat{y}(\mathbf{x}_{\mathsf{query}}) \approx h(\mathbf{x}_{\mathsf{query}})$.

From this high-level description, one can see that at a surface level, the behavior of in-context learning is no different from that of a standard learning algorithm: the learner takes as input a training dataset and returns predictions on test examples. For instance, one can view ordinary least squares as an 'in-context learner' for linear models. However, the rather unique feature of in-context learners is that these learning algorithms can be the solutions to stochastic optimization problems defined over a distribution of prompts. We formalize this notion in the following definition.

Definition 2.3.1 (Trained on in-context examples)**.** *Let $\mathcal{D}_{\mathbf{x}}$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_n, y_n) : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of $(\mathbf{x}, y)$ pairs and let*

$$\mathcal{F}_{\Theta} = \{f_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}, \boldsymbol{\theta} \in \Theta\}$$

be a class of functions parameterized by $\boldsymbol{\theta}$ in some set Θ. For $N > 0$, we say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ is trained on in-context examples of functions in $\mathcal{H}$ under loss ℓ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}})$ *if $f = f_{\boldsymbol{\theta}^*}$ where $\boldsymbol{\theta}^* \in \Theta$ satisfies*

$$\boldsymbol{\theta}^* \in \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{P=(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_N, h(\mathbf{x}_N), \mathbf{x}_{\mathsf{query}})} \left[\ell\left(f_{\boldsymbol{\theta}}(P), h(\mathbf{x}_{\mathsf{query}})\right)\right], \tag{2.1}$$

where $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathbf{x}}$ and $h \sim \mathcal{D}_{\mathcal{H}}$ are independent. We call N the length of the prompts seen during training.

As mentioned above, this definition naturally leads to a method for *learning a learning algorithm from data*: Sample independent prompts by sampling a random function $h \sim \mathcal{D}_{\mathcal{H}}$ and feature vectors $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathbf{x}}$, and then minimize the objective function appearing in (2.1) using stochastic gradient descent or other stochastic optimization algorithms. This procedure returns a model that is learned from in-context examples and can form predictions for test (query) examples given a sequence of training data. This leads to the following natural definition that quantifies how well such a model performs on in-context examples corresponding to a particular hypothesis class.

Definition 2.3.2 (In-context learning of a hypothesis class)**.** Let $\mathcal{D}_{\mathbf{x}}$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a class of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_n, y_n) : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$

be the set of finite-length sequences of $(\mathbf{x}, y)$ pairs. We say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ defined on prompts of the form $P = (\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_M, h(\mathbf{x}_M), \mathbf{x}_{\mathsf{query}})$ *in-context learns a hypothesis class* $\mathcal{H}$ *under loss* ℓ *with respect to* $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}})$ if there exists a function $M_{\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}}}(\varepsilon) : (0,1) \to \mathbb{N}$ such that for every $\varepsilon \in (0,1)$, and for every prompt P of length $M \geq M_{\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}}}(\varepsilon)$,

$$\mathbb{E}_{P=(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_M, h(\mathbf{x}_M), \mathbf{x}_{\mathsf{query}})} \left[\ell\Big(f(P), h\left(\mathbf{x}_{\mathsf{query}}\right) \Big) \right] \leq \varepsilon, \tag{2.2}$$

where the expectation is over the randomness in $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathbf{x}}$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

Note that in order for a model to in-context learn a hypothesis class, it must be expressive enough to achieve arbitrarily small error when sampling a random prompt whose labels are governed by some hypothesis h.

With these two definitions in hand, we can formulate the following questions: suppose a function class $\mathcal{F}_\Theta$ is given and $\mathcal{D}_{\mathcal{H}}$ corresponds to random instances of hypotheses in a hypothesis class $\mathcal{H}$. Can a model from $\mathcal{F}_\Theta$ that is trained on in-context examples of functions in $\mathcal{H}$ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}})$ in-context learn the hypothesis class $\mathcal{H}$ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}})$? How large must the training prompts be in order for this to occur? Do standard gradient-based optimization algorithms suffice for training the model from in-context examples? How many in-context examples $M_{\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}}}(\varepsilon)$ are needed to achieve error ε? In the remaining sections, we shall answer these questions for the case of one-layer transformers with linear self-attention modules when the hypothesis class is linear models, the loss of interest is the squared loss, and the marginals are (possibly anisotropic) Gaussian marginals.

Linear self-attention networks

Before describing the particular transformer models we analyze in this chapter, we first recall the definition of the softmax-based single-head self-attention module (Vaswani et al., 2017). Let $\mathbf{E} \in \mathbb{R}^{d_e \times d_N}$ be an embedding matrix formed using a prompt $(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N, \mathbf{x}_{\mathsf{query}})$ of length N. The user has the freedom to determine how this embedding matrix is formed from the prompt. One natural way to form $\mathbf{E}$ is to stack $(\mathbf{x}_i, y_i)^\top \in \mathbb{R}^{d+1}$ as the first N columns of $\mathbf{E}$ and to let the final column be $(\mathbf{x}_{\mathsf{query}}, 0)^\top$; if $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we would then have $d_e = d + 1$ and $d_N = N + 1$. Let $\boldsymbol{W}^K, \boldsymbol{W}^Q \in \mathbb{R}^{d_k \times d_e}$ and $\boldsymbol{W}^V \in \mathbb{R}^{d_v \times d_e}$ be the key, query, and value weight matrices, $\boldsymbol{W}^P \in \mathbb{R}^{d_e \times d_v}$ the projection matrix, and $\rho > 0$ a normalization factor. The softmax self-attention module takes as input an embedding matrix $\mathbf{E}$ of width d_N and outputs a matrix of the same size,

$$f_{\mathsf{Attn}}(\mathbf{E}; \boldsymbol{W}^K, \boldsymbol{W}^Q, \boldsymbol{W}^V, \boldsymbol{W}^P) = \mathbf{E} + \boldsymbol{W}^P \boldsymbol{W}^V \mathbf{E} \cdot \mathrm{softmax}\left(\frac{(\boldsymbol{W}^K \mathbf{E})^\top \boldsymbol{W}^Q \mathbf{E}}{\rho} \right),$$

where softmax is applied column-wise and, given a vector input of v, the i-th entry of $\mathrm{softmax}(v)$ is given by $\exp(v_i) / \sum_s \exp(v_s)$. The $d_N \times d_N$ matrix appearing inside the softmax

is referred to as the *self-attention matrix*. Note that f_{Attn} can take as its input a sequence of arbitrary length.

In this chapter, we consider a simplified version of the single-layer self-attention module, which is more amenable to theoretical analysis and yet is still capable of in-context learning linear models. In particular, we consider a single-layer linear self-attention (LSA) model, which is a modified version of f_{Attn} where we remove the softmax nonlinearity, merge the projection and value matrices into a single matrix $\boldsymbol{W}^{PV} \in \mathbb{R}^{d_e \times d_e}$, and merge the query and key matrices into a single matrix $\boldsymbol{W}^{KQ} \in \mathbb{R}^{d_e \times d_e}$. We concatenate these matrices into $\boldsymbol{\theta} = (\boldsymbol{W}^{KQ}, \boldsymbol{W}^{PV})$ and denote

$$f_{\mathsf{LSA}}(\mathbf{E}; \boldsymbol{\theta}) = \mathbf{E} + \boldsymbol{W}^{PV}\mathbf{E} \cdot \frac{\mathbf{E}^\top \boldsymbol{W}^{KQ}\mathbf{E}}{\rho}. \tag{2.3}$$

We note that recent theoretical works on understanding transformers looked at identical models (Von Oswald et al., 2023; Li et al., 2023b; Ahn et al., 2023c). It is noteworthy that recent empirical work has shown that state-of-the-art trained vision transformers with standard softmax-based attention modules are such that $(\boldsymbol{W}^K)^\top \boldsymbol{W}^Q$ and $\boldsymbol{W}^P \boldsymbol{W}^V$ are nearly multiples of the identity matrix (Trockman and Kolter, 2023), which can be represented under the parameterization we consider.

The user has the flexibility to determine the method for constructing the embedding matrix from a prompt $P = (\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N, \mathbf{x}_{\mathsf{query}})$. In this chapter, for a prompt of length N, we shall use the following embedding, which stacks $(\mathbf{x}_i, y_i)^\top \in \mathbb{R}^{d+1}$ into the first N columns with $(\mathbf{x}_{\mathsf{query}}, 0)^\top \in \mathbb{R}^{d+1}$ as the last column:

$$\mathbf{E} = \mathbf{E}(P) = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{\mathsf{query}} \\ y_1 & y_2 & \cdots & y_N & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \tag{2.4}$$

We take the normalization factor ρ to be the width of the embedding matrix $\mathbf{E}$ minus one, i.e., $\rho = d_N - 1$, since each element in $\mathbf{E} \cdot \mathbf{E}^\top$ is an inner product of two vectors of length d_N. Under the above token embedding, we take $\rho = N$. We note that there are alternative ways to form the embedding matrix with this data, e.g. by padding all inputs and labels into vectors of equal length and arranging them into a matrix (Akyürek et al., 2022), or by stacking columns that are linear transformations of the concatenation $(\mathbf{x}_i, y_i)$ (Garg et al., 2022), although the dynamics of in-context learning will differ under alternative parameterizations.

The network's prediction for the token $\mathbf{x}_{\mathsf{query}}$ will be the bottom-right entry of the matrix output by f_{LSA}, namely,

$$\widehat{y}_{\mathsf{query}} = \widehat{y}_{\mathsf{query}}(\mathbf{E}; \boldsymbol{\theta}) = [f_{\mathsf{LSA}}(\mathbf{E}; \boldsymbol{\theta})]_{(d+1),(N+1)}.$$

Hereafter, we may occasionally suppress dependence on $\boldsymbol{\theta}$ and write $\widehat{y}_{\mathsf{query}}(\mathbf{E}; \boldsymbol{\theta})$ as $\widehat{y}_{\mathsf{query}}$. Since the prediction takes only the right-bottom entry of the token matrix output by the LSA layer, actually only part of $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$ affect the prediction. To see how, let us denote

$$\boldsymbol{W}^{PV} = \begin{pmatrix} \boldsymbol{W}_{11}^{PV} & \boldsymbol{w}_{12}^{PV} \\ (\boldsymbol{w}_{21}^{PV})^\top & \boldsymbol{w}_{22}^{PV} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \quad \boldsymbol{W}^{KQ} = \begin{pmatrix} \boldsymbol{W}_{11}^{KQ} & \boldsymbol{w}_{12}^{KQ} \\ (\boldsymbol{w}_{21}^{KQ})^\top & \boldsymbol{w}_{22}^{KQ} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \tag{2.5}$$

where $\boldsymbol{W}_{11}^{PV} \in \mathbb{R}^{d\times d}; \boldsymbol{w}_{12}^{PV}, \boldsymbol{w}_{21}^{PV} \in \mathbb{R}^d; \boldsymbol{w}_{22}^{PV} \in \mathbb{R}$; and $\boldsymbol{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}; \boldsymbol{w}_{12}^{KQ}, \boldsymbol{w}_{21}^{KQ} \in \mathbb{R}^d; \boldsymbol{w}_{22}^{KQ} \in \mathbb{R}$. Then, the prediction $\widehat{y}_{\mathsf{query}}$ is

$$\widehat{y}_{\mathsf{query}} = \begin{pmatrix} (\boldsymbol{w}_{21}^{PV})^\top & \boldsymbol{w}_{22}^{PV} \end{pmatrix} \cdot \left(\frac{\mathbf{E}\mathbf{E}^\top}{N}\right) \begin{pmatrix} \boldsymbol{W}_{11}^{KQ} \\ (\boldsymbol{w}_{21}^{KQ})^\top \end{pmatrix} \mathbf{x}_{\mathsf{query}}, \tag{2.6}$$

since only the last row of $\boldsymbol{W}^{PV}$ and the first d columns of $\boldsymbol{W}^{KQ}$ affects the prediction, which means we can simply take all other entries zero in the following sections.

Training procedure

In this chapter, we will consider the task of in-context learning linear predictors. We will assume training prompts are sampled as follows. Let $\boldsymbol{\Lambda}$ be a positive definite covariance matrix. Each training prompt, indexed by $\tau \in \mathbb{N}$, takes the form

$$P_\tau = (\mathbf{x}_{\tau,1}, \boldsymbol{H}_\tau(\mathbf{x}_{\tau_1}), \ldots, \mathbf{x}_{\tau,N}, \boldsymbol{H}_\tau(\mathbf{x}_{\tau,N}), \mathbf{x}_{\tau,\mathsf{query}}),$$

where task weights $\boldsymbol{w}_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, inputs $\mathbf{x}_{\tau,i}, \mathbf{x}_{\tau,\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$, and labels $h_\tau(\mathbf{x}) = \langle \boldsymbol{w}_\tau, \mathbf{x}\rangle$.

Each prompt corresponds to an embedding matrix $\mathbf{E}_\tau$, formed using the transformation (2.4):

$$\mathbf{E}_\tau := \begin{pmatrix} \mathbf{x}_{\tau,1} & \mathbf{x}_{\tau,2} & \cdots & \mathbf{x}_{\tau,N} & \mathbf{x}_{\tau,\mathsf{query}} \\ \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,1}\rangle & \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,2}\rangle & \cdots & \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,N}\rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}.$$

We denote the prediction of the LSA model on the query label in the task τ as $\widehat{y}_{\tau,\mathsf{query}}$, which is the bottom-right element of $f_{\mathsf{LSA}}(\mathbf{E}_\tau)$, where f_{LSA} is the linear self-attention model defined in (2.3). The empirical risk over B independent prompts is defined as

$$\widehat{L}(\boldsymbol{\theta}) = \frac{1}{2B}\sum_{\tau=1}^{B}\left(\widehat{y}_{\tau,\mathsf{query}} - \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,\mathsf{query}}\rangle\right)^2. \tag{2.7}$$

We shall consider the behavior of gradient flow-trained networks over the population loss induced by the limit of infinite training tasks/prompts $B \to \infty$:

$$L(\boldsymbol{\theta}) = \lim_{B\to\infty} \widehat{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{w}_\tau, \mathbf{x}_{\tau,1}, \cdots, \mathbf{x}_{\tau,N}, \mathbf{x}_{\tau,\mathsf{query}}}\left[(\widehat{y}_{\tau,\mathsf{query}} - \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,\mathsf{query}}\rangle)^2\right] \tag{2.8}$$

Above, the expectation is taken w.r.t. the covariates $\{\mathbf{x}_{\tau,i}\}_{i=1}^N \cup \{\mathbf{x}_{\mathsf{query}}\}$ in the prompt and the weight vector $\boldsymbol{w}_\tau$, i.e. over $\mathbf{x}_{\tau,i}, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$ and $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$. Gradient flow captures the behavior of gradient descent with infinitesimal step size and has dynamics given by the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta} = -\nabla L(\boldsymbol{\theta}). \tag{2.9}$$

We will consider gradient flow with an initialization that satisfies the following.

Assumption 2.3.3 (Initialization)**.** Let $\sigma > 0$ be a parameter, and let $\boldsymbol{\Theta} \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\|_F = 1$ and $\boldsymbol{\Theta}\boldsymbol{\Lambda} \neq \mathbf{0}_{d\times d}$. We assume

$$\boldsymbol{W}^{PV}(0) = \sigma \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 1 \end{pmatrix}, \quad \boldsymbol{W}^{KQ}(0) = \sigma \begin{pmatrix} \boldsymbol{\Theta}\boldsymbol{\Theta}^\top & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{pmatrix}. \tag{2.10}$$

This initialization is satisfied for a particular class of random initialization schemes: if $\mathbf{M}$ has i.i.d. entries from a continuous distribution, then by setting $\Theta\Theta^\top = \mathbf{M}\mathbf{M}^\top/\|\mathbf{M}\mathbf{M}^\top\|_F$, the assumption is satisfied almost surely. The reason we use this particular initialization scheme will be made more clear in Section 2.5 when we describe the proof, but at a high-level this is due to the fact that the predictions (2.6) can be viewed as the output of a two-layer linear network, and initializations satisfying Assumption 2.3.3 allow for the layers to be 'balanced' throughout the gradient flow trajectory. Random initializations that induce this balancedness condition have been utilized in a number of theoretical works on deep linear networks (Du et al., 2018; Arora et al., 2018; Arora et al., 2019; Azulay et al., 2021). We leave the question of convergence under alternative random initialization schemes for future work.

2.4 Main Results

In this section, we present the main results of this chapter. First, in Section 2.4, we prove the gradient flow on the population loss will converge to a specific global optimum. We characterize the prediction error of the trained transformer at this global minimum when given a prompt from a new prediction task. Our characterization allows for the possibility that this new prompt comes from a nonlinear prediction task. We then instantiate our results for well-specified linear regression prompts and characterize the number of samples needed to achieve small prediction error, showing that transformers can in-context learn linear models when trained on in-context examples of linear models.

Next, in Section 2.4, we analyze the behavior of the trained transformer under a variety of distribution shifts. We show the transformer is robust to a number of distribution shifts, including task shift (when the labels in the prompt are not deterministic linear functions of their input) and query shift (when the query example $\mathbf{x}_{\mathsf{query}}$ has a possibly different distribution than the test prompt). On the other hand, we show that the transformer suffers from covariate distribution shifts, i.e. when the training prompt covariate distribution differs from the test prompt covariate distribution.

Finally, motivated by the failure of the trained transformer under covariate distribution shift, we consider in Section 2.4 the setting of training on in-context examples with varying covariate distributions across prompts. We prove that transformers with a single linear self-attention layer trained by gradient flow converge to a global minimum of the population objective, but that the trained transformer still fails to perform well on new prompts. We complement our proof in the linear self-attention case with experiments on large, nonlinear transformer architectures which we show are more robust under covariate shifts.

Convergence of gradient flow and prediction error for new tasks

First, we prove that under suitable initialization, gradient flow will converge to a global optimum.

Theorem 2.4.1 (Convergence and limits)**.** *Consider gradient flow of a linear self-attention network f_{LSA} defined in (2.3) over the population loss (2.8). Suppose the initialization satisfies Assumption 2.3.3 with initialization scale $\sigma > 0$ satisfying $\sigma^2 \|\mathbf{\Gamma}\|_{op}\sqrt{d} < 2$ where we have defined*

$$\mathbf{\Gamma} := \left(1 + \frac{1}{N}\right)\mathbf{\Lambda} + \frac{1}{N}\operatorname{tr}(\mathbf{\Lambda})\mathbf{I}_d \in \mathbb{R}^{d\times d}.$$

Then gradient flow converges to a global minimum of the population loss (2.8). Moreover, $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$ converge to $\boldsymbol{W}^{PV}_*$ and $\boldsymbol{W}^{KQ}_*$ respectively, where

$$\boldsymbol{W}^{KQ}_* = \left[\operatorname{tr}\left(\mathbf{\Gamma}^{-2}\right)\right]^{-\frac{1}{4}} \begin{pmatrix} \mathbf{\Gamma}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{pmatrix}, \qquad \boldsymbol{W}^{PV}_* = \left[\operatorname{tr}\left(\mathbf{\Gamma}^{-2}\right)\right]^{\frac{1}{4}} \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 1 \end{pmatrix}. \tag{2.11}$$

The full proof of this theorem appears in Appendix 2.7. We note that if we restrict our setting to $\mathbf{\Lambda} = \mathbf{I}_d$, then the limiting solution described found by gradient flow is quite similar to the construction of Von Oswald et al. (2023). Since the prediction of the transformer is the same if we multiply $\boldsymbol{W}^{PV}$ by a constant $c \neq 0$ and simultaneously multiply $\boldsymbol{W}^{KQ}$ by c^{-1}, the only difference (up to scaling) is that the top-left entry of their $\boldsymbol{W}^{KQ}$ matrix is $\mathbf{I}_d$ rather than the $(1 + (d+1)/N)^{-1}\mathbf{I}_d$ that we find for the case $\mathbf{\Lambda} = \mathbf{I}_d$.

Next, we would like to characterize the prediction error of the trained network described above when the network is given a new prompt. Let us consider a prompt of the form $(\boldsymbol{x}_1, \langle \boldsymbol{w}, \boldsymbol{x}_1\rangle, \ldots, \boldsymbol{x}_M, \langle \boldsymbol{w}, \boldsymbol{x}_M\rangle, \mathbf{x}_{\mathsf{query}})$ where $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$. A simple calculation shows that the prediction $\hat{y}_{\mathsf{query}}$ at the global optimum with parameters $\boldsymbol{W}^{KQ}_*$ and $\boldsymbol{W}^{PV}_*$ is given by

$$\begin{aligned} \widehat{y}_{\mathsf{query}} &= \begin{pmatrix} \mathbf{0}_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i\boldsymbol{x}_i^\top + \frac{1}{M}\boldsymbol{x}_{\mathsf{query}}\boldsymbol{x}_{\mathsf{query}}^\top & \frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{w} \\ \frac{1}{M}\sum_{i=1}^M \boldsymbol{w}^\top \boldsymbol{x}_i\boldsymbol{x}_i^\top & \frac{1}{M}\sum_{i=1}^M \boldsymbol{w}^\top \boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{w} \end{pmatrix} \begin{pmatrix} \mathbf{\Gamma}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{\mathsf{query}} \\ 0 \end{pmatrix} \\ &= \boldsymbol{x}_{\mathsf{query}}^\top \mathbf{\Gamma}^{-1} \left(\frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i\boldsymbol{x}_i^\top\right) w. \end{aligned} \tag{2.12}$$

When the length of prompts seen during training N is large, $\mathbf{\Gamma}^{-1} \approx \mathbf{\Lambda}^{-1}$, and when the test prompt length M is large, $\frac{1}{M}\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top \approx \mathbf{\Lambda}$, so that $\hat{y}_{\mathsf{query}} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{w}$. Thus, for sufficiently large prompt lengths, *the trained transformer indeed in-context learns the class of linear predictors.*

In fact, we can generalize the above calculation for test prompts which could take a significantly different form than the training prompts. Consider prompts that are of the form

$(\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_n, y_n, \boldsymbol{x}_{\mathsf{query}})$ where, for some joint distribution $\mathcal{D}$ over $(\mathbf{x}, y)$ pairs with marginal distribution $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, we have $(\boldsymbol{x}_i, y_i) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and $\mathbf{x}_{\mathsf{query}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ independently. Note that this allows for a label y_i to be a nonlinear function of the input $\boldsymbol{x}_i$. The prediction of the trained transformer for this prompt is then

$$\begin{aligned}\widehat{y}_{\mathsf{query}} &= \begin{pmatrix}\mathbf{0}_d^\top & 1\end{pmatrix}\begin{pmatrix}\frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i\boldsymbol{x}_i^\top + \frac{1}{M}\boldsymbol{x}_{\mathsf{query}}\boldsymbol{x}_{\mathsf{query}}^\top & \frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i y_i \\ \frac{1}{M}\sum_{i=1}^M \boldsymbol{x}_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2\end{pmatrix}\begin{pmatrix}\mathbf{\Gamma}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{x}_{\mathsf{query}} \\ 0\end{pmatrix} \\ &= \boldsymbol{x}_{\mathsf{query}}^\top \mathbf{\Gamma}^{-1}\left(\frac{1}{M}\sum_{i=1}^M y_i \boldsymbol{x}_i\right). \end{aligned} \tag{2.13}$$

Just as before, when N is large we have $\mathbf{\Gamma}^{-1} \approx \mathbf{\Lambda}^{-1}$, and so when M is large as well this implies

$$\hat{y}_{\mathsf{query}} \approx \boldsymbol{x}_{\mathsf{query}}^\top \mathbf{\Lambda}^{-1}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y\mathbf{x}] = \mathbf{x}_{\mathsf{query}}^\top\left(\operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(y - \langle \boldsymbol{w}, \mathbf{x}\rangle)^2]\right). \tag{2.14}$$

This suggests that trained transformers in-context learn the *best linear predictor* over a distribution when the test prompt consists of i.i.d. samples from a joint distribution over feature-response pairs. In the following theorem, we formalize the above and characterize the prediction error when prompts take this form.

Theorem 2.4.2. *Let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on $\mathbf{x}$ is $\mathcal{D}_{\mathbf{x}} = \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$. Assume $\mathbb{E}_{\mathcal{D}}[y], \mathbb{E}_{\mathcal{D}}[\mathbf{x}y], \mathbb{E}_{\mathcal{D}}[y^2\mathbf{x}\mathbf{x}^\top]$ exist and are finite. Assume the test prompt is of the form $P = (\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_M, y_M, \boldsymbol{x}_{\mathsf{query}})$, where $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Let f^*_{LSA} be the LSA model with parameters $\boldsymbol{W}_*^{PV}$ and $\boldsymbol{W}_*^{KQ}$ in (2.11), and $\widehat{y}_{\mathsf{query}}$ is the prediction for $\boldsymbol{x}_{\mathsf{query}}$ given the prompt. If we define*

$$\mathbf{a} := \mathbf{\Lambda}^{-1}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbf{x}y\right], \qquad \mathbf{\Sigma} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\Big(\mathbf{x}y - \mathbb{E}\left(\mathbf{x}y\right)\Big)\Big(\mathbf{x}y - \mathbb{E}\left(\mathbf{x}y\right)\Big)^\top\right], \tag{2.15}$$

then, for $\mathbf{\Gamma} = \mathbf{\Lambda} + \frac{1}{N}\mathbf{\Lambda} + \frac{1}{N}\mathrm{tr}(\mathbf{\Lambda})\mathbf{I}_d$. we have,

$$\begin{aligned}\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 &= \underbrace{\min_{\boldsymbol{w}\in\mathbb{R}^d}\mathbb{E}\left(\langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\textit{Error of best linear predictor}} \\ &\quad + \frac{1}{M}\mathrm{tr}\left[\mathbf{\Sigma}\mathbf{\Gamma}^{-2}\mathbf{\Lambda}\right] + \frac{1}{N^2}\left[\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}^3} + 2\mathrm{tr}(\mathbf{\Lambda})\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}^2} + \mathrm{tr}(\mathbf{\Lambda})^2\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}}\right], \end{aligned} \tag{2.16}$$

where the expectation is over $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

The full proof is deferred to Appendix 2.8. Let us now make a few remarks on the above theorem before considering particular instances of $\mathcal{D}$ where we may provide more explicit bounds on the prediction error.

First, this theorem shows that, provided the length of prompts seen during training (N) and the length of the test prompt (M) is large enough, a transformer trained by gradient flow from in-context examples achieves prediction error competitive with the best linear model. Next, our bound shows that the length of prompts seen during training and the length of prompts seen at test-time have different effects on the expected prediction error: ignoring dimension and covariance-dependent factors, the prediction error is at most $O(1/M + 1/N^2)$, decreasing more rapidly as a function of the training prompt length N compared to the test prompt length M.

Let us now consider when $\mathcal{D}$ corresponds to noiseless linear models, so that for some $\boldsymbol{w} \in \mathbb{R}^d$, we have $(\boldsymbol{x}, y) = (\boldsymbol{x}, \langle \boldsymbol{w}, \boldsymbol{x} \rangle)$, in which case the prediction of the trained transformer is given by (2.12). Moreover, a simple calculation shows that the $\boldsymbol{\Sigma}$ from Theorem 2.4.2 takes the form $\boldsymbol{\Sigma} = \|\boldsymbol{w}\|_{\boldsymbol{\Lambda}}^2 \boldsymbol{\Lambda} + \boldsymbol{\Lambda} \boldsymbol{w} \boldsymbol{w}^\top \boldsymbol{\Lambda}$. Hence Theorem 2.4.2 implies the prediction error for the prompt $P = (\mathbf{x}_1, \langle \boldsymbol{w}, \mathbf{x}_1 \rangle, \ldots, \mathbf{x}_M, \langle \boldsymbol{w}, \mathbf{x}_M \rangle, \mathbf{x}_{\mathsf{query}})$ is

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M, \boldsymbol{x}_{\mathsf{query}}} \left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}} \rangle \right)^2 \\
&= \frac{1}{M} \left\{ \|\boldsymbol{w}\|_{\boldsymbol{\Gamma}^{-2} \boldsymbol{\Lambda}^3}^2 + \operatorname{tr}(\boldsymbol{\Gamma}^{-2} \boldsymbol{\Lambda}^2) \|\boldsymbol{w}\|_{\boldsymbol{\Lambda}}^2 \right\} \\
&\quad + \frac{1}{N^2} \left\{ \|\boldsymbol{w}\|_{\boldsymbol{\Gamma}^{-2} \boldsymbol{\Lambda}^3}^2 + 2 \|\boldsymbol{w}\|_{\boldsymbol{\Gamma}^{-2} \boldsymbol{\Lambda}^2}^2 \operatorname{tr}(\boldsymbol{\Lambda}) + \|\boldsymbol{w}\|_{\boldsymbol{\Gamma}^{-2} \boldsymbol{\Lambda}}^2 \operatorname{tr}(\boldsymbol{\Lambda})^2 \right\} \\
&\leq \frac{d+1}{M} \|\boldsymbol{w}\|_{\boldsymbol{\Lambda}}^2 + \frac{1}{N^2} \left[\|\boldsymbol{w}\|_{\boldsymbol{\Lambda}}^2 + 2 \|\boldsymbol{w}\|_2^2 \operatorname{tr}(\boldsymbol{\Lambda}) + \|\boldsymbol{w}\|_{\boldsymbol{\Lambda}^{-1}}^2 \operatorname{tr}(\boldsymbol{\Lambda})^2 \right].
\end{aligned}
$$

The inequality above uses that $\boldsymbol{\Gamma} \succ \boldsymbol{\Lambda}$. Finally, if we assume that $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and denote κ as the condition number of $\boldsymbol{\Lambda}$, then by taking expectations over $\boldsymbol{w}$ we get the following:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M, \boldsymbol{x}_{\mathsf{query}}, \boldsymbol{w}} \left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}} \rangle \right)^2 \\
&\leq \frac{(d+1)\operatorname{tr}(\boldsymbol{\Lambda})}{M} + \frac{1}{N^2} \left[\operatorname{tr}(\boldsymbol{\Lambda}) + 2d \operatorname{tr}(\boldsymbol{\Lambda}) + \operatorname{tr}(\boldsymbol{\Lambda}^{-1}) \operatorname{tr}(\boldsymbol{\Lambda})^2 \right] \\
&\leq \frac{(d+1)\operatorname{tr}(\boldsymbol{\Lambda})}{M} + \frac{(1 + 2d + d^2 \kappa) \operatorname{tr}(\boldsymbol{\Lambda})}{N^2},
\end{aligned}
$$

From the upper bound above, we can see the rate w.r.t M and N are still at most $O(1/M)$ and $O(1/N^2)$ respectively. Moreover, the generalization risk also scales with dimension d, $\operatorname{tr}(\boldsymbol{\Lambda})$ and the condition number κ. This suggests that for in-context examples involving covariates of greater variance, or a more ill-conditioned covariance matrix, the generalization risk will be higher for the same lengths of training and testing prompts. Putting the above together with Theorem 2.4.2, Definition 2.3.1 and Definition 2.3.2, we get the following corollary.

Corollary 2.4.3. *The transformer f_{LSA} trained on length-N prompts of in-context examples of functions in $\{\mathbf{x} \mapsto \langle \boldsymbol{w}, \mathbf{x} \rangle\}$ w.r.t. $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $\mathcal{D}_{\mathbf{x}} = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$ by gradient flow on the population loss (2.8) for initializations satisfying Assumption 2.3.3 converges to the model $f_{\mathsf{LSA}}(\cdot\,; \boldsymbol{W}_*^{KQ}, \boldsymbol{W}_*^{PV})$. This model takes a prompt $P = (\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_M, y_M, \boldsymbol{x}_{\mathsf{query}})$ and*

returns a prediction $\hat{y}_{\mathsf{query}}$ *for* $\mathbf{x}_{\mathsf{query}}$ *given by*

$$\hat{y}_{\mathsf{query}} = [f_{\mathsf{LSA}}(P; \boldsymbol{W}_*^{KQ}, \boldsymbol{W}_*^{PV})]_{d+1,M+1} = \boldsymbol{x}_{\mathsf{query}}^\top \left(\boldsymbol{\Lambda} + \frac{1}{N}\boldsymbol{\Lambda} + \frac{\mathrm{tr}(\boldsymbol{\Lambda})}{N}\mathbf{I}_d\right)^{-1} \left(\frac{1}{M}\sum_{i=1}^M y_i \boldsymbol{x}_i\right).$$

This model in-context learns the class of linear models $\{x \mapsto \langle \boldsymbol{w}, \mathbf{x}\rangle\}$ *with respect to* $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ *and* $\mathcal{D}_{\mathbf{x}} = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$ *up to error* $\eta := (1+2d+d^2\kappa)\mathrm{tr}(\boldsymbol{\Lambda})/N^2$ *(where* κ *is the condition number of* $\boldsymbol{\Lambda}$*): provided* $M \geq (d+1)\mathrm{tr}(\boldsymbol{\Lambda})\varepsilon^{-1}$*, the model achieves prediction error at most* $\eta + \varepsilon$*.*

It is worth emphasizing that the transformer $f_{\mathsf{LSA}}(\cdot\,; \boldsymbol{W}_*^{KQ}, \boldsymbol{W}_*^{PV})$ only learns the function class up to error $\eta = O(1/N^2)$ in the sense of Definition 2.3.2. In particular, training on finite-length prompts leads to prediction error bounded away from zero.

Behavior of trained transformer under distribution shifts

Using the identity (2.13), it is straightforward to characterize the behavior of the trained transformer under a variety of distribution shifts. In this section, we shall examine a number of shifts that were first explored empirically for transformer architectures by Garg et al. (2022). Although their experiments were for transformers trained by gradient descent, we find that (in the case of linear models) many of the behaviors of the trained transformers under distribution shift are identical to those predicted by our theoretical characterizations of the performance of transformers with a single linear self-attention layer trained by gradient flow on the population.

Following Garg et al. (2022), for prompts of the form $(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_N, h(\mathbf{x}_N), \mathbf{x}_{\mathsf{query}})$, let us assume for training prompts that $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\boldsymbol{x}}^{\mathsf{train}}$ and $h \sim \mathcal{D}_{\mathcal{H}}^{\mathsf{train}}$, while for test prompts $\mathbf{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\boldsymbol{x}}^{\mathsf{test}}$, $\mathbf{x}_{\mathsf{query}} \sim \mathcal{D}_{\mathsf{query}}^{\mathsf{test}}$, and $h \sim \mathcal{D}_{\mathcal{H}}^{\mathsf{test}}$. We will consider the following distinct categories of shifts:

- Task shifts: $\mathcal{D}_{\mathcal{H}}^{\mathsf{train}} \neq \mathcal{D}_{\mathcal{H}}^{\mathsf{test}}$.
- Query shifts: $\mathcal{D}_{\mathsf{query}}^{\mathsf{test}} \neq \mathcal{D}_{\boldsymbol{x}}^{\mathsf{test}}$.
- Covariate shifts: $\mathcal{D}_{\boldsymbol{x}}^{\mathsf{train}} \neq \mathcal{D}_{\boldsymbol{x}}^{\mathsf{test}}$.

In the following, we shall fix $\mathcal{D}_{\boldsymbol{x}}^{\mathsf{train}} = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$ and vary the other distributions. Recall from (2.13) that the prediction for a test prompt $(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N, \mathbf{x}_{\mathsf{query}})$ is given by (for N large),

$$\hat{y}_{\mathsf{query}} = \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Gamma}^{-1} \left(\frac{1}{M}\sum_{i=1}^M y_i \mathbf{x}_i\right) \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M}\sum_{i=1}^M y_i \mathbf{x}_i\right). \tag{2.17}$$

Task shifts. These shifts are easily tolerated by the trained transformer. As Theorem 2.4.2 shows, the trained transformer is competitive with the best linear model provided the prompt length during training and at test time is large enough. In particular, even if the prompt is such that the labels y_i are not given by $\langle \boldsymbol{w}, \mathbf{x}_i \rangle$ for some $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, the trained transformer will compute a prediction which has error competitive with the best linear model that fits the test prompt.

For example, consider a prompt corresponding to a noisy linear model, so that the prompt consists of a sequence of $(\mathbf{x}_i, y_i)$ pairs where $y_i = \langle \boldsymbol{w}, \mathbf{x}_i \rangle + \varepsilon_i$ for some arbitrary vector $\boldsymbol{w} \in \mathbb{R}^d$ and independent sub-Gaussian noise ε_i. Then from (2.17), the prediction of the transformer on query examples is

$$\hat{y}_{\mathsf{query}} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M} \sum_{i=1}^M y_i \mathbf{x}_i \right) = \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M} \sum_{i=1}^M \mathbf{x}_i \mathbf{x}_i^\top \right) \boldsymbol{w} + \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M} \sum_{i=1}^M \varepsilon_i \mathbf{x}_i \right).$$

Since ε_i is mean zero and independent of $\mathbf{x}_i$, this is approximately $\mathbf{x}_{\mathsf{query}}^\top \boldsymbol{w}$ when M is large. And note that this calculation holds for an *arbitrary* vector $\boldsymbol{w}$, not just those which are sampled from an isotropic Gaussian or those with a particular norm. This behavior coincides with that of the trained transformers observed by Garg et al. (2022).

Query shifts. Continuing from (2.17), since $y_i = \langle \boldsymbol{w}, \mathbf{x}_i \rangle$,

$$\hat{y}_{\mathsf{query}} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M} \sum_{i=1}^M \mathbf{x}_i \mathbf{x}_i^\top \right) \boldsymbol{w}.$$

From this, we see that whether query shifts can be tolerated hinges upon the distribution of the $\mathbf{x}_i$'s. Since $\mathcal{D}_{\boldsymbol{x}}^{\mathsf{train}} = \mathcal{D}_{\boldsymbol{x}}^{\mathsf{test}}$, if M is large then

$$\hat{y}_{\mathsf{query}} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{\Lambda} \boldsymbol{w} = \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{w}. \tag{2.18}$$

Thus, very general shifts in the query distribution can be tolerated. On the other hand, very different behavior can be expected if M is not large and the query example depends on the training data. For example, if the query example is orthogonal to the subspace spanned by the $\mathbf{x}_i$'s, the prediction will be zero, as was observed with transformer architectures by Garg et al. (2022).

Covariate shifts. In contrast to task and query shifts, covariate shifts cannot be fully tolerated in the transformer. This can be easily seen due to the identity (2.13): when $\mathcal{D}_{\boldsymbol{x}}^{\mathsf{train}} \neq \mathcal{D}_{\boldsymbol{x}}^{\mathsf{test}}$, then the approximation in (2.18) does not hold as $\frac{1}{M} \sum_{i=1}^M \mathbf{x}_i \mathbf{x}_i^\top$ will not cancel $\boldsymbol{\Gamma}^{-1}$ when M and N are large. For instance, if we consider test prompts where the covariates are scaled by a constant $c \neq 1$, then

$$\hat{y}_{\mathsf{query}} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} \left(\frac{1}{M} \sum_{i=1}^M \mathbf{x}_i \mathbf{x}_i^\top \right) \boldsymbol{w} \approx \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{\Lambda}^{-1} c^2 \boldsymbol{\Lambda} \boldsymbol{w} = c^2 \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{w} \neq \mathbf{x}_{\mathsf{query}}^\top \boldsymbol{w}.$$

This failure mode of the trained transformer with linear self-attention was also observed in the trained transformer architectures by Garg et al. (2022). This suggests that although the predictions of the transformer may look similar to those of ordinary least squares in some settings, the algorithm implemented by the transformer is not the same since ordinary least squares is robust to scaling of the features by a constant.

It may seem surprising that a transformer trained on linear regression tasks fails in settings where ordinary least squares performs well. However, both the linear self-attention transformer we consider and the transformers considered by Garg et al. (2022) were trained on instances of linear regression when the covariate distribution $\mathcal{D}_{\mathbf{x}}$ over the features was fixed across instances. This leads to the natural question of what happens if the transformers instead are trained on prompts where the covariate distribution varies across instances, which we explore in the following section.

Transformers trained on prompts with random covariate distributions

In this section, we will consider a variant of training on in-context examples (in the sense of Definition 2.3.1) where the distribution $\mathcal{D}_{\mathbf{x}}$ is itself sampled randomly from a distribution, and training prompts are of the form $(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_N, h(\mathbf{x}_N), \mathbf{x}_{\mathsf{query}})$ where $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathbf{x}}$ and $h \sim \mathcal{D}_{\mathcal{H}}$. More formally, we can generalize Definition 2.3.1 as follows.

Definition 2.4.4 (In-context training with random covariate distributions)**.** Let Δ be a distribution over distributions $\mathcal{D}_{\mathbf{x}}$ defined on an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_n, y_n) : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs and let

$$\mathcal{F}_\Theta = \{f_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}, \boldsymbol{\theta} \in \Theta\}$$

be a class of functions parameterized by some set Θ. We say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ is *trained on in-context examples of functions in $\mathcal{H}$ under loss ℓ w.r.t. $\mathcal{D}_{\mathcal{H}}$ and distribution over covariate distributions* Δ if $f = f_{\boldsymbol{\theta}^*}$ where $\boldsymbol{\theta}^* \in \Theta$ satisfies

$$\boldsymbol{\theta}^* \in \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{P=(\mathbf{x}_1, h(\mathbf{x}_1), \ldots, \mathbf{x}_N, h(\mathbf{x}_N), \mathbf{x}_{\mathsf{query}})} \left[\ell \left(f_{\boldsymbol{\theta}}(P), h(\mathbf{x}_{\mathsf{query}}) \right) \right], \tag{2.19}$$

where $\mathcal{D}_{\mathbf{x}} \sim \Delta$, $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathbf{x}}$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

We recover the previous definition of training on in-context examples by taking Δ to be concentrated on a singleton, $\operatorname{supp}(\Delta) = \{\mathcal{D}_{\mathbf{x}}\}$. The natural question is then, if a model f is trained on in-context examples from a function class $\mathcal{H}$ w.r.t. $\mathcal{D}_{\mathcal{H}}$ and a *distribution* Δ over covariate distributions, and if one then samples some covariate distribution $\mathcal{D}_{\mathbf{x}} \sim \Delta$, does f in-context learn $\mathcal{H}$ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathbf{x}})$ for that $\mathcal{D}_{\mathbf{x}}$ (cf. Definition 2.3.2)? Since $\mathcal{D}_{\mathbf{x}}$ is random, we can hope that this may hold in expectation or with high probability over the sampling of

the covariate distribution. In the remainder of this section, we will explore this question for transformers with a linear self-attention layer trained by gradient flow on the population loss.

We shall again consider the case where the covariates have Gaussian marginals, $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, but we shall now assume that within each prompt we first sample a random covariance matrix $\mathbf{\Lambda}$. For simplicity, we will restrict our attention to the case where $\mathbf{\Lambda}$ is diagonal. More formally, we shall assume training prompts are sampled as follows. For each independent task indexed by $\tau \in [B]$, we first sample $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. Then, for each task τ and coordinate $i \in [d]$, we sample $\mathbf{\Lambda}_{\tau,i}$ independently such that the distribution of each $\mathbf{\Lambda}_{\tau,i}$ is fixed and has finite third moments and is strictly positive almost surely. We then form a diagonal matrix

$$\mathbf{\Lambda}_\tau = \mathrm{diag}(\mathbf{\Lambda}_{\tau,1}, \ldots, \mathbf{\Lambda}_{\tau,d}).$$

Thus the diagonal entries of $\mathbf{\Lambda}_\tau$ are independent but could have different distributions, and $\mathbf{\Lambda}_\tau$ is identically distributed for $\tau = 1, \ldots, B$. Then, conditional on $\mathbf{\Lambda}_\tau$, we sample independent and identically distributed $\mathbf{x}_{\tau,1}, \ldots, \mathbf{x}_{\tau,N}, \mathbf{x}_{\tau,\mathsf{query}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_\tau)$. A training prompt is then given by $P_\tau = (\mathbf{x}_{\tau,1}, \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,1}\rangle, \ldots, \mathbf{x}_{\tau,N}, \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,N}\rangle, \mathbf{x}_{\tau,\mathsf{query}})$ Notice that here, $\mathbf{x}_{\tau,i}, \mathbf{x}_{\tau,\mathsf{query}}$ are conditionally independent given the covariance matrix $\mathbf{\Lambda}_\tau$, but not independent in general. We consider the same token embedding matrix as (2.4) and linear self-attention network, which forms the prediction $\hat{y}_{\mathsf{query},\tau}$ as in (2.6). The empirical risk is the same as before (see (2.7)), and as in (2.8), we then take $B \to \infty$ and consider the gradient flow on the population loss. The population loss now includes an expectation over the distribution of the covariance matrices in addition to the task weight $\boldsymbol{w}_\tau$ and covariate distributions, and is given by

$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{w}_\tau, \mathbf{\Lambda}_\tau, \mathbf{x}_{\tau,1}, \cdots, \mathbf{x}_{\tau,N}, \mathbf{x}_{\tau,\mathsf{query}}} \left[(\widehat{y}_{\tau,\mathsf{query}} - \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,\mathsf{query}} \rangle)^2 \right]. \tag{2.20}$$

In the main result for this section, we show that gradient flow with a suitable initialization converges to a global minimum, and we characterize the limiting solution. The proof will be deferred to Appendix 2.9.

Theorem 2.4.5 (Global convergence with random covariance)**.** *Consider gradient flow of the linear self-attention network f_{LSA} defined in (2.3) over the population loss (2.20), where $\mathbf{\Lambda}_\tau$ are diagonal with independent diagonal entries which are strictly positive a.s. and have finite third moments. Suppose the initialization satisfies Assumption 2.3.3, $\|\mathbb{E}\mathbf{\Lambda}_\tau \mathbf{\Theta}\|_F \neq 0$, with initialization scale $\sigma > 0$ satisfying*

$$\sigma^2 < \frac{2\,\|\mathbb{E}\mathbf{\Lambda}_\tau \mathbf{\Theta}\|_F^2}{\sqrt{d}\left[\mathbb{E}\,\|\mathbf{\Gamma}_\tau\|_{op}\,\|\mathbf{\Lambda}_\tau\|_F^2\right]}. \tag{2.21}$$

Then gradient flow converges to a global minimum of the population loss (2.20). Moreover,

$\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$ converge to $\boldsymbol{W}_*^{PV}$ and $\boldsymbol{W}_*^{KQ}$ respectively, where

$$\begin{aligned}\boldsymbol{W}_*^{KQ} &= \left\|\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\mathbb{E}\left[\boldsymbol{\Lambda}_\tau^2\right]\right\|_F^{-\frac{1}{2}}\cdot\begin{pmatrix}\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right] & \mathbf{0}_d\\ \mathbf{0}_d^\top & 0\end{pmatrix},\\ \boldsymbol{W}_*^{PV} &= \left\|\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\mathbb{E}\left[\boldsymbol{\Lambda}_\tau^2\right]\right\|_F^{\frac{1}{2}}\cdot\begin{pmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_d\\ \mathbf{0}_d^\top & 1\end{pmatrix},\end{aligned} \tag{2.22}$$

where $\boldsymbol{\Gamma}_\tau = \frac{N+1}{N}\boldsymbol{\Lambda}_\tau + \frac{1}{N}\mathrm{tr}(\boldsymbol{\Lambda}_\tau)\mathbf{I}_d \in \mathbb{R}^{d\times d}$ and the expectations above are over the distribution of $\boldsymbol{\Lambda}_\tau$.

From this result, we can see why the trained transformer fails in the random covariance case. Suppose we have a new prompt corresponding to a weight matrix $\boldsymbol{w} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Lambda}_{\mathsf{new}}$, sampled from the same distribution as the covariance matrices for training prompts, so that conditionally on $\boldsymbol{\Lambda}_{\mathsf{new}}$ we have $\mathbf{x}_i, \mathbf{x}_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_{\mathsf{new}})$. The ground-truth labels are given by $y_i = \langle \boldsymbol{w}, \mathbf{x}_i\rangle, i \in [M]$ and $y_{\mathsf{query}} = \langle \boldsymbol{w}, \mathbf{x}_{\mathsf{query}}\rangle$. At convergence, the prediction by the trained transformer on the new task will be

$$\begin{aligned}&\widehat{y}_{\mathsf{query}}\\ &= \begin{pmatrix}\mathbf{0}_d^\top & 1\end{pmatrix}\begin{pmatrix}\frac{1}{M}\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top + \frac{1}{M}\mathbf{x}_{\mathsf{query}}\mathbf{x}_{\mathsf{query}}^\top & \frac{1}{M}\sum_{i=1}^M \mathbf{x}_i y_i\\ \frac{1}{M}\sum_{i=1}^M \mathbf{x}_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2\end{pmatrix}\begin{pmatrix}\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\mathbb{E}\boldsymbol{\Lambda}_\tau^2 & \mathbf{0}_d\\ \mathbf{0}_d^\top & 0\end{pmatrix}\begin{pmatrix}\mathbf{x}_{\mathsf{query}}\\ 0\end{pmatrix}\\ &= \mathbf{x}_{\mathsf{query}}^\top\cdot\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right]\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\cdot\left[\frac{1}{M}\sum_{i=1}^M \mathbf{x}_i\mathbf{x}_i^\top\right]\boldsymbol{w}\\ &\to \mathbf{x}_{\mathsf{query}}^\top\cdot\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right]\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\cdot\boldsymbol{\Lambda}_{\mathsf{new}}\boldsymbol{w} \quad \text{almost surely when } M\to\infty.\end{aligned} \tag{2.23}$$

The last line comes from the strong law of large numbers. Thus, in order for the prediction on the query example to be close to the ground-truth $\mathbf{x}_{\mathsf{query}}^\top\boldsymbol{w}$, we need $\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right]\left[\mathbb{E}\boldsymbol{\Gamma}_\tau\boldsymbol{\Lambda}_\tau^2\right]^{-1}\cdot\boldsymbol{\Lambda}_{\mathsf{new}}$ to be close to the identity. When $\boldsymbol{\Lambda}_\tau \equiv \boldsymbol{\Lambda}_{\mathsf{new}}$ is deterministic, this indeed is the case as we know from Theorem 2.4.2. However, this clearly does not hold in general when $\boldsymbol{\Lambda}_\tau$ is random.

To make things concrete, let us assume for simplicity that $M, N \to \infty$ so that $\boldsymbol{\Gamma}_\tau \to \boldsymbol{\Lambda}_\tau$ and the identity (2.23) holds (conditionally on $\boldsymbol{\Lambda}_{\mathsf{new}}$). Then, taking expectation over $\boldsymbol{\Lambda}_{\mathsf{new}}$ in (2.23), we obtain

$$\mathbb{E}\left[\widehat{y}_{\mathsf{query}}\middle|\,\mathbf{x}_{\mathsf{query}}, \boldsymbol{w}\right] \to \mathbf{x}_{\mathsf{query}}^\top\cdot\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right]\left[\mathbb{E}\boldsymbol{\Lambda}_\tau^3\right]^{-1}\cdot\left[\mathbb{E}\boldsymbol{\Lambda}_\tau\right]\boldsymbol{w}.$$

If we consider the case $\boldsymbol{\Lambda}_{\tau,i} \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$, so that $\mathbb{E}[\boldsymbol{\Lambda}_\tau] = \mathbf{I}_d$, $\mathbb{E}[\boldsymbol{\Lambda}_\tau^2] = 2\mathbf{I}_d$, and $\mathbb{E}[\boldsymbol{\Lambda}_\tau^3] = 6\mathbf{I}_d$, we get

$$\mathbb{E}\widehat{y}_{\mathsf{query}} \to \frac{1}{3}\langle \boldsymbol{w}, \mathbf{x}_{\mathsf{query}}\rangle.$$

This shows that for transformers with a single linear self-attention layer, training on in-context examples with random covariate distributions does not allow for in-context learning of a hypothesis class with varying covariate distributions.

Experiments with large, nonlinear transformers. We have shown that even when trained on prompts with random covariance matrices, transformers with a single linear self-attention layer fail to in-context learn linear models with random covariance matrices. We now investigate the behavior of more complex transformer architectures that are trained on in-context examples of linear models, both in the fixed-covariance case and in the random-covariance case.

We examine the performance of transformers with a GPT2 architecture (Radford et al., 2019) that are trained on linear regression tasks with mean-zero Gaussian features with either a fixed covariance matrix or random covariance matrices. For the fixed covariance case, the covariance matrix is fixed to the identity matrix across prompts. For the random covariance case, covariates are drawn from $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, c\mathbf{\Lambda})$ where $\mathbf{\Lambda}$ is diagonal with $\mathbf{\Lambda}_i \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$ and $c > 0$ is a scaling factor. We set $c = 1$ during training and vary this value at test time. The transformer is trained using the procedure of Garg et al. (2022) (see Appendix 2.11 for more details). We consider linear models in $d = 20$ dimensions and we train on prompt lengths of $N = 40, 70, 100$ with either fixed or random covariance matrices. The performance of these trained models, when tested on new data with fixed covariance or random covariance matrices ($c = 1, 4, 9$), is represented in six curves in Figure 2.1. Using the calculation (2.23), we can compare the prediction error for the linear self-attention networks in the $M \to \infty$, $N \to \infty$ limit (the black dash line) to those of GPT2 architectures. We additionally compare these models to the ordinary least-squares solution which is optimal for this task.

From the figure, we can see that the GPT2 model trained on fixed covariance succeeds in the random covariance setting if the variance is not too large, which shows that the larger nonlinear model is able to generalize better than the model with a single linear self-attention layer. However, when the variance is large ($c = 4, 9$ for the bottom two figures), the GPT2 model trained with fixed covariance is unsuccessful. When trained on random covariance, the model performs better for test prompts from higher-variance random covariance matrices, but still fails to match least squares when the scaling is largest ($c = 9$).

Furthermore, we notice some surprising behaviors when the test prompt length exceeds the training prompt length (i.e., $M > N$): there is an evident spike in prediction error, regardless of whether training and testing were performed on fixed or random covariance, and the spike appears to decrease when evaluated on prompts with higher variance. Although we are unsure of why the spike should decrease with higher-variance prompts, the failure of large language models to generalize to larger contexts than they were trained on is a well-known problem (Dai et al., 2019; Anil et al., 2022). In our setting, we conjecture that this spike in error comes from the absolute positional encodings in the GPT2 architecture. The positional encodings are randomly-initialized and are learnable parameters but the encoding for position i is only updated if the transformer encounters a prompt which has a context of length i.

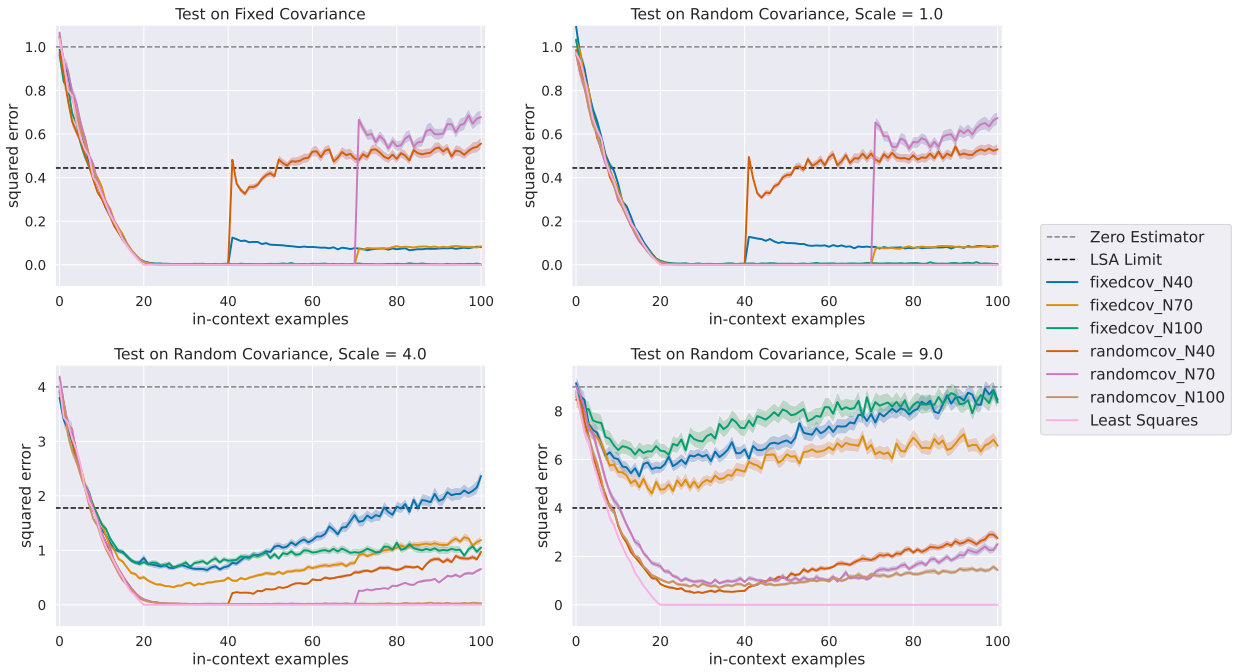

Figure 2.1: Normalized prediction error for transformers with GPT2 architectures as a function of the number of in-context test examples M when trained on in-context examples of linear models in $d = 20$ dimensions. Colored lines correspond to different training context lengths ($N \in \{40, 70, 100\}$) and different training procedures (either a fixed identity covariance matrix or random diagonal covariance matrices with each diagonal element sampled i.i.d. from the standard exponential distribution). The four figures correspond to evaluating on either fixed covariance or random covariance matrices of different scales. The gray dashed line shows the prediction error of zero estimator and the black dashed line the prediction error of LSA model when $M, N \to \infty$. The GPT2 models achieve smaller error when they are trained on random covariance matrices with larger contexts, but their prediction error spikes when evaluated on contexts larger than those they were trained on.

Thus, when evaluating on prompts of length $M > N$, the model is relying upon random positional encodings for $M - N$ samples. We note that a concurrent work has explored the performance of transformers with GPT2 architectures for in-context learning of linear models and found that removing positional encoders improves performance when evaluating on larger contexts (Panwar et al., 2023a). We leave further investigation of this behavior for future work.

2.5 Proof Ideas

In this section, we briefly outline the proof sketch of Theorem 2.4.1. The full proof of this theorem is left for Appendix 2.7.

Equivalence to a quadratic optimization problem

We recall each task τ corresponds to a weight vector $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. The prompt inputs for this task are $\mathbf{x}_{\tau,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$, which are also independent of $\boldsymbol{w}_\tau$. The corresponding labels are $y_{\tau,j} = \langle \boldsymbol{w}_\tau, \mathbf{x}_{\tau,j} \rangle$. For each task τ, we can form the prompt into a token matrix $\mathbf{E}_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ as in (2.4), with the right-bottom entry being zero.

The first key step in our proof is to recognize that the prediction $\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau; \boldsymbol{\theta})$ in the linear self-attention model can be written as the output of a quadratic function $\mathbf{u}^\top \boldsymbol{H}_\tau \mathbf{u}$ for some matrix $\boldsymbol{H}_\tau$ depending on the token embedding matrix $\mathbf{E}_\tau$ and for some vector $\mathbf{u}$ depending on $\boldsymbol{\theta} = (\boldsymbol{W}^{KQ}, \boldsymbol{W}^{PV})$. This is shown in the following lemma, the proof of which is provided in Appendix 2.7.

Lemma 2.5.1. *Let $\mathbf{E}_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ be an embedding matrix corresponding to a prompt of length N and weight $\boldsymbol{w}_\tau$. Then the prediction $\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau; \boldsymbol{\theta})$ for the query covariate can be written as the output of a quadratic function,*

$$\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau; \boldsymbol{\theta}) = \boldsymbol{u}^\top \boldsymbol{H}_\tau \boldsymbol{u},$$

where the matrix $\boldsymbol{H}_\tau$ is defined as,

$$\boldsymbol{H}_\tau = \frac{1}{2}\boldsymbol{X}_\tau \otimes \left(\frac{\mathbf{E}_\tau \mathbf{E}_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad \boldsymbol{X}_\tau = \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{x}_{\tau,\mathsf{query}} \\ (\mathbf{x}_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \tag{2.24}$$

and

$$\boldsymbol{u} = \mathrm{Vec}(\boldsymbol{U}) \in \mathbb{R}^{(d+1)^2}, \quad \boldsymbol{U} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $\mathbf{U}_{11} = \boldsymbol{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}, \mathbf{u}_{12} = \boldsymbol{w}_{21}^{PV} \in \mathbb{R}^{d\times 1}, \mathbf{u}_{21} = \boldsymbol{w}_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = \boldsymbol{w}_{22}^{PV} \in \mathbb{R}$ correspond to particular components of $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$, defined in (2.5).

This implies that we can write the original loss function (2.7) as

$$\widehat{L} = \frac{1}{2B}\sum_{\tau=1}^{B}\left(\mathbf{u}^\top \boldsymbol{H}_\tau \mathbf{u} - \boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}}\right)^2. \tag{2.25}$$

Thus, our problem is reduced to *understanding the dynamics of an optimization algorithm defined in terms of a quadratic function*. We also note that this quadratic optimization problem is an instance of a rank-one matrix factorization problem, a problem well-studied in

the deep learning theory literature (Gunasekar et al., 2017; Arora et al., 2019; Li et al., 2018; Chi et al., 2019; Belabbas, 2020; Li et al., 2020; Jin et al., 2023; Soltanolkotabi et al., 2023).

Note, however, this quadratic function is non-convex. To see this, we will show that $\boldsymbol{H}_\tau$ has negative eigenvalues. By standard properties of the Kronecker product, the eigenvalues of $\boldsymbol{H}_\tau = \frac{1}{2}\boldsymbol{X}_\tau \otimes \left(\frac{\mathbf{E}_\tau \mathbf{E}_\tau^\top}{N}\right)$ are the products of the eigenvalues of $\frac{1}{2}\boldsymbol{X}_\tau$ and the eigenvalues of $\frac{\mathbf{E}_\tau \mathbf{E}_\tau^\top}{N}$. Since $\mathbf{E}_\tau \mathbf{E}_\tau^\top$ is symmetric and positive semi-definite, all of its eigenvalues are nonnegative. Since $\mathbf{E}_\tau \mathbf{E}_\tau^\top$ is nonzero almost surely, it thus has at least one strictly positive eigenvalue. Thus, if $\boldsymbol{X}_\tau$ has any negative eigenvalues, $\boldsymbol{H}_\tau$ does as well. The characteristic polynomial of $\boldsymbol{X}_\tau$ is given by,

$$\det(\mu \mathbf{I} - \boldsymbol{X}_\tau) = \det \begin{pmatrix} \mu \mathbf{I}_d & -\mathbf{x}_{\tau,\mathsf{query}} \\ -\mathbf{x}_{\tau,\mathsf{query}}^\top & \mu \end{pmatrix} = \mu^{d-1}\left(\mu^2 - \|\mathbf{x}_{\tau,\mathsf{query}}\|_2^2\right).$$

Therefore, we know almost surely, $\boldsymbol{X}_\tau$ has one negative eigenvalue. Thus $\boldsymbol{H}_\tau$ has at least $d+1$ negative eigenvalues, and hence the quadratic form $\mathbf{u}^\top \boldsymbol{H}_\tau \mathbf{u}$ is non-convex.

Dynamical system of gradient flow

We now describe the dynamical system for the coordinates of $\boldsymbol{u}$ above. We prove the following lemma in Appendix 2.7.

Lemma 2.5.2. *Let* $\boldsymbol{u} = \mathrm{Vec}(\boldsymbol{U}) := \mathrm{Vec}\begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix}$ *as in Lemma 2.5.1. Consider gradient flow over*

$$L := \frac{1}{2}\mathbb{E}\left(\boldsymbol{u}^\top \boldsymbol{H}_\tau \boldsymbol{u} - \boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}}\right)^2 \tag{2.26}$$

with respect to $\boldsymbol{u}$ *starting from an initial value satisfying Assumption 2.3.3. Then the dynamics of* $\boldsymbol{U}$ *follows*

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t) &= -u_{-1}^2 \boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda} + u_{-1}\boldsymbol{\Lambda}^2 \\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -\operatorname{tr}\left[u_{-1}\boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top - \boldsymbol{\Lambda}^2(\mathbf{U}_{11})^\top\right], \end{aligned} \tag{2.27}$$

and $\mathbf{u}_{12}(t) = \mathbf{0}_d, \mathbf{u}_{21}(t) = \mathbf{0}_d$ *for all* $t \geq 0$, *where* $\boldsymbol{\Gamma} = \left(1 + \frac{1}{N}\right)\boldsymbol{\Lambda} + \frac{1}{N}\operatorname{tr}(\boldsymbol{\Lambda})\mathbf{I}_d \in \mathbb{R}^{d\times d}$.

We see that the dynamics are governed by a complex system of d^2+1 coupled differential equations. Moreover, basic calculus (for details, see Lemma 2.7.1) shows that these dynamics are the same as those of gradient flow on the following objective function:

$$\tilde{\ell} : \mathbb{R}^{d\times d} \times \mathbb{R} \to \mathbb{R}, \quad \tilde{\ell}(\mathbf{U}_{11}, u_{-1}) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2 \boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top - u_{-1}\boldsymbol{\Lambda}^2(\mathbf{U}_{11})^\top\right]. \tag{2.28}$$

Actually, the loss function $\tilde{\ell}$ is simply the loss function L in (2.26) plus some constants that do not depend on the parameter u. Therefore our problem is reduced to studying the dynamics of gradient flow on the above objective function.

Our next key observation is that the set of global minima for $\tilde{\ell}$ satisfies the condition $u_{-1}\mathbf{U}_{11} = \mathbf{\Gamma}^{-1}$. Thus, if we can establish global convergence of gradient flow over the above objective function $\tilde{\ell}$, then we have that $u_{-1}(t)\mathbf{U}_{11}(t) \to \mathbf{\Gamma}^{-1} \approx_{N\to\infty} \mathbf{\Lambda}^{-1}$.

Lemma 2.5.3. *For any global minimum of $\tilde{\ell}$, we have*

$$u_{-1}\mathbf{U}_{11} = \mathbf{\Gamma}^{-1}. \tag{2.29}$$

Putting this together with Lemma 2.5.2, we see that at those global minima of the population objective satisfying $\mathbf{U}_{11} = (c\mathbf{\Gamma})^{-1}$, $u_{-1} = c$ and $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$, the transformer's predictions for a new linear regression task prompt are given by

$$\widehat{y}_{\mathsf{query}}(\mathbf{E};\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} y_i \mathbf{x}_i^\top \mathbf{\Gamma}^{-1}\mathbf{x}_{\mathsf{query}} = \boldsymbol{w}^\top \left(\frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{\Gamma}^{-1}\mathbf{x}_{\mathsf{query}} \approx \boldsymbol{w}^\top \mathbf{x}_{\mathsf{query}}.$$

Thus, the only remaining task is to show global convergence when gradient flow has an initialization satisfying Assumption 2.3.3.

PL inequality and global convergence

We now show that although the optimization problem is non-convex, a Polyak-Łojasiewicz (PL) inequality holds, which implies that gradient flow converges to a global minimum. Moreover, we can exactly calculate the limiting value of $\mathbf{U}_{11}$ and u_{-1}.

Lemma 2.5.4. *Suppose the initialization of gradient flow satisfies Assumption 2.3.3 with initialization scale satisfying $\sigma^2 < \frac{2}{\sqrt{d}\|\mathbf{\Gamma}\|_{op}}$ for $\mathbf{\Gamma} = (1+\frac{1}{N})\mathbf{\Lambda} + \frac{\mathrm{tr}(\mathbf{\Lambda})}{N}\mathbf{I}_d$. If we define*

$$\mu := \frac{\sigma^2}{\sqrt{d}\,\|\mathbf{\Lambda}\|_{op}^2 \,\mathrm{tr}\left(\mathbf{\Gamma}^{-1}\mathbf{\Lambda}^{-1}\right)\mathrm{tr}\left(\mathbf{\Lambda}^{-1}\right)}\,\|\mathbf{\Lambda}\mathbf{\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\,\|\mathbf{\Gamma}\|_{op}\right] > 0, \tag{2.30}$$

then gradient flow on $\tilde{\ell}$ with respect to $\mathbf{U}_{11}$ and u_{-1} satisfies, for any $t \geq 0$,

$$\begin{aligned}\left\|\nabla\tilde{\ell}(\mathbf{U}_{11}(t), u_{-1}(t))\right\|_2^2 :=& \left\|\frac{\partial\tilde{\ell}}{\partial\mathbf{U}_{11}}\right\|_F^2 + \left|\frac{\partial\tilde{\ell}}{\partial u_{-1}}\right|^2 \\ \geq& \mu\left(\tilde{\ell}(\mathbf{U}_{11}(t), u_{-1}(t)) - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right).\end{aligned}$$

Moreover, gradient flow converges to the global minimum of $\tilde{\ell}$, and $\mathbf{U}_{11}$ and u_{-1} converge to the following,

$$\lim_{t\to\infty} u_{-1}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{\frac{1}{2}} \text{ and } \lim_{t\to\infty}\mathbf{U}_{11}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{-\frac{1}{2}}\mathbf{\Gamma}^{-1}. \tag{2.31}$$

With these observations, proving Theorem 2.4.1 becomes a direct application of Lemma 2.5.1, 2.5.2, 2.5.3, and Lemma 2.5.4. It then only requires translating $\mathbf{U}_{11}$ and u_{-1} back to the original parameterization using $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$.

2.6 Discussions and Conclusion

In this chapter, we investigated the dynamics of in-context learning of transformers with a single linear self-attention layer under gradient flow on the population loss. In particular, we analyzed the dynamics of these transformers when trained on prompts consisting of random instances of noiseless linear models over anisotropic Gaussian marginals. We showed that despite non-convexity, gradient flow from a suitable random initialization converges to a global minimum of the population objective. We characterized the prediction error of the trained transformer when given a new prompt that consists of a training dataset where the responses are a nonlinear function of the inputs. We showed how the trained transformer is naturally robust to shifts in the task and query distributions but is brittle to distribution shifts between the covariates seen during training and the covariates seen at test time, matching the empirical observations on trained transformer models of Garg et al. (2022).

There are a number of natural directions for future research. First, our results hold for gradient flow on the population loss with a particular class of random initialization schemes. It is a natural question whether similar results would hold for stochastic gradient descent with finite step sizes and for more general initializations. Further, we restricted our attention to transformers with a single linear self-attention layer. Although this model class is rich enough to allow for in-context learning of linear predictors, we are particularly interested in understanding the dynamics of in-context learning in nonlinear and deep transformers.

Finally, the framework of in-context learning introduced in prior work was restricted to the setting where the marginal distribution over the covariates ($\mathcal{D}_{\mathbf{x}}$) was fixed across prompts. This allows for guarantees akin to distribution-specific PAC learning, where the trained transformer is able to achieve small prediction error when given a test prompt consisting of linear regression data when the marginals over the covariates are fixed. However, other learning algorithms (such as ordinary least squares) are able to achieve small prediction error for prompts corresponding to well-specified linear regression tasks for very general classes of distributions over the covariates. As we showed in Section 2.4, when transformers with a single linear self-attention layer are trained on prompts where the covariate distributions are themselves sampled from a distribution, they do not succeed on test prompts with covariate distributions sampled from the same distribution. By contrast, we demonstrated with experiments that larger, nonlinear transformer architectures appear to be more successful in this setting but are still sub-optimal. Developing a better understanding of the dynamics of in-context learning when the covariate distribution varies across prompts is an intriguing direction for future research.

2.7 Appendix: Proof of Theorem 2.4.1

In this section, we prove Lemma 2.5.1, Lemma 2.5.2, Lemma 2.5.3 and Lemma 2.5.4. Theorem 2.4.1 is a natural corollary of these four lemmas when we translate u_{-1} and $\mathbf{U}_{11}$ back to $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$.

Proof of Lemma 2.5.1

For the reader's convenience, we restate the lemma below.

Lemma 2.5.1. *Let $\mathbf{E}_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ be an embedding matrix corresponding to a prompt of length N and weight $\boldsymbol{w}_\tau$. Then the prediction $\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau;\boldsymbol{\theta})$ for the query covariate can be written as the output of a quadratic function,*

$$\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau;\boldsymbol{\theta}) = \boldsymbol{u}^\top \boldsymbol{H}_\tau \boldsymbol{u},$$

where the matrix $\boldsymbol{H}_\tau$ is defined as,

$$\boldsymbol{H}_\tau = \frac{1}{2}\boldsymbol{X}_\tau \otimes \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad \boldsymbol{X}_\tau = \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{x}_{\tau,\mathsf{query}} \\ (\mathbf{x}_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \tag{2.24}$$

and

$$\boldsymbol{u} = \mathrm{Vec}(\boldsymbol{U}) \in \mathbb{R}^{(d+1)^2}, \quad \boldsymbol{U} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $\mathbf{U}_{11} = \boldsymbol{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}, \mathbf{u}_{12} = \boldsymbol{w}_{21}^{PV} \in \mathbb{R}^{d\times 1}, \mathbf{u}_{21} = \boldsymbol{w}_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = \boldsymbol{w}_{22}^{PV} \in \mathbb{R}$ correspond to particular components of $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$, defined in (2.5).

Proof. First, we decompose $\mathbf{W}_{PV}$ and W_{KQ} in the way above. From the definition, we know $\widehat{y}_{\tau,\mathsf{query}}$ is the right-bottom entry of $f_{\mathsf{LSA}}(\mathbf{E}_\tau)$, which is

$$\widehat{y}_{\tau,\mathsf{query}} = \begin{pmatrix} (\mathbf{u}_{12})^\top & u_{-1} \end{pmatrix} \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right) \begin{pmatrix} \mathbf{U}_{11} \\ (\mathbf{u}_{21})^\top \end{pmatrix} \mathbf{x}_{\tau,\mathsf{query}}.$$

We denote $\mathbf{u}_i \in \mathbb{R}^{d+1}$ as the i-th column of $\begin{pmatrix} \mathbf{U}_{11} \\ (\mathbf{u}_{21})^\top \end{pmatrix}$ and $\mathbf{x}_{\tau,\mathsf{query}}^i$ as the i-th entry of $\mathbf{x}_{\tau,\mathsf{query}}$ for $i \in [d]$. Then, we have

$$\begin{aligned}
&\widehat{y}_{\tau,\mathsf{query}} \\
=&\sum_{i=1}^d \mathbf{x}_{\tau,\mathsf{query}}^i \begin{pmatrix} (\mathbf{u}_{12})^\top & u_{-1} \end{pmatrix} \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right) u_i = \sum_{i=1}^d \mathrm{tr}\left[u_i \begin{pmatrix} (\mathbf{u}_{12})^\top & u_{-1} \end{pmatrix} \cdot \mathbf{x}_{\tau,\mathsf{query}}^i \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right)\right] \\
=&\mathrm{tr}\left[\mathrm{Vec}\left[\begin{pmatrix} \mathbf{U}_{11} \\ (\mathbf{u}_{21})^\top \end{pmatrix}\right] \begin{pmatrix} (\mathbf{u}_{12})^\top & u_{-1} \end{pmatrix} \cdot \mathbf{x}_{\tau,\mathsf{query}}^\top \otimes \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right)\right] \\
=&\frac{1}{2}\mathrm{tr}\left[\mathrm{Vec}\left[\begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix}\right] \mathrm{Vec}^\top\left[\begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix}\right]\right. \\
&\qquad\qquad \left. \cdot \begin{pmatrix} \mathbf{0}_{d(d+1)\times d(d+1)} & \mathbf{x}_{\tau,\mathsf{query}} \otimes \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right) \\ \mathbf{x}_{\tau,\mathsf{query}}^\top \otimes \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right) & 0_{(d+1)\times(d+1)} \end{pmatrix}\right] \\
=&\frac{1}{2}\mathrm{tr}\left[\boldsymbol{u}\boldsymbol{u}^\top \cdot \boldsymbol{X}_\tau \otimes \left(\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right)\right] \\
=&\left\langle \boldsymbol{H}_\tau, \boldsymbol{u}\boldsymbol{u}^\top \right\rangle.
\end{aligned}$$

Here, we use some algebraic facts about matrix vectorization, Kronecker product and trace. For reference, we refer to (Petersen and Pedersen, 2008). □

Proof of Lemma 2.5.2

For the reader's convenience, we restate the lemma below.

Lemma 2.5.2. *Let* $\boldsymbol{u} = \mathrm{Vec}(\boldsymbol{U}) := \mathrm{Vec}\begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix}$ *as in Lemma 2.5.1. Consider gradient flow over*

$$L := \frac{1}{2}\mathbb{E}\left(\boldsymbol{u}^\top \boldsymbol{H}_\tau \boldsymbol{u} - \boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}}\right)^2 \tag{2.26}$$

with respect to $\boldsymbol{u}$ *starting from an initial value satisfying Assumption 2.3.3. Then the dynamics of* $\boldsymbol{U}$ *follows*

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t) &= -u_{-1}^2 \boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda} + u_{-1}\boldsymbol{\Lambda}^2 \\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -\operatorname{tr}\left[u_{-1}\boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top - \boldsymbol{\Lambda}^2(\mathbf{U}_{11})^\top\right], \end{aligned} \tag{2.27}$$

and $\mathbf{u}_{12}(t) = \mathbf{0}_d, \mathbf{u}_{21}(t) = \mathbf{0}_d$ *for all* $t \geq 0$, *where* $\boldsymbol{\Gamma} = \left(1 + \frac{1}{N}\right)\boldsymbol{\Lambda} + \frac{1}{N}\operatorname{tr}(\boldsymbol{\Lambda})\mathbf{I}_d \in \mathbb{R}^{d\times d}$.

Proof. From the definition of L in (2.26) and the dynamics of gradient flow, we calculate the derivatives of $\boldsymbol{u}$. Here, we use the chain rule and some facts about matrix derivatives. See Lemma 2.10.1 for reference.

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -2\mathbb{E}\left(\langle \boldsymbol{H}_\tau, \boldsymbol{u}\boldsymbol{u}^\top\rangle \boldsymbol{H}_\tau\right)\boldsymbol{u} + 2\mathbb{E}\left(\boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right)\boldsymbol{u}. \tag{2.32}$$

Step One: Calculate the Second Term We first calculate the second term. From the definition of $\boldsymbol{H}_\tau$, we have

$$\mathbb{E}\left[\boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right] = \frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right) \otimes \left(\boldsymbol{w}_\tau^i \frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}\right)\right].$$

For ease of notation, we denote

$$\widehat{\boldsymbol{\Lambda}}_\tau := \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{\tau,i}\mathbf{x}_{\tau,i}^\top. \tag{2.33}$$

Then, from the definition of $\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N}$, we know

$$\frac{\mathbf{E}_\tau\mathbf{E}_\tau^\top}{N} = \begin{pmatrix} \widehat{\boldsymbol{\Lambda}}_\tau + \frac{1}{N}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top & \widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau \\ \boldsymbol{w}_\tau\widehat{\boldsymbol{\Lambda}}_\tau & \boldsymbol{w}_\tau^\top\widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau \end{pmatrix}.$$

Since $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is independent of all prompt inputs and query input, we have

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right)\otimes\left(\frac{\boldsymbol{w}_\tau^i}{N}\begin{pmatrix}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}\right)\right]\\
=&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right)\otimes\left(\frac{\boldsymbol{w}_\tau^i}{N}\begin{pmatrix}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}\right)\right]\middle|\mathbf{x}_{\tau,\mathsf{query}}\right]\\
=&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right)\otimes\left(\frac{\mathbb{E}\left[\boldsymbol{w}_\tau^i \mid \mathbf{x}_{\tau,\mathsf{query}}\right]}{N}\begin{pmatrix}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}\right)\right]=0.
\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[\boldsymbol{w}_\tau^\top\mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right]=\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right)\otimes\left(\boldsymbol{w}_\tau^i\begin{pmatrix}\widehat{\boldsymbol{\Lambda}}_\tau & \widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau\\ \boldsymbol{w}_\tau^\top\widehat{\boldsymbol{\Lambda}}_\tau & \boldsymbol{w}_\tau^\top\widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau.\end{pmatrix}\right)\right].$$

Since $\boldsymbol{X}_\tau$ only depends on $\mathbf{x}_{\tau,\mathsf{query}}$ by definition, and $\mathbf{x}_{\tau,\mathsf{query}}$ is independent of $\boldsymbol{w}_\tau$ and $\mathbf{x}_{\tau,i}, i=1,2,...,N$, we have

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{w}_\tau^\top\mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right]&=\frac{1}{2}\sum_{i=1}^{d}\left[\mathbb{E}\left(\mathbf{x}_{\tau,\mathsf{query}}^i \boldsymbol{X}_\tau\right)\otimes\mathbb{E}\left(\boldsymbol{w}_\tau^i\begin{pmatrix}\widehat{\boldsymbol{\Lambda}}_\tau & \widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau\\ \boldsymbol{w}_\tau^\top\widehat{\boldsymbol{\Lambda}}_\tau & \boldsymbol{w}_\tau^\top\widehat{\boldsymbol{\Lambda}}_\tau\boldsymbol{w}_\tau.\end{pmatrix}\right)\right]\\
&=\frac{1}{2}\sum_{i=1}^{d}\left[\begin{pmatrix}\mathbf{0}_{d\times d} & \boldsymbol{\Lambda}_i\\ \boldsymbol{\Lambda}_i^\top & 0\end{pmatrix}\otimes\begin{pmatrix}\mathbb{E}(\boldsymbol{w}_\tau^i)\boldsymbol{\Lambda} & \boldsymbol{\Lambda}\mathbb{E}(\boldsymbol{w}_\tau^i\boldsymbol{w}_\tau)\\ \mathbb{E}(\boldsymbol{w}_\tau^i\boldsymbol{w}_\tau^\top)\boldsymbol{\Lambda} & \mathbb{E}\left(\boldsymbol{w}_\tau^i\boldsymbol{w}_\tau^\top\boldsymbol{\Lambda}\boldsymbol{w}_\tau\right)\end{pmatrix}\right]\\
&=\frac{1}{2}\sum_{i=1}^{d}\begin{pmatrix}\mathbf{0}_{d\times d} & \boldsymbol{\Lambda}_i\\ \boldsymbol{\Lambda}_i^\top & 0\end{pmatrix}\otimes\begin{pmatrix}\mathbf{0}_{d\times d} & \boldsymbol{\Lambda}_i\\ \boldsymbol{\Lambda}_i^\top & 0\end{pmatrix},
\end{aligned}$$

where $\boldsymbol{\Lambda}_i$ denotes $\boldsymbol{\Lambda}_{:i}$. Here, the second line comes from the fact that $\mathbb{E}\widehat{\boldsymbol{\Lambda}}_\tau=\boldsymbol{\Lambda}$, and that $\boldsymbol{w}_\tau$ is independent of all prompt input and query input. The last line comes from the fact that $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. Therefore, simple computation shows that

$$\mathbb{E}\left[\boldsymbol{w}_\tau^\top\mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right]\boldsymbol{u}=\frac{1}{2}\begin{pmatrix}\mathbf{0}_{d(d+1)\times d(d+1)} & \mathbf{A}\\ \mathbf{A}^\top & \mathbf{0}_{(d+1)\times(d+1)}\end{pmatrix}\cdot\boldsymbol{u}, \tag{2.34}$$

where

$$\mathbf{A}=\begin{pmatrix}\mathbf{V}_1+\mathbf{V}_1^\top\\ \mathbf{V}_2+\mathbf{V}_2^\top\\ ...\\ \mathbf{V}_d+\mathbf{V}_d^\top\end{pmatrix}\in\mathbb{R}^{d(d+1)\times(d+1)},\quad \mathbf{V}_j=\begin{pmatrix}\mathbf{0}_{d\times d} & \boldsymbol{\Lambda}\boldsymbol{\Lambda}_j\\ \mathbf{0} & 0\end{pmatrix}\in\mathbb{R}^{(d+1)\times(d+1)}. \tag{2.35}$$

Step Two: Calculate the First Term Next, we compute the first term in (2.32), namely

$$\mathbf{D}:=2\mathbb{E}\left(\langle\boldsymbol{H}_\tau,\boldsymbol{u}\boldsymbol{u}^\top\rangle\boldsymbol{H}_\tau\boldsymbol{u}\right).$$

For simplicity, we denote $\mathbf{Z}_\tau := \frac{1}{N}\mathbf{E}_\tau\mathbf{E}_\tau^\top$. Using the definition of $\boldsymbol{H}_\tau$ in (2.24) and Lemma 2.10.1, we have

$$\begin{aligned}
\mathbf{D} &= 2\mathbb{E}\left(\langle \boldsymbol{H}_\tau, \boldsymbol{u}\boldsymbol{u}^\top\rangle \boldsymbol{H}_\tau \boldsymbol{u}\right) && \text{(definition)}\\
&= \frac{1}{2}\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{X}_\tau \otimes \mathbf{Z}_\tau \operatorname{Vec}(\boldsymbol{U}) \operatorname{Vec}(\boldsymbol{U})^\top\right)(\boldsymbol{X}_\tau \otimes \mathbf{Z}_\tau)\operatorname{Vec}(\boldsymbol{U})\right]\\
&= \frac{1}{2}\mathbb{E}\left[\operatorname{tr}\left(\operatorname{Vec}(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)\operatorname{Vec}(\boldsymbol{U})^\top\right)\operatorname{Vec}(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)\right]\\
&= \frac{1}{2}\mathbb{E}\left[\operatorname{Vec}(\boldsymbol{U})^\top \cdot \operatorname{Vec}(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)\cdot \operatorname{Vec}(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)\right]\\
&= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d+1}\left((\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{ij}\boldsymbol{U}_{ij}\right)\operatorname{Vec}(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)\right].
\end{aligned}$$

Step Three: $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ Vanish We first prove that if $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$, then $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}_{12} = \mathbf{0}_d$ and $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}_{21} = \mathbf{0}_d$. If this is true, then these two blocks will be zero all the time since we assume they are zero at the initial time in Assumption 2.3.3. We denote $\mathbf{A}_{k:}$ and $\mathbf{A}_{:k}$ as the k-th row and k-th column of matrix $\mathbf{A}$, respectively.

Under the assumption that $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$, we first compute

$$(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau) = \begin{pmatrix} \widehat{\boldsymbol{\Lambda}}_\tau \boldsymbol{w}_\tau u_{-1}\mathbf{x}_{\tau,\mathsf{query}}^\top & \left(\widehat{\boldsymbol{\Lambda}}_\tau + \frac{1}{N}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top\right)\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}} \\ \boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\boldsymbol{w}_\tau u_{-1}\mathbf{x}_{\tau,\mathsf{query}}^\top & \boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\end{pmatrix}.$$

Written in an entry-wise manner, it will be

$$(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{kl} = \begin{cases} \left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{k:}\boldsymbol{w}_\tau u_{-1}\mathbf{x}_{\tau,\mathsf{query}}^l & k,l\in[d] \\ \left(\widehat{\boldsymbol{\Lambda}}_\tau + \frac{1}{N}\mathbf{x}_{\tau,\mathsf{query}}\cdot\mathbf{x}_{\tau,\mathsf{query}}^\top\right)_{k:}\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}} & k\in[d], l = d+1 \\ \boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\boldsymbol{w}_\tau u_{-1}\mathbf{x}_{\tau,\mathsf{query}}^l & l\in[d], k = d+1 \\ \boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}} & k = l = d+1\end{cases}. \tag{2.36}$$

We use D_{ij} to denote the (i,j)-th entry of the $(d+1)\times(d+1)$ matrix $\bar{D}$ such that $\operatorname{Vec}(\bar{D}) = D$. Now we fix a $k\in[d]$, then

$$\begin{aligned}
D_{k,d+1} &= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d+1}\left((\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{ij}\boldsymbol{U}_{ij}\right)(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{k,d+1}\right]\\
&= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d}\left((\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{ij}\boldsymbol{U}_{ij}\right)(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{k,d+1}\right]\\
&\qquad + \frac{1}{2}\mathbb{E}\left[\left((\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{d+1,d+1}\,u_{-1}\right)(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau)_{k,d+1}\right],
\end{aligned} \tag{2.37}$$

since $\boldsymbol{U}_{i,d+1} = \boldsymbol{U}_{d+1,i} = 0$ for any $i \in [d]$. For the first term in the right hand side of last equation, we fix $i, j \in [d]$ and have

$$\begin{aligned}
&\mathbb{E}\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{ij} \boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{k,d+1} \\
=&\mathbb{E}\left(\boldsymbol{U}_{ij}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:} \boldsymbol{w}_\tau u_{-1} \mathbf{x}_{\tau,\mathsf{query}}^j \cdot \left(\widehat{\boldsymbol{\Lambda}}_\tau + \frac{1}{N}\mathbf{x}_{\tau,\mathsf{query}} \cdot \mathbf{x}_{\tau,\mathsf{query}}^\top\right)_{k:} \mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\right) = 0,
\end{aligned}$$

since $\boldsymbol{w}_\tau$ is independent with all prompt input and query input, namely all $\mathbf{x}_{\tau,i}$ for $i \in [\mathsf{query}]$, and $\boldsymbol{w}_\tau$ is mean zero. Similarly, for the second term of (2.37), we have

$$\begin{aligned}
&\mathbb{E}\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{d+1,d+1} u_{-1}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{k,d+1} \\
=&\mathbb{E}\left(u_{-1}\boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}} \cdot \left(\widehat{\boldsymbol{\Lambda}}_\tau + \frac{1}{N}\mathbf{x}_{\tau,\mathsf{query}} \cdot \mathbf{x}_{\tau,\mathsf{query}}\right)_{k:} \mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\right) = 0
\end{aligned}$$

since $\mathbb{E}\left(\boldsymbol{w}_\tau^\top\right) = 0$ and $\boldsymbol{w}_\tau$ is independent of all $\mathbf{x}_{\tau,i}$ for $i \in [\mathsf{query}]$. Therefore, we have $D_{k,d+1} = 0$ for $k \in [d]$. Similar calculation shows that $D_{d+1,k} = 0$ for $k \in [d]$.

For $k \in [d]$, to calculate the derivative of $\boldsymbol{U}_{k,d+1}$, it suffices to further calculate the inner product of the $d(d+1)+k$ th row of $\mathbb{E}\left[\boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}} \boldsymbol{H}_\tau\right]$ and $\boldsymbol{u}$. From (2.34), we know this is

$$\frac{1}{2}\sum_{j=1}^d \boldsymbol{\Lambda}_k^\top \boldsymbol{\Lambda}_j \boldsymbol{U}_{d+1,j} = 0$$

given that $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$. Therefore, we conclude that the derivative of $\boldsymbol{U}_{k,d+1}$ will vanish given $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$. Similarly, we conclude the same result for $\boldsymbol{U}_{d+1,k}$ for $k \in [d]$. Therefore, we know $\mathbf{u}_{12} = \mathbf{0}_d$ and $\mathbf{u}_{21} = \mathbf{0}_d$ for all time $t \geq 0$.

Step Four: Dynamics of $\mathbf{U}_{11}$ Next, we calculate the derivatives of $\mathbf{U}_{11}$ given $\mathbf{u}_{12} = \mathbf{u}_{21} = \mathbf{0}_d$. For a fixed pair of $k, l \in [d]$, we have

$$D_{kl} = \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^d \left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{ij} \boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{kl} + \left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{d+1,d+1} u_{-1}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{kl}\right].$$

For fixed $i, j \in [d]$, we have

$$\begin{aligned}
\mathbb{E}\left[\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{ij} \boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{kl}\right] &= \boldsymbol{U}_{ij} u_{-1}^2 \mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:} \boldsymbol{w}_\tau \mathbf{x}_{\tau,\mathsf{query}}^j \mathbf{x}_{\tau,\mathsf{query}}^l \boldsymbol{w}_\tau^\top \left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{:k}\right] \\
&= \boldsymbol{U}_{ij} u_{-1}^2 \mathbb{E}\left[\mathbf{x}_{\tau,\mathsf{query}}^j \mathbf{x}_{\tau,\mathsf{query}}^l\right] \cdot \mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{:k}\right] \\
&= \boldsymbol{U}_{ij} u_{-1}^2 \boldsymbol{\Lambda}_{\tau,jl} \mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{:k}\right].
\end{aligned}$$

Therefore, we sum over $i, j \in [d]$ to get

$$\frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^d \left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{ij} \boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{kl}\right] = \frac{1}{2}u_{-1}^2 \mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{k:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\right)\mathbf{U}_{11}\boldsymbol{\Lambda}_l$$

For the last term, we have

$$\frac{1}{2}\mathbb{E}\left[\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{d+1,d+1} u_{-1}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U} \boldsymbol{X}_\tau\right)_{kl}\right] = \frac{1}{2}u_{-1}^2 \mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{k:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\right)\mathbf{U}_{11}\boldsymbol{\Lambda}_l.$$

So we have

$$\mathbf{D}_{kl} = u_{-1}^2 \mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{k:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\right)\mathbf{U}_{11}\boldsymbol{\Lambda}_l.$$

Additionally, we have

$$\begin{aligned}
2\left[\mathbb{E}\left(\boldsymbol{w}_\tau^\top \mathbf{x}_{\tau,\mathsf{query}} \boldsymbol{H}_\tau\right)\boldsymbol{u}\right]_{(l-1)(d+1)+k} &= \left[\begin{pmatrix} \mathbf{0}_{d(d+1)\times d(d+1)} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0}_{(d+1)\times(d+1)} \end{pmatrix}\cdot \boldsymbol{u}\right]_{(l-1)(d+1)+k} && \text{(definition)} \\
&= \begin{pmatrix} \mathbf{0}_{(d+1)\times d(d+1)} & \mathbf{V}_l + \mathbf{V}_l^\top \end{pmatrix}_{k:} \cdot \boldsymbol{U} && \text{(definition of } \mathbf{A} \text{ in (2.35))} \\
&= \boldsymbol{\Lambda}_k^\top \boldsymbol{\Lambda}_l u_{-1}. && \text{(definition of } \mathbf{V}_i \text{ in (2.35))}
\end{aligned}$$

Therefore, we have that for $k, l \in [d]$, the dynamics of $\boldsymbol{U}_{kl}$ is

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_{kl} = -u_{-1}^2 \mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{k:}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\right)\mathbf{U}_{11}\boldsymbol{\Lambda}_l + u_{-1}\boldsymbol{\Lambda}_k^\top \boldsymbol{\Lambda}_l,$$

which implies

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11} = -u_{-1}^2 \mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\right)\mathbf{U}_{11}\boldsymbol{\Lambda} + u_{-1}\boldsymbol{\Lambda}^2.$$

From the definition of $\widehat{\boldsymbol{\Lambda}}_\tau$ (equation (2.33)), the independence and Gaussianity of $\mathbf{x}_{\tau,i}$ and Lemma 2.10.2, we compute

$$\begin{aligned}
\mathbb{E}\left(\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\right) &= \mathbb{E}\left(\left(\frac{1}{N}\sum_{i=1}^N \mathbf{x}_{\tau,i}\mathbf{x}_{\tau,i}^\top\right)^2\right) && \text{(definition (2.33))} \\
&= \frac{N-1}{N}\left[\mathbb{E}\left(\mathbf{x}_{\tau,1}\mathbf{x}_{\tau,1}^\top\right)\right]^2 + \frac{1}{N}\mathbb{E}\left(\mathbf{x}_{\tau,1}\mathbf{x}_{\tau,1}^\top\mathbf{x}_{\tau,1}\mathbf{x}_{\tau,1}^\top\right) \\
& && \text{(independence between prompt input)} \\
&= \frac{N+1}{N}\boldsymbol{\Lambda}^2 + \frac{1}{N}\operatorname{tr}(\boldsymbol{\Lambda})\boldsymbol{\Lambda}. && \text{(Lemma 2.10.2)}
\end{aligned}$$

We define

$$\boldsymbol{\Gamma} := \frac{N+1}{N}\boldsymbol{\Lambda} + \frac{1}{N}\operatorname{tr}(\boldsymbol{\Lambda})\mathbf{I}_d. \tag{2.38}$$

Then, from (2.32), we know the dynamics of $\mathbf{U}_{11}$ is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11} = -u_{-1}^2 \boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda} + u_{-1}\boldsymbol{\Lambda}^2. \tag{2.39}$$

Step Five: Dynamics of u_{-1} Finally, we compute the dynamics of u_{-1}. We have

$$\mathbf{D}_{d+1,d+1} = \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d}\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{ij}\boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}\right] + \frac{1}{2}\mathbb{E}\left[\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}u_{-1}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}\right]. \quad (2.40)$$

For the first term above, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i,j=1}^{d}\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{ij}\boldsymbol{U}_{ij}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}\right]\\
=&u_{-1}\sum_{i,j=1}^{d}\boldsymbol{U}_{ij}\mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\cdot \boldsymbol{w}_\tau\boldsymbol{w}_\tau^\top\cdot\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\cdot \mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\mathbf{x}_{\tau,\mathsf{query}}^j\right] && \text{(from (2.36))}\\
=&u_{-1}\sum_{i,j=1}^{d}\boldsymbol{U}_{ij}\mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\cdot\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\cdot \mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\mathbf{x}_{\tau,\mathsf{query}}^j\right] && \text{(independence and distribution of } \boldsymbol{w}_\tau)\\
=&u_{-1}\sum_{i,j=1}^{d}\boldsymbol{U}_{ij}\mathbb{E}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\cdot\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\cdot \mathbf{U}_{11}\boldsymbol{\Lambda}_j\right] && \text{(independence between prompt covariates)}\\
=&u_{-1}\mathbb{E}\,\mathrm{tr}\left[\sum_{i,j=1}^{d}\boldsymbol{\Lambda}_j\boldsymbol{U}_{ij}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)_{i:}\cdot\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\right] = u_{-1}\mathbb{E}\,\mathrm{tr}\left[\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\mathbf{U}_{11}\right]\\
=&u_{-1}\,\mathrm{tr}\left[\mathbb{E}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\right].
\end{aligned}$$

For the second term in (2.40), we have

$$\begin{aligned}
&\mathbb{E}\left[\left(\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}u_{-1}\right)\left(\boldsymbol{Z}_\tau \boldsymbol{U}\boldsymbol{X}_\tau\right)_{d+1,d+1}\right]\\
=&u_{-1}\mathbb{E}\left[\boldsymbol{w}_\tau^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\mathbf{x}_{\tau,\mathsf{query}}\mathbf{x}_{\tau,\mathsf{query}}^\top(\mathbf{U}_{11})^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\boldsymbol{w}_\tau\right] && \text{(from (2.36))}\\
=&u_{-1}\mathbb{E}\,\mathrm{tr}\left[\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)\right] = u_{-1}\,\mathrm{tr}\left[\mathbb{E}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\right].
\end{aligned}$$

Therefore, we know

$$\mathbf{D}_{d+1,d+1} = u_{-1}\,\mathrm{tr}\left[\mathbb{E}\left(\widehat{\boldsymbol{\Lambda}}_\tau\right)^2\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\right].$$

Additionally, we have

$$\begin{aligned}
2\left[\mathbb{E}\left(\boldsymbol{w}_\tau^\top\mathbf{x}_{\tau,\mathsf{query}}\boldsymbol{H}_\tau\right)\boldsymbol{u}\right]_{(d+1)^2} &= \left[\begin{pmatrix}\mathbf{0}_{d(d+1)\times d(d+1)} & \mathbf{A}\\ \mathbf{A}^\top & \mathbf{0}_{(d+1)\times(d+1)}\end{pmatrix}\cdot\boldsymbol{u}\right]_{(d+1)^2} && \text{(from (2.34))}\\
&= \begin{pmatrix}\mathbf{V}_1+\mathbf{V}_1^\top & \dots & \mathbf{V}_d+\mathbf{V}_d^\top & \mathbf{0}_{(d+1)\times(d+1)}\end{pmatrix}_{d+1:}\cdot U && \text{(definition of } A \text{ in (2.35))}\\
&= \sum_{i,j=1}^{d}\boldsymbol{\Lambda}_i^\top\boldsymbol{\Lambda}_j\boldsymbol{U}_{ji} = \mathrm{tr}\left(\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top\boldsymbol{\Lambda}\right).
\end{aligned}$$

Then, from (2.32), we have the dynamics of u_{-1} is

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{-1} = -\operatorname{tr}\left[u_{-1}\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top - \mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right]. \tag{2.41}$$

□

Proof of Lemma 2.5.3

Lemma 2.5.3 gives the form of global minima of an equivalent loss function. First, we prove that gradient flow on L defined in (2.8) from the initial values satisfying Assumption 2.3.3 is equivalent to gradient flow on another loss function $\tilde{\ell}$ defined below. Then, we derive an expression for the global minima of this loss function.

First, from the dynamics of gradient flow, we can actually recover the loss function up to a constant. We have the following lemma.

Lemma 2.7.1 (Loss Function)**.** *Consider gradient flow over L in (2.26) with respect to $\boldsymbol{u}$ starting from an initial value satisfying Assumption 2.3.3. This is equivalent to doing gradient flow with respect to $\mathbf{U}_{11}$ and u_{-1} on the loss function*

$$\tilde{\ell}\left(\mathbf{U}_{11}, u_{-1}\right) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top - u_{-1}\mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right]. \tag{2.42}$$

Proof. The proof is simply by taking gradient of the loss function in (2.42). For techniques in matrix derivatives, see Lemma 2.10.1. We take the gradient of $\tilde{\ell}$ on $\mathbf{U}_{11}$ to obtain

$$\frac{\partial\tilde{\ell}}{\partial\mathbf{U}_{11}} = \frac{1}{2}u_{-1}^2\mathbf{\Lambda}^\top\mathbf{\Gamma}^\top\mathbf{U}_{11}\mathbf{\Lambda}^\top + \frac{1}{2}u_{-1}^2\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda} - u_{-1}\mathbf{\Lambda}^2 = u_{-1}^2\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda} - u_{-1}\mathbf{\Lambda}^2,$$

since $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are commutable. We take derivatives w.r.t. u_{-1} to get

$$\frac{\partial\tilde{\ell}}{\partial u_{-1}} = \operatorname{tr}\left[u_{-1}\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top - \mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right].$$

Combining this with Lemma 2.5.2, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t) = -\frac{\partial\tilde{\ell}}{\partial\mathbf{U}_{11}}, \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) = -\frac{\partial\tilde{\ell}}{\partial u_{-1}}.$$

□

We remark that actually this is the loss function L up to some constant. This loss function $\tilde{\ell}$ can be negative. But we can still compute its global minima as follows.

Corollary 2.7.2 (Minimum of Loss Function)**.** *The loss function $\tilde{\ell}$ in Lemma 2.7.1 satisfies*

$$\min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}} \tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)=-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}\right]$$

and

$$\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)-\min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}} \tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)=\frac{1}{2}\left\|\mathbf{\Gamma}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\right\|_F^2.$$

Proof. First, we claim that

$$\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)=\frac{1}{2}\operatorname{tr}\left[\mathbf{\Gamma}\cdot\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)^{\top}\right]-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}\right].$$

To calculate this, we just need to expand the terms in the brackets and notice that $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are commutable:

$$\begin{aligned}
&\operatorname{tr}\left[\mathbf{\Gamma}\cdot\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)^{\top}\right]-\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}\right]\\
&\overset{(i)}{=}\operatorname{tr}\left[\mathbf{\Gamma}\cdot\left(u_{-1}^2\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^{\top}\mathbf{\Lambda}^{1/2}-u_{-1}\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{3}{2}}\mathbf{\Gamma}^{-1}+\mathbf{\Gamma}^{-2}\mathbf{\Lambda}^2\right)\right]\\
&\qquad\qquad-\operatorname{tr}[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}]\\
&\overset{(ii)}{=}u_{-1}^2\operatorname{tr}\left[\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^{\top}\right]-2u_{-1}\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}\right]=2\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right).
\end{aligned}$$

Equations (i) and (ii) use that $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ commute.

Since $\mathbf{\Gamma}\succeq 0$ and $\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)^{\top}\succeq 0$, we know from Lemma 2.10.4 that

$$\frac{1}{2}\operatorname{tr}\left[\mathbf{\Gamma}\cdot\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}}-\mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)^{\top}\right]\geq 0,$$

which implies

$$\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)\geq-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}\right].$$

Equality holds when

$$\mathbf{U}_{11}=\mathbf{\Gamma}^{-1},\quad u_{-1}=1,$$

so the minimum of $\tilde{\ell}$ must be $-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Lambda}^2\mathbf{\Gamma}^{-1}\right]$. The expression for $\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)-\min\tilde{\ell}\left(\mathbf{U}_{11},u_{-1}\right)$ comes from the fact that $\operatorname{tr}(A^{\top}A)=\|A\|_F^2$ for any matrix A. □

Lemma 2.5.3 is an immediate consequence of Corollary2.7.2, since the loss will keep the same when we replace $(\mathbf{U}_{11},u_{-1})$ by $(c\mathbf{U}_{11},c^{-1}u_{-1})$ for any non-zero constant c.

Proof of Lemma 2.5.4

In this section, we prove that the dynamical system in Lemma 2.5.2 satisfies a PL inequality. Then, the PL inequality naturally leads to the global convergence of this dynamical system. First, we prove a simple lemma, which says the parameters in the LSA model will keep 'balanced' in the whole trajectory. From the proof of this lemma, we can understand why we assume a balanced parameter at the initial time.

Lemma 2.7.3 (Balanced Parameters)**.** *Consider gradient flow over L in (2.26) with respect to $\boldsymbol{u}$ starting from an initial value satisfying Assumption 2.3.3. For any $t \geq 0$, it holds that*

$$u_{-1}^2 = \operatorname{tr}\left[\mathbf{U}_{11}(\mathbf{U}_{11})^\top\right]. \tag{2.43}$$

Proof. From Lemma 2.5.2, we multiply the first equation in (2.27) by $(\mathbf{U}_{11})^\top$ from the right to get

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t)\right)(\mathbf{U}_{11}(t))^\top = -u_{-1}^2\boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top + u_{-1}\boldsymbol{\Lambda}^2(\mathbf{U}_{11})^\top.$$

Also we multiply the second equation in Lemma 2.5.2 by u_{-1} to obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t) = \operatorname{tr}\left[-u_{-1}^2\boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{U}_{11}\boldsymbol{\Lambda}(\mathbf{U}_{11})^\top + u_{-1}\boldsymbol{\Lambda}^2(\mathbf{U}_{11})^\top\right].$$

Therefore, we have

$$\operatorname{tr}\left[\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t)\right)(\mathbf{U}_{11}(t))^\top\right] = \left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t).$$

Taking the transpose of the equation above and adding to itself gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{tr}\left[\mathbf{U}_{11}(t)(\mathbf{U}_{11}(t))^\top\right] = \frac{\mathrm{d}}{\mathrm{d}t}\left(u_{-1}(t)^2\right).$$

Notice that from Assumption 2.3.3, we know that at $t = 0$,

$$u_{-1}(0)^2 = \sigma^2 = \sigma^2\operatorname{tr}\left[\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right] = \operatorname{tr}\left[\mathbf{U}_{11}(0)(\mathbf{U}_{11}(0))^\top\right].$$

So for any time $t \geq 0$, the equation holds. □

In order to prove the PL inequality, we first prove an important property which says the trajectories of $u_{-1}(t)$ stay away from saddle point at origin. First, we prove that $u_{-1}(t)$ will stay positive along the whole trajectory.

Lemma 2.7.4. *Consider gradient flow over L in (2.26) with respect to $\boldsymbol{u}$ starting from an initial value satisfying Assumption 2.3.3. If the initial scale satisfies*

$$0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\,\|\boldsymbol{\Gamma}\|_{op}}}, \tag{2.44}$$

then, for any $t \geq 0$, it holds that

$$u_{-1} > 0.$$

Proof. From Lemma 2.7.1, we are actually doing gradient flow on the loss $\tilde{\ell}$. The loss function is non-increasing, because

$$\frac{\mathrm{d}\tilde{\ell}}{\mathrm{d}t} = \left\langle \frac{\mathrm{d}\mathbf{U}_{11}}{\mathrm{d}t}, \frac{\partial \tilde{\ell}}{\partial \mathbf{U}_{11}} \right\rangle + \left\langle \frac{\mathrm{d}u_{-1}}{\mathrm{d}t}, \frac{\partial \tilde{\ell}}{\partial u_{-1}} \right\rangle = -\left\| \frac{\mathrm{d}\mathbf{U}_{11}}{\mathrm{d}t} \right\|_F^2 - \left\| \frac{\mathrm{d}u_{-1}}{\mathrm{d}t} \right\|_F^2 \leq 0.$$

We notice that when $u_{-1} = 0$, the loss function $\tilde{\ell} = 0$. Therefore, as long as $\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) < 0$, then for any time, u_{-1} will be non-zero. Further, since $u_{-1}(0) > 0$ and the trajectory of $u_{-1}(t)$ must be continuous, we know $u_{-1}(t) > 0$ for any $t \geq 0$.

Then, it suffices to prove when $0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\boldsymbol{\Gamma}\|_{op}}}$, it holds that $\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) < 0$. From Assumption 2.3.3, we can calculate the loss function at the initial time:

$$\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) = \frac{\sigma^4}{2} \operatorname{tr}\left[\boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right] - \sigma^2 \operatorname{tr}\left[\boldsymbol{\Lambda}^2\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right].$$

From the property of trace, we know

$$\operatorname{tr}\left[\boldsymbol{\Lambda}^2\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right] = \operatorname{tr}\left[\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\boldsymbol{\Lambda}^\top\right] = \|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F^2.$$

From Von-Neumann's trace inequality (Lemma 2.10.3) and the fact that $\left\|\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right\|_F = 1$, we know

$$\operatorname{tr}\left[\boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right] \leq \sqrt{d}\left\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\right\|_F \cdot \|\boldsymbol{\Gamma}\|_{op} \leq \sqrt{d}\,\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F^2\,\|\boldsymbol{\Gamma}\|_{op}.$$

Therefore, we have

$$\begin{aligned}\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) &\leq \frac{\sqrt{d}\sigma^4}{2}\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F^2\,\|\boldsymbol{\Gamma}\|_{op} - \sigma^2\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F^2 \\ &= \frac{\sigma^2}{2}\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F^2\left[\sqrt{d}\sigma^2\|\boldsymbol{\Gamma}\|_{op} - 2\right].\end{aligned}$$

From Assumption 2.3.3, we know $\|\boldsymbol{\Lambda}\boldsymbol{\Theta}\|_F \neq 0$. From (2.38), we know $\|\boldsymbol{\Gamma}\|_{op} > 0$. Therefore, when

$$0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\,\|\boldsymbol{\Gamma}\|_{op}}},$$

we have

$$\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) < 0.$$

□

From the lemma above, we can actually further prove that the $u_{-1}(t)$ can be lower bounded by a positive constant for any $t \geq 0$. This will be a critical property to prove the PL inequality. We have the following lemma.

Lemma 2.7.5. *Consider gradient flow over L in (2.26) with respect to $\boldsymbol{u}$ starting from an initial value satisfying Assumption 2.3.3 with initial scale $0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\mathbf{\Gamma}\|_{op}}}$. For any $t \geq 0$, it holds that*

$$u_{-1} \geq \sqrt{\frac{\sigma^2}{2\sqrt{d}\,\|\mathbf{\Lambda}\|_{op}^2}\,\|\mathbf{\Lambda\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\,\|\mathbf{\Gamma}\|_{op}\right]} > 0. \tag{2.45}$$

Proof. We prove by contradiction. Suppose the claim does not hold. From Lemma 2.7.3, we know $u_{-1}^2 = \operatorname{tr}\left[\mathbf{U}_{11}(\mathbf{U}_{11})^\top\right] = \|\mathbf{U}_{11}\|_F^2$. From Lemma 2.7.4, we know $u_{-1} = \|\mathbf{U}_{11}\|_F$. Recall the definition of loss function:

$$\tilde{\ell}(\mathbf{U}_{11}, u_{-1}) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2\mathbf{\Gamma\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top - u_{-1}\mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right].$$

Since $\mathbf{\Gamma} \succeq 0, \mathbf{\Lambda} \succeq 0$, and they commute, we know from Lemma 2.10.4 that $\mathbf{\Gamma\Lambda} \succeq 0$. Again, since $\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top \succeq 0$, from Lemma 2.10.4 we have $\operatorname{tr}\left[\frac{1}{2}u_{-1}^2\mathbf{\Gamma\Lambda}\mathbf{U}_{11}\mathbf{\Lambda}(\mathbf{U}_{11})^\top\right]$ is non-negative. So

$$\tilde{\ell}(\mathbf{U}_{11}, u_{-1}) \geq -\operatorname{tr}\left[u_{-1}\mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right].$$

From Von-Neumann's trace inequality, we know for any $t \geq 0$,

$$-\operatorname{tr}\left[u_{-1}\mathbf{\Lambda}^2(\mathbf{U}_{11})^\top\right] \geq -\sqrt{d}u_{-1}\left\|\mathbf{\Lambda}^2\right\|_{op}\|\mathbf{U}_{11}\|_F = -\sqrt{d}u_{-1}^2\,\|\mathbf{\Lambda}\|_{op}^2.$$

Therefore, under our assumption that the claim does not hold, we have

$$\tilde{\ell}(\mathbf{U}_{11}, u_{-1}) \geq -\sqrt{d}u_{-1}^2\,\|\mathbf{\Lambda}\|_{op}^2 > -\frac{\sigma^2}{2}\,\|\mathbf{\Lambda\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\,\|\mathbf{\Gamma}\|_{op}\right] \geq \tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)).$$

Here, the last inequality comes from the proof of Lemma 2.7.4. This contradicts the non-increasing property of the loss function in gradient flow. □

Finally, let's prove the PL inequality and further, the global convergence of gradent flow on the loss function $\tilde{\ell}$. We recall the stated lemma from the main text.

Lemma 2.5.4. *Suppose the initialization of gradient flow satisfies Assumption 2.3.3 with initialization scale satisfying $\sigma^2 < \frac{2}{\sqrt{d}\|\mathbf{\Gamma}\|_{op}}$ for $\mathbf{\Gamma} = (1 + \frac{1}{N})\mathbf{\Lambda} + \frac{\operatorname{tr}(\mathbf{\Lambda})}{N}\mathbf{I}_d$. If we define*

$$\mu := \frac{\sigma^2}{\sqrt{d}\,\|\mathbf{\Lambda}\|_{op}^2\operatorname{tr}\left(\mathbf{\Gamma}^{-1}\mathbf{\Lambda}^{-1}\right)\operatorname{tr}\left(\mathbf{\Lambda}^{-1}\right)}\,\|\mathbf{\Lambda\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\,\|\mathbf{\Gamma}\|_{op}\right] > 0, \tag{2.30}$$

then gradient flow on $\tilde{\ell}$ with respect to $\mathbf{U}_{11}$ and u_{-1} satisfies, for any $t \geq 0$,

$$\begin{aligned}\left\|\nabla\tilde{\ell}(\mathbf{U}_{11}(t), u_{-1}(t))\right\|_2^2 :=& \left\|\frac{\partial\tilde{\ell}}{\partial\mathbf{U}_{11}}\right\|_F^2 + \left|\frac{\partial\tilde{\ell}}{\partial u_{-1}}\right|^2 \\ \geq& \mu\left(\tilde{\ell}(\mathbf{U}_{11}(t), u_{-1}(t)) - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right).\end{aligned}$$

Moreover, gradient flow converges to the global minimum of $\tilde{\ell}$, and $\mathbf{U}_{11}$ and u_{-1} converge to the following,

$$\lim_{t\to\infty} u_{-1}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{\frac{1}{2}} \text{ and } \lim_{t\to\infty} \mathbf{U}_{11}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{-\frac{1}{2}} \mathbf{\Gamma}^{-1}. \tag{2.31}$$

Proof. From the definition and Lemma 2.7.5, we have

$$\begin{aligned}&\|\nabla\ell(\mathbf{U}_{11}, u_{-1})\|_2^2 \geq \left\|\frac{\partial\ell}{\partial\mathbf{U}_{11}}\right\|_F^2 = \left\|u_{-1}^2\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{U}_{11}\mathbf{\Lambda} - u_{-1}\mathbf{\Lambda}^2\right\|_F^2 \\ &= u_{-1}^2\left\|\mathbf{\Gamma}\mathbf{\Lambda}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\mathbf{\Lambda}^{\frac{1}{2}}\right\|_F^2 \\ &\geq \frac{\sigma^2}{2\sqrt{d}\,\|\mathbf{\Lambda}\|_{op}^2}\|\mathbf{\Lambda}\mathbf{\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\|\mathbf{\Gamma}\|_{op}\right]\left\|\mathbf{\Gamma}\mathbf{\Lambda}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\mathbf{\Lambda}^{\frac{1}{2}}\right\|_F^2.\end{aligned} \tag{2.46}$$

To see why the second line is true, recall that $u_{-1} \in \mathbb{R}$ and $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ commute. The last line comes from the lower bound of u_{-1} in Lemma 2.7.5. From Corollary 2.7.2, we know

$$\begin{aligned}\ell - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \ell(\mathbf{U}_{11}, u_{-1}) &= \frac{1}{2}\operatorname{tr}\left[\mathbf{\Gamma}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)^\top\right] \\ &= \frac{1}{2}\left\|\mathbf{\Gamma}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\right\|_F^2.\end{aligned}$$

Therefore, we know that

$$\begin{aligned}&\ell - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \ell(\mathbf{U}_{11}, u_{-1}) \\ &\leq \frac{1}{2}\left\|\mathbf{\Gamma}\mathbf{\Lambda}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\mathbf{\Lambda}^{\frac{1}{2}}\right\|_F^2 \cdot \left\|\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{\Lambda}^{-\frac{1}{2}}\right\|_F^2\left\|\mathbf{\Lambda}^{-\frac{1}{2}}\right\|_F^2 \\ &= \frac{1}{2}\left\|\mathbf{\Gamma}\mathbf{\Lambda}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\mathbf{\Lambda}^{\frac{1}{2}}\right\|_F^2 \cdot \operatorname{tr}\left(\mathbf{\Gamma}^{-1}\mathbf{\Lambda}^{-1}\right)\operatorname{tr}\left(\mathbf{\Lambda}^{-1}\right).\end{aligned} \tag{2.47}$$

We compare (2.46) and (2.47) to obtain that in order to make the PL condition hold, one needs to let

$$\mu := \frac{\sigma^2}{\sqrt{d}\,\|\mathbf{\Lambda}\|_{op}^2\operatorname{tr}\left(\mathbf{\Gamma}^{-1}\mathbf{\Lambda}^{-1}\right)\operatorname{tr}\left(\mathbf{\Lambda}^{-1}\right)}\|\mathbf{\Lambda}\mathbf{\Theta}\|_F^2\left[2 - \sqrt{d}\sigma^2\|\mathbf{\Gamma}\|_{op}\right] > 0.$$

Once we set this μ, we get the PL inequality. The μ is positive due to the assumption for σ in the lemma.

From the dynamics of gradient flow and the PL condition, we know

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\left(\tilde{\ell} - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right) = \left\langle \frac{\mathrm{d}\mathbf{U}_{11}}{\mathrm{d}t}, \frac{\partial\tilde{\ell}}{\partial\mathbf{U}_{11}}\right\rangle + \left\langle \frac{\mathrm{d}u_{-1}}{\mathrm{d}t}, \frac{\partial\tilde{\ell}}{\partial u_{-1}}\right\rangle \\
= &-\left\|\frac{\mathrm{d}\mathbf{U}_{11}}{\mathrm{d}t}\right\|_F^2 - \left|\frac{\mathrm{d}u_{-1}}{\mathrm{d}t}\right|^2 \le -\mu\left(\tilde{\ell} - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right).
\end{aligned}$$

Therefore, we have when $t \to \infty$,

$$\begin{aligned}
0 \le\ & \tilde{\ell} - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1}) \\
\le\ & \exp(-\mu t)\left[\tilde{\ell}(\mathbf{U}_{11}(0), u_{-1}(0)) - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right] \to 0,
\end{aligned}$$

which implies

$$\lim_{t\to\infty}\left[\tilde{\ell} - \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(\mathbf{U}_{11}, u_{-1})\right] = 0.$$

From Corollary 2.7.2, we know this is

$$\left\|\mathbf{\Gamma}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\right\|_F^2 \to 0.$$

Since $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are non-singular and positive definite, and they commute, we know

$$\left\|u_{-1}\mathbf{U}_{11} - \mathbf{\Gamma}^{-1}\right\|_F^2 \le \left\|\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{\Lambda}^{-\frac{1}{2}}\right\|_F^2 \left\|\mathbf{\Gamma}^{\frac{1}{2}}\left(u_{-1}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{11}\mathbf{\Lambda}^{\frac{1}{2}} - \mathbf{\Lambda}\mathbf{\Gamma}^{-1}\right)\right\|_F^2 \left\|\mathbf{\Lambda}^{-\frac{1}{2}}\right\|_F^2 \to 0.$$

This implies $u_{-1}\mathbf{U}_{11} - \mathbf{\Gamma}^{-1} \to \mathbf{0}_{d\times d}$ entry-wise. Since $u_{-1} = \|\mathbf{U}_{11}\|_F$, we know

$$u_{-1}^2 = \|u_{-1}\mathbf{U}_{11}\|_F \to \left\|\mathbf{\Gamma}^{-1}\right\|_F.$$

Therefore, we know

$$\lim_{t\to\infty} u_{-1}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{\frac{1}{2}} \text{ and } \lim_{t\to\infty} \mathbf{U}_{11}(t) = \left\|\mathbf{\Gamma}^{-1}\right\|_F^{-\frac{1}{2}} \mathbf{\Gamma}^{-1}.$$

□

2.8 Appendix: Proof of Theorem 2.4.2

In this section, we prove Theorem 2.4.2, which characterizes the excess risk of the prediction of a trained LSA layer with respect to the risk of best linear predictor, on a new task which is possibly non-linear. First, we restate the theorem.

Theorem 2.4.2. *Let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on $\mathbf{x}$ is $\mathcal{D}_{\mathbf{x}} = \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$. Assume $\mathbb{E}_{\mathcal{D}}[y], \mathbb{E}_{\mathcal{D}}[\mathbf{x}y], \mathbb{E}_{\mathcal{D}}[y^2\mathbf{x}\mathbf{x}^\top]$ exist and are finite. Assume the test prompt is of the form $P = (\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_M, y_M, \boldsymbol{x}_{\mathsf{query}})$, where $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Let f^*_{LSA} be the LSA model with parameters $\boldsymbol{W}^{PV}_*$ and $\boldsymbol{W}^{KQ}_*$ in (2.11), and $\widehat{y}_{\mathsf{query}}$ is the prediction for $\boldsymbol{x}_{\mathsf{query}}$ given the prompt. If we define*

$$\mathbf{a} := \mathbf{\Lambda}^{-1}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbf{x}y\right], \qquad \mathbf{\Sigma} := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\Big(\mathbf{x}y - \mathbb{E}\left(\mathbf{x}y\right)\Big)\Big(\mathbf{x}y - \mathbb{E}\left(\mathbf{x}y\right)\Big)^\top\right], \tag{2.15}$$

then, for $\mathbf{\Gamma} = \mathbf{\Lambda} + \frac{1}{N}\mathbf{\Lambda} + \frac{1}{N}\mathrm{tr}(\mathbf{\Lambda})\mathbf{I}_d$. we have,

$$\begin{aligned}\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 = &\underbrace{\min_{\boldsymbol{w}\in\mathbb{R}^d}\mathbb{E}\left(\langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\textit{Error of best linear predictor}} \\ &+ \frac{1}{M}\mathrm{tr}\left[\mathbf{\Sigma}\mathbf{\Gamma}^{-2}\mathbf{\Lambda}\right] + \frac{1}{N^2}\left[\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}^3} + 2\mathrm{tr}(\mathbf{\Lambda})\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}^2} + \mathrm{tr}(\mathbf{\Lambda})^2\|\mathbf{a}\|^2_{\mathbf{\Gamma}^{-2}\mathbf{\Lambda}}\right], \end{aligned}\tag{2.16}$$

where the expectation is over $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

Proof. We denote $\mathbb{E}$ as the expectation over $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ unless otherwise specified. Since when $(\mathbf{x}, y) \sim \mathcal{D}$, we assume $\mathbb{E}[\mathbf{x}], \mathbb{E}[y], \mathbb{E}[\mathbf{x}y], \mathbb{E}[\mathbf{x}\mathbf{x}^\top], \mathbb{E}[y^2\mathbf{x}\mathbf{x}^\top]$ exist, we know that $\mathbb{E}\left(\langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2$ exists for each $\boldsymbol{w} \in \mathbb{R}^d$. We denote

$$\boldsymbol{a} := \arg\min_{\boldsymbol{w}\in\mathbb{R}^d}\mathbb{E}\left(\langle \boldsymbol{w}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2$$

as the weight of the best linear approximator. Actually, if we denote the function inside the minimum above as $R(\boldsymbol{w})$, we can write it as

$$R(\boldsymbol{w}) = \boldsymbol{w}^\top\mathbf{\Lambda}\boldsymbol{w} - 2\mathbb{E}\left(y_{\mathsf{query}}\cdot\boldsymbol{x}^\top_{\mathsf{query}}\right)\boldsymbol{w} + \mathbb{E}y^2_{\mathsf{query}}.$$

Since the Hessian matrix $\frac{\partial^2}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}R(\boldsymbol{w})$ is $\mathbf{\Lambda}$, which is positive definite, we know that this function is strictly convex and hence, the global minimum can be achieved at the unique first-order stationary point. This is

$$\mathbf{a} = \mathbf{\Lambda}^{-1}\mathbb{E}\left(y_{\mathsf{query}}\cdot\boldsymbol{x}_{\mathsf{query}}\right). \tag{2.48}$$

We also define a similar vector for ease of computation:

$$\boldsymbol{b} = \mathbf{\Gamma}^{-1}\mathbb{E}\left(y_{\mathsf{query}}\cdot\boldsymbol{x}_{\mathsf{query}}\right). \tag{2.49}$$

Therefore, we can decompose the risk as

$$\begin{aligned}\mathbb{E}\left(\widehat{y}_{\mathsf{query}}-y_{\mathsf{query}}\right)^2 = &\underbrace{\mathbb{E}\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\mathrm{I}} + \underbrace{\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)^2}_{\mathrm{II}} \\ &+ \underbrace{\mathbb{E}\left(\langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle - \langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)^2}_{\mathrm{III}} + \underbrace{2\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)}_{\mathrm{IV}} \\ &+ \underbrace{2\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle - \langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)}_{\mathrm{V}} \\ &+ \underbrace{2\mathbb{E}\left(\langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle - \langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)}_{\mathrm{VI}}\end{aligned}$$

The term I is the first term on the right hand side of (2.16). So it suffices to calculate II to VI.

First, from the tower property of conditional expectation, we have

$$\begin{aligned}\mathrm{V} &= 2\mathbb{E}\left[\mathbb{E}\left(\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle - \langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\middle| \boldsymbol{x}_{\mathsf{query}}\right)\right] \\ &= 2\mathbb{E}\left[\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\middle| \boldsymbol{x}_{\mathsf{query}}\right)\left(\langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle - \langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\right] = 0,\end{aligned}$$

since

$$\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\middle| \boldsymbol{x}_{\mathsf{query}}\right) = \left(\mathbb{E}\frac{1}{M}\sum_{i=1}^{M} y_i \boldsymbol{\Gamma}^{-1}\boldsymbol{x}_i - \boldsymbol{b}\right)^{\top} \boldsymbol{x}_{\mathsf{query}} = 0.$$

Similarly, for IV, we have

$$\begin{aligned}\mathrm{IV} &= 2\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right) \\ &= 2\mathbb{E}\left[\mathbb{E}\left(\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)\middle| \boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}\right)\right] \\ &= 2\mathbb{E}\left[\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle \boldsymbol{b}, \boldsymbol{x}_{\mathsf{query}}\rangle\middle| \boldsymbol{x}_{\mathsf{query}}, y_{\mathsf{query}}\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)\right] \\ &= 0.\end{aligned}$$

For VI, we have

$$\begin{aligned}\mathrm{VI} &= 2\mathbb{E}\mathrm{tr}\left[\left(\boldsymbol{b}-\mathbf{a}\right)\left(\langle \mathbf{a}, \boldsymbol{x}_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)\boldsymbol{x}_{\mathsf{query}}^{\top}\right] \\ &= 2\mathrm{tr}\left[(\boldsymbol{b}-\mathbf{a})\mathbf{a}^{\top}\boldsymbol{\Lambda}\right] - 2\mathrm{tr}\left[(\boldsymbol{b}-\mathbf{a})\mathbb{E}\left(y_{\mathsf{query}}\boldsymbol{x}_{\mathsf{query}}^{\top}\right)\right] = 0,\end{aligned}$$

where the last line comes from the definition of a. Therefore, all cross terms vanish and it suffices to consider II and III.

For II, from the definition we have

$$
\begin{aligned}
&\text{II} \\
=&\mathbb{E}\left(\frac{1}{M}\sum_{i=1}^{M} y_i \boldsymbol{x}_i - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)^{\top} \boldsymbol{\Gamma}^{-1} \boldsymbol{x}_{\mathsf{query}} \boldsymbol{x}_{\mathsf{query}}^{\top} \boldsymbol{\Gamma}^{-1} \left(\frac{1}{M}\sum_{i=1}^{M} y_i \boldsymbol{x}_i - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right) \\
=&\mathbb{E}\mathrm{tr}\left(\frac{1}{M}\sum_{i=1}^{M} y_i \boldsymbol{x}_i - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)\left(\frac{1}{M}\sum_{i=1}^{M} y_i \boldsymbol{x}_i - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)^{\top} \boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda} \\
&\qquad\qquad \text{(property of trace and the fact that } \boldsymbol{\Gamma} \text{ and } \boldsymbol{\Lambda} \text{ commute)} \\
=&\frac{1}{M^2}\sum_{i,j=1}^{M} \mathbb{E}\mathrm{tr}\left\{\left(y_i \boldsymbol{x}_i - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)\left(y_j \boldsymbol{x}_j - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)^{\top} \boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}\right\} \\
=&\frac{1}{M}\mathbb{E}\mathrm{tr}\left\{\left(y_1 \boldsymbol{x}_1 - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)\left(y_1 \boldsymbol{x}_1 - \mathbb{E}\left(y_{\mathsf{query}} \cdot \boldsymbol{x}_{\mathsf{query}}\right)\right)^{\top} \boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}\right\} \\
&\qquad\qquad \text{(all cross terms vanish due to the independence of } \boldsymbol{x}_i\text{)} \\
=&\frac{1}{M}\mathrm{tr}\left[\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}\right].
\end{aligned}
$$

The last line comes from the definition of $\boldsymbol{\Sigma}$.

For III, we have

$$
\begin{aligned}
\text{III} &= \mathbb{E}(\boldsymbol{b}-\mathbf{a})^{\top}\boldsymbol{x}_{\mathsf{query}}\boldsymbol{x}_{\mathsf{query}}^{\top}(\boldsymbol{b}-\mathbf{a}) = \mathbf{a}^{\top}\boldsymbol{\Lambda}(\boldsymbol{\Gamma}^{-1}-\boldsymbol{\Lambda}^{-1})\boldsymbol{\Lambda}(\boldsymbol{\Gamma}^{-1}-\boldsymbol{\Lambda}^{-1})\boldsymbol{\Lambda}\mathbf{a} \\
&= \mathrm{tr}\left[\left(\mathbf{I}_d - \boldsymbol{\Gamma}\boldsymbol{\Lambda}^{-1}\right)^2 \boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}^3 \mathbf{a}\mathbf{a}^{\top}\right] \\
&\qquad\qquad \text{(property of trace and the fact that } \boldsymbol{\Gamma} \text{ and } \boldsymbol{\Lambda} \text{ commute)} \\
&= \frac{1}{N^2}\mathrm{tr}\left[\left(\mathbf{I}_d + \mathrm{tr}(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\right)^2 \boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}^3 \mathbf{a}\mathbf{a}^{\top}\right] \\
&= \frac{1}{N^2}\left[\mathrm{tr}(\boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}^3\mathbf{a}\mathbf{a}^{\top}) + 2\mathrm{tr}(\boldsymbol{\Lambda})\mathrm{tr}(\boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}^2\mathbf{a}\mathbf{a}^{\top}) + \mathrm{tr}(\boldsymbol{\Lambda})^2\mathrm{tr}(\boldsymbol{\Gamma}^{-2}\boldsymbol{\Lambda}\mathbf{a}\mathbf{a}^{\top})\right].
\end{aligned}
$$

Combining all terms above, we conclude. □

2.9 Appendix: Proof of Theorem 2.4.5

The proof of Theorem 2.4.5 is very similar to that of Theorem 2.4.1. The first step is to explicitly write out the dynamical system. In order to do so, we notice that the Lemma 2.5.1 does not depend on the training data and data-generating distribution and hence, it still holds in the case of a random covariance matrix. Therefore, we know when we input the embedding matrix $\mathbf{E}_\tau$ to the linear self-attention layer with parameter $\boldsymbol{\theta} = (\boldsymbol{W}^{KQ}, \boldsymbol{W}^{PV})$, the prediction will be

$$
\widehat{y}_{\mathsf{query}}(\mathbf{E}_\tau; \boldsymbol{\theta}) = \boldsymbol{u}^{\top}\boldsymbol{H}_\tau\boldsymbol{u},
$$

where the matrix $\boldsymbol{H}_\tau$ is defined as,

$$\boldsymbol{H}_\tau = \frac{1}{2}\boldsymbol{X}_\tau \otimes \left(\frac{\mathbf{E}_\tau \mathbf{E}_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad \boldsymbol{X}_\tau = \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{x}_{\tau,\mathsf{query}} \\ (\mathbf{x}_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}$$

and

$$\boldsymbol{u} = \mathrm{Vec}(\boldsymbol{U}) \in \mathbb{R}^{(d+1)^2}, \quad \boldsymbol{U} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $\mathbf{U}_{11} = \boldsymbol{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}, \mathbf{u}_{12} = \boldsymbol{w}_{21}^{PV} \in \mathbb{R}^{d\times 1}, \mathbf{u}_{21} = \boldsymbol{w}_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = \boldsymbol{w}_{22}^{PV} \in \mathbb{R}$ correspond to particular components of $\boldsymbol{W}^{PV}$ and $\boldsymbol{W}^{KQ}$, defined in (2.5).

Dynamical system

The next lemma gives the dynamical system when the covariance matrices in the prompts are i.i.d. sampled from some distribution. Notice that in the lemma below, we do not assume $\mathbf{\Lambda}_\tau$ are almost surely diagonal. The case when the covariance matrices are diagonal can be viewed as a special case of the following lemma.

Lemma 2.9.1. *Consider gradient flow on (2.20) with respect to $\boldsymbol{u}$ starting from an initial value that satisfies Assumption 2.3.3. We assume the covariance matrices $\mathbf{\Lambda}_\tau$ are sampled from some distribution with finite third moment and $\mathbf{\Lambda}_\tau$ are positive definite almost surely. We denote $\boldsymbol{u} = \mathrm{Vec}\,(\boldsymbol{U}) := \mathrm{Vec}\begin{pmatrix} \mathbf{U}_{11} & \mathbf{u}_{12} \\ (\mathbf{u}_{21})^\top & u_{-1} \end{pmatrix}$ and define*

$$\mathbf{\Gamma}_\tau = \left(1 + \frac{1}{N}\right)\mathbf{\Lambda}_\tau + \frac{1}{N}\,\mathrm{tr}(\mathbf{\Lambda}_\tau)\mathbf{I}_d \in \mathbb{R}^{d\times d}.$$

Then the dynamics of $\boldsymbol{U}$ follows

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t) &= -u_{-1}^2\mathbb{E}\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau\right] + u_{-1}\mathbb{E}\left[\mathbf{\Lambda}_\tau^2\right] \\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -u_{-1}\,\mathrm{tr}\,\mathbb{E}\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top\right] + \mathrm{tr}\left(\mathbb{E}\left[\mathbf{\Lambda}_\tau^2\right](\mathbf{U}_{11})^\top\right), \end{aligned} \tag{2.50}$$

and $\mathbf{u}_{12}(t) = \mathbf{0}_d, \mathbf{u}_{21}(t) = \mathbf{0}_d$ for all $t \geq 0$.

Proof. This lemma is a natural corollary of Lemma 2.5.2. Notice that Lemma 2.5.2 holds for any fixed positive definite $\mathbf{\Lambda}_\tau$. So when $\mathbf{\Lambda}_\tau$ is random, if we condition on $\mathbf{\Lambda}_\tau$, the dynamical system will be

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t) &= -u_{-1}^2\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau\right] + u_{-1}\left[\mathbf{\Lambda}_\tau^2\right] \\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -u_{-1}\,\mathrm{tr}\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top\right] + \mathrm{tr}\left(\left[\mathbf{\Lambda}_\tau^2\right](\mathbf{U}_{11})^\top\right), \end{aligned} \tag{2.51}$$

and $\mathbf{u}_{12}(t) = \mathbf{0}_d, \mathbf{u}_{21}(t) = \mathbf{0}_d$ for all $t \geq 0$. Then, we conclude by simply taking expectation over $\mathbf{\Lambda}_\tau$. □

The lemma above gives the dynamical system with general random covariance matrix. When $\mathbf{\Lambda}_\tau$ are diagonal almost surely, we can actually simplify the dynamical system above. In this case, we have the following corollary.

Corollary 2.9.2. *Under the assumptions of Lemma 2.9.1, we further assume the covariance matrix $\mathbf{\Lambda}_\tau$ to be diagonal almost surely. We denote $u_{ij}(t) \in \mathbb{R}$ as the (i,j)-th entry of $\mathbf{U}_{11}(t)$, and further denote*

$$\begin{aligned}
\mathbf{\Gamma}_i &= \mathbb{E}\left[\frac{N+1}{N}\mathbf{\Lambda}_{\tau,i}^3 + \frac{1}{N}\mathbf{\Lambda}_{\tau,i}^2 \cdot \sum_{j=1}^d \mathbf{\Lambda}_{\tau,j}\right], \\
\xi_i &= \mathbb{E}\left[\mathbf{\Lambda}_{\tau,i}^2\right], \\
\zeta_{ij} &= \mathbb{E}\left[\frac{N+1}{N}\mathbf{\Lambda}_{\tau,i}^2\mathbf{\Lambda}_{\tau,j} + \frac{1}{N}\mathbf{\Lambda}_{\tau,i}\mathbf{\Lambda}_{\tau,j} \cdot \sum_{k=1}^d \mathbf{\Lambda}_{\tau,k}\right]
\end{aligned} \tag{2.52}$$

for $i,j \in [d]$, where the expectation is over the distribution of $\mathbf{\Lambda}_\tau$. Then, the dynamical system (2.50) is equivalent to

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}u_{ii}(t) &= -\mathbf{\Gamma}_i u_{-1}^2 u_{ii} + \xi_i u_{-1} \quad \forall i \in [d], \\
\frac{\mathrm{d}}{\mathrm{d}t}u_{ij}(t) &= -\zeta_{ij} u_{-1}^2 u_{ij} \quad \forall i \neq j \in [d], \\
\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -\sum_{i=1}^d \left[\mathbf{\Gamma}_i u_{-1} u_{ii}^2\right] - \sum_{i\neq j} \zeta_{ij} u_{-1} u_{ij}^2 + \sum_{i=1}^d \left[\xi_i u_{ii}\right].
\end{aligned} \tag{2.53}$$

Proof. This is directly obtained by rewriting the equation for each entry of $\mathbf{U}_{11}$ and recalling the assumption that $\mathbf{\Lambda}_\tau$ (and hence $\mathbf{\Gamma}_\tau$) is diagonal almost surely. □

Loss function and global minima

As in the proof of Theorem 2.4.1, we can actually recover the loss function in the random covariance case, up to a constant.

Lemma 2.9.3. *The differential equations in (2.53) are equivalent to gradient flow on the loss function*

$$\begin{aligned}
\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1}) &= \mathbb{E}\mathrm{tr}\left[\frac{1}{2}u_{-1}^2 \mathbf{\Gamma}_\tau \mathbf{\Lambda}_\tau \mathbf{U}_{11} \mathbf{\Lambda}_\tau (\mathbf{U}_{11})^\top - u_{-1}\mathbf{\Lambda}_\tau^2 (\mathbf{U}_{11})^\top\right] \\
&= \frac{1}{2}\sum_{i=1}^d \left[\mathbf{\Gamma}_i u_{-1}^2 u_{ii}^2\right] + \frac{1}{2}\sum_{i\neq j} \zeta_{ij} u_{-1}^2 u_{ij}^2 - \sum_{i=1}^d \left[\xi_i u_{ii} u_{-1}\right]
\end{aligned} \tag{2.54}$$

with respect to $u_{ij} \forall i,j \in [d]$ and u_{-1}, from an initial value that satisfies Assumption 2.3.3.

Proof. This can be verified by simply taking gradient of ℓ_{rdm} to show that

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{ii} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ii}} \quad \forall i \in [d], \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{ij} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}} \quad \forall i \neq j \in [d], \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{-1} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}.$$

□

Next, we solve for the minimum of ℓ_{rdm} and give the expression for all global minima.

Lemma 2.9.4. *Let ℓ_{rdm} be the loss function in (2.54). We denote*

$$\min \ell_{\mathsf{rdm}} := \min_{\mathbf{U}_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \ell_{\mathsf{rdm}}\left(\mathbf{U}_{11}, u_{-1}\right).$$

Then, we have

$$\min \ell_{\mathsf{rdm}} = -\frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i^2}{\mathbf{\Gamma}_i} \tag{2.55}$$

and

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1}) - \min \ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^{d}\mathbf{\Gamma}_i\left(u_{ii}u_{-1} - \frac{\xi_i}{\mathbf{\Gamma}_i}\right)^2 + \frac{1}{2}\sum_{i\neq j}\zeta_{ij}u_{-1}^2u_{ij}^2. \tag{2.56}$$

Moreover, denoting u_{ij} as the (i,j)-entry of $\mathbf{U}_{11}$, all global minima of ℓ_{rdm} satisfy

$$u_{-1}\cdot u_{ij} = \mathbb{I}(i=j)\cdot\frac{\xi_i}{\mathbf{\Gamma}_i}. \tag{2.57}$$

Proof. From the definition of ℓ_{rdm}, we have

$$\ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^{d}\mathbf{\Gamma}_i\left(u_{ii}u_{-1} - \frac{\xi_i}{\mathbf{\Gamma}_i}\right)^2 + \frac{1}{2}\sum_{i\neq j}\zeta_{ij}u_{-1}^2u_{ij}^2 - \frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i^2}{\mathbf{\Gamma}_i} \geq -\frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i^2}{\mathbf{\Gamma}_i}.$$

The equation holds when $u_{ij} = 0$ for $i \neq j \in [d]$ and $u_{-1}u_{ii} = \frac{\xi_i}{\mathbf{\Gamma}_i}$ for each $i \in [d]$. This can be achieved by simply letting $u_{-1} = 1$ and $u_{ii} = \frac{\xi_i}{\mathbf{\Gamma}_i}$ for $i \in [d]$. Of course, when we replace (u_{-1}, u_{ii}) with $(cu_{-1}, c^{-1}u_{ii})$ for any constant $c \neq 0$, we can also achieve this global minimum. □

PL Inequality and global convergence

Finally, to end the proof, we prove a Polyak-Łojasiewicz Inequality on the loss function ℓ_{rdm}, and then prove global convergence. Before that, let's first prove the balanced condition of parameters will hold during the whole trajectory.

Lemma 2.9.5 (Balanced condition)**.** *Under the assumptions of Lemma 2.9.1, for any $t \geq 0$, it holds that*

$$u_{-1}^2 = \mathrm{tr}\left[\mathbf{U}_{11}(\mathbf{U}_{11})^\top\right]. \tag{2.58}$$

Proof. The proof is similar to the proof of Lemma 2.7.3. From Lemma 2.5.2, we multiply the first equation in (2.50) by $(\mathbf{U}_{11})^\top$ from the right to get

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t)\right](\mathbf{U}_{11})^\top = -u_{-1}^2\mathbb{E}\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top\right] + u_{-1}\mathbb{E}\left[\mathbf{\Lambda}_\tau^2(\mathbf{U}_{11})^\top\right].$$

Also we multiply the second equation in Lemma 2.50 by u_{-1} to obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t) = -u_{-1}^2\,\mathrm{tr}\,\mathbb{E}\left[\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top\right] + u_{-1}\,\mathrm{tr}\left(\mathbb{E}\left[\mathbf{\Lambda}_\tau^2\right](\mathbf{U}_{11})^\top\right),$$

Therefore, we have

$$\mathrm{tr}\left[\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_{11}(t)\right)(\mathbf{U}_{11}(t))^\top\right] = \left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t).$$

Taking the transpose of the equation above and adding to itself gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{tr}\left[\mathbf{U}_{11}(t)(\mathbf{U}_{11}(t))^\top\right] = \frac{\mathrm{d}}{\mathrm{d}t}\left(u_{-1}(t)^2\right).$$

Notice that from Assumption 2.3.3, we know that

$$u_{-1}(0)^2 = \sigma^2 = \sigma^2\,\mathrm{tr}\left[\mathbf{\Theta}\mathbf{\Theta}^\top\mathbf{\Theta}\mathbf{\Theta}^\top\right] = \mathrm{tr}\left[\mathbf{U}_{11}(0)(\mathbf{U}_{11}(0))^\top\right].$$

So for any time $t \geq 0$, the equation holds. □

Next, similar to the proof of Theorem 2.4.1, we prove that, as long as the initial scale is small enough, u_{-1} will be positive along the whole trajectory and can be lower bounded by a positive constant, which implies that the trajectories will be away from the saddle point at the origin.

Lemma 2.9.6. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i,j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption 2.3.3 with initial scale*

$$0 < \sigma < \sqrt{\frac{2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2}{\sqrt{d}\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right]}}, \tag{2.59}$$

then for any $t \geq 0$, it holds that

$$u_{-1}(t) > 0. \tag{2.60}$$

Proof. From the dynamics of gradient flow, we know the loss function ℓ_{rdm} is non-increasing:

$$\frac{\mathrm{d}\ell_{\mathsf{rdm}}}{\mathrm{d}t} = \sum_{i,j=1}^{d}\frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{ij}}\cdot\frac{\mathrm{d}u_{ij}}{\mathrm{d}t} + \frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{-1}}\cdot\frac{\mathrm{d}u_{-1}}{\mathrm{d}t} = -\sum_{i,j=1}^{d}\left[\frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{ij}}\right]^2 - \left[\frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{-1}}\right]^2 \leq 0.$$

Since we assume $\mathbf{U}_{11}(0) = \mathbf{\Theta}\mathbf{\Theta}^\top$, we know the loss function at $t = 0$ is

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}(0), u_{-1}(0)) = \mathbb{E}\mathrm{tr}\left[\frac{\sigma^4}{2}\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top - \sigma^2\mathbf{\Lambda}_\tau^2\mathbf{\Theta}\mathbf{\Theta}^\top\right].$$

From the property of trace, we know

$$\mathbb{E}\mathrm{tr}\left[\sigma^2\mathbf{\Lambda}_\tau^2\mathbf{\Theta}\mathbf{\Theta}^\top\right] = \sigma^2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2.$$

From Von-Neumann's trace inequality and the assumption that $\left\|\mathbf{\Theta}\mathbf{\Theta}^\top\right\|_F = 1$, we know

$$\begin{aligned}
&\mathbb{E}\mathrm{tr}\left[\frac{\sigma^4}{2}\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top\right] \le \frac{\sigma^4\sqrt{d}}{2}\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top\mathbf{\Lambda}_\tau\mathbf{\Theta}\mathbf{\Theta}^\top\right\|_F \\
\le\ &\frac{\sigma^4\sqrt{d}\left\|\mathbf{\Theta}\mathbf{\Theta}^\top\right\|_F^2}{2}\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right] = \frac{\sigma^4\sqrt{d}}{2}\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right].
\end{aligned}$$

From the assumptions on $\mathbf{\Theta}$ and $\mathbf{\Lambda}_\tau$ we know $\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta} \neq \mathbf{0}_{d\times d}$ and $\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2 > 0$. Therefore, comparing the two displays above, we know when (2.59) holds, we must have $\ell_{\mathsf{rdm}}(0) < 0$. So from the non-increasing property of the loss function, we know $\ell_{\mathsf{rdm}}(t) < 0$ for any time $t \ge 0$. Notice that when $u_{-1} = 0$, the loss function is also zero, which suggests that $u_{-1}(t) \neq 0$ for any time $t \ge 0$. Since $u_{-1}(0) > 0$ and the trajectory of u_{-1} must be continuous, we know that it stays positive at all times. □

Lemma 2.9.7. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i, j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption 2.3.3 and the initial scale satisfies (2.59). Then, for any $t \ge 0$, it holds that*

$$u_{-1}(t) \ge \sqrt{\frac{\sigma^2}{2\sqrt{d}\left\|\mathbb{E}\mathbf{\Lambda}_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right]\right]} > 0. \tag{2.61}$$

Proof. From the dynamics of gradient flow, we know ℓ_{rdm} is non-increasing (see the proof of Lemma 2.9.6). Recall the definition of the loss function:

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1}) = \mathbb{E}\mathrm{tr}\left[\frac{1}{2}u_{-1}^2\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top - u_{-1}\mathbf{\Lambda}_\tau^2(\mathbf{U}_{11})^\top\right].$$

Since $\mathbf{\Lambda}_\tau$ commutes with $\mathbf{\Gamma}_\tau$ and they are both positive definite almost surely, we know that $\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau \succeq \mathbf{0}_{d\times d}$ almost surely from Lemma 2.10.1. Again, since $\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top \succeq \mathbf{0}_{d\times d}$ almost surely, from Lemma 2.10.1 we have $\mathrm{tr}\left[\frac{1}{2}u_{-1}^2\mathbf{\Gamma}_\tau\mathbf{\Lambda}_\tau\mathbf{U}_{11}\mathbf{\Lambda}_\tau(\mathbf{U}_{11})^\top\right] \ge 0$ almost surely. Therefore, we have

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1}) \ge -\mathbb{E}\mathrm{tr}\left[u_{-1}\mathbf{\Lambda}_\tau^2(\mathbf{U}_{11})^\top\right] = -\mathrm{tr}\left[u_{-1}\left(\mathbb{E}\mathbf{\Lambda}_\tau^2\right)(\mathbf{U}_{11})^\top\right].$$

From Von Neumann's trace inequality (Lemma 2.10.3) and the fact that $u_{-1}(t) > 0$ for any $t \geq 0$ (Lemma 2.9.6), we know $\ell_{\mathsf{rdm}}(\mathbf{U}_{11}(t), u_{-1}(t)) \geq -\sqrt{d}u_{-1} \left\|\mathbb{E}\mathbf{\Lambda}_\tau^2\right\|_{op} \|\mathbf{U}_{11}\|_F$. From Lemma 2.9.5, we know $u_{-1}^2 = \mathrm{tr}(\mathbf{U}_{11}(\mathbf{U}_{11})^\top) = \|\mathbf{U}_{11}\|_F^2$. Since $u_{-1}(t) > 0$ for any time, we know actually $u_{-1}(t) = \|\mathbf{U}_{11}(t)\|_F$. So we have

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}(t), u_{-1}(t)) \geq -\sqrt{d}u_{-1}(t)^2 \left\|\mathbb{E}\mathbf{\Lambda}_\tau^2\right\|_{op}.$$

From the proof of Lemma 2.9.6, we know

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}(t), u_{-1}(t)) \leq \ell_{\mathsf{rdm}}(\mathbf{U}_{11}(0), u_{-1}(0)) \leq \frac{\sigma^4\sqrt{d}}{2}\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right] - \sigma^2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2.$$

Combine the two preceding displays above, we have

$$u_{-1}(t) \geq \sqrt{\frac{\sigma^2}{2\sqrt{d}\left\|\mathbb{E}\mathbf{\Lambda}_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right]\right]} > 0.$$

The last inequality comes from Lemma 2.9.6. □

Finally, we prove the PL Inequality, which naturally leads to the global convergence.

Lemma 2.9.8. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i, j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption 2.3.3 and the initial scale satisfies (2.59). If we denote*

$$\eta = \min\left\{\mathbf{\Gamma}_i, i \in [d]; \zeta_{ij}, i \neq j \in [d]\right\}$$

and

$$\nu := \frac{\eta\cdot\sigma^2}{2\sqrt{d}\left\|\mathbb{E}\mathbf{\Lambda}_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\mathbf{\Lambda}_\tau\mathbf{\Theta}\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\mathbf{\Gamma}_\tau\right\|_{op}\left\|\mathbf{\Lambda}_\tau\right\|_F^2\right]\right] > 0, \tag{2.62}$$

then for any $t \geq 0$, it holds that

$$\left\|\nabla\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1})\right\|_2^2 := \sum_{i,j=1}^d\left|\frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 + \left|\frac{\partial\ell_{\mathsf{rdm}}}{\partial u_{-1}}\right|^2 \geq \nu\left(\ell_{\mathsf{rdm}} - \min\ell_{\mathsf{rdm}}\right). \tag{2.63}$$

Additionally, ℓ_{rdm} converges to the global minimal value, u_{ij} and u_{-1} converge to the following limits,

$$\lim_{t\to\infty} u_{ij}(t) = \mathbb{I}(i=j)\cdot\left[\sum_{i=1}^d\frac{\xi_i^2}{\mathbf{\Gamma}_i^2}\right]^{-\frac{1}{4}}\cdot\frac{\xi_i}{\mathbf{\Gamma}_i} \quad \forall i \in [d], \quad \lim_{t\to\infty} u_{-1}(t) = \left[\sum_{i=1}^d\frac{\xi_i}{\mathbf{\Gamma}_i}\right]^{\frac{1}{4}}. \tag{2.64}$$

Translating back to the original parameterization, we have this is equivalent to

$$\lim_{t\to\infty} \boldsymbol{W}^{KQ}(t) = \begin{pmatrix} \left\| \left[\mathbb{E}\boldsymbol{\Gamma}_\tau \boldsymbol{\Lambda}_\tau^2\right]^{-1} \mathbb{E}\left[\boldsymbol{\Lambda}_\tau^2\right] \right\|_F^{-\frac{1}{2}} \cdot \left[\mathbb{E}\boldsymbol{\Gamma}_\tau \boldsymbol{\Lambda}_\tau^2\right]^{-1} \mathbb{E}\left[\boldsymbol{\Lambda}_\tau^2\right] & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{pmatrix},$$

$$\lim_{t\to\infty} \boldsymbol{W}^{PV}(t) = \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_d \\ \mathbf{0}_d^\top & \left\| \left[\mathbb{E}\boldsymbol{\Gamma}_\tau \boldsymbol{\Lambda}_\tau^2\right]^{-1} \mathbb{E}\left[\boldsymbol{\Lambda}_\tau^2\right] \right\|_F^{\frac{1}{2}} \end{pmatrix},$$

where $\boldsymbol{\Gamma}_\tau = \frac{N+1}{N}\boldsymbol{\Lambda}_\tau + \frac{1}{N}\mathrm{tr}(\boldsymbol{\Lambda}_\tau)\mathbf{I}_d \in \mathbb{R}^{d\times d}$ and $\mathbb{E}$ is over $\boldsymbol{\Lambda}_\tau$.

Proof. First, we prove the PL Inequality. From Lemma 2.9.4, we know

$$\ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1}) - \min \ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^d \boldsymbol{\Gamma}_i \left(u_{ii}u_{-1} - \frac{\xi_i}{\boldsymbol{\Gamma}_i}\right)^2 + \frac{1}{2}\sum_{i\neq j} \zeta_{ij} u_{-1}^2 u_{ij}^2,$$

where $\xi_i, \zeta_{ij}, \boldsymbol{\Gamma}_i$ are defined in (2.52). Meanwhile, we calculate the square norm of the gradient of ℓ_{rdm}:

$$\begin{aligned} \|\nabla \ell_{\mathsf{rdm}}(\mathbf{U}_{11}, u_{-1})\|_2^2 &:= \sum_{i,j=1}^d \left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 + \left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}\right|^2 \geq \sum_{i,j=1}^d \left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 \\ &= \sum_{i=1}^d \boldsymbol{\Gamma}_i^2 u_{-1}^2 \left(u_{ii}u_{-1} - \frac{\xi_i}{\boldsymbol{\Gamma}_i}\right)^2 + \sum_{i\neq j} \zeta_{ij}^2 u_{-1}^4 u_{ij}^2. \end{aligned}$$

Comparing the two displays above, in order to achieve $\|\nabla \ell_{\mathsf{rdm}}\|_2^2 \geq \nu\left(\ell_{\mathsf{rdm}} - \min \ell_{\mathsf{rdm}}\right)$, it suffices to make

$$\begin{aligned} \boldsymbol{\Gamma}_i u_{-1}(t)^2 &\geq \frac{\nu}{2} \quad \forall i \in [d], \\ \zeta_{ij} u_{-1}(t)^2 &\geq \frac{\nu}{2} \quad \forall i \neq j \in [d]. \end{aligned}$$

We define $\eta := \min\left\{\boldsymbol{\Gamma}_i, \zeta_{ij}, i \neq j \in [d]\right\}$, then it is sufficient to make

$$\eta u_{-1}(t)^2 \geq \frac{\nu}{2}.$$

From Lemma 2.9.7, we know that we can actually lower bound u_{-1} from below by a positive constant. Then, the inequality holds if we take

$$\nu := \frac{\eta \cdot \sigma^2}{2\sqrt{d}\left\|\mathbb{E}\boldsymbol{\Lambda}_\tau^2\right\|_{op}} \left[2\left\|\mathbb{E}\boldsymbol{\Lambda}_\tau \boldsymbol{\Theta}\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\boldsymbol{\Gamma}_\tau\right\|_{op}\left\|\boldsymbol{\Lambda}_\tau\right\|_F^2\right]\right] > 0.$$

Therefore, as long as we take ν as above, a PL inequality holds for ℓ_{rdm}.

With an abuse of notation, let us write $\ell_{\mathsf{rdm}}(t) = \ell_{\mathsf{rdm}}(\mathbf{U}_{11}(t), u_{-1}(t))$. Then, from the dynamics of gradient flow and the PL Inequality ((2.63)), we know

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}}\right] = -\left\|\nabla \ell_{\mathsf{rdm}}(t)\right\|_2^2 \leq -\nu\left(\ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}}\right),$$

which by Grönwall's inequality implies

$$0 \leq \ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}} \leq \exp(-\nu t)\left[\ell_{\mathsf{rdm}}(0) - \min \ell_{\mathsf{rdm}}\right] \to 0$$

when $t \to \infty$. From Lemma 2.9.4, we know

$$\sum_{i=1}^{d} \mathbf{\Gamma}_i \left(u_{ii}u_{-1} - \frac{\xi_i}{\mathbf{\Gamma}_i}\right)^2 + \sum_{i \neq j} \zeta_{ij} u_{-1}^2 u_{ij}^2 \to 0 \text{ when } t \to \infty.$$

This implies

$$\begin{aligned} u_{ii}u_{-1} &\to \frac{\xi_i}{\mathbf{\Gamma}_i} \quad \forall i \in [d], \\ u_{ij}u_{-1} &\to 0 \quad \forall i \neq j \in [d]. \end{aligned} \tag{2.65}$$

We take square of $u_{ii}(t)u_{-1}(t)$ and $u_{ij}(t)u_{-1}(t)$, then sum over all $i, j \in [d]$. Then, we get $u_{-1}^2 \sum_{i,j=1}^d u_{ij}^2 \to \sum_{i=1}^d \frac{\xi_i^2}{\mathbf{\Gamma}_i^2}$. From Lemma 2.9.5, we know for any $t \geq 0$, $u_{-1}(t)^2 = \operatorname{tr}\left(\mathbf{U}_{11}(\mathbf{U}_{11})^\top\right) = \sum_{i,j=1}^d u_{ij}^2$. So we have

$$u_{-1}(t)^4 = u_{-1}^2 \sum_{i,j=1}^{d} u_{ij}^2 \to \sum_{i=1}^{d} \frac{\xi_i^2}{\mathbf{\Gamma}_i^2},$$

which implies

$$u_{-1}(t) \to \left[\sum_{i=1}^{d} \frac{\xi_i^2}{\mathbf{\Gamma}_i^2}\right]^{\frac{1}{4}} \tag{2.66}$$

when $t \to \infty$. Combining (2.65) and (2.66), we conclude

$$u_{ij}(t) \to 0 \quad \forall i \neq j \in [d], \quad u_{ii}(t) \to \left[\sum_{i=1}^{d} \frac{\xi_i^2}{\mathbf{\Gamma}_i^2}\right]^{-\frac{1}{4}} \cdot \frac{\xi_i}{\mathbf{\Gamma}_i} \quad \forall i \in [d].$$

□

2.10 Appendix: Additional Supporting Results

Lemma 2.10.1. *(Petersen and Pedersen, 2008) We denote $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X}$ as matrices and $\boldsymbol{x}$ as vectors. Then, we have*

- $\frac{\partial \mathbf{x}^\top \mathbf{B}\mathbf{x}}{\partial \mathbf{x}} = \left(\mathbf{B} + \mathbf{B}^\top\right)\mathbf{x}$.
- $\mathrm{Vec}(\mathbf{AXB}) = \left(\mathbf{B}^\top \otimes \mathbf{A}\right)\mathrm{Vec}(\mathbf{X})$.
- $\mathrm{tr}\left(\mathbf{A}^\top \mathbf{B}\right) = \mathrm{Vec}(\mathbf{A})^\top \mathrm{Vec}(\mathbf{B})$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{X}\mathbf{B}\mathbf{X}^\top\right) = \mathbf{X}\mathbf{B}^\top + \mathbf{X}\mathbf{B}$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{A}\mathbf{X}^\top\right) = \mathbf{A}$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^\top\mathbf{C}\right) = \mathbf{A}^\top\mathbf{C}^\top\mathbf{X}\mathbf{B}^\top + \mathbf{C}\mathbf{A}\mathbf{X}\mathbf{B}$.

Lemma 2.10.2. *If* $\mathbf{X}$ *is Gaussian random vector of* d *dimension, mean zero and covariance matrix* $\mathbf{\Lambda}$*, and* $\mathbf{A} \in \mathbb{R}^{d\times d}$ *is a fixed matrix. Then*

$$\mathbb{E}\left[\mathbf{X}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{X}^\top\right] = \mathbf{\Lambda}\left(\mathbf{A} + \mathbf{A}^\top\right)\mathbf{\Lambda} + \mathrm{tr}(\mathbf{A}\mathbf{\Lambda})\mathbf{\Lambda}.$$

Proof. We denote $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_d)^\top$. Then,

$$\mathbf{X}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{X}^\top = \mathbf{X}(\mathbf{X}^\top\mathbf{A}\mathbf{X})\mathbf{X}^\top = \left(\sum_{i,j=1}^{d} \mathbf{A}_{ij}\mathbf{X}_i\mathbf{X}_j\right)\mathbf{X}\mathbf{X}^\top.$$

So we know $(\mathbf{X}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{X}^\top)_{k,l} = \left(\sum_{i,j=1}^{d} \mathbf{A}_{ij}\mathbf{X}_i\mathbf{X}_j\right)\mathbf{X}_k\mathbf{X}_l$. From Isserlis' Theorem in probability theory (Theorem 1.1 in Michalowicz et al. (2009), originally proposed in Wick (1950)), we know for any $i, j, k, l \in [d]$, it holds that

$$\mathbb{E}\left[\mathbf{X}_i\mathbf{X}_j\mathbf{X}_k\mathbf{X}_l\right] = \mathbf{\Lambda}_{ij}\mathbf{\Lambda}_{kl} + \mathbf{\Lambda}_{ik}\mathbf{\Lambda}_{jl} + \mathbf{\Lambda}_{il}\mathbf{\Lambda}_{jk}.$$

Then, we have for any fixed $k, l \in [d]$,

$$\begin{aligned}\mathbb{E}(\mathbf{X}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{X}^\top)_{k,l} &= \sum_{i,j=1}^{d} \mathbf{A}_{ij}\mathbf{\Lambda}_{ij}\mathbf{\Lambda}_{kl} + \mathbf{A}_{ij}\mathbf{\Lambda}_{ik}\mathbf{\Lambda}_{jl} + \mathbf{A}_{ij}\mathbf{\Lambda}_{il}\mathbf{\Lambda}_{jk} \\ &= \mathrm{tr}(\mathbf{A}\mathbf{\Lambda})\mathbf{\Lambda}_{kl} + \mathbf{\Lambda}_k^\top(\mathbf{A} + \mathbf{A}^\top)\mathbf{\Lambda}_l.\end{aligned}$$

Therefore, we know

$$\mathbb{E}(\mathbf{X}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{X}^\top) = \mathbf{\Lambda}\left(\mathbf{A} + \mathbf{A}^\top\right)\mathbf{\Lambda} + \mathrm{tr}(\mathbf{A}\mathbf{\Lambda})\mathbf{\Lambda}.$$

□

Lemma 2.10.3 (Von-Neumann's Trace Inequality)**.** *Let* $\boldsymbol{U}, \mathbf{V} \in \mathbb{R}^{d\times n}$ *with* $d \le n$*. We have*

$$\operatorname{tr}\left(\boldsymbol{U}^\top \mathbf{V}\right) \le \sum_{i=1}^{d} \sigma_i(\boldsymbol{U})\sigma_i(\mathbf{V}) \le \|\boldsymbol{U}\|_{\mathrm{op}} \times \sum_{i=1}^{d} \sigma_i(\mathbf{V}) \le \sqrt{d} \cdot \|\boldsymbol{U}\|_{\mathrm{op}} \|\mathbf{V}\|_F$$

where $\sigma_1(\mathbf{X}) \ge \sigma_2(\mathbf{X}) \ge \cdots \ge \sigma_d(\mathbf{X})$ *are the ordered singular values of* $\mathbf{X} \in \mathbb{R}^{d\times n}$*.*

Lemma 2.10.4 ((Meenakshi and Rajian, 1999))**.** *For any two positive semidefinite matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times d}$*, we have*

- $\operatorname{tr}[\mathbf{AB}] \ge 0$.
- $\mathbf{AB} \succeq 0$ *if and only if* $\mathbf{A}$ *and* $\mathbf{B}$ *commute.*

2.11 Appendix: Experiment Details

In this section, we provide more details for the experiment in Figure 2.1. Our experimental setup is based on the codebase provided by Garg et al. (2022), with a modification that allows for the possibility that the covariate distribution changes across prompts. We use the standard GPT2 architecture with 256 embedding size, 12 layers and 8 heads (Radford et al., 2018) as implemented by HuggingFace (Wolf et al., 2020). For the GPT2 models, we use the embedding method proposed by Garg et al. (2022), where instead of concatenating x and y into a single token, they are treated as separate tokens. It is also worth noting that the training objective function for the GPT2 model is different from those we consider for the linear self-attention network: for the GPT2 model, the objective function is the average over the full length of the context sequence (predictions for each $\mathbf{x}_i$ using $(\mathbf{x}_k, y_k)_{k<i}$), while in our setting the objective function is only for the final query point. However, in the figure, for both GPT2 and the linear self-attention model the error plotted corresponds to the error for predicting the final query point.

In all experiments, covariates are sampled from a mean-zero Gaussian in $d = 20$ dimensions with either fixed or random covariance matrix. For the fixed covariance case, we fix the covariance matrix to be the identity; for the random case, the covariance matrices are restricted to be diagonal and all diagonal entries are i.i.d. sampled from the standard exponential distribution. The linear weights in all tasks are i.i.d. sampled from a standard Gaussian distribution and also independently of all covariates. We trained the model for 500000 steps using Adam (Kingma and Ba, 2014) with a batch size of 64 and learning rate of 0.0001. We use the same curriculum strategy of Garg et al. (2022) for acceleration.

For testing the trained model, we used ordinary least squares as a baseline which is optimal for noiseless linear regression tasks. For prompts at test time, covariates are sampled i.i.d. from a mean-zero Gaussian distribution. For the fixed-covariance evaluation, the covariance is the identity matrix. In the random-covariance evaluation, the covariance is a random

diagonal matrix with diagonal entries sampled from the standard exponential distribution, multiplied by a scaling coefficient $c \in \{1, 4, 9\}$, i.e. for each task τ, the covariance matrix in the random case is

$$\mathbf{\Lambda}_\tau = c \cdot \mathrm{diag}\left(\mathbf{\Lambda}_{\tau,1}, ..., \mathbf{\Lambda}_{\tau,d}\right)$$

where $\mathbf{\Lambda}_{\tau,i} \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$ for any τ and $i \in [d]$. The plots in Figure 2.1 show the error averaged over 64^2 prompts, where we sample 64 covariance matrices for each curve and 64 prompts for each covariance matrix. We compute 90% confidence interval over 1000 bootstrap trials for each test.

Chapter 3

In-Context Learning of a Linear Transformer Block

3.1 Introduction

A recent line of work quantifies ICL in tractable statistical learning setups such as linear regression with a Gaussian prior (see Garg et al., 2022; Akyürek et al., 2022 and references thereafter). Specifically, an ICL model takes a sequence $(\mathbf{x}_1, y_1, ..., \mathbf{x}_M, y_M, \mathbf{x}_{\mathsf{query}})$ as input and outputs a prediction of y_{query}, where $(\mathbf{x}_i, y_i)_{i=1}^M$ and $(\mathbf{x}_{\mathsf{query}}, y_{\mathsf{query}})$ are independent samples from an unknown task-specific distribution (where the task admits a prior distribution). The ICL model is pre-trained by fitting many empirical observations of such sequence-label pairs. Experiments show that Transformers can achieve an ICL risk close to that achieved by Bayes optimal estimators in many statistical learning tasks (Garg et al., 2022; Akyürek et al., 2022). Chapter 2 also analyzed in-context learning in a transformer consisting of a single linear self-attention layer and characterized the solutions found by gradient-flow training.

A natural next question is the role of the other major component of transformer blocks—the MLP—in enabling or improving in-context adaptation. In this chapter, we study in-context learning in a linear transformer block (LTB) that combines linear attention with a linear MLP, and we characterize when and why the MLP component yields a provable advantage. For ICL of linear regression with a Gaussian prior, the data is generated as

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0},\ \mathbf{H}), \quad y \mid \widetilde{\boldsymbol{\beta}}, \mathbf{x} \sim \mathcal{N}\big(\widetilde{\boldsymbol{\beta}}^\top \mathbf{x},\ \sigma^2\big),$$

where $\widetilde{\boldsymbol{\beta}}$ is a task parameter that satisfies a Gaussian prior, $\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}(\mathbf{0},\ \psi^2\mathbf{I})$. In this setup, Garg et al., 2022; Akyürek et al., 2022 showed in experiments that Transformers can achieve nearly optimal ICL by matching the performance of the Bayes optimal estimator, that is, an optimally tuned ridge regression. Besides, Von Oswald et al., 2023 showed by construction that a single *linear self-attention* (LSA) can implement *one-step gradient descent with zero initialization* (GD-0), offering an insight into the ICL mechanism of the Transformer. Later, theoretical works showed that optimal LSA models effectively correspond to GD-0 models

(Ahn et al., 2023c), trained LSA models converge to GD-0 models (See Chapter 2), and GD-0 models (hence LSA models) can provably achieve nearly Bayes optimal ICL in certain regimes (Wu et al., 2024b).

Our contributions. In this chapter, we consider ICL in a more general setup, that is, linear regression with a Gaussian prior and a *non-zero mean* (that is, $\mathbb{E}[\widetilde{\boldsymbol{\beta}}] = \boldsymbol{\beta}^* \neq 0$). The non-zero mean in the task prior captures a common scenario where tasks share a signal. In this setting, we show that the LSA models considered in prior works (Ahn et al., 2023c; Wu et al., 2024b) and in Chapter 2 incur an irreducible additive approximation error. Furthermore, we show that this approximation error is mitigated by considering a *linear Transformer block* (LTB) that combines a linear multi-layer perceptron (MLP) component and an LSA component. Our results highlight the important role of MLP layers in Transformers in reducing the approximation error, and they suggest that theories about LSA (Ahn et al., 2023c; Wu et al., 2024b) do not fully explain the power of Transformers. Motivated by this understanding, we investigate LTB in depth and obtain the following additional results:

- We show that LTB can implement *one-step gradient descent with learnable initialization*, referred to by us as GD-$\boldsymbol{\beta}$. Additionally, we show that every optimal LTB estimator that minimizes the in-class ICL risk is *effectively* a GD-$\boldsymbol{\beta}$ estimator.
- Moreover, we show that the optimal GD-$\boldsymbol{\beta}$, hence also the optimal LTB, nearly matches the performance of the Bayes optimal estimator for linear regression with a Gaussian prior and a non-zero mean, provided that the signal-to-noise ratio is upper bounded. These two results together suggest that LTB performs nearly optimal ICL by implementing GD-$\boldsymbol{\beta}$.
- Finally, we show that the GD-$\boldsymbol{\beta}$ estimator can be efficiently optimized by gradient descent with an infinitesimal stepsize (that is, gradient flow) under the population ICL risk, despite the non-convexity of the objective.

Chapter organization. The remaining chapter is organized as follows. We conclude this section by introducing a set of notations. We then discuss related papers in Section 3.2. In Section 3.3, we set up our ICL problems and define LTB and LSA models mathematically. In Section 3.4, we show a positive approximation error gap between LSA and LTB, which highlights the importance of the MLP component and motivates our subsequent efforts to study LTB. In Section 3.5, we connect LSA estimators to GD-$\boldsymbol{\beta}$ estimators that are more interpretable for ICL of linear regression. In Section 3.6, we study the in-context learning and training of GD-$\boldsymbol{\beta}$ estimators. We conclude this chapter and discuss future directions in Section 3.7.

Notations. We use lowercase bold letters to denote vectors and uppercase bold letters to denote matrices and tensors. For a vector $\mathbf{x}$ and a positive semi-definite (PSD) matrix $\mathbf{A}$, we write $\|\mathbf{x}\|_{\mathbf{A}} := \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$. We write $\langle \cdot, \cdot \rangle$ for the inner product, which is defined as $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^\top \mathbf{y}$ for vectors and $\langle \mathbf{A}, \mathbf{B} \rangle := \mathrm{tr}(\mathbf{A}\mathbf{B}^\top)$ for matrices. We write $\mathbf{A}[i]$ as the i-th row of the matrix $\mathbf{A}$, $\mathbf{A}_{m,n}$ as the (m,n)-th entry, and $\mathbf{A}_{-1,-1}$ as the right-bottom entry. We write $\mathbf{0}_n, \mathbf{0}_{m\times n}, \mathbf{I}_n$ for the zero vector, zero matrix and identity matrix, respectively. We

denote the Kronecker product as $\otimes$. For two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \otimes \mathbf{B}$ is a linear mapping which operates as $(\mathbf{A} \otimes \mathbf{B}) \circ \mathbf{C} = \mathbf{B}\mathbf{C}\mathbf{A}^\top$. For a positive semi-definite matrix $\mathbf{A}$, we write $\mathbf{A}^{\frac{1}{2}}$ as its principal square root, which is the unique positive semi-definite matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B} = \mathbf{B}\mathbf{B}^\top = \mathbf{A}$. We also write $\mathbf{A}^{-\frac{1}{2}} = (\mathbf{A}^{\frac{1}{2}})^+$, where $(\cdot)^+$ denotes the Moore Penrose pseudo-inverse. For two sets A, B we write $A + B$ for the Minkowski sum, which is defined as $\{a + b : a \in A, b \in B\}$. We also write $a + B = \{a\} + B$ for an element a and a set B. We write $\mathsf{null}(\cdot)$ for the null set of a matrix or a tensor.

3.2 Related Works

Empirical results for ICL in controlled settings. The work by Garg et al., 2022 first considered ICL in controlled statistical learning setups. For noiseless linear regression, Garg et al., 2022 showed in experiments that Transformers match the performance of the optimal estimator (that is, *Ordinary Least Square*). Subsequent works by Akyürek et al., 2022; Li et al., 2023b extended their result to noisy linear regression with a Gaussian prior and showed that Transformers can match the performance of the Bayesian optimal estimator (that is, an optimally tuned ridge regression). Besides, Raventós et al., 2023 showed that the above holds even when Transformers are pretrained on a limited number of linear regression tasks. These papers only considered a Gaussian prior with a zero mean. In contrast, we consider a Gaussian prior with a non-zero mean. Our setup better captures the common scenarios where the tasks share a signal.

The empirical investigation of ICL in tractable statistical learning setups goes beyond linear regression settings. For more examples, researchers empirically studied the ICL ability of Transformers for decision trees (see Garg et al., 2022, they also considered two-layer networks), algorithm selection (Bai et al., 2023), linear mixture models (Pathak et al., 2024), learning Fourier series (Panwar et al., 2023a), discrete boolean functions (Bhattamishra et al., 2024), representation learning (Fu et al., 2023b; Guo et al., 2024), and reinforcement learning (Lee et al., 2023; Lin et al., 2024). Among all these settings, Transformers can either compete with the Bayes optimal estimators or expert-designed strong benchmarks. These works are not directly comparable with our chapter.

Transformer implements gradient descent. A line of work interpreted the ICL of Transformers by their abilities to implement *gradient descent* (GD) (Von Oswald et al., 2023; Akyürek et al., 2022; Dai et al., 2023; Ahn et al., 2023c; Zhang et al., 2024c; Bai et al., 2023; Wu et al., 2024b). In experiments, Von Oswald et al., 2023; Dai et al., 2023 showed that (multi-layer) Transformer outputs are close to (multi-step) GD outputs. When specialized to linear regression tasks, Von Oswald et al., 2023 constructed a single *linear self-attention* (LSA) that implements *one-step gradient descent with zero initialization* (GD-0). Subsequently, Ahn et al., 2023c showed that optimal LSA models effectively correspond to GD-0 models, Zhang et al., 2024c proved that trained LSA models converge to GD-0 models (the results are also covered in Chapter 2), Wu et al., 2024b showed that GD-0 models (hence LSA models) can provably achieve nearly Bayes optimal ICL in certain regimes and provided a sharp task

complexity analysis of the pre-training. These papers focused on LSA models. Instead, we consider a linear Transformer block that also utilizes the MLP layer.

From an approximation theory perspective, Akyürek et al., 2022; Bai et al., 2023 showed that Transformers can implement multi-step GD under general losses. In comparison, we consider a limited setting of linear regression and show the LSA models can implement one-step GD with learnable initialization (GD-$\boldsymbol{\beta}$), moreover, every optimal LTB model is effectively an GD-$\boldsymbol{\beta}$ model. Both of our constructions utilize MLP layers in Transformers, highlighting its importance in reducing approximation error. Different from their results, we also show a negative approximation result that reveals the limitation of LSA models.

3.3 Preliminaries

Model input. We use $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$ to denote a feature vector and its label, respectively. Throughout the chapter, we assume a fixed number of context examples, denoted by $M > 0$. We denote the context examples by $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{M\times d} \times \mathbb{R}^M$, where each row represents a context example, denoted by $(\mathbf{x}_i^\top, y_i)$, $i = 1, \ldots, M$. To formalize an ICL problem, the input of a model is a *token matrix* given by (Ahn et al., 2023c) and by Chapter 2:

$$\mathbf{E} := \begin{pmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{y}^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(M+1)}. \tag{3.1}$$

The output of a model corresponds to a prediction of y.

A Transformer block. Modern large language models are often made by stacking basic Transformer blocks (see, for example, Vaswani et al., 2017). A basic Transformer block consists of a *self-attention* layer and a *multi-layer perceptron* (MLP) layer (Vaswani et al., 2017), $\mathbf{E} \mapsto \mathsf{MLP}\left[\mathsf{Attn}\left(\mathbf{E}\right)\right]$. Here, the MLP layer is defined as $\mathsf{MLP}\left(\mathbf{E}\right) := \mathbf{W}_2^\top \mathrm{ReLU}\left(\mathbf{W}_1\mathbf{E}\right)$, where $\mathrm{ReLU}(\cdot)$ refers to the entrywise *rectified linear unit* (ReLU) activation function, and $\mathbf{W}_1,\ \mathbf{W}_2 \in \mathbb{R}^{d_f\times(d+1)}$ are two weight matrices. The self-attention layer (we focus on the single-head version in this chapter) is defined as $\mathsf{Attn}(\mathbf{E}) := \mathbf{E} + (\boldsymbol{W}^P)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \cdot \mathrm{softmax}\Big((\boldsymbol{W}^K\mathbf{E})^\top \boldsymbol{W}^Q\mathbf{E}\Big)$, where $\mathrm{softmax}(\cdot)$ refers to the row-wise softmax operator, $\boldsymbol{W}^K,\ \boldsymbol{W}^Q \in \mathbb{R}^{d_k\times(d+1)}$ are the key and query matrices, respectively, $\boldsymbol{W}^P,\ \boldsymbol{W}^V \in \mathbb{R}^{d_v\times(d+1)}$ are the projection and value matrices, respectively, and $\mathbf{M}$ is a fixed masking matrix given by

$$\mathbf{M} := \begin{pmatrix} \mathbf{I}_M & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{(M+1)\times(M+1)}. \tag{3.2}$$

This mask matrix is included to reflect the asymmetric structure of a prompt since the label of query input $\mathbf{x}$ is not included in the token matrix (Ahn et al., 2023c; Mahankali et al., 2024). In the above formulation, d_k, d_v, d_f are three hyperparameters controlling the key size, value size, and width of the MLP layer, respectively. In the single-head case, it is common to set $d_k = d_v = d + 1$ and $d_f = 4(d + 1)$, where $d + 1$ corresponds to the embedding size (see, for example, Vaswani et al., 2017). Our formulation of a basic transformer block ignores all

bias parameters and some popular techniques (such as layer normalization and dropout) to focus solely on the benefits provided by the model structure.

A linear Transformer block. To facilitate theoretical analysis, we ignore the non-linearities in the Transformer block (specifically, softmax and ReLU) and work with a *linear Transformer block* (LTB) defined as

$$f_{\mathsf{LTB}} : \mathbb{R}^{(d+1)\times(M+1)} \to \mathbb{R}, \quad \mathbf{E} \mapsto \left[\mathbf{W}_2^\top \mathbf{W}_1 \left(\mathbf{E} + \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right)\right]_{-1,-1}, \tag{3.3}$$

where $\mathbf{E}$ is the token matrix given by (3.1), $[\,\cdot\,]_{-1,-1}$ refers to the bottom right entry of a matrix, $\mathbf{M}$ is the fixed masking matrix given by (3.2). Here, the trainable parameters are $\boldsymbol{W}^P, \boldsymbol{W}^V \in \mathbb{R}^{d_v \times (d+1)}$, $\boldsymbol{W}^K, \boldsymbol{W}^Q \in \mathbb{R}^{d_k \times (d+1)}$, and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_f \times (d+1)}$. We use the bottom right entry of the transformed token matrix as the model output to form a prediction of the label y. The $1/M$ factor is a normalization factor in linear attention and can be absorbed into trainable parameters. Finally, we denote the hypothesis class formed by LTB models as $\mathcal{F}_{\mathsf{LTB}} := \{f_{\mathsf{LTB}} : \ \boldsymbol{W}^K, \boldsymbol{W}^Q, \boldsymbol{W}^V, \boldsymbol{W}^P, \mathbf{W}_1, \mathbf{W}_2, d_k \geq d, \ d_v \geq d+1, \ d_f \geq 1\}$, where f_{LTB} is defined in (3.3).

A linear self-attention. We will also consider a *linear self-attention* (LSA) defined as

$$f_{\mathsf{LSA}} : \mathbb{R}^{(d+1)\times(M+1)} \to \mathbb{R}, \quad \mathbf{E} \mapsto \left[\mathbf{E} + \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right]_{-1,-1}, \tag{3.4}$$

where $\boldsymbol{W}^K, \boldsymbol{W}^Q, \boldsymbol{W}^P, \boldsymbol{W}^V$ are trainable parameters. An LSA model can be viewed as an LTB model without the MLP layer (setting $d_f = d+1$ and $\mathbf{W}_1 = \mathbf{W}_2 = \mathbf{I}$). We remark that, unlike LTB, the residual connection in LSA plays no role because the bottom right entry of the prompt $\mathbf{E}$ is zero (see (3.1)). A variant of the LSA model has been studied by Ahn et al., 2023c; Wu et al., 2024b and Chapter 2, where $\left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q$ and $\left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V$ are respectively merged into one matrix parameter. Similarly, we denote the hypothesis class formed by LSA models as $\mathcal{F}_{\mathsf{LSA}} := \{f_{\mathsf{LSA}} : \boldsymbol{W}^K, \boldsymbol{W}^Q, \boldsymbol{W}^V, \boldsymbol{W}^P, d_k \geq d, d_v \geq 1\}$, where f_{LSA} is defined in (3.4).

Linear regression tasks with a shared signal. Assume that data and context examples are generated as follows.

Assumption 3.3.1 (Distributional conditions)**.** Assume that $(\mathbf{X}, \mathbf{y}, \mathbf{x}, y)$ are generated by:

- First, a task parameter is independently generated by $\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^*, \boldsymbol{\Psi})$.
- The feature vectors are independently generated by $\mathbf{x}, \mathbf{x}_1, \ldots \mathbf{x}_M \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{H})$.
- Then, the labels are generated by $y = \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x} \rangle + \varepsilon, \quad y_i = \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x}_i \rangle + \varepsilon_i, \ i = 1, \ldots, M$, where ε and ε_i's are independently generated by $\varepsilon, \varepsilon_1, \ldots, \varepsilon_M \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2)$.

Here, $\sigma^2 \geq 0$, $\mathbf{H} \succeq \mathbf{0}$, $\mathbf{\Psi} \succeq \mathbf{0}$, and $\boldsymbol{\beta}^* \in \mathbb{R}^d$ are fixed but unknown quantities that govern the data distribution. We denote $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_M)^\top$.

We emphasize the importance of the mean of the task parameter $\boldsymbol{\beta}^*$ in Assumption 3.3.1. A non-zero $\boldsymbol{\beta}^*$ represents a shared signal across tasks, which is arguably common in practice. This assumption is implicitly used in Finn et al., 2017 where they assumed task parameters are close to a meta parameter. In comparison, the prior works for ICL of linear regression (Ahn et al., 2023c; Wu et al., 2024b) and Chapter 2 only considered a special case where $\boldsymbol{\beta}^* = 0$. In this special case, they showed that a single LSA layer can achieve nearly optimal ICL by approximating one-step gradient descent from zero initialization (GD-0). In more general cases where $\boldsymbol{\beta}^*$ is non-zero, we will show in Section 3.4 that LSA is insufficient to learn the shared signal and must incur an irreducible approximation error compared to LTB models. This sets our results apart from the prior works.

ICL risk. We measure the ICL risk of a model f by the mean squared error,

$$\mathcal{R}(f) := \mathbb{E}(f(\mathbf{E}) - y)^2, \tag{3.5}$$

where $\mathbf{E}$ is defined in (3.1) and the expectation is over $\mathbf{E}$ (equivalent to over $\mathbf{X}$, $\mathbf{y}$, and $\mathbf{x}$) and y.

3.4 Benefits of the MLP Component

Our first main result separates the approximation abilities of the LTB and LSA models for ICL. Recall that an LSA model can be viewed as a special case of the LTB model with $\mathbf{W}_2 = \mathbf{W}_1 = \mathbf{I}$. So the best ICL risk achieved by LTB models is no larger than the best ICL risk achieved by LSA models. However, our next theorem shows a strictly positive gap between the best ICL risks achieved by those two model classes. This result highlights the benefits of the MLP layer for reducing approximation error in Transformer.

Theorem 3.4.1 (Approximation gap)**.** *Consider the ICL risk defined by (3.5) and the two hypothesis classes $\mathcal{F}_{\mathsf{LTB}}$ and $\mathcal{F}_{\mathsf{LSA}}$. Suppose that Assumption 3.3.1 holds. Then we have*

- $\inf_{f \in \mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}\,(f)$ *is independent of* $\boldsymbol{\beta}^*$.
- $\inf_{f \in \mathcal{F}_{\mathsf{LSA}}} \mathsf{Risk}\,(f)$ *is a function of* $\boldsymbol{\beta}^*$. *Moreover,*

$$\inf_{f \in \mathcal{F}_{\mathsf{LSA}}} \mathsf{Risk}\,(f) - \inf_{f \in \mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}\,(f) \geq \max\left\{ \frac{2}{3\,(M+1)}, \frac{\left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right)^2}{(M+1)^2 \operatorname{tr}\left((\mathbf{H}\mathbf{\Psi})^2\right)} \right\} \|\boldsymbol{\beta}^*\|_{\mathbf{H}}^2 .$$

The proof of Theorem 3.4.1 is deferred to Section 3.9. Theorem 3.4.1 reveals a gap in terms of the approximation abilities between the hypothesis set of LTB models and that of LSA models. Specifically, the best ICL performance achieved by LTB models is independent of $\boldsymbol{\beta}^*$, while the best ICL performance achieved by LSA models is sensitive to the norm of

$\boldsymbol{\beta}^*$. In particular, when $\|\boldsymbol{\beta}^*\|_{\mathbf{H}}^2 = \Omega(M)$, the best ICL risk of the former is smaller than that of the latter by at least a *constant* additive term. So when $\boldsymbol{\beta}^*$ is large, the hypothesis class formed by LSA models is restricted in its ability to perform effective ICL. We will show in Section 3.5 that the hypothesis class formed by LTB models can achieve nearly optimal ICL in this case.

We also remark that the $\Theta(1/M)$ factor in the lower bound in Theorem 3.4.1 is not improvable. This is because LSA models can implement a one-step GD algorithm that is *consistent* for linear regression tasks (that is, the risk converges to the Bayes risk as the number of context examples goes to infinity), with an excess risk bound of $\Theta(1/M)$ (see, e.g., Wu et al. (2024b); see also Chapter 2). So the approximation error gap is at most $\Theta(1/M)$. Nonetheless, the $\Omega(\|\boldsymbol{\beta}^*\|_{\mathbf{H}}^2/M)$ approximation gap between LSA models and LTB models shown by Theorem 3.4.1 suggests that $\boldsymbol{\beta}^*$ is not learnable by LSA models during pre-training. In contrast, we will show in Section 3.6 that LTB models can learn $\boldsymbol{\beta}^*$ during pre-training.

We emphasize that the ability of LTB to learn non-zero mean is a joint effect of an MLP component and a skip connection. Note that LSA also has a skip connection, but its skip connection is inactive. In comparison, the MLP component in LTB activates the skip connection. Therefore, we attribute the ability to learn non-zero mean to the MLP component, which is the only difference between LTB and LSA. Nonetheless, one can attribute the ability to learn non-zero mean to the skip connection: without a skip connection, LTB reduces to LSA with a potential rank constraint on the parameter, which cannot learn non-zero mean as we have proved. The above two explanations take different perspectives to interpret the same phenomenon. Finally, we remark that Theorem 3.4.1 holds even when $\mathbf{H}$ and $\boldsymbol{\Psi}$ in Assumption 3.3.1 are not full rank.

Does scratchpad help? We have demonstrated that employing a single-layer LSA introduces an additional approximation error in the in-context learning problem for the linear regression task defined in Assumption 3.3.1. Nonetheless, are there alternative structures, apart from MLPs, that could potentially reduce this approximation error? One plausible strategy is to include a "scratchpad" in the input token. Specifically, we construct the token matrix as follows:

$$\mathbf{E} := \begin{pmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{1}_M^\top & 1 \\ \mathbf{y}^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+2)\times(M+1)} \tag{3.6}$$

which we then input into the LSA layer. We discuss the limitation of this scheme in Section 3.17. Notably, this method does not successfully recover the GD-$\boldsymbol{\beta}$ estimator defined in Section 3.5. We leave it as future work to see whether the token matrix with scratchpad could implement other types of estimators that more effectively address the linear regression tasks defined in Assumption 3.3.1, as well as whether additional structures could help alleviate this approximation error.

Experiments on GPT2. Theorem 3.4.1 shows the importance of the MLP component in LSA models for reducing approximation error. We also empirically validate this result by training a more complex GPT2 model (Garg et al., 2022) for the ICL tasks specified by Assumption 3.3.1. In the experiments, we use a GPT2-small model (with or without the

Table 3.1: Losses of GPT2 with or without MLP component for linear regression with a shared signal.

Model	GPT2	GPT2-noMLP
$\boldsymbol{\beta}^* = 0 \times \mathbf{1}$	0.003	0.024
$\boldsymbol{\beta}^* = 10 \times \mathbf{1}$	0.013	8.871

MLP component) with 6 layers and 4 heads in each layer. The experiments follow the setting in Garg et al., 2022, except that we train the model using a token matrix defined in (3.1). We considere two ICL settings, which instantiates Assumption 3.3.1 with $\boldsymbol{\beta}^* = (0, 0, ..., 0)^\top$ and $\boldsymbol{\beta}^* = (10, 10, ..., 10)^\top$, respectively. We set $d = 20, M = 40, \sigma = 0, \boldsymbol{\Psi} = \mathbf{H} = \mathbf{I}_d$ in both settings. More experimental details are in Section 3.16. In each setting, we train and test the model using the same data distribution. The experimental results are presented in Table 3.1. From Table 3.1,

we observe that both models (with or without the MLP component) achieve a nearly zero loss when the task mean is zero. However, when the task mean is set away from zero, the GPT2 model with MLP component still performs relatively well while the GPT2 model without MLP component incurs a significantly larger loss. These empirical observations are consistent with our Theorem 3.4.1, indicating the benefits of MLP layers in reducing approximation error for ICL of linear regression with a shared signal.

Experiments on LSA and LTB. We trained both the LSA and LTB layers for $\boldsymbol{\beta}^* = (1, 1, ..., 1)^\top$, and found that the trained LTB layer consistently achieved a significantly lower ICL risk than the LSA layer (see Figure 3.1). In these experiments, we adhered strictly to the previously defined LSA and LTB structures, setting the parameters as follows: $d = 5$, $M = 5$, $\sigma = 0$, and $\boldsymbol{\Psi} = \mathbf{H} = \mathbf{I}_d$. At each training step, we sampled $B = 128$ new linear regression tasks.

In this part, we have shown that LSA models, the primary focus in previous ICL theory literature (see, e.g., Ahn et al., 2023c; Wu et al., 2024b, Chapter 2 and references therein), are not sufficiently expressive for ICL of linear regression with a shared signal. In what follows, we will show LTB models are sufficient for this ICL problem.

3.5 LTB Implements One-Step GD with Learnable Initialization

To understand the expressive power of the LTB models, we build a connection between $\mathcal{F}_{\mathsf{LTB}}$ and its subset of models which we call *one-step GD with learnable initialization* (GD-$\boldsymbol{\beta}$). We will first introduce GD-$\boldsymbol{\beta}$ models and then show that the best LTB models that minimize the ICL risk effectively belong to GD-$\boldsymbol{\beta}$ models.

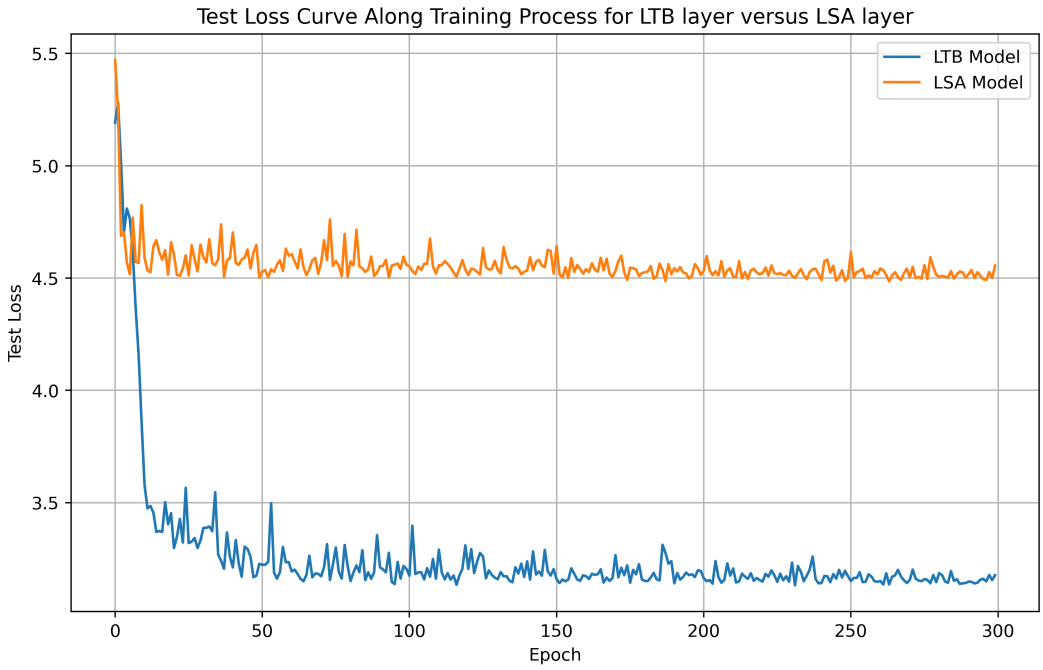

Figure 3.1: The test loss along the training process for LTB and LSA layer.

The GD-$\boldsymbol{\beta}$ models. A GD-$\boldsymbol{\beta}$ model is defined as

$$f_{\text{GD-}\boldsymbol{\beta}} : \mathbb{R}^{(d+1)\times(M+1)} \to \mathbb{R}, \quad \mathbf{E} \mapsto \left\langle \boldsymbol{\beta} - \boldsymbol{\Gamma} \cdot \frac{1}{M} \mathbf{X}^\top \left(\mathbf{X}\boldsymbol{\beta} - \mathbf{y} \right), \mathbf{x} \right\rangle, \tag{3.7}$$

where $\mathbf{E}$ is the token matrix given by (3.1), $\boldsymbol{\Gamma} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\beta} \in \mathbb{R}^d$ are trainable parameters. Similarly, we define the function class formed by GD-$\boldsymbol{\beta}$ models as $\mathcal{F}_{\text{GD-}\boldsymbol{\beta}} := \{f_{\text{GD-}\boldsymbol{\beta}} : \ \boldsymbol{\beta} \in \mathbb{R}^d, \ \boldsymbol{\Gamma} \in \mathbb{R}^{d\times d}\}$. A GD-$\boldsymbol{\beta}$ model computes a parameter that fits the context examples $(\mathbf{X}, \mathbf{y})$ and uses that parameter to make a linear prediction of label y on feature $\mathbf{x}$. More specifically, the first step is by using one gradient descent step with a *matrix stepsize* $\boldsymbol{\Gamma}$ and an *initialization* $\boldsymbol{\beta}$ on the least square objective formed by context examples $(\mathbf{X}, \mathbf{y})$.

LTB implements GD-$\boldsymbol{\beta}$. A GD-$\boldsymbol{\beta}$ model (3.7) is a special case of an LTB model (3.3) by setting

$$\mathbf{W}_2^\top \mathbf{W}_1 = \begin{pmatrix} * & * \\ \boldsymbol{\beta}^\top & 1 \end{pmatrix}, \quad \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V = \begin{pmatrix} -\mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d^\top & 1 \end{pmatrix}, \quad \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q = \begin{pmatrix} \boldsymbol{\Gamma} & * \\ \mathbf{0}_d^\top & * \end{pmatrix},$$

where $*$ denotes the entries that do not affect the model output (hence can be set to anything). Note that $\boldsymbol{\Gamma}$ and $\boldsymbol{\beta}$ are *free* provided that $d_v \geq d+1$, $d_k \geq d$ and $d_f \geq 1$ (as required in $\mathcal{F}_{\text{LTB}}$). In sum, we have proved the following lemma showing that the set of GD-$\boldsymbol{\beta}$ models belongs to the set of LTB models.

Lemma 3.5.1. *We have* $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}} \subseteq \mathcal{F}_{\mathsf{LTB}}$. *Therefore,* $\inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \geq \inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}(f)$.

Optimal GD-$\boldsymbol{\beta}$ models for ICL. We now consider $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ and its optimal ICL risk. We have the following theorem that computes the globally minimal ICL risk over $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ and specifies the sufficient and necessary conditions for a global minimizer. The proof is deferred to Appendix 3.10.

Theorem 3.5.2 (Optimal GD-$\boldsymbol{\beta}$ models)**.** *Consider the ICL risk defined by* (3.5)*. Suppose that Assumption 3.3.1 holds. Then we have*

- *The minimal ICL risk of* $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ *is*

$$\inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) = \sigma^2 + \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\left(\mathbf{I} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\right), \tag{3.8}$$

  where $\boldsymbol{\Omega} := [(M+1)\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + (\operatorname{tr}(\mathbf{H}\boldsymbol{\Psi}) + \sigma^2)\mathbf{I}_d]/M$.

- *The global optimal parameters for* $f_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ *that attain the minimum* (3.8) *take the following form:*

$$\boldsymbol{\beta} = \boldsymbol{\beta}^* + \mathsf{null}(\mathbf{H}), \quad \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^* + \mathsf{null}\left(\mathbf{H}^{\otimes 2}\right), \quad \text{where} \quad \boldsymbol{\Gamma}^* := \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}} \tag{3.9}$$

  and $\boldsymbol{\beta}^*$ *is the mean of the task parameter in Assumption 3.3.1 Here,*

$$\mathsf{null}(\mathbf{H}) := \left\{\mathbf{z}\in\mathbb{R}^d : \mathbf{H}\mathbf{z} = \mathbf{0}\right\}, \quad \mathsf{null}\left(\mathbf{H}^{\otimes 2}\right) := \left\{\mathbf{Z}\in\mathbb{R}^{d\times d} : \mathbf{H}\mathbf{Z}\mathbf{H} = \mathbf{0}\right\}$$

  are the null space of $\mathbf{H}$*, and the null space of* $\mathbf{H}^{\otimes 2}$*, respectively. In particular, when* $\mathbf{H}$ *is positive definite, the global optimal parameter is unique,* $(\boldsymbol{\beta}, \boldsymbol{\Gamma}) = (\boldsymbol{\beta}^*, \boldsymbol{\Gamma}^*)$.

- *Under Assumption 3.3.1, the global optimal* $f_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ *that attains the minimum* (3.8) *is unique as a function of* $\mathbf{E}$ *(given by* (3.1)*) and takes the following form:*

$$f^*(\mathbf{E}) = \left\langle \boldsymbol{\beta}^* - \frac{1}{M}\boldsymbol{\Gamma}^*\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta}^* - \mathbf{y}), \mathbf{x}\right\rangle. \tag{3.10}$$

Theorem 3.5.2 characterizes the optimal GD-$\boldsymbol{\beta}$ models for ICL. In the above theorem, $\boldsymbol{\Gamma}^* \to \mathbf{H}^{-1}$ as the context length M goes to infinity (assuming that $\mathbf{H}$ is positive definite). In this case, the optimal GD-$\boldsymbol{\beta}$ function (3.8) implements one Newton step from initialization $\boldsymbol{\beta}^*$. With a finite context length M, (3.8) implements one regularized Newton step from initialization $\boldsymbol{\beta}^*$.

Optimal LTB models for ICL. Lemma 3.5.1 shows that the best ICL risk achieved by an GD-$\boldsymbol{\beta}$ model is no smaller than the best ICL risk achieved by an LTB model. Surprisingly, our next theorem shows that the best ICL risk achieved by a GD-$\boldsymbol{\beta}$ is *equal* to that achieved by an LTB model. Therefore, the hypothesis set $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ is diverse enough to match the approximation ability of the larger hypothesis set $\mathcal{F}_{\mathsf{LTB}}$ for ICL.

Theorem 3.5.3 (Optimal LTB models)**.** *Consider the ICL risk defined by (3.5). Suppose that Assumption 3.3.1 holds and that* $\operatorname{rank}\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}^{\frac{1}{2}}\right) \geq 2$. *Then we have*

- *The minimal ICL risk of* $\mathcal{F}_{\mathsf{LTB}}$ *and of* $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ *are equal,*

$$\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}(f) = \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) = \text{ RHS of (3.8)}$$

- *Rewrite an LTB model as* f_{LTB} *in (3.3) with parameters (* $*$ *denotes parameters that do not affect the output)*

$$\mathbf{W}_2^\top \mathbf{W}_1 = \begin{pmatrix} * & * \\ \boldsymbol{\gamma}^\top & * \end{pmatrix}, \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q = \begin{pmatrix} \mathbf{V}_{11} & * \\ \mathbf{v}_{12}^\top & * \end{pmatrix}, \mathbf{W}_2^\top \mathbf{W}_1 \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V = \begin{pmatrix} * & * \\ \mathbf{v}_{21}^\top & v_{-1} \end{pmatrix}.$$

  Then the sufficient and necessary conditions for $f_{\mathsf{LTB}} \in \arg\min_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}(f)$ *are*

$$v_{-1} \neq 0, \quad v_{-1}\mathbf{v}_{12} \in \mathsf{null}\left(\mathbf{H}\right), \quad \mathbf{v}_{21} \in -v_{-1}\boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right), \quad \boldsymbol{\gamma} \in \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right),$$
$$v_{-1}\mathbf{V}_{11} \in \mathbf{\Gamma}^{*\top} - v_{-1}\boldsymbol{\beta}^*\mathbf{v}_{12}^\top + \mathsf{null}\left(\mathbf{H}^{\otimes 2}\right),$$

  where $\boldsymbol{\beta}^*$ *is defined in Assumption 3.3.1 and* $\mathbf{\Gamma}^*$ *is defined in (3.9). In particular, when* $\mathbf{H}$ *is positive definite, the globally optimal parameter represented by* $(\mathbf{V}_{11}, \mathbf{v}_{12}, \mathbf{v}_{21}, v_{-1}, \boldsymbol{\gamma})$ *is unique up to a rescaling of* v_{-1}.

- *Under Assumption 3.3.1, the globally optimal LTB model (i.e., a function in* $\arg\min_{f\in\mathcal{F}_{\mathsf{LTB}}}$*) is unique as a function of* $\mathbf{E}$ *(given by (3.1)) and takes the form of (3.10) almost surely.*

Lemma 3.5.1 and Theorem 3.5.3 together show that $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ is a representative subset of $\mathcal{F}_{\mathsf{LTB}}$ that does not incur additional approximation error. In addition, every optimal LTB model is effectively an optimal $\mathsf{GD}\text{-}\boldsymbol{\beta}$ model when restricted to all possible token matrices. Note that the optimal model parameters for LTB or $\mathsf{GD}\text{-}\boldsymbol{\beta}$ are not unique because of redundant parameterization. But the optimal LTB and $\mathsf{GD}\text{-}\boldsymbol{\beta}$ models are unique as a function of all possible token matrices. The above holds even when $\mathbf{H}$ and $\mathbf{\Psi}$ are potentially rank deficient.

Comparison with prior works. A line of papers considers LSA models for ICL under Assumption 3.3.1 with $\boldsymbol{\beta}^* = 0$ (Von Oswald et al., 2023; Ahn et al., 2023c; Zhang et al., 2024c; Wu et al., 2024b). They show that LSA models can (effectively) implement all possible GD-0 models that specialize $\mathsf{GD}\text{-}\boldsymbol{\beta}$ models by fixing $\boldsymbol{\beta} = \mathbf{0}$. In addition, they show that every optimal LSA model is (effectively) a GD-0 model for ICL under Assumption 3.3.1 with $\boldsymbol{\beta}^* = \mathbf{0}$ (see, for example, Theorem 1 in Ahn et al., 2023c). In comparison, we consider a harder ICL problem that allows a large shared signal in tasks (that is, a large $\boldsymbol{\beta}^*$ in Assumption 3.3.1). In this setting, our Theorem 3.4.1 shows that $\mathcal{F}_{\mathsf{LSA}}$ (hence its subset formed by GD-0 models), as a subet of $\mathcal{F}_{\mathsf{LTB}}$, incurs an additional approximation error proportional to $\|\boldsymbol{\beta}^*\|_{\mathbf{H}}^2$ compared with $\mathcal{F}_{\mathsf{LTB}}$. In contrast, $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$, as a subset of $\mathcal{F}_{\mathsf{LTB}}$, does not incur additional approximation error according to our Theorem 3.5.3. Thus the LSA and GD-0 models considered by Von Oswald et al., 2023; Ahn et al., 2023c; Wu et al., 2024b and by Chapter 2 are not capable of learning the shared signal $\boldsymbol{\beta}^*$, while an LTB model can learn $\boldsymbol{\beta}^*$ through implementing $\mathsf{GD}\text{-}\boldsymbol{\beta}$ and encoding $\boldsymbol{\beta}^*$ in the initialization parameter.

3.6 Training and In-Context Learning of GD-beta

We have shown that $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ is a representative subset of $\mathcal{F}_{\mathsf{LTB}}$ that effectively contains every optimal LTB model. We now examine the ICL and training of GD-$\boldsymbol{\beta}$ models.

Nearly optimal ICL with GD-$\boldsymbol{\beta}$. We will compare the best ICL risk achieved by GD-$\boldsymbol{\beta}$ with the best ICL risk achieved by any estimator. The following lemma is an extension of Proposition 5.1 and Corollary 5.2 in Wu et al., 2024b (which is based on Tsigler and Bartlett, 2023) that characterizes the Bayes optimal ICL risk among all estimators.

Lemma 3.6.1 (Bayes optimal ICL)**.** *Given a task-specific dataset* $(\mathbf{X}, \mathbf{y}, \mathbf{x}, y)$ *sampled according to Assumption 3.3.1, let* $g(\mathbf{X}, \mathbf{y}, \mathbf{x})$ *be an arbitrary estimator for* y *and measure the average linear regression risk by* $\ell(g; \mathbf{X}) := \mathbb{E}[(g(\mathbf{X}, \mathbf{y}, \mathbf{x}) - y)^2 \mid \mathbf{X}]$*. It is clear that* $\mathbb{E}\ell(g; \mathbf{X}) = \mathsf{Risk}(g)$*. Then,*

- *The optimal estimator that minimizes the average linear regression risk* $\ell(\cdot; \mathbf{X})$ *is*

$$g^*(\mathbf{X}, \mathbf{y}, \mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}^* + \mathbf{x}^\top \boldsymbol{\Psi}^{\frac{1}{2}} \left(\boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{X}^\top \mathbf{X} \boldsymbol{\Psi}^{\frac{1}{2}} + \sigma^2 \mathbf{I}_d \right)^{-1} \boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{X}^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*).$$

- *Assume the signal-to-noise ratio is upper bounded, that is,* $\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi}) \lesssim \sigma^2$*, then with probability at least* $1 - \exp(-\Omega(M))$ *over the randomness of* $\mathbf{X}$*, it holds that* $\ell(g^*; \mathbf{X}) - \sigma^2 \simeq \sum_{i=1}^d \min\{\bar{\phi}, \phi_i\}$*, where* $\bar{\phi} \eqsim \frac{\sigma^2}{M}$*, and* $(\phi_i)_{i \geq 1}$ *are the eigenvalues of* $\boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{H} \boldsymbol{\Psi}^{\frac{1}{2}}$*.*

The proof is deferred to Appendix 3.13. Lemma 3.6.1 shows that the Bayes optimal estimator is a ridge regression estimator centered at $\boldsymbol{\beta}^*$. This is consistent with Wu et al., 2024b where the Bayes optimal estimator is a ridge regression estimator as they assumed $\boldsymbol{\beta}^* = 0$. The following corollary of Theorem 3.5.2 computes the rate of the ICL risk achieved by the optimal GD-$\boldsymbol{\beta}$ model.

Corollary 3.6.2. *Under the setup of Theorem 3.5.2, additionaly assume the signal-to-noise ratio is upper bounded, that is,* $\mathrm{tr}(\boldsymbol{\Psi}\mathbf{H}) \lesssim \sigma^2$*. Then we have* $\inf_{f \in \mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) - \sigma^2 \simeq \sum_{i=1}^d \min\left\{\bar{\phi}, \phi_i\right\}$*, where* $(\phi_i)_{i \geq 0}$ *are the eigenvalues of* $\boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{H} \boldsymbol{\Psi}^{\frac{1}{2}}$ *and* $\bar{\phi} := [\mathrm{tr}(\boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{H} \boldsymbol{\Psi}^{\frac{1}{2}}) + \sigma^2]/M \simeq \sigma^2/M$*.*

The optimal (expected) ICL risk achieved by $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ in Corollary 3.6.2 matches the (high probability) Bayes optimal ICL risk in Lemma 3.6.1 ignoring constant factors, provided that the signal-to-noise ratio is upper bounded. Therefore $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ achieves nearly Bayes optimal ICL risk. As a consequence, the larger hypothesis set $\mathcal{F}_{\mathsf{LTB}}$ also achieves nearly Bayes optimal ICL of linear regression under Assumption 3.3.1.

For the simplicity of discussion, we assume a fixed context length during pretraining and inference. Our discussions can be extended to allow a different context length during pretraining and inference using techniques in Wu et al., 2024b. However, this is not the main focus of this chapter.

Optimization of GD-$\boldsymbol{\beta}$ with infinite tasks. We have shown that $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ is a representative subset of $\mathcal{F}_{\mathsf{LTB}}$ that covers the optimal LTB models and achieves nearly optimal ICL risk. We now consider the optimization in the parameter space specified by $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$. For simplicity, we follow Chapter 2 and consider gradient descent with an infinitesimal stepsize on the ICL objective with an infinite number of tasks. That is, we consider the optimization of gradient flow on the population ICL risk under the parameterization of GD-$\boldsymbol{\beta}$,

$$\frac{\mathrm{d}\boldsymbol{\beta}(t)}{\mathrm{d}t} = -\frac{\partial}{\partial \boldsymbol{\beta}}\mathcal{R}(f_{\mathsf{GD}\text{-}\boldsymbol{\beta}}), \quad \frac{\mathrm{d}\boldsymbol{\Gamma}(t)}{\mathrm{d}t} = -\frac{\partial}{\partial \boldsymbol{\Gamma}}\mathcal{R}(f_{\mathsf{GD}\text{-}\boldsymbol{\beta}}), \tag{3.11}$$

where Risk is defined by (3.5) and $f_{\mathsf{GD}\text{-}\boldsymbol{\beta}}$ is defined by (3.7).

The following theorem guarantees the global convergence of gradient flow. We introduce some notation to accommodate cases when $\mathbf{H}$ is rank deficient. Let $\mathcal{P}_{\mathcal{S}}$ be the orthogonal projection operator onto a subspace $\mathcal{S}$. Let $\mathcal{H} = \mathsf{Im}(\mathbf{H})$ be the image space of matrix $\mathbf{H}$ (viewing $\mathbf{H}$ as a linear map) and $\mathcal{H}^{\perp} := \mathsf{null}(\mathbf{H})$ be its orthogonal complement. Similarly, let $\mathcal{Z} := \mathsf{Im}(\mathbf{H}^{\otimes 2}) = \{\mathbf{HZH}, \mathbf{Z} \in \mathbb{R}^{d\times d}\}$ be the image space of the operator $\mathbf{H}^{\otimes 2}$ and $\mathcal{Z}^{\perp}$ be its orthogonal complement. Then we have the following theorem.

Theorem 3.6.3. *Consider the gradient flow defined by (3.11) with initialization $\boldsymbol{\beta}(0), \boldsymbol{\Gamma}(0)$. We have,*

$$\begin{aligned} &\mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}(t)) \to \mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}^*), \quad \mathcal{P}_{\mathcal{H}^{\perp}}(\boldsymbol{\beta}(t)) = \mathcal{P}_{\mathcal{H}^{\perp}}(\boldsymbol{\beta}(0)), \\ &\mathcal{P}_{\mathcal{Z}}(\boldsymbol{\Gamma}(t)) \to \mathcal{P}_{\mathcal{Z}}(\boldsymbol{\Gamma}^*), \quad \mathcal{P}_{\mathcal{Z}^{\perp}}(\boldsymbol{\Gamma}(t)) = \mathcal{P}_{\mathcal{Z}^{\perp}}(\boldsymbol{\Gamma}(0)) \end{aligned}$$

as $t \to \infty$. In particular, if $\mathbf{H}$ is positive definite, the gradient flow converges to the unique global minimizer of ICL risk over GD-$\boldsymbol{\beta}$ class, that is, $\boldsymbol{\beta}(t) \to \boldsymbol{\beta}^*$ and $\boldsymbol{\Gamma}(t) \to \boldsymbol{\Gamma}^*$ as $t \to \infty$.

The proof, as well as the convergence rate, is deferred to Appendix 3.14. We remark that (3.11) is a complex dynamical system on a non-convex potential function of $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$. We briefly discuss our proof techniques assuming that $\mathbf{H}$ is full rank. The rank-deficient cases can be handled in the same way by applying appropriate project operators. To conquer the non-convex optimization issue, we observe that for every fixed $\boldsymbol{\Gamma}$, the potential as a function of $\boldsymbol{\beta}$ is smooth and strongly convex with a uniformly bounded condition number. This observation allows us to establish a uniform convergence for $\boldsymbol{\beta}$. When $\boldsymbol{\beta}$ is sufficiently close to $\boldsymbol{\beta}^*$, the potential as a function of $\boldsymbol{\Gamma}$ is approximately convex, allowing us to track the convergence of $\boldsymbol{\Gamma}$.

Theorem 3.6.3 shows that optimization of GD-$\boldsymbol{\beta}$ can be done efficiently by gradient flow without suffering from non-convexity. However, as we have shown in previous sections, LTB utilizes a more complex parameterization than GD-$\boldsymbol{\beta}$. So Theorem 3.6.3 does not imply optimization of LTB is easy. We leave it as future work to study the optimization and statistical complexity for directly learning LTB models.

3.7 Discussion and Conclusion

In this chapter, we study the in-context learning of linear regression with a shared signal represented by a Gaussian prior with a non-zero mean. We show that although the linear self-attention layer discussed in prior works is consistent for this more complex task, its risk has an inevitable gap compared to that of the linear Transformer block (LTB), which is a linear self-attention layer followed by a linear multi-layer perceptron (MLP) layer. Next, we show that the effectiveness of the LTB arises because it can implement the one-step gradient descent estimator with learnable initialization (GD-$\boldsymbol{\beta}$). Moreover, all global minimizers in the LTB class are equivalent to the unique global minimizer in the GD-$\boldsymbol{\beta}$ class, which can achieve nearly Bayes optimal in-context learning risk. Finally, we consider training on in-context examples and prove global convergence over the GD-$\boldsymbol{\beta}$ class of gradient flow on the population loss. Several future directions are worth discussing.

Optimization and statistical complexity. This chapter provides an approximation theory of LTB and an optimization theory of GD-$\boldsymbol{\beta}$. However, the statistical complexity of learning LTB or GD-$\boldsymbol{\beta}$ is not considered. The work by Wu et al., 2024b provided techniques for analyzing the statistical task complexity for pre-training GD-0. An interesting direction is to extend their method to study the statistical complexity of learning GD-$\boldsymbol{\beta}$. However, their method crucially relies on the convexity of the risk induced GD-0, while we have shown that the risk induced by GD-$\boldsymbol{\beta}$ is non-convex. New ideas for dealing with non-convexity are needed here.

From LTB to Transformer block. We focus on LTB in this chapter, which simplifies a vanilla Transformer block by removing the non-linearities from the softmax self-attention and the ReLU activation in the MLP layers. This simplification allows us to obtain precise theoretical results for LTB (such as its connection to GD-$\boldsymbol{\beta}$). On the other hand, non-linearities are arguably necessary for Transformers to work well in practice. An important next step is to further consider the theoretical benefits of non-linearities based on our current results.

Roles of MLP layers. The work by Geva et al., 2021 (and references thereafter) empirically found that MLP layers operate as key-value memories that store human-interpretable patterns in some pre-trained Transformers. Their work motivated a method for locating and editing information stored in language models by modifying their MLP layers (see, e.g., Dai et al., 2022; Meng et al., 2022). This chapter proves that the MLP component enables LTB to learn the shared signal in linear regression tasks, which cannot be done by a single LSA component. We leave it as future work to theoretically clarify the information stored in the MLP component.

Chapter 2 and Chapter 3 focus on explaining in-context learning phenomena in Transformers through simplified but representative models. We next turn to a complementary set of questions on the training side: modern deep learning often operates outside classical stability regimes, yet still converges rapidly. Chapter 4 studies such non-monotone optimization dynamics in a canonical logistic regression with linearly separable data setting and develops a minimax-optimal analysis for gradient descent with large, adaptive stepsizes.

3.8 Appendix: Notation and Variable Transformation

In order to simplify the proof, let's first describe a variable transformation scheme for our model and data coming from Assumption 3.3.1.

Definition 3.8.1 (Variable Transformation)**.** We fix M as the length of the contexts. Recall that $\mathbf{X}, \mathbf{x}$ are, respectively, the features in the context and the query input, while $\widetilde{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^*$ are the true task parameter for the inference prompt and its expectation, respectively. Following the Assumption 3.3.1, we have $\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^*, \boldsymbol{\Psi})$. From the definition for multivariate Gaussian distribution, we know there exists a random vector $\widetilde{\boldsymbol{\theta}}$ such that

$$\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}, \tag{3.12}$$

where

$$\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{3.13}$$

Recall the noise vector is defined as $\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ and is generated from $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I}_M)$.

Rank deficient case Note that, even when $\boldsymbol{\Psi}$ is rank-deficient, the variable transformation in (3.12) and (3.13) still hold. This can be seen from the definition of the multivariate Gaussian distribution (Def 20.11 and the discussion below in (Seber, 2008)): A vector $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ with mean $\boldsymbol{\beta}^*$ and covariance matrix $\boldsymbol{\Psi}$ has a multivariate normal distribution, if it has the same distribution as $\mathbf{A}\widetilde{\boldsymbol{\theta}} + \boldsymbol{\beta}^*$ where $\mathbf{A}$ is any $d \times m$ matrix satisfying $\mathbf{A}\mathbf{A}^\top = \boldsymbol{\Psi}$ and $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$. Here, we can recover the variable transformation above if we take $m = d$ and $\mathbf{A} = \boldsymbol{\Psi}^{\frac{1}{2}}$.

Notation Before we delve into the detailed proof, let's repeat the notation part with some additional notations. We use lowercase bold letters to note vectors and uppercase bold letters to denote matrices and tensors. For a vector $\mathbf{x}$ and a positive semi-definite (PSD) matrix $\mathbf{A}$, we denote $\|x\|_{\mathbf{A}} := \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$. We denote $\langle \cdot, \cdot \rangle$ as the inner product, which is defined as $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^\top \mathbf{y}$ for vectors and $\langle \mathbf{A}, \mathbf{B} \rangle := \operatorname{tr}\left(\mathbf{A}\mathbf{B}^\top\right)$ for matrices. For a matrix $\mathbf{A}$, we denote $\|\mathbf{A}\|_{op}, \|\mathbf{A}\|_F$ as the operator norm and the Frobenius norm, respectively. We denote $\mathbf{A}[i]$ as the i-th row of the matrix $\mathbf{A}$, $\mathbf{A}_{m,n}$ as the (m,n)-th entry, and $\mathbf{A}_{-1,-1}$ as the right-bottom entry. We denote $\mathbf{0}_n, \mathbf{0}_{m\times n}, \mathbf{I}_n$ as the zero vector, zero matrix and identity matrix, respectively.

For a positive semi-definite matrix $\mathbf{A}$, we denote $\mathbf{A}^{\frac{1}{2}}$ for the principal square root of $\mathbf{A}$, which is defined as the unique real matrix $\mathbf{B}$ such that $\mathbf{B}$ is positive semi-definite and $\mathbf{B}\mathbf{B} = \mathbf{B}\mathbf{B}^\top = \mathbf{A}$. For positive definite $\mathbf{A}$, its principle square root is also positive definite. We denote $\mathbf{A}^+$ as the Moore-Penrose pseudo-inverse for any matrix $\mathbf{A}$. We also denote $\mathbf{A}^{-\frac{1}{2}} = \left(\mathbf{A}^{\frac{1}{2}}\right)^+$. We denote $\otimes$ as the Kronecker product. For compatible matrices of proper size $\mathbf{A}, \mathbf{B}, \mathbf{C}$, $\mathbf{B}^\top \otimes \mathbf{A}$ is a linear mapping which is defined by $\left(\mathbf{B}^\top \otimes \mathbf{A}\right) \circ \mathbf{C} = \mathbf{A}\mathbf{C}\mathbf{B}$. We

denote $\mathbf{\Lambda} := \mathbf{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{\Psi}^{\frac{1}{2}}$ and $\phi_1 \geq \phi_2 \geq ... \geq \phi_d \geq 0$ are its ordered eigenvalues. We also denote $\mathbf{\Lambda}_1 \geq \mathbf{\Lambda}_2 \geq ... \geq \mathbf{\Lambda}_d \geq 0$ as the ordered eigenvalues of $\mathbf{H}$, and $\mathbf{\Lambda}min > 0$ as its minimal positive eigenvalue. Finally, we denote another important matrix

$$\mathbf{\Omega} := \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}} + \frac{\operatorname{tr}(\mathbf{H}\mathbf{\Psi}) + \sigma^2}{M} \cdot \mathbf{I}_d. \tag{3.14}$$

Note that, under this definition and our assumption on $\mathbf{H}$ and $\mathbf{\Psi}$, we have $\mathbf{\Omega}$ is invertible.

3.9 Appendix: Proof of Theorem 3.4.1

Proof. The fact that $\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}(f)$ does not depend on the vector $\boldsymbol{\beta}^*$ is subsumed in the Theorem 3.5.3, so we do not prove it here. We are going to prove the inequality in the theorem. First, from Theorem 3.5.3 we know that,

$$\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}(f) = \sigma^2 + \operatorname{tr}(\mathbf{H}\mathbf{\Psi}) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \mathbf{\Omega}^{-1}\right), \tag{3.15}$$

where $\mathbf{\Omega}$ is defined in (3.14). Then, it suffices to lower bound $\inf_{f\in\mathcal{F}_{\mathsf{LSA}}} \mathsf{Risk}(f)$. So let's take an arbitrary $f \in \mathcal{F}_{\mathsf{LSA}}$, which is denoted as

$$f(\mathbf{E}) = \left[\mathbf{E} + \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M}\frac{\mathbf{E}^\top \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right]_{-1,-1},$$

where $\mathbf{E}$ is the token matrix defined in (3.1). Since the prediction is the right-bottom entry of the output matrix, we know only the last row of the product $\left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V$ attends the prediction. Similarly, only the last column of the $\mathbf{E}$ on the far right in the above equation attends the prediction. Since the last column of $\mathbf{E}$ is $\begin{pmatrix}\mathbf{x}^\top & 0\end{pmatrix}^\top$, we know that only the first d rows of the product $\left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q$ enter the calculation (since other parts are multiplied by zero). Therefore, we denote

$$\left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V = \begin{pmatrix} * & * \\ \mathbf{u}_{21}^\top & u_{-1}\end{pmatrix}, \quad \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q = \begin{pmatrix}\mathbf{U}_{11} & * \\ \mathbf{u}_{12}^\top & *\end{pmatrix},$$

where $\mathbf{U}_{11} \in \mathbb{R}^{d\times d}, \mathbf{u}_{12}, \mathbf{u}_{21} \in \mathbb{R}^{d\times 1}, u_{-1} \in \mathbb{R}$, and $*$ denotes entries that do not enter the final prediction. The model prediction can be written as

$$\begin{aligned} f(\mathbf{E}) &= \begin{pmatrix}\mathbf{u}_{21}^\top & u_{-1}\end{pmatrix} \cdot \frac{\mathbf{E}\mathbf{M}_M\mathbf{E}^\top}{M} \cdot \begin{pmatrix}\mathbf{U}_{11} \\ \mathbf{u}_{12}^\top\end{pmatrix} \cdot \mathbf{x} \\ &= \left[\mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{X} \cdot \mathbf{U}_{11} + \mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{y} \cdot \mathbf{u}_{12}^\top + u_{-1} \cdot \frac{1}{M}\mathbf{y}^\top\mathbf{X} \cdot \mathbf{U}_{11} + u_{-1} \cdot \frac{1}{M}\mathbf{y}^\top\mathbf{y} \cdot \mathbf{u}_{12}^\top\right] \cdot \mathbf{x}. \end{aligned}$$

Step 1: simplify the risk function. We use $\widetilde{\boldsymbol{\beta}}$ to denote the task parameter. From the Assumption 3.3.1 and Definition 3.8.1, we have

$$\mathbf{y} = \mathbf{X}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}, \quad y = \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x} \rangle + \varepsilon, \quad \widetilde{\boldsymbol{\beta}} \sim \mathcal{N}\left(\boldsymbol{\beta}^*, \boldsymbol{\Psi}\right), \quad \widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}};$$

and

$$\mathbf{X}[i], \mathbf{x} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\mathbf{0}, \mathbf{H}\right), \quad \boldsymbol{\varepsilon}[i], \varepsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\mathbf{0}, \sigma^2\right), \quad \widetilde{\boldsymbol{\theta}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right).$$

Then the model output can be written as

$$\begin{aligned} f(\mathbf{E}) &= \left[\mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{X}\mathbf{U}_{11} + \mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{y}\mathbf{u}_{12}^\top + u_{-1} \cdot \frac{1}{M}\mathbf{y}^\top\mathbf{X}\mathbf{U}_{11} + u_{-1} \cdot \frac{1}{M}\mathbf{y}^\top\mathbf{y}\mathbf{u}_{12}^\top\right] \cdot \mathbf{x} \\ &= \left[\left(\mathbf{u}_{21} + u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{X} \cdot \left(\mathbf{U}_{11} + \widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\right] \cdot \mathbf{x} \\ &\quad + \left[\mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\boldsymbol{\varepsilon} \cdot \mathbf{u}_{12}^\top + u_{-1} \cdot \frac{1}{M}\boldsymbol{\varepsilon}^\top\mathbf{X} \cdot \mathbf{U}_{11} + u_{-1} \cdot \frac{2}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top + \frac{1}{M}\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon} \cdot u_{-1}\mathbf{u}_{12}^\top\right] \cdot \mathbf{x}. \end{aligned}$$

To simplify the presentation, we denote

$$\begin{aligned} \mathbf{z}_1^\top &= \left(\mathbf{u}_{21} + u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{X} \cdot \left(\mathbf{U}_{11} + \widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right), \\ \mathbf{z}_2^\top &= \mathbf{u}_{21}^\top \cdot \frac{1}{M}\mathbf{X}^\top\boldsymbol{\varepsilon} \cdot \mathbf{u}_{12}^\top + u_{-1} \cdot \frac{1}{M}\boldsymbol{\varepsilon}^\top\mathbf{X} \cdot \mathbf{U}_{11} + u_{-1} \cdot \frac{2}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top \\ \mathbf{z}_3^\top &= \frac{1}{M}\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon} \cdot u_{-1}\mathbf{u}_{12}^\top. \end{aligned}$$

Since $\mathbf{x}, \mathbf{X}, \boldsymbol{\varepsilon}, \widetilde{\boldsymbol{\beta}}$ are independent, we have

$$\begin{aligned} \mathsf{Risk}\left(f\right) &= \mathbb{E}\left(f(\mathbf{E}) - \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x} \rangle - \varepsilon\right)^2 \\ &= \sigma^2 + \mathbb{E}\left(f(\mathbf{E}) - \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x} \rangle\right)^2 \ (\varepsilon \text{ is independent from other variables and zero-mean}) \\ &= \mathbb{E}\left[\langle \mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 - \widetilde{\boldsymbol{\beta}}, \mathbf{x} \rangle^2\right] + \sigma^2 \\ &= \langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 - \widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 - \widetilde{\boldsymbol{\beta}}\right)^\top \rangle + \sigma^2. \end{aligned}$$

Note that $\mathbf{z}_1$ does not contain $\boldsymbol{\varepsilon}$, $\mathbf{z}_2$ is a linear form of $\boldsymbol{\varepsilon}$, and $\mathbf{z}_3$ is a quadratic form of $\boldsymbol{\varepsilon}$. Using $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$, we have $\mathbb{E}[(\mathbf{z}_1 - \widetilde{\boldsymbol{\beta}}) \cdot \mathbf{z}_2^\top] = \mathbf{0}$ and $\mathbb{E}[\mathbf{z}_2\mathbf{z}_3^\top] = \mathbf{0}$. Therefore, we have

$$\begin{aligned} \mathsf{Risk}\left(f\right) - \sigma^2 = &\underbrace{\langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 - \widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{z}_1 - \widetilde{\boldsymbol{\beta}}\right)^\top \rangle}_{S_1} + \underbrace{\langle \mathbf{H}, \mathbb{E}\mathbf{z}_2\mathbf{z}_2^\top \rangle}_{S_2} + \underbrace{\langle \mathbf{H}, \mathbb{E}\mathbf{z}_3\mathbf{z}_3^\top \rangle}_{S_3} \\ &+ \underbrace{2\langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 - \widetilde{\boldsymbol{\beta}}\right)\mathbf{z}_3^\top \rangle}_{S_4}. \end{aligned} \tag{3.16}$$

Step 2: compute S_1. By Lemma 2.10.2 and that $\mathbf{X}$ is independent of all other random variables, we have

$$\begin{aligned}
&\left\langle \mathbf{H}, \mathbb{E}\mathbf{z}_1\mathbf{z}_1^\top \right\rangle \\
=&\mathbb{E}\mathrm{tr}\left[\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)^\top \frac{\mathbf{X}^\top\mathbf{X}}{M}\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \frac{\mathbf{X}^\top\mathbf{X}}{M}\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\right] \\
=&\frac{M+1}{M}\mathbb{E}_{\widetilde{\boldsymbol{\beta}}}\mathrm{tr}\left[\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)^\top\cdot\mathbf{H}\cdot\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top\cdot\mathbf{H}\cdot\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\right] \\
+&\frac{1}{M}\mathbb{E}_{\widetilde{\boldsymbol{\beta}}}\mathrm{tr}\left[\mathrm{tr}\left(\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{u}_{21}+u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top\mathbf{H}\right)\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)^\top\mathbf{H}\left(\mathbf{U}_{11}+\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\right].
\end{aligned}$$

We denote

$$\boldsymbol{b}:=\mathbf{u}_{21}+u_{-1}\boldsymbol{\beta}^*\in\mathbb{R}^d,\quad \mathbf{A}:=\mathbf{U}_{11}+\boldsymbol{\beta}^*\mathbf{u}_{12}^\top\in\mathbb{R}^{d\times d}, \tag{3.17}$$

then applying $\widetilde{\boldsymbol{\beta}}=\boldsymbol{\beta}^*+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}$, we get

$$\begin{aligned}
&\left\langle \mathbf{H}, \mathbb{E}\mathbf{z}_1\mathbf{z}_1^\top \right\rangle \\
=&\frac{M+1}{M}\mathbb{E}\mathrm{tr}\left[\left(\mathbf{A}+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right)^\top\mathbf{H}\left(\boldsymbol{b}+u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)\left(\boldsymbol{b}+u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)^\top\mathbf{H}\left(\mathbf{A}+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\right] \\
+&\frac{1}{M}\mathbb{E}\mathrm{tr}\left(\left(\boldsymbol{b}+u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)\left(\boldsymbol{b}+u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)^\top\mathbf{H}\right)\mathrm{tr}\left[\left(\mathbf{A}+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right)^\top\mathbf{H}\left(\mathbf{A}+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\right],
\end{aligned}$$

where the expectation is over $\widetilde{\boldsymbol{\theta}}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)$. Note that the first and third moments of $\widetilde{\boldsymbol{\theta}}$ are zero. So the above equation only involves the zeroth, second, and fourth moments of $\widetilde{\boldsymbol{\theta}}$. Therefore we have

$$\left\langle \mathbf{H}, \mathbb{E}\mathbf{z}_1\mathbf{z}_1^\top \right\rangle = \underbrace{\frac{M+1}{M}\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\boldsymbol{b}\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right)+\frac{1}{M}\mathrm{tr}\left(\boldsymbol{b}\boldsymbol{b}^\top\mathbf{H}\right)\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right)}_{\text{The leading term}}+T_2+T_4, \tag{3.18}$$

where T_2 and T_4 are the second order and the fourth order term, respectively. More concretely, the second order term T_2 is

$$
\begin{aligned}
&\frac{M+1}{M}\mathbb{E}\mathrm{tr}\Big\{u_{-1}\mathbf{A}^\top\mathbf{H}\boldsymbol{b}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}+u_{-1}\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\\
&+u_{-1}\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{b}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{A}\mathbf{H}+u_{-1}\mathbf{A}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\\
&+\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{b}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}+u_{-1}^2\mathbf{A}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{A}\mathbf{H}\Big\}\\
&+\frac{1}{M}\mathbb{E}\Big\{u_{-1}^2\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\cdot\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right)+\boldsymbol{b}^\top\mathbf{H}\boldsymbol{b}\cdot\mathrm{tr}\left(\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\right)\\
&+2u_{-1}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\cdot\mathrm{tr}\left(\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{A}\mathbf{H}\right)+2u_{-1}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\cdot\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\right)\Big\}\\
&=\frac{M+1}{M}\Big\{2u_{-1}\mathrm{tr}(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}})\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{A}^\top\mathbf{H}\boldsymbol{b}+2u_{-1}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12}+\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b}\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}\\
&+u_{-1}^2\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\right)\Big\}+\frac{1}{M}\Big\{u_{-1}^2\mathrm{tr}\left(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\right)\cdot\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right)\\
&+\mathrm{tr}\left(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\right)\cdot\boldsymbol{b}^\top\mathbf{H}\boldsymbol{b}\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}+4u_{-1}\cdot\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12}\Big\}\\
&=\frac{2(M+1)}{M}u_{-1}\boldsymbol{b}^\top\left[\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}+\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\right]\mathbf{A}\mathbf{H}\mathbf{u}_{12}+\boldsymbol{b}^\top\left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H}+\frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\boldsymbol{b}\cdot\\
&\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}+u_{-1}^2\mathrm{tr}\left(\mathbf{A}^\top\left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H}+\frac{\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})}{M}\mathbf{H}\right)\mathbf{A}\mathbf{H}\right)+\frac{4u_{-1}}{M}\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12}. \qquad (3.19)
\end{aligned}
$$

The fourth order term is

$$
\begin{aligned}
T_4&=\frac{M+1}{M}\mathbb{E}\mathrm{tr}\Big\{u_{-1}^2\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\Big\}\\
&\qquad+\frac{1}{M}\mathbb{E}\mathrm{tr}\Big\{u_{-1}^2\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\cdot\mathrm{tr}\left(\mathbf{u}_{12}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\right)\Big\}\\
&=\frac{M+2}{M}u_{-1}^2\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}\cdot\mathbb{E}\left(\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)^2\\
&=\frac{M+2}{M}u_{-1}^2\cdot\left(2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right)+\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)^2\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}. \qquad (3.20)
\end{aligned}
$$

For the cross term in S_1, we have

$$\begin{aligned}
&\left\langle \mathbf{H}, \mathbb{E}\mathbf{z}_1\widetilde{\boldsymbol{\beta}}^\top \right\rangle \\
=&\mathbb{E}\left\{ \left(\mathbf{u}_{21} + u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \cdot \frac{1}{M}\mathbf{X}^\top\mathbf{X} \cdot \left(\mathbf{U}_{11} + \widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\widetilde{\boldsymbol{\beta}} \right\} && (3.21)\\
=&\mathbb{E}\left\{ \left(\mathbf{u}_{21} + u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \mathbf{H}\left(\mathbf{U}_{11} + \widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\widetilde{\boldsymbol{\beta}} \right\} && \text{(From the distribution of } \mathbf{X})\\
=&\mathbb{E}\left\{ \left(\boldsymbol{b} + u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)^\top \mathbf{H}\left(\mathbf{A} + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right)\mathbf{H}\left(\boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right) \right\} && \text{(By (3.17))}\\
=&\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\beta}^* + \mathbb{E}\left\{ u_{-1}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\beta}^* + u_{-1}\widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}} + \boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}} \right\}\\
=&\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\beta}^* + u_{-1}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\beta}^* + u_{-1}\mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}\right) + \mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b}, && (3.22)
\end{aligned}$$

where the last row comes from the fact that $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right)$. Moreover, we have

$$\left\langle \mathbf{H}, \mathbb{E}\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\beta}}^\top \right\rangle = \boldsymbol{\beta}^{*\top}\mathbf{H}\boldsymbol{\beta}^* + \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right). \tag{3.23}$$

Combining (3.18), (3.19), (3.20), (3.22), (3.23), we have

$$\begin{aligned}
S_1 =& \frac{M+1}{M}\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\boldsymbol{b}\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right) + \frac{1}{M}\mathrm{tr}\left(\boldsymbol{b}\boldsymbol{b}^\top\mathbf{H}\right)\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right)\\
&+ \frac{M+2}{M}u_{-1}^2\cdot\left(2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)^2\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}\\
&+ \frac{2(M+1)}{M}u_{-1}\boldsymbol{b}^\top\left[\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H} + \mathbf{H}\boldsymbol{\Psi}\mathbf{H}\right]\mathbf{A}\mathbf{H}\mathbf{u}_{12}\\
&+ u_{-1}^2\mathrm{tr}\left(\mathbf{A}^\top\left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\mathbf{A}\mathbf{H}\right) + \frac{4}{M}u_{-1}\cdot\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12}\\
&- 2\Big[\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\beta}^* + u_{-1}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\beta}^* + u_{-1}\mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}\right)\\
&+ \mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b}\Big] + \boldsymbol{\beta}^{*\top}\mathbf{H}\boldsymbol{\beta}^* + \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \boldsymbol{b}^\top\left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\boldsymbol{b}\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}.
\end{aligned} \tag{3.24}$$

Step 3: other terms. Let us compute S_2, S_3 and S_4. Using definitions, we rewrite $\mathbf{z}_2$ as

$$\begin{aligned}
\mathbf{z}_2^\top &= \mathbf{u}_{21}^\top\cdot\frac{1}{M}\mathbf{X}^\top\boldsymbol{\varepsilon}\cdot\mathbf{u}_{12}^\top + u_{-1}\cdot\frac{1}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\cdot\mathbf{U}_{11} + u_{-1}\cdot\frac{2}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\\
&= \boldsymbol{b}^\top\cdot\frac{1}{M}\mathbf{X}^\top\boldsymbol{\varepsilon}\cdot\mathbf{u}_{12}^\top + u_{-1}\cdot\frac{1}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\cdot\mathbf{A} + u_{-1}\cdot\frac{2}{M}\boldsymbol{\varepsilon}^\top\mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top.
\end{aligned} \tag{3.25}$$

Since $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M), \widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and they are independent, all terms in S_2 vanish except if the term contains even orders of $\widetilde{\boldsymbol{\theta}}$ or $\boldsymbol{\varepsilon}$. So we have

$$\begin{aligned} S_2 &= \left\langle \mathbf{H}, \mathbb{E}\mathbf{z}_2\mathbf{z}_2^\top \right\rangle = \frac{1}{M^2}\mathbb{E}\Big\{ \boldsymbol{b}^\top \mathbf{X}^\top \boldsymbol{\varepsilon} \cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \cdot \boldsymbol{\varepsilon}^\top \mathbf{X} \cdot \boldsymbol{b} + u_{-1}^2 \boldsymbol{\varepsilon}^\top \mathbf{X}\mathbf{A}\mathbf{H}\mathbf{A}^\top \mathbf{X}^\top \boldsymbol{\varepsilon} \\ &\quad + 2u_{-1} \cdot \boldsymbol{\varepsilon}^\top \mathbf{X}\mathbf{A}\mathbf{H}\mathbf{u}_{12} \cdot \boldsymbol{\varepsilon}^\top \mathbf{X}\boldsymbol{b} + 4u_{-1}^2 \cdot \boldsymbol{\varepsilon}^\top \mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}} \cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \cdot \widetilde{\boldsymbol{\theta}}^\top \boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{X}^\top \boldsymbol{\varepsilon} \Big\} \\ &= \frac{\sigma^2}{M}\Big\{ \boldsymbol{b}^\top \mathbf{H}\boldsymbol{b} \cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} + u_{-1}^2 \mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{A}^\top\right) \\ &\qquad\qquad + 2u_{-1}\mathbf{u}_{12}^\top \mathbf{H}\mathbf{A}^\top \mathbf{H}\boldsymbol{b} + 4u_{-1}^2 \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)\mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \Big\} \end{aligned} \tag{3.26}$$

For the other two terms, we have

$$S_3 = \frac{1}{M^2} u_{-1}^2 \mathbb{E}\Big\{ \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \Big\} = \frac{\sigma^4(M+2)}{M} u_{-1}^2 \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \tag{3.27}$$

and

$$\begin{aligned} S_4 &= 2\langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 - \widetilde{\boldsymbol{\beta}}\right)\mathbf{z}_3^\top \rangle \\ &= 2\mathbb{E}\Big\{ \left[\left(\mathbf{u}_{21} + u_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \cdot \frac{1}{M}\mathbf{X}^\top \mathbf{X} \cdot \left(\mathbf{U}_{11} + \widetilde{\boldsymbol{\beta}}\mathbf{u}_{12}^\top\right) - \widetilde{\boldsymbol{\beta}}^\top\right] \mathbf{H} \cdot \frac{1}{M}\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \cdot u_{-1}\mathbf{u}_{12}^\top \Big\} \\ &= 2\sigma^2 u_{-1}\mathbb{E}\Big\{ \left[\left(\boldsymbol{b} + u_{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)^\top \cdot \mathbf{H} \cdot \left(\mathbf{A} + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\mathbf{u}_{12}^\top\right) - \widetilde{\boldsymbol{\beta}}^{*\top} - \widetilde{\boldsymbol{\theta}}^\top \boldsymbol{\Psi}^{\frac{1}{2}}\right] \mathbf{H}\mathbf{u}_{12} \Big\} \\ &= 2\sigma^2 u_{-1}\left[\boldsymbol{b}^\top \mathbf{H}\mathbf{A} - \widetilde{\boldsymbol{\beta}}^{*\top}\right]\mathbf{H}\mathbf{u}_{12} + 2\sigma^2 u_{-1}^2 \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) \cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12}. \end{aligned} \tag{3.28}$$

Step 4: combine all parts. Combining the four parts (3.24), (3.26), (3.27) and (3.28), we have

$$\mathsf{Risk}(f) - \sigma^2 = S_1 + S_2 + S_3 + S_4$$

We remark that in the (3.24), (3.26), (3.27) and (3.28), the parameters are $\mathbf{A}, \boldsymbol{b}, u_{-1}$ and $\mathbf{u}_{12}$, instead of the original parameter $\mathbf{U}_{11}, \mathbf{u}_{12}, \mathbf{u}_{21}, u_{-1}$. From the definitions of $\mathbf{A}$ and $\boldsymbol{b}$, we know there is a bijective map between $(\mathbf{U}_{11}, \mathbf{u}_{12}, \mathbf{u}_{21}, u_{-1})$ and $(\mathbf{A}, \boldsymbol{b}, \mathbf{u}_{12}, u_{-1})$, so these two parameterizations are equivalent when computing the minimal risk achieved by a model in a hypothesis class. From the expression of S_1, S_2, S_3, S_4, we can rewrite the risk as

$$\mathsf{Risk}(f) - \sigma^2 = \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV}, \tag{3.29}$$

where

$$
\begin{aligned}
\mathrm{I} &= \boldsymbol{b}^\top \left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\boldsymbol{b}\cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} + \frac{\sigma^2}{M}\cdot \boldsymbol{b}^\top \mathbf{H}\boldsymbol{b}\cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12} \\
&\quad + u_{-1}^2 \mathrm{tr}\left(\mathbf{A}^\top\left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\mathbf{A}\mathbf{H}\right) + \frac{\sigma^2}{M}\cdot \mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{A}^\top\right) - 2\mathbf{u}_{12}^\top \mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b} \\
&\quad + 2u_{-1}\boldsymbol{b}^\top\left[\frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H} + \frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{\sigma^2}{M}\cdot\mathbf{H}\right]\mathbf{A}\mathbf{H}\mathbf{u}_{12} - 2u_{-1}\mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}\right); \\
\mathrm{II} &= \frac{1}{M}\left[\boldsymbol{b}^\top\mathbf{H}\boldsymbol{b}\cdot\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{H}\right) + 4u_{-1}\cdot\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12} + 4u_{-1}^2\cdot\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}\right]; \\
\mathrm{III} &= \boldsymbol{b}^\top\left(\frac{M+1}{M}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{A}^\top\mathbf{H}\right)\boldsymbol{b} \\
&\quad - 2\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\beta}^* + 2u_{-1}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\cdot\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12} + 2\sigma^2\boldsymbol{b}^\top\mathbf{H}\mathbf{A}\mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right); \\
\mathrm{IV} &= \boldsymbol{\beta}^{*\top}\mathbf{H}\boldsymbol{\beta}^* - 2\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)\cdot\boldsymbol{\beta}^{*\top}\mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right) + \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + u_{-1}^2\cdot\left[\left(2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right)\right.\right. \\
&\quad \left.\left. + \frac{M+2}{M}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)^2\right)\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12} + \left(2+\frac{4}{M}\right)\sigma^2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \frac{M+2}{M}\sigma^4\right]\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12}
\end{aligned}
$$

Step 5: lower bound the risk function. Let's first define a new matrix, which by definition is invertible:

$$
\boldsymbol{\Omega} := \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \frac{\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2}{M}\mathbf{I}_d. \tag{3.30}
$$

So we can write I as

$$
\begin{aligned}
\mathrm{I} &= \boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\boldsymbol{b}\cdot\mathbf{u}_{12}^\top\mathbf{H}\mathbf{u}_{12} + u_{-1}^2\mathrm{tr}\left(\mathbf{A}^\top\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}\right) + 2u_{-1}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}\mathbf{u}_{12} \\
&\quad - 2u_{-1}\mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}\right) - 2\mathbf{u}_{12}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b} \\
&= \mathrm{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\mathbf{u}_{12}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}} + u_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\right. \\
&\qquad \left.\cdot\left(\mathbf{H}^{\frac{1}{2}}\mathbf{u}_{12}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}} + u_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right] - \mathrm{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&\geq -\mathrm{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&= -\mathrm{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right),
\end{aligned}
$$

where the last line comes from the fact that $\boldsymbol{\Omega}$ and $\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}$ commute.

Next, we claim II ≥ 0. To see this, it suffices to notice that

$$\begin{aligned}
4u_{-1}\cdot \boldsymbol{b}^\top \mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{u}_{12} &\geq -4\left\|\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|u_{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{u}_{12}\right\|_F \\
&\geq -\left\|\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}}\right\|_F^2 \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 - 4\left\|u_{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{u}_{12}\right\|_F^2 \\
&\geq -\boldsymbol{b}^\top \mathbf{H}\boldsymbol{b}\cdot \operatorname{tr}\left(\mathbf{A}^\top \mathbf{H}\mathbf{A}\mathbf{H}\right) - 4u_{-1}^2 \cdot \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right)\cdot \mathbf{u}_{12}^\top \mathbf{H}\mathbf{u}_{12}, \quad (3.31)
\end{aligned}$$

where the last line comes from the fact that $\|\mathbf{A}\|_F^2 = \operatorname{tr}\left(\mathbf{A}\mathbf{A}^\top\right)$ for any matrix $\mathbf{A}$.

Then, let's consider III. Notice that actually, III can be viewed as a quadratic function of $\mathbf{A}^\top \mathbf{H}\boldsymbol{b}$, which can be easily minimized. More concretely, we have

$$\begin{aligned}
&\mathrm{III} + \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right) \\
=\ &\left[\mathbf{A}^\top \mathbf{H}\boldsymbol{b} - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right)\right]^\top \left(\frac{M+1}{M}\mathbf{H}\right) \\
&\qquad \cdot \left[\mathbf{A}^\top \mathbf{H}\boldsymbol{b} - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right)\right] \\
\geq\ &0. \quad (3.32)
\end{aligned}$$

Therefore, one has

$$\mathrm{III} \geq -\frac{M}{M+1}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right).$$

Combining the three parts above, one has

$$\begin{aligned}
&\mathsf{Risk}\left(f\right) - \sigma^2 \\
\geq\ &\mathrm{IV} - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \boldsymbol{\Omega}^{-1}\right) \\
&\qquad - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)u_{-1}\mathbf{u}_{12}\right) \\
=\ &\underbrace{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \boldsymbol{\Omega}^{-1}\right)}_{\inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f)-\sigma^2} + \frac{1}{M+1}\boldsymbol{\beta}^{*\top}\mathbf{H}\boldsymbol{\beta}^* - \frac{2\left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)}{M+1}\boldsymbol{\beta}^{*\top}\mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right) \\
&\qquad + \left[2\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)^2\right]\left(u_{-1}\mathbf{u}_{12}\right)^\top \mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right).
\end{aligned}$$

Therefore, one has

$$\begin{aligned}
\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f) \geq\ &\frac{1}{M+1}\boldsymbol{\beta}^{*\top}\mathbf{H}\boldsymbol{\beta}^* - \frac{2\left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)}{M+1}\boldsymbol{\beta}^{*\top}\mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right) \\
&+ \left[2\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)^2\right]\left(u_{-1}\mathbf{u}_{12}\right)^\top \mathbf{H}\left(u_{-1}\mathbf{u}_{12}\right).
\end{aligned}$$

The right hand side in the above inequality is a quadratic function of $u_{-1}\mathbf{u}_{12}$, so we can take its global minimizer:

$$u_{-1}\mathbf{u}_{12} = \frac{\frac{\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)}{M+1}}{2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}\boldsymbol{\beta}^*$$

to lower bound the risk gap as

$$\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f) \geq \left[\frac{1}{M+1} - \frac{\frac{\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)^2}{(M+1)^2}}{2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}\right]\left\|\boldsymbol{\beta}^*\right\|_{\mathbf{H}}^2.$$

Finally, to simplify the results, we notice that on the one hand,

$$\frac{\frac{\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)^2}{(M+1)^2}}{2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2} \leq \frac{\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}{2(M+1)^2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right)}.$$

On the other hand, one has

$$\frac{\frac{\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)^2}{(M+1)^2}}{2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2} \leq \frac{M}{(M+1)(3M+2)} \leq \frac{1}{3(M+1)}.$$

Therefore, we have

$$\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f) \geq \max\left\{\frac{2}{3(M+1)}, \frac{1}{M+1} - \frac{\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}{2(M+1)^2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right)}\right\}\cdot\left\|\boldsymbol{\beta}^*\right\|_{\mathbf{H}}^2.$$

When $\frac{1}{M+1} - \frac{\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)^2}{2(M+1)^2\mathrm{tr}(\mathbf{H\Psi H\Psi})} \geq \frac{2}{3(M+1)}$, we have $\frac{1}{M+1} \geq \frac{3\left(\mathrm{tr}(\mathbf{H\Psi})+\sigma^2\right)^2}{2(M+1)^2\mathrm{tr}(\mathbf{H\Psi H\Psi})}$, which implies

$$\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f) \geq \frac{\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}{(M+1)^2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right)}\cdot\left\|\boldsymbol{\beta}^*\right\|_{\mathbf{H}}^2.$$

Therefore, we finally have

$$\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{LTB}}}\mathsf{Risk}(f) \geq \max\left\{\frac{2}{3(M+1)}, \frac{\left(\mathrm{tr}\left(\mathbf{H\Psi}\right)+\sigma^2\right)^2}{(M+1)^2\mathrm{tr}\left(\mathbf{H\Psi H\Psi}\right)}\right\}\cdot\left\|\boldsymbol{\beta}^*\right\|_{\mathbf{H}}^2.$$

Note that this holds for an arbitrary $f \in \mathcal{F}_{\mathsf{LSA}}$, so the proof finishes by taking infimum on the left hand side. □

3.10 Appendix: Proof of Theorem 3.5.2

Proof. Recall that from Assumption 3.3.1,

$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{H}), \quad y = \mathbf{x}^\top \widetilde{\boldsymbol{\beta}} + \varepsilon, \quad \mathbf{X}[i] \sim \mathcal{N}(0, \mathbf{H}), \quad \mathbf{y} = \mathbf{X}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\varepsilon},$$

where

$$\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}\left(\boldsymbol{\beta}^*, \boldsymbol{\Psi}\right), \quad \varepsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\right), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \cdot \mathbf{I}_M\right).$$

Step 1: compute the risk function. From the independence of $\varepsilon, \boldsymbol{\varepsilon}$ with other random variables, we have

$$\begin{aligned}
\mathcal{R}_M\left(\boldsymbol{\beta}, \boldsymbol{\Gamma}\right) &= \mathbb{E}\left(\mathbf{x}^\top\left(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) + \varepsilon + \mathbf{x}^\top \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\left(\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}\right) - \mathbf{x}^\top \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \boldsymbol{\varepsilon}}{M}\right)^2 \\
&= \mathbb{E}\left(\mathbf{x}^\top\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right) \cdot \left(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right)^2 + \frac{1}{M^2}\mathbb{E}\left(\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{X}^\top \boldsymbol{\varepsilon}\right)^2 + \sigma^2 \\
&= \left\langle \mathbf{H}, \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right)^{\otimes 2} \circ \mathbb{E}\left(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)^{\otimes 2} \right\rangle + \frac{1}{M^2}\mathbb{E}\left(\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{X}^\top \boldsymbol{\varepsilon}\right)^2 + \sigma^2, \quad (3.33)
\end{aligned}$$

where the last line comes from the independence between $\widetilde{\boldsymbol{\beta}}$ and $\mathbf{X}, \mathbf{x}$, as well as the fact that $\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{H}\right)$ and the property of tensor product $\circ$. Note that, $\boldsymbol{\beta}$ ad $\boldsymbol{\Gamma}$ are learnable parameters here, and we should set them apart from the task vector $\widetilde{\boldsymbol{\beta}}$ or the prior mean vector $\widetilde{\boldsymbol{\beta}}^*$. Recall that $\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}\left(\boldsymbol{\beta}^*, \boldsymbol{\Psi}\right)$, we have $\mathbb{E}\left(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)^{\otimes 2} = \left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^{\otimes 2} + \boldsymbol{\Psi}$. Moreover, using the following

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{H}\right), \quad \mathbf{X}[i] \sim \mathcal{N}\left(\mathbf{0}, \mathbf{H}\right), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \cdot \mathbf{I}_M\right),$$

we have that

$$\mathbb{E}\left(\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{X}^\top \boldsymbol{\varepsilon}\right)^2 = \sigma^2 \mathbb{E}\mathrm{tr}\left(\mathbf{X}\boldsymbol{\Gamma}^\top \mathbf{x}\mathbf{x}^\top \boldsymbol{\Gamma}\mathbf{X}^\top\right) = M\sigma^2 \cdot \left\langle \mathbf{H}\boldsymbol{\Gamma}\mathbf{H}, \boldsymbol{\Gamma}^\top \right\rangle.$$

Bridging the two terms above into (3.33), we get

$$\begin{aligned}
\mathcal{R}_M\left(\boldsymbol{\beta}, \boldsymbol{\Gamma}\right) &= \left\langle \mathbf{H}, \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right)^{\otimes 2} \circ \left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^{\otimes 2} \right\rangle \\
&\quad + \left\langle \mathbf{H}, \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right)^{\otimes 2} \circ \boldsymbol{\Psi} + \frac{\sigma^2}{M}\boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Gamma}^\top \right\rangle + \sigma^2 \\
&= \underbrace{\left\langle \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right)^{\otimes 2} \circ \mathbf{H}, \left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^{\otimes 2} \right\rangle}_{V_1} \\
&\quad + \underbrace{\left\langle \mathbf{H}, \mathbb{E}\left(\left(\mathbf{I}_d - \boldsymbol{\Gamma} \cdot \frac{\mathbf{X}^\top \mathbf{X}}{M}\right)^\top\right)^{\otimes 2} \circ \boldsymbol{\Psi} + \frac{\sigma^2}{M}\boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Gamma}^\top \right\rangle}_{V_2} + \sigma^2 \quad (3.34)
\end{aligned}$$

First, let's compute V_1. From the definition above, we have

$$V_1 = (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}_{\boldsymbol{\Gamma}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*),$$

where

$$\begin{aligned}
\mathbf{H}_{\boldsymbol{\Gamma}} :&= \mathbb{E}\left(\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)^\top\right)^{\otimes 2}\circ\mathbf{H} && (3.35)\\
&= \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)^\top \mathbf{H}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right) \quad \text{(the definition of the tensor product)}\\
&= \mathbf{H} - \mathbf{H}\left(\boldsymbol{\Gamma}+\boldsymbol{\Gamma}^\top\right)\mathbf{H} + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\right)}{M}\mathbf{H} + \frac{M+1}{M}\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\mathbf{H}\\
&\qquad (\mathbf{X}[i]\sim\mathcal{N}(\mathbf{0},\mathbf{H}) \text{ and Lemma 2.10.2})\\
&= (\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H})^\top\mathbf{H}(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}) + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\right)}{M}\mathbf{H} + \frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\mathbf{H} \succeq \mathbf{0}. && (3.36)
\end{aligned}$$

Then, let's compute V_2. We have

$$\begin{aligned}
&\mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)^{\otimes 2}\circ\boldsymbol{\Psi} + \frac{\sigma^2}{M}\boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Gamma}^\top\\
&= \mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)\boldsymbol{\Psi}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)^\top + \frac{\sigma^2}{M}\boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Gamma}^\top\\
&= \boldsymbol{\Psi} - \boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Psi} - \boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}\left(\frac{\operatorname{tr}(\mathbf{H}\boldsymbol{\Psi})+\sigma^2}{M}\cdot\mathbf{H} + \frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\right)\boldsymbol{\Gamma}^\top.\\
&\qquad (\mathbf{X}[i]\overset{\text{i.i.d.}}{\sim}\mathcal{N}(\mathbf{0},\mathbf{H}) \text{ and the Lemma 2.10.2})
\end{aligned}$$

From the definition of $\boldsymbol{\Omega}$ in (3.14), we know that it is invertible. Moreover, it holds that

$$\begin{aligned}
&\mathbb{E}\left(\mathbf{I}_d - \boldsymbol{\Gamma}\cdot\frac{\mathbf{X}^\top\mathbf{X}}{M}\right)^{\otimes 2}\circ\boldsymbol{\Psi} + \frac{\sigma^2}{M}\boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Gamma}^\top = \boldsymbol{\Psi} - \boldsymbol{\Gamma}\mathbf{H}\boldsymbol{\Psi} - \boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^\top\\
&= \left(\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\left(\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top + \boldsymbol{\Psi} - \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
V_2 = &\operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right]\\
&+ \operatorname{tr}(\mathbf{H}\boldsymbol{\Psi}) - \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)
\end{aligned}$$

Step 2: solve the global minimum. Now we have

$$\begin{aligned}
&\mathsf{Risk}\left(\boldsymbol{\beta}, \boldsymbol{\Gamma}\right) - \sigma^2 \\
&= \left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top \left[\left(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}\right)^\top \mathbf{H} \left(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}\right) + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\right)}{M}\mathbf{H} + \frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^\top\mathbf{H}\boldsymbol{\Gamma}\mathbf{H}\right]\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right) \\
&+ \operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right] \\
&+ \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \qquad (3.37) \\
&\geq \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&= \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right),
\end{aligned}$$

where the last line comes from the fact that $\boldsymbol{\Omega}$ and $\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}$ commute. Taking infimum over all $\boldsymbol{\beta}, \boldsymbol{\Gamma}$ in the left hand side, we have

$$\inf_{f\in\mathcal{F}_{\text{GD-}\beta}} \mathsf{Risk}(f) - \sigma^2 = \inf_{\boldsymbol{\beta},\boldsymbol{\Gamma}} \mathsf{Risk}\left(\boldsymbol{\beta}, \boldsymbol{\Gamma}\right) - \sigma^2 \geq \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right).$$

On the other hand, we take

$$\boldsymbol{\beta} = \boldsymbol{\beta}^*, \quad \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^* := \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}, \qquad (3.38)$$

where $\mathbf{H}^{\frac{1}{2}}$ is the principal square root of $\mathbf{H}$ and $\mathbf{H}^{-\frac{1}{2}}$ is its Moore-Penrose pseudo inverse (see notation part in Section 3.8). Then, we have

$$\begin{aligned}
&\mathsf{Risk}\left(\boldsymbol{\beta}^*, \boldsymbol{\Gamma}^*\right) - \sigma^2 \\
&= \operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right] \\
&+ \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right).
\end{aligned}$$

Since

$$\begin{aligned}
\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1} &= \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1} \\
&= \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} \\
&\qquad (\boldsymbol{\Omega} \text{ and } \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} \text{ commute}) \\
&= \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}. \\
&\qquad (\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}})
\end{aligned}$$

Therefore, we have

$$\mathsf{Risk}\left(\boldsymbol{\beta}^*, \boldsymbol{\Gamma}^*\right) - \sigma^2 = \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right) \geq \inf_{\boldsymbol{\beta},\boldsymbol{\Gamma}} \mathsf{Risk}\left(\boldsymbol{\beta}, \boldsymbol{\Gamma}\right) - \sigma^2.$$

Combining both directions, we conclude that

$$\inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f) - \sigma^2 = \inf_{\boldsymbol{\beta},\boldsymbol{\Gamma}} \mathsf{Risk}\left(\boldsymbol{\beta},\boldsymbol{\Gamma}\right) - \sigma^2 = \operatorname{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right). \tag{3.39}$$

Step 3: identify all global minimizers Above we show that $\mathsf{Risk}\left(\boldsymbol{\beta}^*,\boldsymbol{\Gamma}^*\right)$ achieves the global minimum of the ICL risk over GD-$\boldsymbol{\beta}$ class. Now we will figure out the sufficient and necessary condition to achieve the global minimal risk. From the proof above, we know that for any $\boldsymbol{\beta}\in\mathbb{R}^d, \boldsymbol{\Gamma}\in\mathbb{R}^{d\times d}$, it holds that

$$\begin{aligned}
&\mathsf{Risk}\left(\boldsymbol{\beta},\boldsymbol{\Gamma}\right) = \inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \\
\iff &\begin{cases} V_1 = \left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}_{\boldsymbol{\Gamma}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right) = 0 \\ \operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right] = 0 \end{cases}
\end{aligned}$$

Since $\boldsymbol{\Omega}$ is invertible, the second equation is equivalent to

$$\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1} = \mathbf{0}_{d\times d},$$

which is a linear system of $\boldsymbol{\Gamma}$. Since $\boldsymbol{\Gamma}^*$ is proved to be one solution, all solutions of this equation are

$$\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^* + \left\{\mathbf{Z}\in\mathbb{R}^{d\times d} : \mathbf{H}^{\frac{1}{2}}\mathbf{Z}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}\right\} = \boldsymbol{\Gamma}^* + \mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right),$$

where the last equation comes from Lemma 3.15.2. Here, $+$ denotes Minkowski sum. This is defined as $A+B=\{a+b, a\in A, b\in B\}$ for two sets A, B and $a+B=\{a\}+B$ for an entry a and a set B. Under this condition, we know that

$$\mathbf{H}_{\boldsymbol{\Gamma}} = \mathbf{H} - \mathbf{H}\left(\boldsymbol{\Gamma}^* + \boldsymbol{\Gamma}_M^{*\top}\right)\mathbf{H} + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\right)}{M}\mathbf{H} + \frac{M+1}{M}\mathbf{H}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}.$$

Consider the following inequality:

$$\begin{aligned}
&\mathbf{I}_d - \mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\Gamma}^* + \boldsymbol{\Gamma}^{*\top}\right)\mathbf{H}^{\frac{1}{2}} + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\right)}{M}\mathbf{I}_d + \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} \\
\succeq\ &\mathbf{I}_d - \mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\Gamma}^* + \boldsymbol{\Gamma}^{*\top}\right)\mathbf{H}^{\frac{1}{2}} + \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} \\
=\ &\left(\sqrt{\frac{M}{M+1}}\mathbf{I}_d - \sqrt{\frac{M+1}{M}}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)\cdot\left(\sqrt{\frac{M}{M+1}}\mathbf{I}_d - \sqrt{\frac{M+1}{M}}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)^\top + \frac{M}{M+1}\mathbf{I}_d \\
\succ\ &\mathbf{0}_{d\times d}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^\top \mathbf{H}_{\boldsymbol{\Gamma}} (\boldsymbol{\beta}-\boldsymbol{\beta}^*) = 0 \\
\iff &\left[(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^\top \mathbf{H}^{\frac{1}{2}}\right] \left(\mathbf{I}_d - \mathbf{H}^{\frac{1}{2}} \left(\boldsymbol{\Gamma}^* + \boldsymbol{\Gamma}^{*\top}\right) \mathbf{H}^{\frac{1}{2}} + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\right)}{M}\mathbf{I}_d \right. \\
&\left. + \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}_M^{*\top}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} \right) \left[\mathbf{H}^{\frac{1}{2}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)\right] = 0 \\
\iff &\mathbf{H}^{\frac{1}{2}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*) = \mathbf{0}_d \iff \boldsymbol{\beta} = \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}}\right) \iff \boldsymbol{\beta} = \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right),
\end{aligned}$$

where $\mathsf{null}(\cdot)$ denotes the null space of a matrix and $+$ denotes the Minkowski sum. The last equivalence comes from Lemma 3.15.2. Therefore, we conclude that the $f_{\boldsymbol{\beta},\boldsymbol{\Gamma}}$ achieves the minimal ICL risk in GD-$\boldsymbol{\beta}$ class if and only if

$$\boldsymbol{\beta} = \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right), \quad \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^* + \mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right).$$

Specially, if $\mathbf{H}$ is positive definite, there is unique global minimizer in GD-$\boldsymbol{\beta}$ class with parameters

$$\boldsymbol{\beta} = \boldsymbol{\beta}^*, \quad \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^*.$$

Step 4: equivalence in the hypothesis class Finally, we show when $\mathbf{H}$ is rank-deficient, any global minimizer actually corresponds to one single function in $\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}$ almost surely. For an arbitrary global minimizer, we assume $\boldsymbol{\beta} = \boldsymbol{\beta}^* + \mathbf{h}, \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^* + \mathbf{Z}$, where $\mathbf{h} \in \mathsf{null}(\mathbf{H}), \mathbf{Z} \in \mathsf{Im}(\mathbf{H}^{\otimes 2})$. Suppose we have a prompt $\mathbf{X}, \mathbf{y}, \mathbf{x}, y$ which follows the Assumption 3.3.1 and the token matrix is formed by (3.1), we have

$$\begin{aligned}
f_{\boldsymbol{\beta},\boldsymbol{\Gamma}}(\mathbf{E}) &= \left\langle \boldsymbol{\beta} - \frac{\boldsymbol{\Gamma}}{M}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta}-\mathbf{y}), \mathbf{x} \right\rangle \\
&= \left\langle \boldsymbol{\beta}^* - \frac{\boldsymbol{\Gamma}^*}{M}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta}^*-\mathbf{y}), \mathbf{x} \right\rangle \\
&\qquad + \mathbf{h}^\top\mathbf{x} - \frac{1}{M}\mathbf{x}^\top\mathbf{Z}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}^* - \frac{1}{M}\mathbf{x}^\top\mathbf{Z}\mathbf{X}^\top\mathbf{X}\mathbf{h} - \frac{1}{M}\mathbf{x}^\top\boldsymbol{\Gamma}^*\mathbf{X}^\top\mathbf{X}\mathbf{h} + \mathbf{x}^\top\mathbf{Z}\mathbf{X}^\top\mathbf{y}.
\end{aligned}$$

Notice that $\mathbf{X}[i], \mathbf{x} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d, \mathbf{H})$, we have

$$\mathbb{E}\left(\mathbf{h}^\top\mathbf{x}\right)^2 = \mathbf{h}^\top\mathbf{H}\mathbf{h} = 0$$

since $\mathbf{h} \in \mathsf{null}(\mathbf{H})$. Therefore, we know $\mathbf{h}^\top\mathbf{x} = 0$ almost surely, and similarly, $\mathbf{X}\mathbf{h} = \mathbf{0}_d$ almost surely. Finally, we have

$$\mathbb{E}\left[\left(\frac{1}{M}\mathbf{x}^\top\mathbf{Z}\mathbf{X}^\top\right)\cdot\left(\frac{1}{M}\mathbf{x}^\top\mathbf{Z}\mathbf{X}^\top\right)^\top\right] = \frac{1}{M}\mathbb{E}\operatorname{tr}\left[\mathbf{H}\mathbf{Z}\mathbf{H}\right] = 0,$$

which imples $\mathbf{x}^\top \mathbf{Z}\mathbf{X}^\top = \mathbf{0}_d$ almost surely. Therefore, we conclude

$$f_{\boldsymbol{\beta},\boldsymbol{\Gamma}}(\mathbf{E}) = f_{\boldsymbol{\beta}^*,\boldsymbol{\Gamma}^*}(\mathbf{E}) \quad \text{almost surely.}$$

□

3.11 Appendix: Proof of Theorem 3.5.3

Proof. Recall the definition of Linear Transformer Block class:

$$f_{\mathsf{LTB}} : \mathbb{R}^{(d+1)\times(M+1)} \to \mathbb{R}$$
$$\mathbf{E} \mapsto \left[\mathbf{W}_2^\top \mathbf{W}_1 \left(\mathbf{E} + \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right)\right]_{-1,-1},$$

where $\mathbf{E}$ is the token matrix defined by (3.1). Let's take an arbitrary function f in the LTB class with trainable matrices

$$\boldsymbol{W}^K,\ \boldsymbol{W}^Q \in \mathbb{R}^{d_k\times(d+1)}, \quad \boldsymbol{W}^P,\ \boldsymbol{W}^V \in \mathbb{R}^{d_v\times(d+1)}, \quad \mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_f\times(d+1)}.$$

Similar to the proof of the Theorem 3.4.1, we know that only the last row of $\mathbf{W}_2^\top \mathbf{W}_1$ and the first d-columns of $\left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q$ attend the prediction. Therefore, we denote

$$\mathbf{W}_2^\top \mathbf{W}_1 = \begin{pmatrix} * & * \\ \boldsymbol{\gamma}^\top & * \end{pmatrix}, \quad \mathbf{W}_2^\top \mathbf{W}_1 \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V = \begin{pmatrix} * & * \\ \mathbf{v}_{21}^\top & v_{-1} \end{pmatrix}, \quad \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q = \begin{pmatrix} \mathbf{V}_{11} & * \\ \mathbf{v}_{12}^\top & * \end{pmatrix}, \tag{3.40}$$

where $\boldsymbol{\gamma}, \mathbf{v}_{12}, \mathbf{v}_{21}, \in \mathbb{R}^d, v_{-1} \in \mathbb{R}, \mathbf{V}_{11} \in \mathbb{R}^{d\times d}$, and $*$ denotes entries that do not enter the prediction. Then, the prediction of LTB function can be written as

$$\begin{aligned} f(\mathbf{E}) &= \boldsymbol{\gamma}^\top \mathbf{x} + \begin{pmatrix} \mathbf{v}_{21}^\top & v_{-1} \end{pmatrix} \cdot \frac{\mathbf{E}\mathbf{M}_M\mathbf{E}^\top}{M} \cdot \begin{pmatrix} \mathbf{V}_{11} \\ \mathbf{v}_{12}^\top \end{pmatrix} \mathbf{x} \\ &= \left[\boldsymbol{\gamma}^\top + \mathbf{v}_{21}^\top \frac{\mathbf{X}^\top\mathbf{X}}{M}\mathbf{V}_{11} + \mathbf{v}_{21}^\top \frac{\mathbf{X}^\top\mathbf{y}}{M}\mathbf{v}_{12}^\top + v_{-1}\frac{\mathbf{y}^\top\mathbf{X}}{M}\mathbf{V}_{11} + v_{-1}\frac{\mathbf{y}^\top\mathbf{y}}{M}\mathbf{v}_{12}^\top\right]\mathbf{x}. \end{aligned}$$

Step 1: simplify the risk function. We use $\widetilde{\boldsymbol{\beta}}$ to denote the task parameter. From the Assumption 3.3.1 and Definition 3.8.1, we have

$$\mathbf{y} = \mathbf{X}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}, \quad y = \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle + \varepsilon, \quad \widetilde{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^*, \boldsymbol{\Psi}), \quad \widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}};$$

and

$$\mathbf{X}[i], \mathbf{x} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{H}), \quad \boldsymbol{\varepsilon}[i], \varepsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right), \quad \widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d).$$

Then the model output can be written as

$$\begin{aligned} f(\mathbf{E}) &= \left[\boldsymbol{\gamma}^\top + \mathbf{v}_{21}^\top \frac{\mathbf{X}^\top \mathbf{X}}{M} \mathbf{V}_{11} + \mathbf{v}_{21}^\top \frac{\mathbf{X}^\top \mathbf{y}}{M} \mathbf{v}_{12}^\top + v_{-1} \frac{\mathbf{y}^\top \mathbf{X}}{M} \mathbf{V}_{11} + v_{-1} \frac{\mathbf{y}^\top \mathbf{y}}{M} \mathbf{v}_{12}^\top\right] \mathbf{x} \\ &= \left[\boldsymbol{\gamma}^\top + \left(\mathbf{v}_{21} + v_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \frac{\mathbf{X}^\top \mathbf{X}}{M} \left(\mathbf{V}_{11} + \widetilde{\boldsymbol{\beta}} \mathbf{v}_{12}^\top\right)\right] \mathbf{x} \\ &\quad + \left[\mathbf{v}_{21}^\top \frac{\mathbf{X}^\top \boldsymbol{\varepsilon}}{M} \mathbf{v}_{12}^\top + v_{-1} \frac{\boldsymbol{\varepsilon}^\top \mathbf{X}}{M} \mathbf{V}_{11} + v_{-1} \cdot \frac{2}{M} \boldsymbol{\varepsilon}^\top \mathbf{X} \widetilde{\boldsymbol{\beta}} \mathbf{v}_{12}^\top + \frac{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}}{M} \cdot v_{-1} \mathbf{v}_{12}^\top\right] \mathbf{x}. \end{aligned}$$

To simplify the presentation, we denote

$$\begin{aligned} \mathbf{z}_1^\top &= \left(\mathbf{v}_{21} + v_{-1}\widetilde{\boldsymbol{\beta}}\right)^\top \cdot \frac{1}{M} \mathbf{X}^\top \mathbf{X} \cdot \left(\mathbf{V}_{11} + \widetilde{\boldsymbol{\beta}} \mathbf{v}_{12}^\top\right), \\ \mathbf{z}_2^\top &= \mathbf{v}_{21}^\top \cdot \frac{1}{M} \mathbf{X}^\top \boldsymbol{\varepsilon} \cdot \mathbf{v}_{12}^\top + v_{-1} \cdot \frac{1}{M} \boldsymbol{\varepsilon}^\top \mathbf{X} \cdot \mathbf{V}_{11} + v_{-1} \cdot \frac{2}{M} \boldsymbol{\varepsilon}^\top \mathbf{X} \widetilde{\boldsymbol{\beta}} \mathbf{v}_{12}^\top \\ \mathbf{z}_3^\top &= \frac{1}{M} \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \cdot v_{-1} \mathbf{v}_{12}^\top. \end{aligned}$$

Since $\mathbf{x}, \mathbf{X}, \boldsymbol{\varepsilon}, \widetilde{\boldsymbol{\beta}}$ are independent, we have

$$\begin{aligned} \mathsf{Risk}\,(f) &= \mathbb{E}\left(f(\mathbf{E}) - \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle - \varepsilon\right)^2 \\ &= \sigma^2 + \mathbb{E}\left(f(\mathbf{E}) - \langle \widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle\right)^2 \ (\varepsilon \text{ is independent from other variables and zero-mean}) \\ &= \mathbb{E}\left[\langle \mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle^2\right] + \sigma^2 \\ &= \langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}\right)^\top \rangle + \sigma^2. \end{aligned}$$

Note that $\mathbf{z}_1$ does not contain $\boldsymbol{\varepsilon}$, $\mathbf{z}_2$ is a linear form of $\boldsymbol{\varepsilon}$, and $\mathbf{z}_3$ is a quadratic form of $\boldsymbol{\varepsilon}$. Using $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$, we have $\mathbb{E}[(\mathbf{z}_1 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}) \cdot \mathbf{z}_2^\top] = \mathbf{0}$ and $\mathbb{E}[\mathbf{z}_2 \mathbf{z}_3^\top] = \mathbf{0}$. Therefore, we have

$$\begin{aligned} \mathsf{Risk}\,(f) - \sigma^2 = &\underbrace{\langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}\right)\left(\mathbf{z}_1 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}\right)^\top \rangle}_{S_1'} + \underbrace{\langle \mathbf{H}, \mathbb{E}\mathbf{z}_2 \mathbf{z}_2^\top \rangle}_{S_2'} + \underbrace{\langle \mathbf{H}, \mathbb{E}\mathbf{z}_3 \mathbf{z}_3^\top \rangle}_{S_3'} \\ &+ \underbrace{2\langle \mathbf{H}, \mathbb{E}\left(\mathbf{z}_1 + \boldsymbol{\gamma} - \widetilde{\boldsymbol{\beta}}\right) \mathbf{z}_3^\top \rangle}_{S_4'}. \end{aligned} \tag{3.41}$$

Step 2: compute the risk function. Let's compute the risk function. Note that, the risk function is in the same form as the risk function of LSA layer, which is in the proof of Theorem 3.4.1 (see Section 3.9 and equation (3.16)). There are two differences: one is we replace $\mathbf{U}_{11}, \mathbf{u}_{12}, \mathbf{u}_{21}, u_{-1}$ here with $\mathbf{V}_{11}, \mathbf{v}_{12}, \mathbf{v}_{21}, v_{-1}$. The second difference is that we replace $\widetilde{\boldsymbol{\beta}}$ in (3.16) with $\widetilde{\boldsymbol{\beta}} - \boldsymbol{\gamma}$. This is equivalent to replacing $\boldsymbol{\beta}^*$ with $\boldsymbol{\beta}^* - \boldsymbol{\gamma}$, since

$\widetilde{\boldsymbol{\beta}} - \boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\beta}^* - \boldsymbol{\gamma}, \boldsymbol{\Psi})$. Therefore, similar to (3.29), the risk function in (3.41) can be written as

$$\mathsf{Risk}(f) - \sigma^2 = \mathrm{V} + \mathrm{VI} + \mathrm{VII} + \mathrm{VIII}, \tag{3.42}$$

where

$$\begin{aligned}
\mathrm{V} &= \boldsymbol{b}^\top \left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\boldsymbol{b}\cdot \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12} + \frac{\sigma^2}{M}\cdot \boldsymbol{b}^\top \mathbf{H}\boldsymbol{b}\cdot \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12} \\
&\quad + v_{-1}^2 \mathrm{tr}\left(\mathbf{A}^\top \left(\frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H}\right)\mathbf{A}\mathbf{H}\right) + \frac{\sigma^2}{M}\cdot \mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{A}^\top\right) - 2\mathbf{v}_{12}^\top \mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{b} \\
&\quad + 2v_{-1}\boldsymbol{b}^\top \left[\frac{1}{M}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\mathbf{H} + \frac{M+1}{M}\mathbf{H}\boldsymbol{\Psi}\mathbf{H} + \frac{\sigma^2}{M}\cdot \mathbf{H}\right]\mathbf{A}\mathbf{H}\mathbf{v}_{12} - 2v_{-1}\mathrm{tr}\left(\mathbf{H}\mathbf{A}\mathbf{H}\boldsymbol{\Psi}\right); \\
\mathrm{VI} &= \frac{1}{M}\left[\boldsymbol{b}^\top \mathbf{H}\boldsymbol{b}\cdot \mathrm{tr}\left(\mathbf{A}^\top \mathbf{H}\mathbf{A}\mathbf{H}\right) + 4v_{-1}\cdot \boldsymbol{b}^\top \mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{v}_{12} + 4v_{-1}^2\cdot \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right)\cdot \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12}\right]; \\
\mathrm{VII} &= \boldsymbol{b}^\top \left(\frac{M+1}{M}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{A}^\top \mathbf{H}\right)\boldsymbol{b} \\
&\quad - 2\boldsymbol{b}^\top \mathbf{H}\mathbf{A}\mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right) + 2v_{-1}\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})\cdot \boldsymbol{b}^\top \mathbf{H}\mathbf{A}\mathbf{H}\mathbf{v}_{12} + 2\sigma^2 \boldsymbol{b}^\top \mathbf{H}\mathbf{A}\mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right); \\
\mathrm{VIII} &= \left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right) - 2\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)\cdot \left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right) + \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12}\cdot v_{-1}^2 \cdot \\
&\quad \left[\left(2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{M+2}{M}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)^2\right)\cdot \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12} + \left(2 + \frac{4}{M}\right)\sigma^2 \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)\right. \\
&\quad \left. + \frac{M+2}{M}\sigma^4\right] + \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)
\end{aligned}$$

and $\boldsymbol{\Omega} = \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \frac{\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})+\sigma^2}{M}\cdot \mathbf{I}_d$, and $\boldsymbol{b} := \mathbf{v}_{21} + v_{-1}\boldsymbol{\beta}^* \in \mathbb{R}^d, \mathbf{A} := \mathbf{V}_{11} + \boldsymbol{\beta}^* \mathbf{v}_{12}^\top \in \mathbb{R}^{d\times d}$,

Step 3: solve the global minimum of the risk function. This is very similar to step 5 in Appendix 3.9. The V term in (3.42) is actually equal to the I term in (3.29), so we have

$$\begin{aligned}
\mathrm{V} &= \mathrm{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}} + v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\boldsymbol{\Omega}\right. \\
&\qquad \left.\left(\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}} + v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)^\top\right] - \mathrm{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&\geq -\mathrm{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&= -\mathrm{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \boldsymbol{\Omega}^{-1}\right),
\end{aligned}$$

where the last line comes from the fact that $\mathbf{\Omega}$ and $\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}$ commute. For the same reason, we know that the VI term above is equal to the II term in (3.29), which implies VI ≥ 0, since

$$\begin{aligned}
4v_{-1}\cdot \boldsymbol{b}^\top \mathbf{H}\mathbf{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{v}_{12} &\geq -4\left\|\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F \cdot \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F \\
&\geq -\left\|\boldsymbol{b}^\top \mathbf{H}^{\frac{1}{2}}\right\|_F^2 \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 - 4\left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F^2 \\
&= -\boldsymbol{b}^\top \mathbf{H}\boldsymbol{b}\cdot \operatorname{tr}\left(\mathbf{A}^\top \mathbf{H}\mathbf{A}\mathbf{H}\right) - 4v_{-1}^2 \cdot \operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\mathbf{H}\mathbf{\Psi}\right)\cdot \mathbf{v}_{12}^\top \mathbf{H}\mathbf{v}_{12}, \qquad (3.43)
\end{aligned}$$

where the last line comes from the fact that $\|\mathbf{A}\|_F^2 = \operatorname{tr}\left(\mathbf{A}\mathbf{A}^\top\right)$ for any matrix $\mathbf{A}$. The term VII above is equal to III term in (3.29), except that we replace $\boldsymbol{\beta}^*$ with $\boldsymbol{\beta}^* - \boldsymbol{\gamma}$. Therefore, we have

$$\begin{aligned}
&\mathrm{VII} + \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right) \\
= &\left[\mathbf{A}^\top \mathbf{H}\boldsymbol{b} - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right)\right]^\top \left(\frac{M+1}{M}\mathbf{H}\right) \\
&\qquad \cdot \left[\mathbf{A}^\top \mathbf{H}\boldsymbol{b} - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right)\right] \\
\geq &\ 0, \qquad (3.44)
\end{aligned}$$

which implies

$$\mathrm{VII} \geq -\frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right).$$

Combining the three parts above, one has

$$\begin{aligned}
&\mathsf{Risk}\left(f\right) - \sigma^2 \\
\geq\ &\mathrm{VIII} - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \mathbf{\Omega}^{-1}\right) \\
&-\frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right) v_{-1}\mathbf{v}_{12}\right) \\
=\ &\underbrace{\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \mathbf{\Omega}^{-1}\right)}_{\inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f)-\sigma^2} + \frac{1}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right) \\
&\qquad - \frac{2\left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right)}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right) \\
&\qquad + \left[2\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\mathbf{H}\mathbf{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\operatorname{tr}\left(\mathbf{H}\mathbf{\Psi}\right) + \sigma^2\right)^2\right]\left(v_{-1}\mathbf{v}_{12}\right)^\top \mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right).
\end{aligned}$$

Here, the global minimum of GD-$\boldsymbol{\beta}$ class is taken from Theorem 3.5.2, whose proof does not rely on the proof here. Therefore, one has

$$\begin{aligned}
&\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \\
\geq\ & \frac{1}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right) - \frac{2\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)}{M+1}\cdot\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)^\top \mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right) \\
+\ & \left[2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)^2\right]\left(v_{-1}\mathbf{v}_{12}\right)^\top \mathbf{H}\left(v_{-1}\mathbf{v}_{12}\right).
\end{aligned}$$

The right-hand side in the above inequality is a quadratic function of $v_{-1}\mathbf{v}_{12}$, so we can take its global minimizer:

$$v_{-1}\mathbf{v}_{12} = \frac{\frac{\left(\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})+\sigma^2\right)}{M+1}}{2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)^2}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right)$$

to lower-bound the risk gap as

$$\begin{aligned}
&\mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \\
\geq\ & \left[\frac{1}{M+1} - \frac{\frac{\left(\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})+\sigma^2\right)^2}{(M+1)^2}}{2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)^2}\right]\left\|\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right\|_{\mathbf{H}}^2 \geq 0.
\end{aligned}$$

Note that, taking the infimum on the left-hand side, we get

$$\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \geq 0.$$

On the other hand, in the main text we have showed that $\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}} \subset \mathcal{F}_{\mathsf{LTB}}$, which implies

$$\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}\left(f\right) - \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) \leq 0.$$

Therefore, we conclude

$$\inf_{f\in\mathcal{F}_{\mathsf{LTB}}} \mathsf{Risk}\left(f\right) = \inf_{f\in\mathcal{F}_{\mathsf{GD}\text{-}\boldsymbol{\beta}}} \mathsf{Risk}(f) = \mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) - \mathrm{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2\boldsymbol{\Omega}^{-1}\right),$$

where the last equation is from Theorem 3.5.2.

Step 4: sufficient and necessary conditions for global minimizers. Let's now verify the conditions for the global minimizers. For a function in LTB class, the sufficient and necessary condition for it to be a global minimizer is that inequalities in the above step all hold, which are

$$\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}} + v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}, \tag{3.45}$$

$$\left\|\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}\right\|_F^2\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 = 4\left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F^2\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F^2, \tag{3.46}$$

$$v_{-1}\cdot\boldsymbol{b}^\top\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\mathbf{A}\mathbf{H}\mathbf{v}_{12} = \left\|\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}\right\|_F\cdot\left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F\cdot\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F\cdot\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F, \tag{3.47}$$

$$\mathbf{H}^{\frac{1}{2}}\left[\mathbf{A}^\top\mathbf{H}\boldsymbol{b} - \frac{M}{M+1}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma} - \left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right) + \sigma^2\right)v_{-1}\mathbf{v}_{12}\right)\right] = \mathbf{0}_{d\times d}, \tag{3.48}$$

$$v_{-1}\mathbf{v}_{12} = \frac{\frac{\left(\mathrm{tr}(\mathbf{H}\boldsymbol{\Psi})+\sigma^2\right)}{M+1}}{2\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\mathbf{H}\boldsymbol{\Psi}\right) + \frac{3M+2}{M(M+1)}\left(\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Psi}\right)+\sigma^2\right)^2}\left(\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right) \tag{3.49}$$

$$\left\|\boldsymbol{\beta}^* - \boldsymbol{\gamma}\right\|_{\mathbf{H}} = 0. \tag{3.50}$$

Let's first verify the necessary conditions of the system defined above. From (3.50) and Lemma 3.15.1, we know $\boldsymbol{\gamma}\in\boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}}\right) = \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right)$. Then, we have $v_{-1}\mathbf{v}_{12}\in\mathsf{null}\left(\mathbf{H}\right)$. Then, in (3.46), we know

$$\left\|\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}\right\|_F^2\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}}\right\|_F^2 = 4\left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F^2\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F^2 = 4\left\|\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right\|_F^2\left\|v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\right\|_F^2 = 0,$$

which implies either $\mathbf{H}^{\frac{1}{2}}\boldsymbol{b} = \mathbf{0}_d$ or $\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}$. If we assume $\mathbf{H}^{\frac{1}{2}}\mathbf{A}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}$, then (3.45) implies $\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}$. From our assumption, we know $\mathrm{rank}(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}^{\frac{1}{2}})\geq 2$, which implies $\mathrm{rank}(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}})\geq 2$, since for any matrix $\mathbf{Z}$, it holds that $\mathrm{rank}(\mathbf{Z}) = \mathrm{rank}(\mathbf{Z}\mathbf{Z}^\top)$. Since multiplication by an invertible matrix does not change the rank, we know $\mathrm{rank}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\right)\geq 2$, while $\mathbf{H}^{\frac{1}{2}}\mathbf{v}_{12}\boldsymbol{b}^\top\mathbf{H}^{\frac{1}{2}}$ is a matrix of rank at most one, which contradicts with (3.45). Therefore, we have

$$\mathbf{H}^{\frac{1}{2}}\boldsymbol{b} = \mathbf{0}, \tag{3.51}$$

$$v_{-1}\mathbf{H}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}. \tag{3.52}$$

Recall in Theorem 3.5.2, we have defined

$$\boldsymbol{\Gamma}^* := \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}.$$

Simple calculation shows

$$\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} = \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} = \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1},$$

where the second and the last equalities come from the fact that $\boldsymbol{\Omega}$ and $\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}$ commute, and the third equality comes from the property of Moor Penrose pseudo-inverse. Therefore,

one solution of (3.52) is $v_{-1}\mathbf{A} = \mathbf{\Gamma}^{*\top}$. Since (3.52) is a linear equation to $v_{-1}\mathbf{A}$, we know its full solution is

$$v_{-1}\mathbf{A} \in \mathbf{\Gamma}^{*\top} + \left\{\mathbf{Z} : \mathbf{H}^{\frac{1}{2}}\mathbf{Z}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}\right\} = \mathbf{\Gamma}^{*\top} + \mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right).$$

The solution to (3.51) is

$$\mathbf{v}_{21} = -v_{-1}\boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}}\right) = -v_{-1}\boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right).$$

Therefore, we know the necessary conditions for (3.45) to (3.50) are

$$\begin{cases} v_{-1} \neq 0, \\ v_{-1}\mathbf{v}_{12} \in \mathsf{null}\left(\mathbf{H}\right), \\ \mathbf{v}_{21} = -v_{-1}\boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right), \\ v_{-1}\mathbf{V}_{11} \in \mathbf{\Gamma}^{*\top} - v_{-1}\boldsymbol{\beta}^*\mathbf{v}_{12}^\top + \mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right), \\ \boldsymbol{\gamma} = \boldsymbol{\beta}^* + \mathsf{null}\left(\mathbf{H}\right) \end{cases} \tag{3.53}$$

It is easy to verify these equations above are also sufficient conditions of (3.45) to (3.50) by directly replacing each variable with its value and validating equaions from (3.45) to (3.50).

Specially, if $\mathbf{H}$ is positive definite and hence, invertible, the global minimizer is unique up to a scaling to v_{-1} :

$$v_{-1} \neq 0, \quad \mathbf{v}_{12} = \mathbf{0}_d, \quad \mathbf{v}_{21} = -v_{-1}\boldsymbol{\beta}^*, \quad \mathbf{V}_{11} = \frac{1}{v_{-1}} \cdot \mathbf{\Gamma}^{*\top}, \quad \boldsymbol{\gamma} = \boldsymbol{\beta}^*. \tag{3.54}$$

Step 5: equivalence in the hypothesis class. Finally, we will prove that any global minimizer of $\mathcal{F}_{\mathsf{LTB}}$ is actually equivalent to one single function in $\mathcal{F}_{\mathsf{LTB}}$ almost surely. More concretely, let's take a function $f \in \mathcal{F}_{\mathsf{LTB}}$ with parameters $\boldsymbol{W}^K$, $\boldsymbol{W}^Q$, $\boldsymbol{W}^P$, $\boldsymbol{W}^V, \mathbf{W}_1$, $\mathbf{W}_2$ and recall the parameter transformation in (3.40). We assume equations in (3.53) and Assumption 3.3.1 hold. Then, for any vector $\mathbf{a} \in \mathsf{null}\left(\mathbf{H}\right)$, one has $\mathbf{x}^\top\mathbf{a} = \mathbf{x}_i^\top\mathbf{a} = 0$ since $\mathbf{x}, \mathbf{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\mathbf{0}_d, \mathbf{H}\right)$. We denote

$$\mathbf{v}_{21} = -v_{-1}\boldsymbol{\beta}^* + \mathbf{a}_1, \quad v_{-1}\mathbf{V}_{11} = \mathbf{\Gamma}^{*\top} - v_{-1}\boldsymbol{\beta}^*\mathbf{v}_{12}^\top + \mathbf{Z}, \quad \boldsymbol{\gamma} = \boldsymbol{\beta}^* + \mathbf{a}_2,$$

where $\mathbf{a}_1, \mathbf{a}_2 \in \mathbf{H}, \mathbf{H}^{\frac{1}{2}}\mathbf{Z}\mathbf{H}^{\frac{1}{2}} = \mathbf{0}_{d\times d}$. Then, we have

$$\begin{aligned}
f(\mathbf{E}) &= \left[\mathbf{W}_2^\top \mathbf{W}_1 \left(\mathbf{E} + \left(\boldsymbol{W}^P\right)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top \left(\boldsymbol{W}^K\right)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right)\right]_{-1,-1} \\
&= (\boldsymbol{\beta}^* + \mathbf{a}_2)^\top \mathbf{x} + \begin{pmatrix} -v_{-1}\boldsymbol{\beta}^{*\top} + \mathbf{a}_1^\top & v_{-1}\end{pmatrix} \cdot \frac{1}{M}\begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{pmatrix} \cdot \begin{pmatrix} \mathbf{V}_{11} \\ \mathbf{v}_{12}^\top\end{pmatrix} \cdot \mathbf{x} \\
&= \boldsymbol{\beta}^{*\top}\mathbf{x} + \begin{pmatrix} -\boldsymbol{\beta}^{*\top} & 1\end{pmatrix} \cdot \frac{1}{M}\begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{pmatrix} \cdot \begin{pmatrix} v_{-1}\mathbf{V}_{11}\mathbf{x} \\ v_{-1}\mathbf{v}_{12}^\top\mathbf{x}\end{pmatrix} \qquad (\mathbf{X}\mathbf{a}_1 = \mathbf{0}, \mathbf{a}_2^\top\mathbf{x} = 0) \\
&= \boldsymbol{\beta}^{*\top}\mathbf{x} + \begin{pmatrix} -\boldsymbol{\beta}^{*\top} & 1\end{pmatrix} \cdot \frac{1}{M}\begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\Gamma}^{*\top}\mathbf{x} \\ \mathbf{0}_d^\top\end{pmatrix} \qquad (\mathbf{v}_{12}^\top\mathbf{x} = 0, \mathbf{X}\mathbf{V}_{11}\mathbf{x} = \mathbf{0}) \\
&= \left\langle \boldsymbol{\beta}^* - \frac{\boldsymbol{\Gamma}^*}{M}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}), \mathbf{x}\right\rangle = f_{\boldsymbol{\beta}^*, \boldsymbol{\Gamma}^*}(\mathbf{E}),
\end{aligned}$$

where $f_{\boldsymbol{\beta}^*,\boldsymbol{\Gamma}^*}(\cdot)$ is the GD-$\boldsymbol{\beta}$ function defined in (3.7). □

3.12 Appendix: Proof of Corollary 3.6.2

Proof. We denote $\phi_1 \geq \phi_2 \geq ... \geq \phi_d \geq 0$ are ordered eigenvalues of $\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{H}\boldsymbol{\Psi}^{\frac{1}{2}}$. From Theorem 3.5.2, we have

$$\inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f) - \sigma^2 = \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) - \operatorname{tr}\left(\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)^2 \boldsymbol{\Omega}^{-1}\right),$$

where

$$\boldsymbol{\Omega} := \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \frac{\operatorname{tr}(\mathbf{H}\boldsymbol{\Psi}) + \sigma^2}{M}\cdot\mathbf{I}_d.$$

Therefore, we have

$$\begin{aligned}
\inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f) - \sigma^2 &= \operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} \cdot \left(\boldsymbol{\Omega} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)\boldsymbol{\Omega}^{-1}\right) \\
&= \frac{1}{M}\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} \cdot \left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\cdot\mathbf{I}_d\right)\right).
\end{aligned}$$

Since

$$\left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\cdot\mathbf{I}_d \preceq \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\cdot\mathbf{I}_d \preceq 2\left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\cdot\mathbf{I}_d,$$

we have

$$\begin{aligned}
\inf_{f\in\mathcal{F}_{\text{GD-}\boldsymbol{\beta}}} \mathsf{Risk}(f) - \sigma^2 &\simeq \frac{\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2}{M}\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) \\
&= \bar{\phi}\cdot\sum_{i=1}^d \frac{\phi_i}{\frac{M+1}{M}\phi_i + \bar{\phi}} \\
&\simeq \sum_{i=1}^d \min\left\{\phi_i, \bar{\phi}\right\}.
\end{aligned}$$

Therefore, we conclude. □

3.13 Appendix: Proof of Theorem 3.6.1

Proof of first equation

Proof. From the classical bias-variance decomposition, we know

$$\begin{aligned}\ell(g;\mathbf{X}) &:= \mathbb{E}\left[(g(\mathbf{X},\mathbf{y},\mathbf{x})-y)^2 \mid \mathbf{X}\right] \\ &= \mathbb{E}\left[(g(\mathbf{X},\mathbf{y},\mathbf{x})-\mathbb{E}\left[y \mid \mathbf{X},\mathbf{y},\mathbf{x}\right])^2 \mid \mathbf{X}\right] + \mathbb{E}\left[(\mathbb{E}\left[y \mid \mathbf{X},\mathbf{y},\mathbf{x}\right]-y)^2 \mid \mathbf{X}\right]\end{aligned}$$

since the cross term vanishes. The second term does not depend on g and hence, the Bayesian optimal estimator is given by the posterior mean, i.e.,

$$\widehat{y}_{\mathsf{Bayes}} = \mathbb{E}\left[y \mid \mathbf{X},\mathbf{y},\mathbf{x}\right] = \langle \mathbb{E}\left[\widetilde{\boldsymbol{\beta}} \mid \mathbf{X},\mathbf{y},\mathbf{x}\right], \mathbf{x}\rangle.$$

Since $\widetilde{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^*, \boldsymbol{\Psi})$, we have there exists a random vector $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ such that $\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}$ almost surely. Therefore, one has

$$\widehat{y}_{\mathsf{Bayes}} = \langle \boldsymbol{\beta}^*, \mathbf{x}\rangle + \left\langle \mathbb{E}\left[\widetilde{\boldsymbol{\theta}} \mid \mathbf{X},\mathbf{y},\mathbf{x}\right], \boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{x}\right\rangle.$$

To compute $\mathbb{E}\left[\widetilde{\boldsymbol{\theta}} \mid \mathbf{X},\mathbf{y},\mathbf{x}\right]$, it suffices to solve the posterior distribution of $\boldsymbol{\beta}$ given $\mathbf{X},\mathbf{y}$. From $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$, we have

$$\begin{aligned}\mathbb{P}\left(\widetilde{\boldsymbol{\theta}} \mid \mathbf{X},\mathbf{y},\mathbf{x}\right) &\propto \mathbb{P}\left(\widetilde{\boldsymbol{\theta}}\right)\mathbb{P}\left(\mathbf{y} \mid \mathbf{X},\widetilde{\boldsymbol{\theta}}\right) \propto \exp\left(-\frac{\left\|\mathbf{y}-\mathbf{X}\left(\boldsymbol{\beta}^*+\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)\right\|_2^2}{2\sigma^2} - \frac{1}{2}\widetilde{\boldsymbol{\theta}}^\top\cdot\widetilde{\boldsymbol{\theta}}\right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}\left[\widetilde{\boldsymbol{\theta}}^\top\left(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}+\sigma^2\mathbf{I}_d\right)\widetilde{\boldsymbol{\theta}} - 2\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^*\right)^\top\mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right]\right).\end{aligned}$$

Note that, the function above matches the probability density function of a multivariate Gaussian distrbution. Therefore, the posterior mean is given by

$$\mathbb{E}\left[\boldsymbol{\theta} \mid \mathbf{X},\mathbf{y},\mathbf{x}\right] = \left(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}+\sigma^2\mathbf{I}_d\right)^{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{X}^\top\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^*\right).$$

Therefore, we conclude

$$\widehat{y}_{\mathsf{Bayes}} = \mathbf{x}^\top\boldsymbol{\Psi}^{\frac{1}{2}}\left(\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Psi}^{\frac{1}{2}}+\sigma^2\mathbf{I}_d\right)^{-1}\boldsymbol{\Psi}^{\frac{1}{2}}\mathbf{X}^\top\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^*\right) + \mathbf{x}^\top\boldsymbol{\beta}^*. \tag{3.55}$$

□

Proof of second equation

Proof. Let's first do a variable transformation. We denote

$$\widetilde{\mathbf{X}} = \mathbf{X}\mathbf{\Psi}^{\frac{1}{2}}, \quad \widetilde{\mathbf{x}} = \mathbf{\Psi}^{\frac{1}{2}}\mathbf{x}, \quad \widetilde{\mathbf{y}} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*, \quad \mathbf{\Lambda} = \mathbf{\Psi}^{\frac{1}{2}}\mathbf{H}\mathbf{\Psi}^{\frac{1}{2}}. \tag{3.56}$$

Then, we know $\widetilde{\mathbf{X}}[i], \widetilde{\mathbf{x}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d, \mathbf{\Lambda})$. We can write the Bayesian optimal estimator in (3.55) as

$$\widehat{y}_{\mathsf{Bayes}} = \widetilde{\mathbf{x}}^\top \left(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} + \sigma^2\mathbf{I}_d\right)^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{y}} + \mathbf{x}^\top\boldsymbol{\beta}^*.$$

The true label is

$$y = \mathbf{x}^\top\widetilde{\boldsymbol{\beta}} + \varepsilon = \mathbf{x}^\top\left(\boldsymbol{\beta}^* + \mathbf{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right).$$

Therefore, we have

$$\begin{aligned}
\ell\left(\widehat{y}_{\mathsf{Bayes}}; \mathbf{X}\right) - \sigma^2 &= \mathbb{E}\left(\widetilde{\mathbf{x}}^\top\left(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} + \sigma^2\mathbf{I}_d\right)^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{y}} + \mathbf{x}^\top\boldsymbol{\beta}^* - \mathbf{x}^\top\left(\boldsymbol{\beta}^* + \mathbf{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}\right)\right)^2 \\
&= \mathbb{E}\left(\widetilde{\mathbf{x}}^\top\left[\left(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} + \sigma^2\mathbf{I}_d\right)^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{y}} - \widetilde{\boldsymbol{\theta}}\right]\right)^2 = \mathbb{E}\left\|\widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}\right\|_{\mathbf{\Lambda}}^2,
\end{aligned}$$

where

$$\widehat{\boldsymbol{\theta}} := \left(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} + \sigma^2\mathbf{I}_d\right)^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{y}}.$$

Note that, this is equivalent to the risk of estimating the ground-truth linear weight $\widetilde{\boldsymbol{\theta}}$ using $\widehat{\boldsymbol{\theta}}$ under the Gaussian prior $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. The estimator $\widehat{\boldsymbol{\theta}}$ is a function of transformed input-output pairs in the context $\left(\widetilde{\mathbf{X}}, \widetilde{\mathbf{y}}\right)$ and the transformed query input $\mathbf{x}$, and takes the form of the standard ridge estimator with regularization coefficient being σ^2. We then revoke the standard results for the risk of a ridge estimator. Applying the Theorem 1 and 2 in (Tsigler and Bartlett, 2023), we know for a fixed weight vector $\widetilde{\boldsymbol{\theta}}$, with probability at least $1 - \exp(-\Omega(M))$ we have

$$\left\|\widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}\right\|_{\mathbf{H}}^2 \simeq \left(\frac{\sigma^2 + \sum_{i>k^*}\phi_i}{M}\right)^2\left\|\widetilde{\boldsymbol{\theta}}\right\|_{\mathbf{H}_{0:k^*}^{-1}}^2 + \left\|\widetilde{\boldsymbol{\theta}}\right\|_{\mathbf{H}_{k^*:\infty}}^2 + \sigma^2\left(\frac{k^*}{M} + \frac{M\sum_{i>k^*}\phi_i^2}{\sigma^2 + \sum_{i>k^*}\phi_i}\right),$$

where $\phi_1 \geq \phi_2 \geq ... \geq \phi_d \geq 0$ are ordered eigenvalues of $\mathbf{\Lambda}$.

$$k^* := \min\left\{k : \phi_k \geq c \cdot \frac{\sigma^2 + \sum_{i>k^*}\phi_i}{M}\right\}$$

and $c > 1$ is an absolute constant. Here, $\mathbf{H}_{0:k^*}$ is SVD approximation with respect to the largest k^* singular values and $\mathbf{H}_{k^*:\infty}$ is the SVD approximation in the remaining singular values. Namely, if we have the eigen-decomposition of $\mathbf{H} = \boldsymbol{Q} \cdot \mathrm{diag}(\phi_1, \phi_2, ..., \phi_d) \cdot \boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is an orthogonal matrix, then $\mathbf{H}_{0:k^*}$ and $\mathbf{H}_{k^*:\infty}$ are given by

$$\mathbf{H}_{0:k^*} = \boldsymbol{Q} \cdot \mathrm{diag}(\phi_1, \phi_2, ..., \phi_{k^*}, 0, 0, ..., 0) \cdot \boldsymbol{Q}^\top, \quad \mathbf{H}_{k^*:\infty} = \mathbf{H} - \mathbf{H}_{0:k^*}.$$

Taking expectation over $\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we have

$$\begin{aligned}\ell\left(\widehat{y}_{\mathsf{Bayes}}; \mathbf{X}\right) - \sigma^2 &= \mathbb{E}_{\widetilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left\| \widehat{\boldsymbol{\theta}}_{\mathsf{Bayes}} - \widetilde{\boldsymbol{\theta}} \right\|_{\mathbf{H}}^2 \\ &= \left(\frac{\sigma^2 + \sum_{i>k^*} \phi_i}{M}\right)^2 \cdot \sum_{i \le k^*} \frac{1}{\phi_i} + \sum_{i>k^*} \phi_i + \sigma^2 \left(\frac{k^*}{M} + \frac{M \sum_{i>k^*} \phi_i^2}{\sigma^2 + \sum_{i>k^*} \phi_i}\right)\end{aligned}$$

Now we simplify this expression. First, we define

$$\bar{\phi} := c \cdot \frac{\sigma^2 + \sum_{i>k^*} \phi_i}{M}.$$

From the definition, we see $\bar{\phi} \ge \frac{c\sigma^2}{M}$. On the other hand, from the assumption that $\operatorname{tr}(\mathbf{H}) = \operatorname{tr}\left(\boldsymbol{\Psi}^{\frac{1}{2}} \mathbf{H} \boldsymbol{\Psi}^{\frac{1}{2}}\right) = \sum_{i=1}^d \phi_i \lesssim \sigma^2$, we know that $\bar{\phi} \lesssim \frac{c\sigma^2}{M}$. Combining two parts, we get

$$\bar{\phi} \simeq \frac{\sigma^2}{M}. \tag{3.57}$$

Therefore, we have

$$\begin{aligned}\ell\left(\widehat{y}_{\mathsf{Bayes}}; \mathbf{X}\right) - \sigma^2 &\simeq \bar{\phi}^2 \cdot \sum_{i \le k^*} \frac{1}{\phi_i} + \sum_{i>k^*} \phi_i + \frac{\sigma^2}{M} \cdot \left(k^* + \frac{\sum_{i>k^*} \phi_i^2}{\bar{\phi}^2}\right) \\ &\simeq \sum_i \min\left\{\frac{\bar{\phi}^2}{\phi_i}, \phi_i\right\} + \bar{\phi} \cdot \sum_i \min\left\{1, \frac{\phi_i^2}{\bar{\phi}^2}\right\} \qquad \text{(from (3.57))} \\ &\simeq \sum_i \left(\min\left\{\frac{\bar{\phi}^2}{\phi_i}, \phi_i\right\} + \min\left\{\bar{\phi}, \frac{\phi_i^2}{\bar{\phi}}\right\}\right) \\ &\simeq \sum_i \min\left\{\phi_i, \bar{\phi}\right\}.\end{aligned}$$

This finishes the proof. □

3.14 Appendix: Proof of Theorem 3.6.3

Training dynamics

Now we consider doing gradient flow on the risk function, or equivalently, on the excess risk function which differs by only a constant:

$$\frac{\mathrm{d}\boldsymbol{\beta}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\beta}} \left[\mathcal{R}(\boldsymbol{\beta}, \boldsymbol{\Gamma}) - \min \mathcal{R}(\cdot, \cdot)\right]; \tag{3.58}$$

$$\frac{\mathrm{d}\boldsymbol{\Gamma}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\Gamma}} \left[\mathcal{R}(\boldsymbol{\beta}, \boldsymbol{\Gamma}) - \min \mathcal{R}(\cdot, \cdot)\right]. \tag{3.59}$$

First, we have the following corollary of Theorem 3.5.2, which computes the excess risk $\mathcal{R}(\boldsymbol{\beta}, \boldsymbol{\Gamma}) - \min \mathcal{R}(\cdot, \cdot)$.

Corollary 3.14.1 (Excess Risk). *We fix M as the context length. Consider the ICL risk in (3.5), assume the data is generated following Assumption 3.3.1. Then, we have*

$$\begin{aligned}
&\mathcal{R}(\boldsymbol{\beta}, \boldsymbol{\Gamma}) - \min \mathcal{R}(\cdot, \cdot) \\
&= (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \left[(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H})^\top \mathbf{H} (\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}) + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\right)}{M} \mathbf{H} + \frac{1}{M} \mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} \right] (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \\
&+ \operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Gamma} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Psi} \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega}^{-1} \right) \boldsymbol{\Omega} \left(\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Gamma} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Psi} \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega}^{-1} \right)^\top \right]. \qquad (3.60)
\end{aligned}$$

Proof. This is obtained directly from equation (3.37) and (3.39). □

Now, we write out the differential equations explicitly in the following lemma.

Lemma 3.14.2 (Dynamical system). *The dynamical system of gradient flow described in (3.58) and (3.59) is*

$$\frac{\mathrm{d}\boldsymbol{\beta}}{\mathrm{d}t} = -\left[(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H})^\top \mathbf{H} (\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}) + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\right)}{M} \mathbf{H} + \frac{1}{M} \mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} \right] (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \qquad (3.61)$$

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{\Gamma}}{\mathrm{d}t} &= -\left(\mathbf{H}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}\boldsymbol{\Psi}\mathbf{H} \right) - \frac{M+1}{M} \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H} \\
&\quad - \frac{1}{M} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \cdot \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} + \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}. \qquad (3.62)
\end{aligned}$$

Proof. This can be obtained by directly calculating the derivatives over the excess risk (3.60). To write out the dynamics of $\boldsymbol{\beta}$, it suffices to notice that $\boldsymbol{\beta}$ only attends the first term of the RHS of (3.60), which is a standard quadratic function. For the derivatives of $\boldsymbol{\Gamma}$, we first have

$$\begin{aligned}
&\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\Gamma}} \operatorname{tr}\left[\left(\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Gamma} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Psi} \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega}^{-1} \right) \boldsymbol{\Omega} \left(\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Gamma} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Psi} \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega}^{-1} \right)^\top \right] \\
&= \mathbf{H}^{\frac{1}{2}} \left(\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Gamma} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Psi} \mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega}^{-1} \right) \boldsymbol{\Omega}^{\frac{1}{2}} \cdot \boldsymbol{\Omega}^{\frac{1}{2}} \mathbf{H}^{\frac{1}{2}} \qquad \text{(the sixth equation in Lemma 2.10.1)} \\
&= \mathbf{H}\boldsymbol{\Gamma}\mathbf{H}^{\frac{1}{2}} \boldsymbol{\Omega} \mathbf{H}^{\frac{1}{2}} - \mathbf{H}\boldsymbol{\Psi}\mathbf{H}.
\end{aligned}$$

Now it suffices to compute

$$\frac{\partial}{\partial \boldsymbol{\Gamma}} \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \left[(\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H})^\top \mathbf{H} (\mathbf{I}_d - \boldsymbol{\Gamma}\mathbf{H}) + \frac{\operatorname{tr}\left(\mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\right)}{M} \mathbf{H} + \frac{1}{M} \mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} \right] (\boldsymbol{\beta} - \boldsymbol{\beta}^*).$$

Let's compute the partial derivativess separately. From Lemma 2.10.1, we have

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\Gamma}} \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \left(-\mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\right) (\boldsymbol{\beta} - \boldsymbol{\beta}^*) = \frac{\partial}{\partial \boldsymbol{\Gamma}} \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top (-\mathbf{H}\boldsymbol{\Gamma}\mathbf{H}) (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \\
&\qquad = -\frac{1}{2} \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}; \\
&\frac{\partial}{\partial \boldsymbol{\Gamma}} \frac{1}{2} \left(\frac{M+1}{M} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \right) = \frac{M+1}{M} \mathbf{H}\boldsymbol{\Gamma} \cdot \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}; \\
&\frac{\partial}{\partial \boldsymbol{\Gamma}} \frac{1}{2} \left(\frac{\operatorname{tr}\left\{\mathbf{H}\boldsymbol{\Gamma}^\top \mathbf{H}\boldsymbol{\Gamma}\right\}}{M} \cdot (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \right) = \frac{1}{M} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \cdot \mathbf{H}\boldsymbol{\Gamma}\mathbf{H}.
\end{aligned}$$

Summing over the three equations above and applying the definition of gradient flow, we conclude the dynamics of $\boldsymbol{\Gamma}$ in (3.62). □

Proof of the global convergence

Let's now prove Theorem 3.6.3. Since the first part of Theorem 3.6.3 is directly implied by the second part, we only deal with the case with general $\mathbf{H}$. In this section, we denote $\boldsymbol{\Lambda} min > 0$ as the minimal non-zero eigenvalue of $\mathbf{H}$. As in the main text, we define

$$\mathcal{H} := \mathsf{Im}\,(\mathbf{H})$$

and $\mathcal{H}^\perp$ as its orthogonal complement. First, let's prove the convergence of $\boldsymbol{\beta}$.

Lemma 3.14.3 (Convergence of $\boldsymbol{\beta}$)**.** *Under the dynamical system* (3.61) *and* (3.62)*, one has*

$$\left\|\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(t)) - \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}^*)\right\|_2^2 \leq \exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right) \left\|\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(0)) - \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}^*)\right\|_2^2, \tag{3.63}$$

which implies

$$\left\|\mathbf{H}^{\frac{1}{2}} (\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)\right\|_2^2 \leq \boldsymbol{\Lambda}_1 \exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right) \left\|\boldsymbol{\beta}(0) - \boldsymbol{\beta}^*\right\|_2^2, \tag{3.64}$$

and

$$\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(t)) \to \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}^*)$$

when $t \to \infty$, from arbitrary initialization $\boldsymbol{\beta}(0)$ and $\boldsymbol{\Gamma}(0)$. Moreover, one has for any $t > 0$, it holds that

$$\mathcal{P}_{\mathcal{H}^\perp} (\boldsymbol{\beta}(t)) = \mathcal{P}_{\mathcal{H}^\perp} (\boldsymbol{\beta}(0)) \tag{3.65}$$

Proof. We first consider the orthogonal projection operator $\mathcal{P}$. From Lemma 3.15.1 and equation (3.61), we know

$$\frac{\mathrm{d}\mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)}{\mathrm{d}t} = -\mathbf{H}\mathbf{H}^+ \mathbf{H}_{\boldsymbol{\Gamma}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*),$$

where

$$\mathbf{H}_{\mathbf{\Gamma}} := (\mathbf{I}_d - \mathbf{\Gamma}\mathbf{H})^\top \mathbf{H} (\mathbf{I}_d - \mathbf{\Gamma}\mathbf{H}) + \frac{\operatorname{tr}\left(\mathbf{H}\mathbf{\Gamma}^\top\mathbf{H}\mathbf{\Gamma}\right)}{M}\mathbf{H} + \frac{1}{M}\mathbf{H}\mathbf{\Gamma}^\top\mathbf{H}\mathbf{\Gamma}\mathbf{H}.$$

From the property of pseudo-inverse, we know

$$\frac{\mathrm{d}\mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)}{\mathrm{d}t} = -\mathbf{H}_{\mathbf{\Gamma}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) = -\mathbf{H}_{\mathbf{\Gamma}}\mathbf{H}^+\mathbf{H} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) = -\mathbf{H}_{\mathbf{\Gamma}}\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*).$$

Therefore, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\left\|\mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)\right\|_2^2\right] = -\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \cdot \mathbf{H}_{\mathbf{\Gamma}} \cdot \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*).$$

Notice that

$$\mathbf{H}_{\mathbf{\Gamma}} \succeq \left(\sqrt{\frac{M}{M+1}}\mathbf{I} - \sqrt{\frac{M+1}{M}}\mathbf{\Gamma}\mathbf{H}\right)^\top \mathbf{H} \left(\sqrt{\frac{M}{M+1}}\mathbf{I} - \sqrt{\frac{M+1}{M}}\mathbf{\Gamma}\mathbf{H}\right) + \frac{1}{M+1}\mathbf{H} \succeq \frac{1}{M+1}\mathbf{H}.$$

This implies

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\left\|\mathcal{P}_{\mathcal{H}}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)\right\|_2^2\right] &\leq -\frac{1}{M+1}\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \cdot \mathbf{H} \cdot \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \\ &\leq -\frac{\mathbf{\Lambda}min}{M+1}\left\|\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)\right\|_2^2,\end{aligned}$$

where the last line comes from Lemma 3.15.1 and $\mathbf{\Lambda}min$ is the minimal non-zero eigenvalue of $\mathbf{H}$. Via standard integration method in ODE, we know this suggests

$$\left\|\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)\right\|_2^2 \leq \exp\left(\frac{-2\mathbf{\Lambda}mint}{M+1}\right)\left\|\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(0) - \boldsymbol{\beta}^*)\right\|_2^2 \to 0 \quad \text{when } t \to \infty.$$

This indicates that $\mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}(t)) \to \mathcal{P}_{\mathcal{H}} (\boldsymbol{\beta}^*)$ when $t \to \infty$. Finally, we have

$$\frac{\mathrm{d}\mathcal{P}_{\mathcal{H}^\perp}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*)}{\mathrm{d}t} = -\left(\mathbf{I}_d - \mathbf{H}\mathbf{H}^+\right)\mathbf{H}_{\mathbf{\Gamma}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) = \mathbf{0}_d,$$

which implies $\mathcal{P}_{\mathcal{H}^\perp}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*) = \mathcal{P}_{\mathcal{H}^\perp}(\boldsymbol{\beta}(0) - \boldsymbol{\beta}^*)$ and hence, $\mathcal{P}_{\mathcal{H}^\perp}(\boldsymbol{\beta}(t)) = \mathcal{P}_{\mathcal{H}^\perp}(\boldsymbol{\beta}(0))$ for any $t > 0$. □

Next, let's consider the convergence of $\mathbf{\Gamma}$. As defined in the main text, we have

$$\mathcal{Z} := \mathsf{Im}\,(\mathbf{H} \otimes \mathbf{H})$$

and $\mathcal{Z}^\perp$ is its orthogonal complement. We have the following lemma.

Lemma 3.14.4. *Under the dynamical system (3.61) and (3.62), one has*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}_{\mathcal{Z}^\perp}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)=\mathbf{0}_{d\times d}, \tag{3.66}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)=&-\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\\&-\frac{M+1}{M}\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\\&-\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}\\&-\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}-\frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\\&+\frac{1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}+\left(\mathrm{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right)+\sigma^2\right)\mathbf{I}_d\right)\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}.\end{aligned} \tag{3.67}$$

Proof. The first ODE is trivial since the RHS of (3.62) always lies in $\mathcal{Z}:=\mathsf{Im}\left(\mathbf{H}\otimes\mathbf{H}\right)$, so its projection on $\mathcal{Z}^\perp$ always vanishes. To prove the second equation, we can rewrite (3.62) as

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\Gamma}}{\mathrm{d}t}=&-\Bigg(\mathbf{H}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\underbrace{-\mathbf{H}\boldsymbol{\Psi}\mathbf{H}+\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}}_{A}\Bigg)\\&-\frac{M+1}{M}\mathbf{H}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\\&-\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\mathbf{H}+\underbrace{\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}}_{B}\\&-\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\underbrace{-\frac{M+1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}}_{C}.\end{aligned}$$

For term A, recalling $\boldsymbol{\Gamma}^*=\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}$, we have

$$\begin{aligned}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}&=\mathbf{H}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\\&=\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}&&(\boldsymbol{\Omega}\text{ and }\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\text{ commute.})\\&=\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}&&(\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}=\mathbf{H}^{\frac{1}{2}})\\&=\mathbf{H}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}&&(\boldsymbol{\Omega}\text{ and }\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\text{ commute.})\\&=\mathbf{H}\boldsymbol{\Psi}\mathbf{H}.\end{aligned}$$

This suggests $A=0$. For $B+C$, we have

$$B+C=-\frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}+\left(\mathbf{I}_d-\mathbf{H}\boldsymbol{\Gamma}^*\right)\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}.$$

We can compute $(\mathbf{I}_d - \mathbf{H}\boldsymbol{\Gamma}^*)\mathbf{H}$ as

$$\begin{aligned}
(\mathbf{I}_d - \mathbf{H}\boldsymbol{\Gamma}^*)\mathbf{H} &= \mathbf{H}^{\frac{1}{2}}\left(\mathbf{H}^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}\right) \\
&= \mathbf{H}^{\frac{1}{2}}\left(\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}\right) && (\boldsymbol{\Omega} \text{ and } \mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} \text{ commute.}) \\
&= \mathbf{H}^{\frac{1}{2}}\left(\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}\right) && (\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} = \mathbf{H}^{\frac{1}{2}}) \\
&= \mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\Omega}^{-1}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}} - \boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}\right) \\
&= \frac{1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\mathbf{I}_d\right)\mathbf{H}^{\frac{1}{2}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
B + C = &-\frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H} \\
&+ \frac{1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\mathbf{I}_d\right)\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H}.
\end{aligned}$$

Bridging $A = 0$ and the result of $B + C$ into the ODE, we have

$$\begin{aligned}
&\frac{\mathrm{d}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)}{\mathrm{d}t} \\
= &-\mathbf{H}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}} - \frac{M+1}{M}\mathbf{H}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\mathbf{H}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H} \\
&- \frac{1}{M}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\cdot\mathbf{H}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\mathbf{H} \\
&- \frac{1}{M}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\cdot\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H} - \frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H} \\
&+ \frac{1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right) + \sigma^2\right)\mathbf{I}_d\right)\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta} - \boldsymbol{\beta}^*\right)^\top\mathbf{H}. \qquad (3.68)
\end{aligned}$$

Note that, all terms in the RHS above lie in $\mathcal{Z} = \mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right)$, so this summation also equals $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)$. Moreover, since

$$\begin{aligned}
\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\mathbf{H}^{\frac{1}{2}} &= \mathbf{H}^{\frac{1}{2}}\left[\mathcal{P}_{\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}}\otimes\mathbf{H}^{\frac{1}{2}}\right)}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right) + \mathcal{P}_{\mathsf{null}\left(\mathbf{H}^{\frac{1}{2}}\otimes\mathbf{H}^{\frac{1}{2}}\right)}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\right]\mathbf{H}^{\frac{1}{2}} \\
&= \mathbf{H}^{\frac{1}{2}}\cdot\mathcal{P}_{\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}}\otimes\mathbf{H}^{\frac{1}{2}}\right)}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\mathbf{H}^{\frac{1}{2}}. \qquad (3.69)
\end{aligned}$$

Bridging (3.69) into (3.68), we conclude. □

Then, let's upper bound the dynamics of $\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^*\right)\right\|_F^2$ in the following lemma.

Lemma 3.14.5 (Dynamics of Frobenius norm)**.** *Under the dynamical system* (3.61) *and* (3.62)*, one has*

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F^2\right] \leq -A_1\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F^2 + A_2 \exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right)\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F, \tag{3.70}$$

where

$$\begin{aligned} A_1 &= \boldsymbol{\Lambda} min^2 \boldsymbol{\Lambda}_{\min}(\boldsymbol{\Omega}), \\ A_2 &= \left(2+\frac{2}{M}\right)\boldsymbol{\Lambda}_1^2\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2. \end{aligned} \tag{3.71}$$

are two positive constants. Here, $\boldsymbol{\Lambda} min > 0$ *is the minimal positive eigenvalue of* $\mathbf{H}$*,* $\boldsymbol{\Lambda}_1$ *is the maximal eigenvalue of* $\mathbf{H}$*,* $\boldsymbol{\Lambda}_{\min}(\boldsymbol{\Omega})$ *is the minimal eigenvalue of* $\boldsymbol{\Omega}$ *(defined in* (3.14)*) and is strictly positive.*

Proof. From the dynamics of $\mathcal{P}_{\mathcal{Z}}\boldsymbol{\Gamma}$ in (3.67), we know

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F^2\right] &= \mathrm{tr}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right) \\ &= G_1+G_2+G_3, \end{aligned} \tag{3.72}$$

where

$$\begin{aligned} G_1 &= -\mathrm{tr}\left[\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}\mathbf{H}^{\frac{1}{2}}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right], \\ G_2 &= -\mathrm{tr}\left[\frac{M+1}{M}\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right] \\ &\qquad -\mathrm{tr}\left[\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right], \\ G_3 &= -\mathrm{tr}\left[\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right] \\ &\quad +\frac{1}{M}\mathrm{tr}\left[\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right]+\mathrm{tr}\left[\frac{1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\Big(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\right. \\ &\quad \left.+\Big(\mathrm{tr}(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}})+\sigma^2\Big)\mathbf{I}_d\Big)\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right]. \end{aligned} \tag{3.73}$$

From Lemma 2.10.4, we know that $G_2 \leq 0$. Then, we consider G_1 and G_3. For G_1, we have

$$\begin{aligned} G_1 &= -\mathrm{tr}\left[\left(\mathbf{H}^{\frac{1}{2}}\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\mathbf{H}^{\frac{1}{2}}\right)\cdot\left(\mathbf{H}^{\frac{1}{2}}\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\mathbf{H}^{\frac{1}{2}}\right)\cdot\boldsymbol{\Omega}\right] \\ &\leq -\boldsymbol{\Lambda} min^2\mathrm{tr}\left[\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\cdot\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\cdot\boldsymbol{\Omega}\right] \leq -\boldsymbol{\Lambda} min^2\boldsymbol{\Lambda}_{\min}(\boldsymbol{\Omega})\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F^2, \end{aligned}$$

((3) in Lemma 3.15.2 and (2) in Lemma 2.10.4)

where $\boldsymbol{\Lambda}_{\min}(\cdot)$ denote the minimal eigenvalue. Since $\boldsymbol{\Omega}$ is positive definite, we know $\boldsymbol{\Lambda}_{\min}(\boldsymbol{\Omega}) > 0$.

Finally, let's upper bound G_3. First, we have

$$
\begin{aligned}
&-\operatorname{tr}\left[\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot \mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\cdot \mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right]\\
\leq&\frac{\sqrt{d}}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot \left\|\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right\|_2 \left\|\mathbf{H}^{\frac{1}{2}}\cdot \mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top \mathbf{H}^{\frac{1}{2}}\right\|_F \qquad \text{(Lemma 2.10.3)}\\
\leq&\frac{\sqrt{d}\boldsymbol{\Lambda}_1}{M}\exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2 \boldsymbol{\Lambda}_{\max}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)\cdot \|\mathbf{H}\|_F \left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(from equation (3.64))}\\
\leq&\frac{\sqrt{d}\boldsymbol{\Lambda}_1^2}{M}\exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2 \boldsymbol{\Lambda}_{\max}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)\cdot \left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F.
\end{aligned}
$$

For $\boldsymbol{\Lambda}_{\max}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)$, we recall $\boldsymbol{\Gamma}^*=\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}$ and obtain

$$
\begin{aligned}
\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}&=\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}=\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} \qquad (\boldsymbol{\Omega}\text{ and }\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\text{ commute})\\
&=\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}. \qquad (\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}=\mathbf{H}^{\frac{1}{2}})
\end{aligned}
$$

Since $\boldsymbol{\Omega}^{-1}$ and $\mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}$ commute, they are simultaneously diagonalizable. The eigenvalues of $\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}$ are

$$
\frac{\phi_i}{\frac{M+1}{M}\phi_i+\frac{\sum_i\phi_i+\sigma^2}{M}},\quad i=1,2,...,d.
$$

Note that, every eigenvalue is upper bounded by 1, so we simply get

$$
\boldsymbol{\Lambda}_{\max}\left(\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right)\leq 1. \tag{3.74}
$$

Therefore, we have

$$
\begin{aligned}
&-\operatorname{tr}\left[\frac{1}{M}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\cdot \mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\cdot \mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right]\\
\leq&\frac{\sqrt{d}\boldsymbol{\Lambda}_1^2}{M}\exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2\cdot \left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F.
\end{aligned} \tag{3.75}
$$

Similarly, we have

$$
\begin{aligned}
&-\operatorname{tr}\left[\frac{1}{M}\mathbf{H}\boldsymbol{\Gamma}^*\mathbf{H}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}\cdot \mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right]\\
\leq&\frac{\sqrt{d}}{M}\left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)^\top\right\|_F \left\|\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top \mathbf{H}^{\frac{1}{2}}\right\|_F \|\mathbf{H}\|_F \left\|\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Gamma}^*\mathbf{H}^{\frac{1}{2}}\right\|_2 \qquad \text{(Lemma 2.10.3)}\\
\leq&\frac{\sqrt{d}\boldsymbol{\Lambda}_1^2}{M}\exp\left(\frac{-2\boldsymbol{\Lambda} mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2\cdot \left\|\mathcal{P}_{\mathcal{Z}}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}^*\right)\right\|_F,
\end{aligned} \tag{3.76}
$$

and the last term in G_3 can be upper bounded by

$$\begin{aligned}
&\frac{\sqrt{d}}{M}\left\|\mathbf{\Omega}^{-1}\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}+\left(\operatorname{tr}\left(\mathbf{H}^{\frac{1}{2}}\mathbf{\Psi}\mathbf{H}^{\frac{1}{2}}\right)+\sigma^2\right)\mathbf{I}_d\right)\right\|_2\cdot \\
&\qquad\left\|\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}\cdot\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)^\top\mathbf{H}^{\frac{1}{2}}\right\|_F \qquad \text{(Lemma 2.10.3)}\\
\leq\ &\frac{\sqrt{d}}{M}\cdot M\cdot\left\|\mathbf{H}^{\frac{1}{2}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)^\top\mathbf{H}^{\frac{1}{2}}\right\|_F\cdot\|\mathbf{H}\|_F\cdot\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F\\
\leq\ &\sqrt{d}\mathbf{\Lambda}_1^2\exp\left(\frac{-2\mathbf{\Lambda}mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2\cdot\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F \qquad (3.77)
\end{aligned}$$

Bridging (3.75), (3.76) and (3.77) into (3.73), we get

$$G_3\leq\left(2+\frac{2}{M}\right)\mathbf{\Lambda}_1^2\exp\left(\frac{-2\mathbf{\Lambda}mint}{M+1}\right)\left\|\boldsymbol{\beta}(0)-\boldsymbol{\beta}^*\right\|_2^2\cdot\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F \qquad (3.78)$$

□

Finally, we finish this section by proving the convergence of $\mathbf{\Gamma}$.

Lemma 3.14.6 (Convergence of $\mathbf{\Gamma}$)**.** *Under the dynamical system* (3.61) *and* (3.62)*, one has*

$$\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}(t)\right)\to\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}^*\right),\quad \mathcal{P}_{\mathcal{Z}^\perp}\left(\mathbf{\Gamma}(t)\right)=\mathcal{P}_{\mathcal{Z}^\perp}\left(\mathbf{\Gamma}(0)\right)$$

when $t\to\infty$*, from arbitrary initialization* $\boldsymbol{\beta}(0)$ *and* $\mathbf{\Gamma}(0)$*.*

Proof. We observe that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F^2\right]=\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F\cdot\frac{\mathrm{d}}{\mathrm{d}t}\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F.$$

Combining it with Lemma 3.14.5, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F\leq-A_1\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F+A_2\exp\left(\frac{-2\mathbf{\Lambda}mint}{M+1}\right),$$

where $A_1>0, A_2>0$ are defined in (3.71). Simple calculation shows that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\exp\left(A_1t\right)\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}-\mathbf{\Gamma}^*\right)\right\|_F\right]\leq A_2\exp\left[\left(A_1-\frac{2\mathbf{\Lambda}min}{M+1}\right)t\right]. \qquad (3.79)$$

When $A_1\neq\frac{2\mathbf{\Lambda}min}{M+1}$, we integrate both sides from $t=0$ to $t=T$, then divide them by $\exp\left(A_1T\right)$. This gives

$$\begin{aligned}
\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}(T)-\mathbf{\Gamma}^*\right)\right\|_F&\leq\frac{\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}(0)-\mathbf{\Gamma}^*\right)\right\|_F}{\exp\left(A_1T\right)}+\frac{A_2}{A_1-\frac{2\mathbf{\Lambda}min}{M+1}}\cdot\frac{\exp\left[\left(A_1-\frac{2\mathbf{\Lambda}min}{M+1}\right)T\right]-1}{\exp\left(A_1T\right)}\\
&=\frac{\left\|\mathcal{P}_{\mathcal{Z}}\left(\mathbf{\Gamma}(0)-\mathbf{\Gamma}^*\right)\right\|_F}{\exp\left(A_1T\right)}+\frac{A_2}{A_1-\frac{2\mathbf{\Lambda}min}{M+1}}\cdot\left[\exp\left(-\frac{2\mathbf{\Lambda}minT}{M+1}\right)-\exp\left(-A_1T\right)\right]\to0
\end{aligned}$$

when $T \to \infty$. Otherwise, if $A_1 = \frac{2\mathbf{\Lambda}min}{M+1}$, the right hand side of (3.79) reduces to a constant, which implies $\exp(A_1 t) \left\| \mathcal{P}_{\mathcal{Z}} (\mathbf{\Gamma}(t) - \mathbf{\Gamma}^*) \right\|_F$ grows at most at a linear rate, which implies $\left\| \mathcal{P}_{\mathcal{Z}} (\mathbf{\Gamma}(t) - \mathbf{\Gamma}^*) \right\|_F \to 0$ when $t \to \infty$. Together, in all cases, we have $\left\| \mathcal{P}_{\mathcal{Z}} (\mathbf{\Gamma}(t) - \mathbf{\Gamma}^*) \right\|_F \to 0$. From (3.66), we soon get $\mathcal{P}_{\mathcal{Z}} (\mathbf{\Gamma}(t)) = \mathcal{P}_{\mathcal{Z}} (\mathbf{\Gamma}(0))$ for all $t > 0$. Therefore, we conclude. □

The Theorem 3.6.3 is proved by simplly combining Lemma 3.14.3 and Lemma 3.14.6.

3.15 Appendix: Technical Lemmas

Lemma 3.15.1 (Linear Algebra)**.** *Suppose* $\mathbf{H}$ *is a (non-zero) positive semi-definite matrix in* $\mathbb{R}^{d \times d}$ *and* $\mathbf{H}^{\frac{1}{2}}$ *is its principal square root. We denote* $\mathbf{\Lambda}min$ *as the minimal non-zero eigenvalue of* $\mathbf{H}$*. Then, we have*

- *1.* $\mathsf{null}(\mathbf{H}) = \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}}\right), \mathsf{Im}(\mathbf{H}) = \mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}}\right)$.
- *2.* $\mathbf{H}\mathbf{H}^{+} = \mathbf{H}^{+}\mathbf{H}$*, where* $(\cdot)^{+}$ *denotes the Moore-Penrose pseudo-inverse.*
- *3. For any vector* $\boldsymbol{\alpha} \in \mathbb{R}^d$*, the orthogonal projection operator on* $\mathsf{Im}(\mathbf{H})$ *and* $\mathsf{null}(\mathbf{H})$ *are respectively*

$$\mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha}) = \mathbf{H}\mathbf{H}^{+}\boldsymbol{\alpha} = \mathbf{H}^{+}\mathbf{H}\boldsymbol{\alpha}, \quad \mathcal{P}_{\mathsf{null}(\mathbf{H})}(\boldsymbol{\alpha}) = \left(\mathbf{I}_d - \mathbf{H}\mathbf{H}^{+}\right)\boldsymbol{\alpha} = \left(\mathbf{I}_d - \mathbf{H}^{+}\mathbf{H}\right)\boldsymbol{\alpha}.$$

- *4. For any vector* $\boldsymbol{\alpha} \in \mathbb{R}^d$*, we have*

$$\mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha})^{\top} \mathbf{H} \mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha}) \geq \mathbf{\Lambda}min \left\| \mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha}) \right\|_2^2$$

Proof. We consider the eigen-decomposition of matrix $\mathbf{H}$, which is

$$\mathbf{H} = \boldsymbol{Q}\mathbf{D}\boldsymbol{Q}^{\top}, \tag{3.80}$$

where $\mathbf{D} = \mathrm{diag}(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, ..., \mathbf{\Lambda}_d)$ is a diagonal matrix with diagonal entries being the eigenvalues of $\mathbf{H}$, and $\boldsymbol{Q}$ is an orthogonal matrix. Then, from the definition of principal square root, we know

$$\mathbf{H}^{\frac{1}{2}} = \boldsymbol{Q}\mathbf{D}^{\frac{1}{2}}\boldsymbol{Q}^{\top}, \tag{3.81}$$

where $\mathbf{D}^{\frac{1}{2}} = \mathrm{diag}\left(\sqrt{\mathbf{\Lambda}_1}, \sqrt{\mathbf{\Lambda}_2}, ..., \sqrt{\mathbf{\Lambda}_d}\right)$. We denote columns of $\boldsymbol{Q}$ as $\mathbf{q}_1, ...\mathbf{q}_d$. We know they are the eigenvectors of $\mathbf{H}$. From the eigen-decomposition of $\mathbf{H}$ and $\mathbf{H}^{\frac{1}{2}}$, we know they share the same set of eigenvectors. Without loss of generality, we assume $\mathrm{rank}(\mathbf{H}) = r$ and $\mathbf{\Lambda}_1 \geq \mathbf{\Lambda}_2 \geq ... \geq \mathbf{\Lambda}_r > 0, \mathbf{\Lambda}_i = 0, i = r+1, ..., d$. Then, we denote $\mathcal{H}$ as the linear vector space spanned by $\mathbf{q}_1, ..., \mathbf{q}_r$ and $\mathcal{H}^{\perp}$ as its orthogonal complement (which is the subspace spanned by $\mathbf{q}_{r+1}, ..., \mathbf{q}_d$). For any vector $\boldsymbol{\alpha} \in \mathbb{R}^d$, we can write its orthogonal decomposition as $\boldsymbol{\alpha} = \sum_{i=1}^{d} a_i \mathbf{q}_i$, so we have

$$\mathbf{H}\boldsymbol{\alpha} = \sum_{i=1}^{d} a_i \mathbf{H}\mathbf{q}_i = \sum_{i=1}^{r} a_i \mathbf{\Lambda}_i \mathbf{q}_i \in \mathcal{H},$$

which implies $\mathsf{Im}(\mathbf{H}) \subset \mathcal{H}$. On the other hand, we know for any $\boldsymbol{\alpha} \in \mathcal{H}$, we can write it as $\boldsymbol{\alpha} = \sum_{i=1}^{r} a_i \mathbf{q}_i$ for some $a_1, ..., a_r$, then we have

$$\boldsymbol{\alpha} = \sum_{i=1}^{r} \frac{a_i}{\boldsymbol{\Lambda}_i} \cdot \boldsymbol{\Lambda}_i \mathbf{q}_i = \mathbf{H}\left(\sum_{i=1}^{r} \frac{a_i}{\boldsymbol{\Lambda}_i} \mathbf{q}_i\right) \in \mathsf{Im}(\mathbf{H}),$$

which implies $\mathcal{H} \subset \mathsf{Im}(\mathbf{H})$. This shows

$$\mathsf{Im}(\mathbf{H}) = \mathcal{H} = \mathsf{span}\{\mathbf{q}_1, ...\mathbf{q}_r\},$$

and

$$\mathsf{null}(\mathbf{H}) = \mathcal{H} = \mathsf{span}\{\mathbf{q}_{r+1}, ...\mathbf{q}_d\},$$

since $\mathsf{null}(\mathbf{H})$ is the orthogonal complement of $\mathsf{Im}(\mathbf{H})$. Then, (1) is proved by noticing that $\mathbf{H}$ and $\mathbf{H}^{\frac{1}{2}}$ share the same set of eigenvectors. To prove (2), it suffices to notice that

$$\mathbf{H}\mathbf{H}^{+} = \boldsymbol{Q} \cdot \mathrm{diag}(\underbrace{1, 1, ..., 1}_{\mathrm{r}}, 0, 0, .., 0) \cdot \boldsymbol{Q}^{\top} = \mathbf{H}^{+}\mathbf{H}.$$

To prove (3), it suffices to notice that actually $\mathbf{H}\mathbf{H}^{+}\boldsymbol{\alpha} \in \mathsf{Im}(\mathbf{H})$ and

$$\left\langle \mathbf{H}\mathbf{H}^{+}\boldsymbol{\alpha}, \left(\mathbf{I}_d - \mathbf{H}\mathbf{H}^{+}\right)\boldsymbol{\alpha} \right\rangle = \left\langle \mathbf{H}^{+}\mathbf{H}\boldsymbol{\alpha}, \left(\mathbf{I}_d - \mathbf{H}\mathbf{H}^{+}\right)\boldsymbol{\alpha} \right\rangle = \boldsymbol{\alpha}^{\top}\left(\mathbf{H}\mathbf{H}^{+} - \mathbf{H}\mathbf{H}^{+}\mathbf{H}\mathbf{H}^{+}\right)\boldsymbol{\alpha} = 0.$$

Finally, to prove (4), we can write $\boldsymbol{\alpha} = \sum_{i=1}^{d} a_i \mathbf{q}_i$ for some $a_1, ..., a_d$. Then, by orthogonality we have

$$\begin{aligned}\mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha})^{\top}\mathbf{H}\mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha}) &= \left(\sum_{i=1}^{d} a_i \mathbf{q}_i\right)^{\top}\mathbf{H}\left(\sum_{i=1}^{d} a_i \mathbf{q}_i\right) = \sum_{i=1}^{r} a_i^2 \boldsymbol{\Lambda}_i \mathbf{q}_i^{\top}\mathbf{q}_i \geq \boldsymbol{\Lambda}min \cdot \sum_{i=1}^{r} a_i^2 \mathbf{q}_i^{\top}\mathbf{q}_i \\ &= \boldsymbol{\Lambda}min \left\|\mathcal{P}_{\mathsf{Im}(\mathbf{H})}(\boldsymbol{\alpha})\right\|_2^2.\end{aligned}$$

Therefore, we conclude. □

Lemma 3.15.2 (Tensor product)**.** *Suppose* $\mathbf{H}$ *is a (non-zero) positive semi-definite matrix in* $\mathbb{R}^{d\times d}$ *and* $\mathbf{H}^{\frac{1}{2}}$ *is its principal square root. We denote* $\otimes$ *as Kronecker product, which is defined as*

$$(\mathbf{A} \otimes \mathbf{B}) \circ \mathbf{C} = \mathbf{B}\mathbf{C}\mathbf{A}^{\top}.$$

We define

$$\mathsf{Im}(\mathbf{H} \otimes \mathbf{H}) := \left\{(\mathbf{H} \otimes \mathbf{H}) \circ \mathbf{Z} : \mathbf{Z} \in \mathbb{R}^{d\times d}\right\}, \mathsf{null}(\mathbf{H} \otimes \mathbf{H}) := \left\{\mathbf{Z} \in \mathbb{R}^{d\times d} : (\mathbf{H} \otimes \mathbf{H}) \circ \mathbf{Z} = \mathbf{0}\right\},$$

and define $\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right)$ *and* $\mathsf{null}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right)$ *similarly. Then, we have*

- *1.* $\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{Im}\left(\mathbf{H} \otimes \mathbf{H}\right), \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{null}\left(\mathbf{H} \otimes \mathbf{H}\right).$
- *2. We denote* $\mathcal{Z} = \mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{Im}\left(\mathbf{H} \otimes \mathbf{H}\right)$ *and* $\mathcal{Z}^{\perp} = \mathsf{null}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{null}\left(\mathbf{H} \otimes \mathbf{H}\right)$. *For any matrix* $\mathbf{Z} \in \mathbb{R}^{d\times d}$, *we have*

$$\mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right) = \left(\mathbf{H}\mathbf{H}^{+}\right)^{\otimes 2} \circ \mathbf{Z} = \mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H} = \mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{Z}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} \tag{3.82}$$

$$\mathcal{P}_{\mathcal{Z}^{\perp}}\left(\mathbf{Z}\right) = \left[\mathbf{I}_d^{\otimes 2} - \left(\mathbf{H}\mathbf{H}^{+}\right)^{\otimes 2}\right] \circ \mathbf{Z} = \mathbf{Z} - \mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H} = \mathbf{Z} - \mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{Z}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}}. \tag{3.83}$$

- *3. For any matrix* $\mathbf{Z} \in \mathbb{R}^{d\times d}$, *it holds that*

$$\left(\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)\right) \cdot \left(\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)\right)^{\top} \succeq \mathbf{\Lambda}min^{2}\mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right) \cdot \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)^{\top}, \tag{3.84}$$

where $\mathbf{\Lambda}min$ *is the minimal positive eigenvector of* $\mathbf{H}$.

Proof. Let's first prove the second part. From the definition of tensor product and the fact that $\mathbf{H}\mathbf{H}^{+} = \mathbf{H}^{+}\mathbf{H}$ (see Lemma 3.15.1), we know the second equation in 3.82 holds. To prove the first equation, it suffices to notice that for any matrix $\mathbf{Z} \in \mathbb{R}^{d\times d}$, the matrix $\mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}$ is in $\mathsf{Im}\left(\mathbf{H}^{\otimes 2}\right)$, together with the fact that

$$\begin{aligned}
\left\langle \mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}, \mathbf{Z} - \mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H} \right\rangle &= \mathrm{tr}\left(\mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}\mathbf{Z}^{\top}\right) - \mathrm{tr}\left(\mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}\mathbf{H}\mathbf{H}^{+}\mathbf{Z}^{\top}\mathbf{H}^{+}\mathbf{H}\right) \\
&= \mathrm{tr}\left(\mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}\mathbf{Z}^{\top}\right) - \mathrm{tr}\left(\mathbf{H}\mathbf{H}^{+}\mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H}\mathbf{H}\mathbf{H}^{+}\mathbf{H}\mathbf{Z}^{\top}\right) \\
&\qquad (\mathbf{H}\mathbf{H}^{+} = \mathbf{H}^{+}\mathbf{H}) \\
&= 0.
\end{aligned}$$

The proof of (3.83) is similar. Then, from Lemma 3.15.1, we know $\mathsf{Im}\left(\mathbf{H}\right) = \mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}}\right)$, which implies the projection operator onto those two subspace are identical, indicating $\mathbf{H}\mathbf{H}^{+} = \mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}$. Therefore, we have

$$\mathcal{P}_{\mathsf{Im}(\mathbf{H}\otimes\mathbf{H})}\left(\mathbf{Z}\right) = \left(\mathbf{H}\mathbf{H}^{+}\right)^{\otimes 2} \circ \mathbf{Z} = \mathbf{H}\mathbf{H}^{+}\mathbf{Z}\mathbf{H}^{+}\mathbf{H} = \mathbf{H}^{\frac{1}{2}}\mathbf{H}^{-\frac{1}{2}}\mathbf{Z}\mathbf{H}^{-\frac{1}{2}}\mathbf{H}^{\frac{1}{2}} = \mathcal{P}_{\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}}\otimes\mathbf{H}^{\frac{1}{2}}\right)}\left(\mathbf{Z}\right).$$

Since this holds for any matrix $\mathbf{Z} \in \mathbb{R}^{d\times d}$, this suggests $\mathsf{Im}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{Im}\left(\mathbf{H} \otimes \mathbf{H}\right)$. Similarly, we also have $\mathsf{null}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) = \mathsf{null}\left(\mathbf{H} \otimes \mathbf{H}\right)$. Finally, to prove (3), we use the eigen-decomposition of $\mathbf{H}$ and $\mathbf{H}^{\frac{1}{2}}$:

$$\mathbf{H} = \boldsymbol{Q}\mathbf{D}\boldsymbol{Q}^{\top}, \quad \mathbf{H}^{\frac{1}{2}} = \boldsymbol{Q}\mathbf{D}^{\frac{1}{2}}\boldsymbol{Q}^{\top},$$

where $\boldsymbol{Q} = (\mathbf{q}_1\ \mathbf{q}_2\ ...\ \mathbf{q}_d)$ is an orthogonal matrix and $\mathbf{D} = \mathrm{diag}\left(\mathbf{\Lambda}_1, ..., \mathbf{\Lambda}_d\right)$, where $\mathbf{\Lambda}_1 \geq \mathbf{\Lambda}_2 \geq ....\mathbf{\Lambda}_d \geq 0$ are ordered eigenvalues. We assume $\mathrm{rank}(\mathbf{H}) = r$ and $\mathbf{\Lambda}_r > 0 = \mathbf{\Lambda}_{r+1}$. Then,

from the property of Kronecker product (eg. see (Petersen and Pedersen, 2008)), the eigenvalues of $\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}$ are $\left\{\sqrt{\mathbf{\Lambda}_i \mathbf{\Lambda}_j} : 1 \leq i, j \leq d\right\}$. The eigenvectors are $\{\mathbf{q}_i \otimes \mathbf{q}_j : 1 \leq i, j \leq d\}$, which forms an orthogonal unit basis of $\mathbb{R}^{d \times d}$. We define

$$\mathcal{S} := \mathsf{span}\left\{\mathbf{q}_i \otimes \mathbf{q}_j : 1 \leq i, j \leq r\right\}$$

as the eigenspace spanned by all eigenvectors corresponding to positive eigenvalues. For any matrix $\mathbf{Z}$, we can decompose it as

$$\mathbf{Z} = \sum_{i,j} a_{i,j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j,$$

so

$$\begin{aligned}(\mathbf{H} \otimes \mathbf{H}) \circ \mathbf{Z} &= \sum_{i,j=1}^{d} a_{i,j} \left(\mathbf{H} \otimes \mathbf{H}\right) \circ \left(\mathbf{q}_i \otimes \mathbf{q}_j\right) = \sum_{i,j} \left(\mathbf{H}\mathbf{q}_i\right) \otimes \left(\mathbf{H}\mathbf{q}_j\right) \\ &= \sum_{i,j=1}^{d} a_{i,j} \mathbf{\Lambda}_i \mathbf{\Lambda}_j \mathbf{q}_i \otimes \mathbf{q}_j = \sum_{i,j=1}^{r} a_{i,j} \mathbf{\Lambda}_i \mathbf{\Lambda}_j \mathbf{q}_i \otimes \mathbf{q}_j \in \mathcal{S},\end{aligned}$$

which implies $\mathcal{Z} \subset \mathcal{S}$. On the other hand, for any $\mathbf{Z} \in \mathcal{S}$, we can write it as $\mathbf{Z} = \sum_{i,j=1}^{r} b_{i,j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j$, so we have

$$\mathbf{Z} = \sum_{i,j=1}^{r} \frac{b_{i,j}}{\mathbf{\Lambda}_i \mathbf{\Lambda}_j} \cdot \left(\mathbf{H}\mathbf{q}_i\right) \otimes \left(\mathbf{H}\mathbf{q}_j\right) = \left(\mathbf{H} \otimes \mathbf{H}\right) \circ \left[\sum_{i,j=1}^{r} \frac{b_{i,j}}{\mathbf{\Lambda}_i \mathbf{\Lambda}_j} \left(\mathbf{q}_i \otimes \mathbf{q}_j\right)\right] \in \mathcal{Z},$$

which implies $\mathcal{S} \subset \mathcal{Z}$. Combining two directions, we have $\mathcal{S} = \mathcal{Z}$. Therefore, for any matrix $\mathbf{Z} \in \mathbb{R}^{d \times d}$, we can write it as $\mathbf{Z} = \sum_{i,j=1}^{d} c_{i,j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j$ for some $c_{i,j}$. Then, we have

$$\begin{aligned}\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right) &= \left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{S}}\left(\mathbf{Z}\right) = \left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \sum_{i,j=1}^{r} c_{i,j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j \\ &= \sum_{i,j=1}^{r} c_{i,j} \left(\mathbf{H}^{\frac{1}{2}} \mathbf{q}_i\right) \otimes \left(\mathbf{H}^{\frac{1}{2}} \mathbf{q}_j\right) = \sum_{i,j=1}^{r} c_{i,j} \sqrt{\mathbf{\Lambda}_i \mathbf{\Lambda}_j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j.\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\left(\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)\right) \left(\left(\mathbf{H}^{\frac{1}{2}} \otimes \mathbf{H}^{\frac{1}{2}}\right) \circ \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)\right)^{\top} \\ &= \left(\sum_{i,j=1}^{r} c_{i,j} \sqrt{\mathbf{\Lambda}_i \mathbf{\Lambda}_j} \cdot \mathbf{q}_i \otimes \mathbf{q}_j\right) \left(\sum_{k,l=1}^{r} c_{k,l} \sqrt{\mathbf{\Lambda}_k \mathbf{\Lambda}_l} \cdot \mathbf{q}_k \otimes \mathbf{q}_l\right)^{\top} \\ &= \sum_{i,j=1}^{r} c_{i,j}^2 \mathbf{\Lambda}_i \mathbf{\Lambda}_j \left(\mathbf{q}_i \otimes \mathbf{q}_j\right) \left(\mathbf{q}_i \otimes \mathbf{q}_j\right)^{\top} \qquad \text{(By orthogonality)} \\ &\succeq \mathbf{\Lambda} min^2 \sum_{i,j=1}^{r} c_{i,j}^2 \left(\mathbf{q}_i \otimes \mathbf{q}_j\right) \left(\mathbf{q}_i \otimes \mathbf{q}_j\right)^{\top} \\ &= \mathbf{\Lambda} min^2 \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right) \cdot \mathcal{P}_{\mathcal{Z}}\left(\mathbf{Z}\right)^{\top}.\end{aligned}$$

Therefore, we conclude. □

3.16 Appendix: Experiment Details

Our experiments on GPT2 mostly follow the setting in (Garg et al., 2022), except that we use the token matrix defined in (3.1). In our experiments, we use $d = 20, M = 40, \mathbf{\Psi} = \mathbf{H} = \mathbf{I}_d$ and $\sigma = 0$. We train the GPT2 with and without MLP layers on two settings: $\boldsymbol{\beta}^* = (0, 0, ..., 0)^\top$ and $\boldsymbol{\beta}^* = (10, 10, ..., 10)^\top$. For GPT2 with and without MLP layers, we initialize the $\boldsymbol{W}^V$ and $\mathbf{W}_2$ by normal distribution with standard deviation $0.02/\sqrt{\text{number of residual connections}}$, where the number of residual connections equals the number of layers for GPT2 without MLP layers. For GPT2 with MLP layers, this is 2 times the number of layers. Initialization for other matrices follow the default setting in (Wolf et al., 2020). We use Adam with learning rate 0.0001. We train the model for 200000 steps and we sample a batch of 256 new tasks for each step.

3.17 Appendix: Does Scratchpad Help?

In this section, we show the limitations of adding a scratchpad to the token matrix. We will show that by using a single LSA layer and the token matrix with scratchpad, one cannot recover the GD-$\boldsymbol{\beta}$ estimator defined in Section 3.5. We leave it as future work to see whether the token matrix with scratchpad could implement other types of estimators that more effectively address the linear regression tasks defined in Assumption 3.3.1, as well as whether additional structures could help alleviate this approximation error. We follow the notations in previous sections. The token matrix with scratchpad is defined as

$$\mathbf{E} := \begin{pmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{1}_M^\top & 1 \\ \mathbf{y}^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+2)\times(M+1)}. \tag{3.85}$$

where $\mathbf{1}_M \in \mathbb{R}^M$ refers to the vector filled with ones.

A LSA model f, is defined by

$$\begin{aligned} f(\mathbf{E}) &:= \left[\mathbf{E} + (\boldsymbol{W}^P)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top (\boldsymbol{W}^K)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right]_{-1,-1} \\ &= \left[(\boldsymbol{W}^P)^\top \boldsymbol{W}^V \mathbf{E}\mathbf{M} \frac{\mathbf{E}^\top (\boldsymbol{W}^K)^\top \boldsymbol{W}^Q \mathbf{E}}{M}\right]_{-1,-1}, \end{aligned}$$

where the second equality is because the bottom right entry of $\mathbf{E}$ is zero (see (3.1)). Note that the prediction is the bottom right entry of the output matrix. So only the last row of $(\boldsymbol{W}^P)^\top \boldsymbol{W}^V$ and the last column of $\mathbf{E}$ attend the prediction. Denote

$$(\boldsymbol{W}^P)^\top \boldsymbol{W}^V = \begin{pmatrix} * & * & * \\ \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix}, \quad (\boldsymbol{W}^K)^\top \boldsymbol{W}^Q = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{b} & * \\ \boldsymbol{q}_1^\top & \boldsymbol{b}_1 & * \\ \boldsymbol{q}_2^\top & \boldsymbol{b}_2 & * \end{pmatrix},$$

where

$$\boldsymbol{w} \in \mathbb{R}^d,\ \boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R},\ \boldsymbol{Q} \in \mathbb{R}^{d\times d},\ \boldsymbol{b}, \boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{R}^d,\ \boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R},$$

and $*$ denotes entries that do not enter the final prediction. Then we have

$$\begin{aligned} f(\mathbf{E}) &= \begin{pmatrix} \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix} \frac{\mathbf{E}\mathbf{M}_M\mathbf{E}^\top}{M} \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{b} & * \\ \boldsymbol{q}_1^\top & \boldsymbol{b}_1 & * \\ \boldsymbol{q}_2^\top & \boldsymbol{b}_2 & * \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix} \frac{\mathbf{E}\mathbf{M}_M\mathbf{E}^\top}{M} \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{b} \\ \boldsymbol{q}_1^\top & \boldsymbol{b}_1 \\ \boldsymbol{q}_2^\top & \boldsymbol{b}_2 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix} \frac{1}{M} \begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{1}_M & \mathbf{X}^\top\mathbf{y} \\ \mathbf{1}_M^\top\mathbf{X} & M & \mathbf{1}_M^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{1}_M & \mathbf{y}^\top\mathbf{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{b} \\ \boldsymbol{q}_1^\top & \boldsymbol{b}_1 \\ \boldsymbol{q}_2^\top & \boldsymbol{b}_2 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}. \end{aligned}$$

Following the Assumption 3.1, we use $\widetilde{\boldsymbol{\beta}}$ to refer to the task parameter, then

$$\widetilde{\boldsymbol{\beta}} := \boldsymbol{\beta}^* + \boldsymbol{\Psi}^{\frac{1}{2}}\widetilde{\boldsymbol{\theta}}, \quad \text{where} \quad \widetilde{\boldsymbol{\theta}} \sim \mathcal{N}\left(\mathbf{0}_d, \mathbf{I}_d\right).$$

We then decompose $f(\mathbf{E})$ as

$$f(\mathbf{E}) = \underbrace{\begin{pmatrix} \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix} \frac{1}{M} \begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{1}_M & \mathbf{X}^\top\mathbf{y} \\ \mathbf{1}_M^\top\mathbf{X} & M & \mathbf{1}_M^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{1}_M & \mathbf{y}^\top\mathbf{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{Q} \\ \boldsymbol{q}_1^\top \\ \boldsymbol{q}_2^\top \end{pmatrix}}_{\mathrm{I}} \cdot \mathbf{x} + \mathrm{II},$$

where terms I and II are independent of $\mathbf{x}$. Therefore, we have

$$\begin{aligned} \mathcal{R}(f) &:= \mathbb{E}\left(f(\mathbf{E}) - y\right)^2 \\ &= \mathbb{E}\left(f(\mathbf{E}) - \langle\widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle\right)^2 + \sigma^2 && \text{since } y|\widetilde{\boldsymbol{\beta}}, \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2) \\ &= \mathbb{E}\left(\mathrm{I}\cdot\mathbf{x} + \mathrm{II} - \langle\widetilde{\boldsymbol{\beta}}, \mathbf{x}\rangle\right)^2 + \sigma^2 && \text{since } f(\mathbf{E}) = \mathrm{I}\cdot\mathbf{x} + \mathrm{II} \\ &= \mathbb{E}\|\mathrm{I}^\top - \widetilde{\boldsymbol{\beta}}\|_{\mathbf{H}}^2 + \mathbb{E}\mathrm{II}^2 + \sigma^2 && \text{since } \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{H}) \\ &\geq \mathbb{E}\|\mathrm{I}^\top - \widetilde{\boldsymbol{\beta}}\|_{\mathbf{H}}^2 + \sigma^2. \end{aligned}$$

Note that the above equation holds if $\mathrm{II} = 0$, which can be easily achieved by setting $\boldsymbol{b} = \mathbf{0}_d, \boldsymbol{b}_1 = \boldsymbol{b}_2 = 0$. Therefore, without loss of generality, we can consider the LSA function which takes the following form:

$$f(\mathbf{E}) = \mathrm{I}\cdot\mathbf{x} = \begin{pmatrix} \boldsymbol{w}^\top & \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix} \frac{1}{M} \begin{pmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{1}_M & \mathbf{X}^\top\mathbf{y} \\ \mathbf{1}_M^\top\mathbf{X} & M & \mathbf{1}_M^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{1}_M & \mathbf{y}^\top\mathbf{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{Q} \\ \boldsymbol{q}_1^\top \\ \boldsymbol{q}_2^\top \end{pmatrix} \cdot \mathbf{x}. \tag{3.86}$$

Let's then try to determine whether such a function can effectively represent a GD-$\boldsymbol{\beta}$ function, which takes the form of

$$f_{\text{GD-}\boldsymbol{\beta}}(\mathbf{E}) = \langle \boldsymbol{\beta}^*, \mathbf{x} \rangle - \langle \frac{\boldsymbol{\Gamma}^* \mathbf{X}^\top (\mathbf{X}\boldsymbol{\beta}^* - \mathbf{y})}{M}, \mathbf{x} \rangle, \tag{3.87}$$

where $\boldsymbol{\beta}^*$ is the prior mean in Assumption 3.1 in our submission, and $\boldsymbol{\Gamma}^* = \boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Omega}^{-1}\mathbf{H}^{\frac{1}{2}}$ and $\boldsymbol{\Omega} = \frac{M+1}{M}\mathbf{H}^{\frac{1}{2}}\boldsymbol{\Psi}\mathbf{H}^{\frac{1}{2}} + \frac{\sigma^2+\text{tr}(\mathbf{H}\boldsymbol{\Psi})}{M}\mathbf{I}_d$ is defined in (5.3) in our submission. In order to achieve this, one simple way is to let

$$\boldsymbol{w} = -\boldsymbol{\beta}^*, \quad \boldsymbol{a}_1 = \boldsymbol{a}_2 = 1, \quad \boldsymbol{Q} = (\boldsymbol{\Gamma}^*)^\top, \quad \boldsymbol{q}_1 = \boldsymbol{\beta}^*, \quad \boldsymbol{q}_2 = \mathbf{0}_d. \tag{3.88}$$

However, this will incur some additive terms and will potentially enlarge the ICL risk. Inserting the parameters above, we have

$$f(\mathbf{E}) = f_{\text{GD-}\boldsymbol{\beta}}(\mathbf{E}) + \frac{\mathbf{1}_M^\top}{M}\left(\mathbf{X}(\boldsymbol{\Gamma}^*)^\top - \mathbf{x}\boldsymbol{\beta}^*(\boldsymbol{\beta}^*)^\top + \mathbf{y}(\boldsymbol{\beta}^*)^\top\right) \cdot \mathbf{x}$$

This shows that the extended token with a single LSA cannot easily implement the GD-$\boldsymbol{\beta}$ function class and may incur an additive ICL risk depending on $\boldsymbol{\beta}^*$ (as shown in Theorem 3.4.1).

Chapter 4

Minimax Optimal Convergence of Gradient Descent in Logistic Regression via Large and Adaptive Stepsizes

4.1 Introduction

A large class of optimization methods in machine learning are variants of *gradient descent* (GD). In this paper, we study the convergence of GD with *adaptive* stepsizes given by

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \nabla \mathcal{L}(\boldsymbol{w}_t), \quad t \geq 0, \tag{GD}$$

where $\mathcal{L}(\cdot)$ is the objective to be minimized, $\boldsymbol{w}_t \in \mathbb{R}^d$ is the trainable parameters, and $(\eta_t)_{t\geq 0}$ are the stepsizes. We allow the stepsize η_t to adapt to the current risk $\mathcal{L}(\boldsymbol{w}_t)$.

Classical analyses of GD require the stepsizes to be sufficiently small so that the risk decreases monotonically (Nesterov, 2018). This is often referred to as the *descent lemma*. For example, the descent lemma is satisfied for stepsizes such that $\eta_t < 2/\sup_{\mathcal{I}} \lambda_{\max}(\nabla^2 \mathcal{L}(\cdot))$, where the supremum is taken over the interval $\mathcal{I}$ connecting $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t+1}$ and $\lambda_{\max}(\cdot)$ is the maximum eigenvalue of a symmetric matrix. In this regime, a large volume of theory has been developed to show the convergence of GD in a variety of settings (see Lan, 2020, for example).

However, practical deep learning models trained by GD often converge in the long run while suffering from a locally oscillatory risk (Wu, Ma, et al., 2018; Xing et al., 2018; Lewkowycz et al., 2020; Cohen et al., 2020). This oscillation occurs when the stepsizes are too large and violate the descent lemma. This unstable convergence phenomenon is referred to by Cohen et al. (2020) as the *edge of stability* (EoS). To obtain a reasonable optimization and generalization performance in practice, it is often necessary to use large stepsizes so that GD enters the EoS regime (Wu, Ma, et al., 2018; Cohen et al., 2020), violating the seemingly theoretically desirable descent lemma.

Surprisingly, a recent line of works showed that GD provably converges faster by violating the descent lemma in various settings (see Altschuler and Parrilo, 2025b; Wu et al., 2024a, for examples, other related works are discussed later in Section 4.1). Specifically, Altschuler and Parrilo (2025b) proposed a stepsize scheduler in which GD violates the descent lemma but (occasionally) achieves an improved convergence rate for smooth convex optimization. The work by Wu et al. (2024a) focused on GD with a constant stepsize for logistic regression with linearly separable data. They showed that GD with a large stepsize can achieve an accelerated convergence rate, while this is impossible if the stepsize is small and enables the descent lemma. Note that the stepsizes considered in (Altschuler and Parrilo, 2025b; Wu et al., 2024a) are *oblivious*, which are determined before the GD run and do not adapt to the evolving risk.

Our results

This work complements the prior theory by considering the convergence of GD with large and *adaptive* stepsizes. We focus on logistic regression with linearly separable data. Specifically, the risk to be minimized is given by

$$\mathcal{L}(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i \mathbf{x}_i^\top \boldsymbol{w}), \quad \ell \in \Big\{\ell_{\mathsf{exp}} : z \mapsto \exp(-z),\ \ell_{\mathsf{log}} : z \mapsto \ln(1+\exp(-z))\Big\}, \tag{4.1}$$

where the loss function can be exponential or logistic losses and the dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfies the following standard conditions (Novikoff, 1962):

Assumption 4.1.1 (Linear separability)**.** Assume the dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfies

A. for every $i = 1, \dots, n$, $\|\mathbf{x}_i\| \le 1$ and $y_i \in \{\pm 1\}$;

B. there is a margin $\gamma > 0$ and a unit vector $\mathbf{w}^*$ such that $y_i \mathbf{x}_i^\top \mathbf{w}^* \ge \gamma$ for every $i = 1, \dots, n$.

Under Assumption 4.1.1, the minimizer of the risk in Equation (4.1) is at infinity; moreover, the landscape becomes flatter as the risk decreases. To compensate for this effect, we consider GD with the following *adaptive* stepsizes (proposed by Nacson et al., 2019; Ji and Telgarsky, 2021),

$$\eta_t := \eta \cdot (-\ell^{-1})' \circ \mathcal{L}(\boldsymbol{w}_t) \approx \eta / \mathcal{L}(\boldsymbol{w}_t),$$

where $\eta > 0$ is a constant hyperparameter.

We make the following significant contributions.

Benefits of large and adaptive stepsizes. We show that GD with adaptive stepsizes achieves improved convergence by entering the EoS regime. Specifically, we show that after at most $1/\gamma^2$ burn-in steps, GD attains a risk upper bounded by $\exp(-\Theta(\eta))$, which is arbitrarily small by setting η large enough. By doing so, however, the risk evolution could be non-monotonic. On the other hand, if the hyperparameter η is set such that GD does not

enter the EoS regime, we provide examples in which GD needs at least $\Omega(\ln(1/\varepsilon))$ steps to achieve an ε-risk. These together justify the benefits of operating in the EoS regime.

In comparison, prior works on the same problem focused on GD with a large but constant stepsize (Wu et al., 2024a) or adaptive but small stepsizes (Ji and Telgarsky, 2021). They established $\widetilde{\mathcal{O}}\big(1/(\gamma^2\sqrt{\varepsilon})\big)$ and $\mathcal{O}(\ln(1/\varepsilon)/\gamma^2)$ step complexity for GD to attain an ε-risk respectively (see Tables 4.1 and 4.2 for comparisons with other related works). In stark contrast, we show that, using large and adaptive steps together, $1/\gamma^2$ steps are sufficient.

Minimax lower bounds. Furthermore, we construct hard datasets with margin γ, in which *any* batch first-order methods must take $\Omega(\min\{\ln n, 1/\gamma^2\})$ steps to identify a linear separator of the dataset, while any online first-order method needs at least $\Omega(\min\{n, 1/\gamma^2\})$ updates. Thus, GD with large and adaptive stepsize is minimax optimal (ignoring constant factors) among all first-order batch methods when the sample size n is unrestricted. It is worth noting that the seminal *Perceptron* algorithm (Novikoff, 1962), a first-order online method, also takes $1/\gamma^2$ steps to find a linear separator. This matches GD even with constants.

General losses and two-layer networks. Finally, we extend our results beyond logistic regression. We establish conditions on the loss functions that enable our analysis. We also obtain similar results for a class of two-layer networks for fitting linearly separable data.

Related works

In this part, we review and discuss additional related papers.

Edge of stability. Our research is motivated by the empirically observed *edge of stability* (EoS) phenomenon (see Cohen et al., 2020, and references therein). Specifically, in practice, GD often induces a convergent yet oscillatory risk, implying the stepsizes are large such that the descent lemma is violated. A growing body of papers investigates the theoretical mechanism of EoS under various scenarios (Kong and Tao, 2020; Wang et al., 2022b; Wang et al., 2023d; Lyu et al., 2022; Wang et al., 2022c; Ma et al., 2022; Zhu et al., 2023; Damian et al., 2022; Ahn et al., 2022; Ahn et al., 2023b; Chen and Bruna, 2023; Kreisler et al., 2023; Even et al., 2023; Andriushchenko et al., 2023; Lu et al., 2023; Chen et al., 2024a). We are less interested in characterizing the mechanism of EoS; rather, we focus on understanding the benefits of operating in the EoS regime for improving optimization efficiency.

Aggressive stepsize schedulers. For smooth and (strongly) convex optimization, a recent line of research showed that GD with certain stepsize schedulers converges faster than GD with a constant stepsize (Kelner et al., 2022; Altschuler and Parrilo, 2025a; Altschuler and Parrilo, 2024; Altschuler and Parrilo, 2025b; Grimmer, 2024; Grimmer et al., 2023; Zhang and Jiang, 2024; Zhang et al., 2025c; Grimmer et al., 2025). Similar to our results, they obtained the acceleration because their stepsize schedulers violate the descent lemma (occasionally).

However, there are several notable differences. First, the considered problem classes are not directly comparable. They considered smooth and (strongly) convex optimization problems where the minimizers are finite. In contrast, we study the problem of finding a linear separator of a linearly separable dataset by minimizing a convex loss; this includes logistic regression as a special case. Although our logistic regression problem is smooth and convex, it does not admit a finite minimizer and, thus, does not belong to their problem class. Moreover, their stepsize schedulers are *oblivious*, determined before the GD run, while our stepsizes are *adaptive* to the current risk.

Logistic regression. Logistic regression with linearly separable data is a standard testbed for studying the implicit bias of GD (see Soudry et al., 2018; Ji and Telgarsky, 2018, and references thereafter). Specifically, this means that GD with a small constant stepsize (satisfying the descent lemma) diverges to infinity in norm but converges to the direction that maximizes the margin (Soudry et al., 2018; Ji and Telgarsky, 2018). The same result is later extended to GD with an arbitrarily large constant stepsize by Wu et al. (2023). Their results imply a risk convergence rate of $\widetilde{\mathcal{O}}(1/t)$ for GD with a small constant stepsize, where t is the number of steps (Soudry et al., 2018; Ji and Telgarsky, 2018). Remarkably, Wu et al. (2024a) showed that with a large constant stepsize, GD enters the EoS regime and attains an accelerated $\widetilde{\mathcal{O}}(1/t^2)$ rate. Our work is directly motivated by Wu et al. (2024a), although we consider GD with large and adaptive stepsizes while they focused on GD with a large constant stepsize.

Compared to the constant stepsize, it is known that adaptive stepsizes improve the risk (and margin) convergence rate of GD for logistic regression with separable data (Nacson et al., 2019; Ji and Telgarsky, 2021). Specifically, Nacson et al. (2019) and Ji and Telgarsky (2021) showed risk convergence rates of $\exp(-\Omega(\sqrt{t}))$ and $\exp(-\Omega(t))$, respectively. But they both considered GD with adaptive but small stepsizes that satisfy the descent lemma. In comparison, for the same algorithm, we show that simply letting the stepsizes be large can greatly accelerate the risk convergence.

Additionally, the risk (and margin) convergence rate can be further accelerated by using momentum techniques with GD (Ji et al., 2021; Wang et al., 2022a). The work by Ji et al. (2021) obtained $\exp(-\Omega(t^2 - \ln(t)))$ risk convergence rate (see Proposition 4.2.5), where the $\ln(t)$ term was later removed by Wang et al. (2022a). Their results are again limited to small stepsizes such that a certain potential decreases monotonically (see the discussions after Proposition 4.2.5). In contrast, we show that not using momentum but just using large stepsizes can lead to faster convergence.

Finally, we highlight that our lower bounds suggest that GD with large and adaptive stepsizes is minimax optimal. In contrast, GD with a large but constant stepsize (Wu et al., 2024a) or adaptive but small stepsizes (Ji and Telgarsky, 2021) or momentum (Ji et al., 2021; Wang et al., 2022a) are all suboptimal.

Perceptron. The seminal paper by Novikoff (1962) showed that the *Perceptron* algorithm takes $1/\gamma^2$ steps to find a linear separator of a dataset with margin γ. Our lower bound matches this rate up to constants in the online first-order setting. Note that Perceptron is an *online* method while GD for logistic regression considered in this paper is a *batch* method. In our analysis, entering the EoS regime is a key factor enabling GD to be minimax optimal. Interestingly, Perceptron can be viewed as one-pass stochastic gradient descent with stepsize 1 under the hinge loss, $z \mapsto \max\{0, -z\}$. Since the hinge loss is nonsmooth, stepsize 1 violates the descent lemma, so Perceptron effectively operates in the EoS regime, too. It seems that operating in the EoS regime might be necessary for achieving first-order minimax optimality in this problem.

A recent paper by Tyurin (2025) proposed a batch version of the Perceptron algorithm, attaining the same step complexity of $1/\gamma^2$ when translated to our notation. This is also minimax optimal according to our lower bound. Their method can be viewed as GD with adaptive stepsizes. Their stepsize scheme is motivated by Perceptron, while our stepsize scheme is motivated by the loss landscape (Nacson et al., 2019; Ji and Telgarsky, 2021). We also point out that Tyurin (2025) did not provide minimax lower bounds.

A recent paper by Kornowski and Shamir (2025) considered lower bounds for finding linear separators of a linearly separable dataset from a game-theoretic perspective. Our lower bounds are connected to theirs but with several differences. First, our lower bound for batch methods (Theorem 4.3.2) is comparable with their Theorem 4.2: their Theorem 4.2 covers more algorithms, but only showing an $\Omega(\gamma^{-2/3})$ lower bound (ignoring dependence on sample size), while our Theorem 4.3.2 shows an $\Omega(\gamma^{-2})$ lower bound but only covers first-order batch methods. Second, their Theorem 4.1 matches our lower bound in Theorem 4.3.4 for online methods while covering more algorithms. They both can be viewed as solutions to Shalev-Shwartz and Ben-David (2014, Section 9.6, Exercise 3). We choose to keep our version as it is easier to use in our context.

4.2 Logistic Regression

In this section, we present an improved analysis for GD with large and adaptive stepsizes for logistic regression on linearly separable data.

Convergence of GD with large and adaptive stepsizes

Recall the definitions of Equation (GD) and the objective Equation (4.1). Recall the adaptive stepsizes are defined as

$$\eta_t := \eta \cdot \left(-\ell^{-1}\right)' \circ \mathcal{L}(\boldsymbol{w}_t) = \begin{cases} \dfrac{\eta}{\mathcal{L}(\boldsymbol{w}_t)} & \ell = \ell_{\mathsf{exp}}, \\ \dfrac{\eta \exp(\mathcal{L}(\boldsymbol{w}_t))}{\exp\left(\mathcal{L}(\boldsymbol{w}_t)\right) - 1} & \ell = \ell_{\mathsf{log}}. \end{cases} \tag{4.2}$$

It is easy to check that $\|\nabla^2 \mathcal{L}(\boldsymbol{w})\| \leq \mathcal{L}(\boldsymbol{w})$ under Assumption 4.1.1. Thus, the landscape becomes flatter as the risk $\mathcal{L}(\boldsymbol{w})$ decreases. The adaptive stepsizes Equation (4.2) are designed to compensate for the flattened curvature (Nacson et al., 2019; Ji and Telgarsky, 2021). Moreover, we point out that GD with adaptive stepsizes Equation (4.2) for logistic regression Equation (4.1) is equivalent to (Ji and Telgarsky, 2021)

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta \nabla \phi(\boldsymbol{w}_t), \quad \phi(\boldsymbol{w}) := -\ell^{-1}\left(\mathcal{L}(\boldsymbol{w})\right). \tag{4.3}$$

This is GD with a constant stepsize η under a transformed objective $\phi(\cdot)$. Equivalently, $\phi(\cdot)$ can be viewed as a smooth surrogate of the unnormalized margin (Hardy et al., 1952). This has been exploited by Ji and Telgarsky (2021), where they established a primal-dual analysis of GD and obtained an improved margin convergence rate. Unlike Ji and Telgarsky (2021), we focus on the risk convergence and obtain the following improved results.

Theorem 4.2.1 (GD with large and adaptive stepsizes)**.** *Consider* (GD) *with adaptive stepsizes* (4.2) *for logistic regression Equation* (4.1) *under Assumption 4.1.1. Assume without loss of generality that* $\boldsymbol{w}_0 = \boldsymbol{0}$*. Then for every* $t \geq 1$ *and* $\eta > 0$*, we have*

$$\mathcal{L}\left(\overline{\boldsymbol{w}}_t\right) \leq \exp\left(-\frac{(\gamma^2(t+1))^2 - 1}{4\gamma^2(t+1)}\eta\right), \quad \textit{where} \quad \overline{\boldsymbol{w}}_t := \frac{1}{t+1}\sum_{k=0}^{t} \boldsymbol{w}_k.$$

In particular, after $1/\gamma^2$ *burn-in steps, for every* $\eta > 0$*, we have*

$$\mathcal{L}(\overline{\boldsymbol{w}}_t) \leq \exp\left(-\frac{\gamma^2\eta}{4}\right) = \exp\left(-\Theta\left(\eta\right)\right), \quad t \geq \frac{1}{\gamma^2}.$$

The proof of Theorem 4.2.1 is deferred to Section 4.2. Theorem 4.2.1 provides a sharp risk convergence bound for GD with large and adaptive stepsizes. Note that the hyperparameter η can be chosen arbitrarily large. Therefore, with a sufficiently large η, GD achieves an arbitrarily small risk right after $1/\gamma^2$ burn-in steps. In this case, however, the evolution of the risk might not be monotonic.

In other words, to attain an ε-risk, GD with adaptive and large stepsize only needs $1/\gamma^2$ steps, where the step complexity is independent of targeted risk ε (but the smallest base stepsize depends on ε). This is in stark contrast to the step complexity of GD with a large but constant stepsize (Wu et al., 2024a) or adaptive but small stepsizes (Ji and Telgarsky, 2021) (see Table 4.1 and a detailed discussion later in Section 4.2). Moreover, we will show that this $1/\gamma^2$ step complexity is minimax optimal up to constant factors in Section 4.3. We note that Theorem 4.2.1 is stated for the averaged iterate $\overline{\boldsymbol{w}}_t$. The same proof also yields a comparable best-iterate guarantee for $\min_{0\leq k\leq t} \mathcal{L}(\boldsymbol{w}_k)$, but it does not provide a comparable last-iterate guarantee for $\mathcal{L}(\boldsymbol{w}_t)$ during the non-monotone phase induced by large adaptive stepsizes. We treat this as one important future research direction.

Benefits of EoS. In Theorem 4.2.1, GD achieves the $1/\gamma^2$ step complexity by using large stepsizes and entering the EoS regime. Our next theorem suggests this is necessary by providing a lower bound on the convergence rate for adaptive stepsize GD that avoids the EoS phase.

Theorem 4.2.2 (A lower bound for GD in the stable regime)**.** *Consider* (GD) *with adaptive stepsizes* (4.2) *for logistic regression Equation* (4.1) *with the following dataset*

$$\mathbf{x}_1 = (\gamma, \sqrt{1-\gamma^2}), \quad \mathbf{x}_2 = (\gamma, -\sqrt{1-\gamma^2}), \quad y_1 = y_2 = 1,$$

where $0 < \gamma < 0.1$. *This dataset satisfies Assumption 4.1.1. Let* $\boldsymbol{w}_0 = \mathbf{0}$. *For all hyperparameter* η *such that* $(\mathcal{L}(\boldsymbol{w}_t))_{t\geq 0}$ *is nonincreasing, we have*

$$\mathcal{L}(\overline{\boldsymbol{w}}_t),\ \mathcal{L}(\boldsymbol{w}_t) \geq \exp(-ct), \quad t \geq 1,$$

where $c > 0$ *is a parameter that depends on* γ *but is independent of* t *and* η.

The proof of Theorem 4.2.2 is deferred to Section 4.7. Theorem 4.2.2 is motivated by Theorem 3 in (Wu et al., 2024a), which provides a risk lower bound for GD with a constant stepsize that satisfies the descent lemma. Theorem 4.2.2 shows that if GD with adaptive stepsizes satisfies the descent lemma, then the step complexity is at least $\Omega(\ln(1/\varepsilon))$ in the worst case. By contrast, Theorem 4.2.1 shows that GD achieves an ε-independent step complexity by violating the descent lemma. Theorems 4.2.1 and 4.2.2 together justify the optimization benefits of adaptive stepsize GD to operate in the EoS regime.

Comparisons with Prior Results

In this part, we review representative existing results on variants of GD for logistic regression with linearly separable data (Ji and Telgarsky, 2018; Ji and Telgarsky, 2021; Ji et al., 2021; Wu et al., 2024a) and compare their results with ours. Table 4.1 provides an overview of the comparisons. We discuss each result in detail below.

A constant stepsize. The work by Ji and Telgarsky (2018) considered GD with a small constant stepsize satisfying the descent lemma and obtained a $\tilde{\mathcal{O}}(1/(\gamma^2 t))$ convergence rate (see their Theorem 3.1). This translates to a $\tilde{\mathcal{O}}(1/(\gamma^2\varepsilon))$ step complexity. Later, the work by Wu et al. (2024a) obtained a faster rate by considering GD with a large constant stepsize that violates the descent lemma. Specifically, they showed the following.

Proposition 4.2.3 (Corollary 2 in (Wu et al., 2024a))**.** *Consider* (GD) *with constant stepsize* $\eta_t = \eta > 0$ *for logistic regression Equation* (4.1) *with logistic loss* $\ell_{\log}$ *under Assumption 4.1.1. Let* $\boldsymbol{w}_0 = 0$. *For a given step budget* $T \geq \max\{e, n\}/\gamma^2$, *there exists* $\eta = \Theta(T)$ *such that*

$$\mathcal{L}(\boldsymbol{w}_T) \leq C\frac{\ln^2(T)}{\gamma^4 T^2},$$

where $C > 1$ *is a numerical constant.*

Table 4.1: Step complexities for GD with various designs to achieve an ε-risk for logistic regression.

stepsize/momentum design	step complexity
small, constant (Ji and Telgarsky, 2018, Theorem 3.1)	$\widetilde{\mathcal{O}}\left(1/\left(\gamma^2\varepsilon\right)\right)$
large, constant (Wu et al., 2024a, Corollary 2)	$\widetilde{\mathcal{O}}\left(1/\left(\gamma^2\sqrt{\varepsilon}\right)\right)$ for $\varepsilon < \Theta(1/n)$
small, adaptive (Ji and Telgarsky, 2021, Theorem 2.2)	$\mathcal{O}\left(\ln\left(1/\varepsilon\right)/\gamma^2\right)$
small, adaptive, and momentum (Ji et al., 2021, Theorem 3.1)	$\mathcal{O}\Big(\sqrt{\ln(1/\varepsilon) + \ln(n)\ln\ln(n)}/\gamma\Big)$
large, adaptive (Theorem 4.2.1)	$\leq 1/\gamma^2$
minimax lower bound (Theorem 4.3.2)	$\Omega(1/\gamma^2)$

Proposition 4.2.3 leads to an improved step complexity, $\widetilde{\mathcal{O}}(1/(\gamma^2\sqrt{\varepsilon}))$, for GD with a large constant stepsize. There are three caveats. First, this improvement only happens for $\varepsilon < \Theta(1/n)$ (hence, it does not help with finding a linear separator; see Section 4.3). Second, this improvement works under the logistic loss but not under the exponential loss; in fact, Wu et al. (2023) constructed a separable dataset where GD with a large constant stepsize under the exponential loss does not converge (see their Theorem 4.2). This gap stems from the logistic loss being Lipschitz while the exponential loss is not. Finally, the stepsize is a function of the step budget, meaning that the algorithm needs to be rerun from the beginning when the step budget is changed.

Compared to their results for GD with a constant stepsize (Ji and Telgarsky, 2018; Wu et al., 2024a), we show that GD with large and adaptive stepsizes Equation (4.2) achieves a strictly better step complexity of $1/\gamma^2$. Interestingly, our analysis allows for both logistic and exponential losses. This is not contradictory to the counter-example offered by Wu et al. (2023). Recall that GD with adaptive stepsizes Equation (4.2) can be viewed as GD with a constant stepsize under a transformed objective $\phi(\cdot)$ defined in Equation (4.3). While the exponential loss is not Lipschitz, the transformed objective $\phi(\cdot)$ is Lipschitz (see Lemma 4.6.1 in Section 4.6), enabling large stepsizes. Finally, unlike Proposition 4.2.3, the adaptive stepsize scheduler Equation (4.2) used in Theorem 4.2.1 is independent of the step budget (as long as setting η sufficiently large).

Small adaptive stepsizes. Before our paper, the best analysis for GD with adaptive stepsizes Equation (4.2) is by Ji and Telgarsky (2021). Their results only allow small stepsizes, summarized as follows.

Proposition 4.2.4 (Consequences of Theorem 2.2 in (Ji and Telgarsky, 2021))**.** *Consider* (GD) *with adaptive stepsizes* (4.2) *for logistic regression Equation* (4.1) *with exponential loss* $\ell_{\exp}$ *under Assumption 4.1.1. Then the transformed loss* ϕ *(defined in* (4.3)*) is* 1*-smooth with respect to* ℓ_∞*-norm. Let* $\boldsymbol{w}_0 = 0$*. Then for every* $\eta \leq 1$*, we have* $\mathcal{L}(\boldsymbol{w}_t)$ *decreases monotonically and*

$$\mathcal{L}(\boldsymbol{w}_t) \leq C \exp\left(-\gamma^2 \eta t\right),$$

where $C > 1$ *is a numerical constant.*

We note that the main focus of Ji and Telgarsky (2021) is to prove a fast margin convergence rate, and Proposition 4.2.4 is merely a side product of their results. Proposition 4.2.4 leads to an $\mathcal{O}(\ln(1/\varepsilon)/\gamma^2)$ step complexity for GD with small adaptive stepsizes. In comparison, we analyze the same algorithm, but our Theorem 4.2.1 applies to both large and small adaptive stepsizes. Our results suggest that the step complexity reduces to $1/\gamma^2$ when the stepsizes are sufficiently large.

Momentum. The work by Ji et al. (2021) considered a version of momentum GD for logistic regression, achieving a faster margin convergence rate compared to GD. Their results also imply a risk convergence rate, summarized as follows.

Proposition 4.2.5 (Consequences of Lemmas C.7 and C.12 in (Ji et al., 2021))**.** *Consider logistic regression Equation* (4.1) *with exponential loss* $\ell_{\exp}$ *under Assumption 4.1.1. For a version of momentum GD (see Algorithm 1 in Ji et al. (2021)) with* $\boldsymbol{w}_0 = 0$ *and suitably chosen stepsizes, we have*

$$\mathcal{L}(\boldsymbol{w}_t) \leq \exp\left(-C\left(\gamma^2 t^2 - \ln(t+1)\ln(n)\right)\right), \quad t \geq 1,$$

where $C > 0$ *is a numerical constant.*

Proposition 4.2.5 implies an $\mathcal{O}\left(\sqrt{\ln(1/\varepsilon) + \ln(n)\ln\ln(n)}/\gamma\right)$ step complexity for momentum GD. We remark that with a more careful momentum design, Wang et al. (2022a) improved the $\ln(t+1)\ln(n)$ term in the above proposition to $\ln(n)$, which leads to a slightly improved step complexity of $\mathcal{O}\left(\sqrt{\ln(1/\varepsilon) + \ln(n)}/\gamma\right)$. As the improvement is small, we choose to mainly compare with the earlier results by Ji et al. (2021).

Recall that our Theorem 4.2.1 suggests a $1/\gamma^2$ step complexity for GD with large and adaptive stepsizes. In comparison, the step complexity of their momentum GD has a better dependence on γ but has a worse dependence on ε and n. As shown later in Section 4.3, our step complexity is minimax optimal if there are no restrictions on the sample size n. Note that the analysis by Ji et al. (2021) still relies on the monotonic decreasing of a potential (see Ji et al., 2021, Equation (B.9) in the proof of Lemma B.3), which needs the stepsize to be small. As their momentum GD is designed to minimize a dual objective with a finite minimizer, it remains unclear whether their momentum GD can be used with large stepsizes. We leave this as future work.

Proof of Theorem 4.2.1

Proof. **(of Theorem 4.2.1)** As explained in Equation (4.3), it is equivalent to considering GD with a constant stepsize under a transformed objective $\phi(\cdot)$. We can check that $\phi(\cdot)$ is convex (see Lemma 5.2 in (Ji and Telgarsky, 2021) or Lemma 4.6.3 in Section 4.6) and 1-Lipschitz (see Lemma 4.6.1 in Section 4.6). We then use the split optimization technique developed by Wu et al. (2024a). Specifically, for a comparator $\mathbf{u} := \mathbf{u}_1 + \mathbf{u}_2$, we have

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \mathbf{u}\|^2 &= \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u} - \boldsymbol{w}_t\rangle + \eta^2 \|\nabla\phi(\boldsymbol{w}_t)\|^2 \\
&= \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_1 - \boldsymbol{w}_t\rangle + \eta \left[2\langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_2\rangle + \eta \|\nabla\phi(\boldsymbol{w}_t)\|^2\right] \\
&\le \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_1 - \boldsymbol{w}_t\rangle \qquad (4.4) \\
&\le \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \left(\phi(\mathbf{u}_1) - \phi(\boldsymbol{w}_t)\right), \qquad (4.5)
\end{aligned}$$

where Equation (4.4) is by the following inequality (see Lemma 4.6.2 in Section 4.6)

$$2\langle \nabla\phi(\boldsymbol{w}), \mathbf{u}_2\rangle + \eta \|\nabla\phi(\boldsymbol{w})\|^2 \le 0 \quad \text{for} \quad \mathbf{u}_2 := \frac{\eta}{2\gamma}\mathbf{w}^*, \tag{4.6}$$

and Equation (4.5) is by the convexity of $\phi(\cdot)$. Rearranging (4.5) and telescoping the sum, we obtain

$$\frac{\|\boldsymbol{w}_{t+1} - \mathbf{u}\|^2}{2\eta(t+1)} + \frac{1}{t+1}\sum_{k=0}^{t} \phi(\boldsymbol{w}_k) \le \phi(\mathbf{u}_1) + \frac{\|\mathbf{u}\|^2}{2\eta(t+1)}. \tag{4.7}$$

For $\mathbf{u}_1 \propto \boldsymbol{w}^*$, we have $\phi(\mathbf{u}_1) \le -\gamma\|\mathbf{u}_1\|$ by Assumption 4.1.1. Further setting $\|\mathbf{u}_1\| = \gamma\eta(t+1)/2$, we get

$$\frac{1}{t+1}\sum_{k=0}^{t} \phi(\boldsymbol{w}_k) \le -\gamma\|\mathbf{u}_1\| + \frac{\|\mathbf{u}_1 + \mathbf{u}_2\|^2}{2\eta(t+1)} \le -\frac{(\gamma^2(t+1))^2 - 1}{4\gamma^2(t+1)}\eta. \tag{4.8}$$

We complete the proof by applying the convexity of $\phi(\cdot)$ and the fact that $\mathcal{L}(\cdot) = \ell(-\phi(\cdot))$. □

4.3 Minimax Lower Bounds for First-Order Methods

In this section, we consider the task of finding a linear separator for a linearly separable dataset. Specifically, this means to find a parameter $\hat{\boldsymbol{w}}$ such that

$$\min_{i\in[n]} y_i \mathbf{x}_i^\top \hat{\boldsymbol{w}} > 0.$$

We point out the fact that an optimization method can find a linear separator by solving the logistic regression problem Equation (4.1) sufficiently well.

Fact 1. In logistic regression Equation (4.1), if $\mathcal{L}(\hat{\boldsymbol{w}}) < \ell(0)/n$, then $\min_{i\in[n]} y_i \mathbf{x}_i^\top \hat{\boldsymbol{w}} > 0$.

In this section, we establish minimax lower bounds on the step complexity needed by any first-order methods to solve this task. We then compare the performance for GD with various designs in this task, as summarized in Table 4.2 and will be detailed later in this section. We start with first-order batch methods and then discuss first-order online methods.

Table 4.2: Step complexities for first-order methods to find a linear separator.

stepsize/momentum/loss design	type	step complexity
constant (Ji and Telgarsky, 2018, Theorem 3.1)	batch	$\widetilde{\mathcal{O}}\left(n/\gamma^2\right)$
small, adaptive (Ji and Telgarsky, 2021, Theorem 2.2)	batch	$\mathcal{O}\left(\ln(n)/\gamma^2\right)$
small, adaptive, and momentum (Ji et al., 2021, Theorem 3.1)	batch	$\mathcal{O}\Big(\sqrt{\ln(n)\ln\ln(n)/\gamma^2}\Big)$
normalized batch Perceptron (Tyurin, 2025, Theorem 5.3)	batch	$\leq 1/\gamma^2$
large and adaptive (Theorem 4.2.1)	batch	$\leq 1/\gamma^2$
minimax lower bound (Theorem 4.3.2)	batch	$\Omega(\min\{1/\gamma^2, \ln(n)\})$
constant (Wu et al., 2024a, Theorem 4)	online	$\mathcal{O}\left(1/\gamma^2\right)$
Perceptron (Novikoff, 1962)	online	$\leq 1/\gamma^2$
minimax lower bound (Theorem 4.3.4)	online	$\Omega\Big(\min\{1/\gamma^2, n\}\Big)$

A lower bound for first-order batch methods

We formally define first-order batch methods for our problem as follows.

Definition 4.3.1 (First-order batch methods)**.** Let $\ell(\cdot)$ be a locally Lipschitz function. For each z, let $\ell'(z)$ be a unique element from the Clarke subdifferential of $\ell(\cdot)$ at z (Clarke, 1990). For a given dataset $(\mathbf{x}_i, y_i)_{i=1}^n$, define the batch gradient as

$$\nabla\mathcal{L}(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n}\ell'\Big(y_i\mathbf{x}_i^\top\boldsymbol{w}\Big)y_i\mathbf{x}_i,$$

We say $\boldsymbol{w}_t$ is the output of a first-order batch method in t steps with initialization $\boldsymbol{w}_0$ on dataset $(\mathbf{x}_i, y_i)_{i=1}^n$, if it can be generated by

$$\boldsymbol{w}_k \in \boldsymbol{w}_0 + \mathsf{Lin}\left\{\nabla\mathcal{L}(\boldsymbol{w}_0), \ldots, \nabla\mathcal{L}(\boldsymbol{w}_{k-1})\right\}, \quad k = 1, \ldots, t,$$

where "Lin" is the linear span of a vector set and "$+$" is the Minkowski addition.

Definition 4.3.1 characterizes a class of first-order methods for finding linear separators. Compared to the class of first-order methods for smooth convex optimization (Nesterov, 2018, Assumption 2.1.4), our Definition 4.3.1 requires the predictor to be linear—since the goal

is to find a linear separator—but does not require the objective function to be smooth or convex. We establish the following lower bounds.

Theorem 4.3.2 (A lower bound for first-order batch methods)**.** *For every $0 < \gamma < 1/6$, $n > 16$, and $\boldsymbol{w}_0$, there exists a dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfying Assumption 4.1.1 such that the following holds. For any $\boldsymbol{w}_t$ output by a first-order batch method in t-steps with initialization $\boldsymbol{w}_0$ on this dataset, we have*

$$\min_{i\in[n]} y_i \mathbf{x}_i^\top \boldsymbol{w}_t > 0 \quad \textit{implies that} \quad t \geq \min\left\{\frac{\ln n}{8\ln 2}, \frac{1}{30\gamma^2}\right\}.$$

The proof of Theorem 4.3.2 is deferred to Section 4.8. The proof is based on a dimension argument, motivated by the classical lower bounds for first-order methods in smooth convex optimization (Nesterov, 2018, Theorem 2.1.7). We emphasize that Theorem 4.3.2 should be interpreted as a worst-case lower bound for hard high-dimensional instances as the $1/\gamma^2$ branch is active only when $\ln n \gtrsim 1/\gamma^2$, that is, when n can be exponentially large in $1/\gamma^2$.

Minimax optimality. Due to Fact 1 and Theorem 4.2.1, GD with large and adaptive stepsizes takes $1/\gamma^2$ steps to find a linear separator. Our lower bound in Theorem 4.3.2 suggests that this is minimax optimal ignoring constant factors, in the sense that there is no restriction on the sample size n (so n is allowed to be exponential in $1/\gamma$ in the worst case).

Prior results. Applying Fact 1 to the prior convergence results summarized in Table 4.1, we can conclude the step complexities of the variants of GD considered in previous works (Ji and Telgarsky, 2018; Ji and Telgarsky, 2021; Ji et al., 2021; Wu et al., 2024a) for finding a linear separator. Additionally, the work by Tyurin (2025) provided a first-order batch method called normalized batch Perceptron, which achieves $1/\gamma^2$ step complexity for finding the linear separator. These results are summarized in Table 4.2.

We make three remarks. First, since the acceleration effect of a large constant stepsize only appears for $\varepsilon < \Theta(1/n)$ (see Table 4.1), a large constant stepsize considered by Wu et al. (2024a) does not help GD to find a linear separator (but also does not hurt) compared to the results in (Ji and Telgarsky, 2018). Second, when γ is fixed and the sample size n is allowed to be arbitrary, the methods considered in (Ji and Telgarsky, 2018; Ji and Telgarsky, 2021; Ji et al., 2021) are all suboptimal in the worst case, while GD with large and adaptive stepsizes and the normalized batch Perceptron by Tyurin (2025) are minimax optimal. Finally, note that Ji et al. (2021) obtained an $\mathcal{O}(\sqrt{\ln(n)\ln\ln(n)/\gamma^2})$ step complexity by using momentum techniques (note that this can be improved to $\mathcal{O}(\sqrt{\ln(n)/\gamma^2})$ by Wang et al. (2022a) as discussed after Proposition 4.2.5). Their results do not violate the lower bound in Theorem 4.3.2, but suggest that the $1/\gamma^2$ term in our lower bound might be improvable in the regime where $n = \mathrm{poly}(1/\gamma)$.

It turns out that identifying the correct trade-off between n, $1/\gamma$, and the ambient dimension d is challenging. In addition to Theorem 4.3.2, we give an alternative dataset

construction that leads to a lower bound of $\Omega(\min\{\gamma^{-2/3}, n\})$ (see Theorem 4.8.1 in Section 4.8). We leave it as future work to prove the first-order minimax step complexity that is tight for all choices of $1/\gamma$, n, and the ambient dimension d.

A lower bound for first-order online methods

As a side product, we also establish a step complexity lower bound for first-order online methods for finding a linear separator. We formally define a first-order online method as follows.

Definition 4.3.3 (First-order online methods)**.** Let $\ell(\cdot)$ be a locally Lipschitz function. For each z, let $\ell'(z)$ be a unique element from the Clarke subdifferential of $\ell(\cdot)$ at z (Clarke, 1990). We say a sequence $(\boldsymbol{w}_k)_{k=0}^t$ is generated by a first-order online method with initialization $\boldsymbol{w}_0$ on dataset $(\mathbf{x}_i, y_i)_{i=1}^t$, if it satisfies

$$\boldsymbol{w}_k \in \boldsymbol{w}_0 + \mathsf{Lin}\left\{\ell'\Big(y_i \mathbf{x}_i^\top \boldsymbol{w}_{i-1}\Big) y_i \mathbf{x}_i,\ i = 1, \ldots, k\right\}, \quad k = 1, \ldots, t,$$

where "Lin" is the linear span of a vector set and "$+$" is the Minkowski addition.

The following theorem presents our lower bound for the first-order online method. A version of this theorem (that covers more algorithms) also appears in (Kornowski and Shamir, 2025, Theorem 4.1), both of which can be viewed as solutions to Exercise 3 in Section 9.6 of Shalev-Shwartz and Ben-David (2014). Our statement of the lower bound is easier to use in our context.

Theorem 4.3.4 (Lower bounds for online first-order methods)**.** *For every $0 < \gamma < 1/2, n \geq 2$, and $\boldsymbol{w}_0$, there exists a dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfying Assumption 4.1.1 such that the following holds.*

- *For any sequence $(\boldsymbol{w}_k)_{k=0}^t$ generated by a first-order online method with initialization $\boldsymbol{w}_0$ and dataset $\left(\mathbf{x}_{\pi(k)}, y_{\pi(k)}\right)_{k=1}^t$, where $\pi(k) \in [n]$ but is otherwise arbitrary, we have*

$$\min_{i \in [n]} y_i \mathbf{x}_i^\top \boldsymbol{w}_t > 0 \quad \textit{implies that} \quad t \geq \min\left\{\frac{1}{2\gamma^2}, n\right\}.$$

- *For any sequence $(\boldsymbol{w}_k)_{k=0}^n$ generated by a first-order online method with initialization $\boldsymbol{w}_0$ and dataset $(\mathbf{x}_i, y_i)_{i=1}^n$, we have*

$$\sum_{k=1}^t \mathbb{1}\{y_k \mathbf{x}_k^\top \boldsymbol{w}_{k-1} \leq 0\} \geq \min\left\{\frac{1}{2\gamma^2}, t\right\} \quad \textit{for every} \quad 1 \leq t \leq n.$$

We include the proof of Theorem 4.3.4 in Section 4.8 for completeness. We remark that the dataset construction in Theorem 4.3.4 is different from the one in Theorem 4.3.2. From

an offline perspective, Theorem 4.3.4 suggests that any first-order online methods need to make $\Omega(1/\gamma^2)$ updates to find a linear separator of a given dataset (assuming that n is large). From an online perspective, it makes $\Omega(1/\gamma^2)$ mistake steps (or regret in zero-one loss) for a large t in the worst case.

Perceptron. Perceptron is a classical method for finding a linear separator of a linearly separable dataset. It takes the following updates:

$$\boldsymbol{w}_0 = \mathbf{0}, \quad \boldsymbol{w}_t := \boldsymbol{w}_{t-1} + \mathbb{1}\{y_t \mathbf{x}_t^\top \boldsymbol{w}_{t-1} \le 0\} y_t \mathbf{x}_t, \quad t \ge 1.$$

Under Assumption 4.1.1, a seminal analysis by Novikoff (1962) suggests that Perceptron makes at most $1/\gamma^2$ mistake steps when run over a dataset of arbitrary size (that is, the total regret in zero-one loss is at most $1/\gamma^2$ for any t).

Note that Perceptron can be viewed as online stochastic gradient descent under the hinge loss $\ell_{\text{hinge}}(z) := \max\{0, -z\}$ with stepsize 1 and subderivative choice $\ell'_{\text{hinge}}(0) := 1$. Thus, Theorem 4.3.4 applies to Perceptron, suggesting that Perceptron is minimax optimal when the sample size is unrestricted. This is also known from Kornowski and Shamir (2025, Theorem 4.1) and can be viewed as a solution to Exercise 3 in Section 9.6 of Shalev-Shwartz and Ben-David (2014).

We also point out that the hinge loss is non-smooth. So, the stepsize 1 used in Perceptron also violates the descent lemma (although Perceptron is online), similar to our large stepsize GD. We conjecture that violating the descent lemma might be a fundamental property for attaining first-order optimality in this task.

SGD for logistic regression. Similarly to Perceptron, online stochastic gradient descent with a (large) constant stepsize for logistic regression also makes $\mathcal{O}(1/\gamma^2)$ mistake steps (see the proof of Theorem 4 in Wu et al., 2024a, for example). Since the logistic loss is close to the hinge loss when zooming away from zero, we should expect SGD with large stepsizes for logistic regression to behave like Perceptron.

4.4 Extensions

This section extends results in Section 4.2 to a two-layer network and a generic class of loss functions.

Two-Layer Networks

We consider a two-layer network defined as (Brutzkus et al., 2018)

$$f(\boldsymbol{w}; \mathbf{x}) := \frac{1}{m} \sum_{j=1}^{m} a_j \sigma\left(\mathbf{x}^\top \boldsymbol{w}^{(j)}\right), \quad \boldsymbol{w} := \left(\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \ldots, \boldsymbol{w}^{(m)}\right), \tag{4.9}$$

where m is the number of neurons, $a_j \in \{\pm 1\}$ for $j = 1, \ldots, m$ are fixed parameters, $\boldsymbol{w}$ are the trainable parameters, and $\sigma(z) := \max\{z, \alpha z\}$ for $0 < \alpha < 1$ is the leaky ReLU activation. We fix a choice of subderivative $\sigma'(0) := 1$ for clarity. But our results extend to any choice of subderivative $\sigma'(0) \in [\alpha, 1]$. The objective is then given by

$$\mathcal{L}(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i f\left(\boldsymbol{w}; \mathbf{x}_i\right)\right), \tag{4.10}$$

where $\ell(\cdot)$ is exponential loss or logistic loss and the dataset satisfies Assumption 4.1.1. Similarly, we consider GD with adaptive stepsizes (4.2). The next theorem provides an improved convergence analysis for GD with large and adaptive stepsizes.

Theorem 4.4.1 (A two-layer network)**.** *Consider* (GD) *with adaptive stepsizes* (4.2) *for minimizing objective* (4.10) *under Assumption 4.1.1. Assume without loss of generality that* $\boldsymbol{w}_0 = \mathbf{0}$*. Then for every* $t \geq 1$ *and* $\eta > 0$*, we have*

$$\min_{k \leq t} \mathcal{L}(\boldsymbol{w}_k) \leq \exp\left(-\frac{(\alpha\gamma^2(t+1))^2 - 1}{4\gamma^2(t+1)}\eta\right).$$

In particular, after $1/(\alpha\gamma^2)$ *burn-in steps, for every* $\eta > 0$*, we have*

$$\min_{k \leq t} \mathcal{L}(\boldsymbol{w}_k) \leq \exp\left(-\frac{\alpha^2\gamma^2\eta}{4}\right) = \exp\left(-\Theta\left(\eta\right)\right), \quad t \geq \frac{1}{\alpha\gamma^2}.$$

The proof of Theorem 4.4.1 is deferred to Section 4.9, which combines techniques from the proof of Theorem 4.2.1 and the analysis of GD with a large constant stepsize for two-layer networks by Cai et al. (2024). Theorem 4.4.1 suggests GD with large and adaptive stepsizes attains an arbitrarily small risk right after $1/(\alpha\gamma^2)$ burn-in steps. In comparison, the work by Cai et al. (2024) only obtained an $\widetilde{\mathcal{O}}(1/t^2)$ rate. Other discussions for Theorem 4.2.1 also apply here in a similar manner.

We remark that the leaky ReLU activation in Theorem 4.4.1 can be extended to other near-homogeneous activation functions, such as leaky GeLU, leaky Softplus, and leaky SiLU (see Cai et al., 2024). This is done in Section 4.9.

General loss functions

In this part, we extend our results in Section 4.2 from logistic and exponential losses to a generic class of loss functions.

Assumption 4.4.2 (Loss function conditions)**.** Let $\ell : \mathbb{R} \to (0, \infty)$ be a positive and continuously differentiable function. Assume that

A. The loss function ℓ is positive, strictly decreasing, and convex.

B. For $z > 0$, the inverse $\ell^{-1}(z)$ exists, and is differentiable and decreasing.

C. The transformed objective $\phi(\cdot)$ in (4.3) is convex and C_ℓ-Lipschitz for a constant $C_\ell \geq 1$.

Our next theorem shows that Theorem 4.2.1 applies to loss functions beyond the logistic and exponential losses, as long as they satisfy Assumption 4.4.2.

Theorem 4.4.3 (General loss functions)**.** *Consider* (GD) *with adaptive stepsizes* (4.2) *for objective function* $\mathcal{L}(\boldsymbol{w}) := (1/n)\sum_{i=1}^n \ell\left(y_i \mathbf{x}_i^\top \boldsymbol{w}\right)$ *under Assumptions 4.1.1 and 4.4.2. Assume without loss of generality that* $\boldsymbol{w}_0 = \mathbf{0}$*. Then for every* $t \geq 1$ *and* $\eta > 0$*, we have*

$$\mathcal{L}\left(\overline{\boldsymbol{w}}_t\right) \leq \ell\Bigg(- \frac{\left(\gamma^2(t+1)\right)^2 - C_\ell}{4\gamma^2(t+1)}\eta\Bigg), \quad \text{where} \quad \overline{\boldsymbol{w}}_t := \frac{1}{t+1}\sum_{k=0}^{t} \boldsymbol{w}_k.$$

In particular, after C_ℓ/γ^2 *burn-in steps, for every* $\eta > 0$*, we have*

$$\mathcal{L}(\overline{\boldsymbol{w}}_t) \leq \ell\left(-\frac{\gamma^2\eta}{4}\right), \quad t \geq \frac{C_\ell}{\gamma^2}.$$

The proof of Theorem 4.4.3 is deferred to Section 4.10. We conclude this section by providing several examples of loss functions that satisfy Assumption 4.4.2. The proof is also deferred to Section 4.10.

Example 4.4.4. The following loss functions satisfy Assumption 4.4.2.

1. The logistic loss $\ell_{\log} : z \mapsto \ln(1+\exp(-z))$ and the exponential loss $\ell_{\exp} : z \mapsto \exp(-z)$ satisfy Assumption 4.4.2 with $C_\ell = 1$.

2. The polynomial loss of degree $k > 0$ (Ji and Telgarsky, 2021),

$$\ell_{\text{poly}}(z) := \begin{cases} \dfrac{1}{(1+z)^k} & z \geq 0, \\ -2kz + \dfrac{1}{(1-z)^k} & z \leq 0, \end{cases}$$

   satisfies Assumption 4.4.2 with $C_\ell = n^{1/k}$.

3. The semi-circle loss (Shen, 2005),

$$\ell_{\text{semi}}(z) := \frac{-z + \sqrt{z^2+4}}{2},$$

   satisfies Assumption 4.4.2 with $C_\ell = n + 1$.

4.5 Conclusion

We consider logistic regression on linearly separable data with margin γ. We show that GD with large and adaptive stepsizes achieves an arbitrarily small risk within $1/\gamma^2$ steps, although the risk evolution might not be monotonic. This is impossible if GD with adaptive stepsize induces a monotonically decreasing risk. Additionally, we establish batch and online first-order lower bounds of $\Omega(\min\{\ln n, 1/\gamma^2\})$ and $\Omega(\min\{n, 1/\gamma^2\})$, respectively, for finding a linear separator. This shows that GD with large and adaptive stepsizes matches the worst-case dependence and is minimax optimal when the sample size is unrestricted. Finally, our results extend to a broad class of loss functions and certain two-layer networks.

We conclude by discussing the following open questions.

Convergence of the last iterate. Note that Theorem 4.2.1 only controls the averaged iterate of GD with large and adaptive stepsizes. Although this implies the same guarantee for the best iterate, it is unclear whether or not a similar guarantee applies to the last iterate. This is an interesting question left for future research.

Optimal tradeoff between n and $1/\gamma$. Consider the first-order batch methods for identifying a linear separator for a separable dataset (see Definition 4.3.1). The best-known upper bound on the step complexity is given by $\min\{1/\gamma^2, \mathcal{O}(\sqrt{\ln n}/\gamma)\}$, whereas the two branches are achieved by GD with large and adaptive stepsizes (Theorem 4.2.1) and momentum GD with adaptive stepsizes (Ji et al., 2021; Wang et al., 2022a), respectively. In comparison, we constructed hard examples suggesting the step complexity of first-order methods is at least $\Omega\Big(\max\Big\{\min\{\ln n, 1/\gamma^2\}, \min\{n, 1/\gamma^{2/3}\}\Big\}\Big)$, whereas the two branches are given by Theorem 4.3.2 and Theorem 4.8.1, respectively. We leave it as a future direction to close this gap for all n and $1/\gamma$.

Other oracle models. In the exact online first-order setting of Definition 4.3.3, the worst-case mistake complexity is characterized up to constants by Perceptron and Theorem 4.3.4. It would be interesting to understand how this picture changes under other oracle models, such as transductive online algorithms that know the unlabeled sequence in advance (Qian et al., 2026), relaxed-benchmark or covering-based online algorithms (Montasser et al., 2025), or stronger two-sided-oracle access for linear separability (Kornowski and Shamir, 2025), where qualitatively different dependence on n, $1/\gamma$, and the ambient dimension d may be possible.

4.6 Appendix: Missing Parts in the Proof of Theorem 4.2.1 in Section 4.2

We first prove the 1-Lipschitzness of the transformed loss function $\phi(\cdot)$.

Lemma 4.6.1. *For the exponential loss and logistic loss, the transformed loss $\phi(\cdot)$ defined in (4.3) is 1-Lipschitz with respect to $\|\cdot\|$.*

Proof. **(of Lemma 4.6.1)** Recall that $\phi(\boldsymbol{w}) := -\ell^{-1}(\mathcal{L}(\boldsymbol{w}))$ and $\|\mathbf{x}_i\| \leq 1$ for $1 \leq i \leq n$. We have

$$\begin{aligned}\|\nabla\phi(\boldsymbol{w})\| &= \left\|\left(-\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w})) \cdot \frac{1}{n}\sum_{i=1}^{n}\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})y_i\mathbf{x}_i\right\| \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\left|\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})\right| \cdot \left|\left(\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))\right| := \star.\end{aligned}$$

Let $\ell_i = \ell\left(y_i\mathbf{x}_i^\top\boldsymbol{w}\right)$. We rearrange $\star$ as

$$\star = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(-\ell'\right)\left(\ell^{-1}\left(\ell_i\right)\right)}{\left(-\ell'\right)\left(\ell^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\ell_i\right)\right)} = \frac{\frac{1}{n}\sum_{i=1}^{n}h\left(\ell_i\right)}{h\left(\frac{1}{n}\sum_{i=1}^{n}\ell_i\right)},$$

where $h(\cdot)$ is defined as

$$h(z) := \left(-\ell'\right)\left(\ell^{-1}\left(z\right)\right) = \begin{cases} z & \ell = \ell_{\mathsf{exp}}, \\ 1-\exp(-z) & \ell = \ell_{\mathsf{log}}. \end{cases}$$

For both losses, $h(\cdot)$ is concave on $z > 0$. Therefore, $\star \leq 1$. □

Lemma 4.6.2. *Let $\phi(\cdot)$ be the transformed loss function defined in (4.3). Under Assumption 4.1.1, if $\phi(\cdot)$ is C_ℓ-Lipschitz, we have*

$$2\left\langle\nabla\phi(\boldsymbol{w}), \mathbf{u}_2\right\rangle + \eta\left\|\nabla\phi(\boldsymbol{w})\right\|^2 \leq 0, \quad \mathbf{u}_2 := (C_\ell\eta/(2\gamma))\mathbf{w}^*.$$

Proof. **(of Lemma 4.6.2)** Recall that $\phi(\boldsymbol{w}) := -\ell^{-1}(\mathcal{L}(\boldsymbol{w}))$ and $\mathbf{u}_2 = (C_\ell\eta/(2\gamma))\mathbf{w}^*$. We also use the fact that $\langle\mathbf{w}^*, y_i\mathbf{x}_i\rangle \geq \gamma > 0$ and $\|\mathbf{x}_i\| \leq 1$ for $1 \leq i \leq n$ from Assumption 4.1.1. Applying Lemma 4.6.1, we have

$$\begin{aligned}&2\left\langle\nabla\phi(\boldsymbol{w}), \mathbf{u}_2\right\rangle + \eta\left\|\nabla\phi(\boldsymbol{w})\right\|^2 \\ &\leq \frac{2}{n}\left(-\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))\sum_{i=1}^{n}\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})\left\langle\mathbf{u}_2, y_i\mathbf{x}_i\right\rangle + C_\ell\eta\left\|\frac{\left(-\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))}{n}\sum_{i=1}^{n}\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})y_i\mathbf{x}_i\right\| \\ &\qquad\qquad (\phi(\cdot) \text{ is } C_\ell\text{-Lipschitz}) \\ &\leq -\frac{2\gamma\left\|\mathbf{u}_2\right\|}{n}\left|\left(\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))\right|\sum_{i=1}^{n}\left|\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})\right| + \frac{C_\ell\eta}{n}\left(\left|\left(\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))\right|\sum_{i=1}^{n}\left|\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})\right|\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left|\ell'(y_i\mathbf{x}_i^\top\boldsymbol{w})\right| \cdot \left|\left(\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w}))\right| \cdot \left[-2\gamma\left\|\mathbf{u}_2\right\| + C_\ell\eta\right].\end{aligned}$$

Invoking the definition of $\mathbf{u}_2$ completes the proof. □

The following lemma is a restatement of Lemma 5.2 in (Ji and Telgarsky, 2021).

Lemma 4.6.3 (Lemma 5.2 in (Ji and Telgarsky, 2021))**.** *Let $\ell(\cdot)$ be twice continuously differentiable, positive, decreasing, and convex. If $\frac{\ell'(t)^2}{\ell(t)\cdot\ell''(t)}$ is decreasing on $\mathbb{R}$, then $\psi(\mathbf{z}) := -\ell^{-1}(\frac{1}{n}\sum_{i=1}^n \ell(z_i))$ is convex. Moreover, $\phi(\cdot)$ is also convex since $\phi(\boldsymbol{w})$ is a composition between a linear mapping and $\psi(\cdot)$.*

4.7 Appendix: Proof of Theorem 4.2.2

Before proving Theorem 4.2.2, let us first delve into a technical lemma. This lemma shows that for some special datasets, the stepsize must be small if the loss is monotonically decreasing.

Lemma 4.7.1 (A variant of Lemma 14 in (Wu et al., 2024a))**.** *Let $\boldsymbol{w}_0 = \mathbf{0}, y_i = 1$ for $i = 1, 2, ..., n$, and $\bar{\mathbf{x}} := (1/n)\cdot\sum_{i=1}^n \mathbf{x}_i$. Assume there exist constants $r > 0$ and $q \in (0,1)$ such that*

$$\frac{\left|\left\{i \in [n] : \mathbf{x}_i^\top \bar{\mathbf{x}} < -r\right\}\right|}{n} \geq q. \tag{4.11}$$

Suppose we run gradient descent according to (GD). If $\mathcal{L}(\boldsymbol{w}_1) \leq \mathcal{L}(\boldsymbol{w}_0)$, then $\eta \leq \ell(0)/(qr)$.

Proof. **(of Lemma 4.7.1)** Starting from $\boldsymbol{w}_0 = \mathbf{0}$, we have $\mathcal{L}(\boldsymbol{w}_0) = \ell(0)$, $\nabla\mathcal{L}(\boldsymbol{w}_0) = \ell'(0)\cdot\bar{\mathbf{x}}$, $\nabla\phi(\boldsymbol{w}_0) = (-\ell^{-1})'(\ell(0))\cdot\ell'(0)\cdot\bar{\mathbf{x}}$. For both the exponential loss and logistic loss, one can verify that $-(-\ell^{-1})'(\ell(0))\cdot\ell'(0) = 1$, which implies $\boldsymbol{w}_1 = \eta\bar{\mathbf{x}}$. Suppose $\eta > qr/\ell(0)$. Then,

$$\mathcal{L}(\boldsymbol{w}_1) \geq \frac{1}{n}\sum_{i=1}^n \ell(\eta\mathbf{x}_i^\top\bar{\mathbf{x}})\mathbb{I}\left(\mathbf{x}_i^\top\bar{\mathbf{x}} < -r\right) \geq \ell(-\eta r)\cdot\frac{1}{n}\sum_{i=1}^n \mathbb{I}\left(\mathbf{x}_i^\top\bar{\mathbf{x}} < -r\right) \geq \ell(-\eta r)\cdot q$$

(From (4.11))

$$\geq q\cdot\ln\left(1+\exp(\eta r)\right) \geq \eta q r \geq \ell(0) = \mathcal{L}(\boldsymbol{w}_0),$$

a contradiction to $\mathcal{L}(\boldsymbol{w}_1) \leq \mathcal{L}(\boldsymbol{w}_0)$. Thus $\eta \leq \ell(0)/(qr)$. □

Now we prove the Theorem 4.2.2.

Proof. **(of Theorem 4.2.2)** Consider the specific dataset from Theorem 4.2.2, which satisfies the conditions of Lemma 4.7.1 with $r = 0.1$ and $q = 0.5$. So Lemma 4.7.1 implies $\eta \leq c_1$, where c_1 is a constant that depends on the loss (but *not* on t). Next, from the GD iterate on the transformed loss Equation (4.3) and the 1-Lipschitzness of $\phi(\cdot)$ (Lemma 4.6.1), we have

$$\|\boldsymbol{w}_t\| = \left\|\eta\sum_{k=0}^{t-1}\nabla\phi(\boldsymbol{w}_k)\right\| \leq \eta\cdot\sum_{k=0}^{t-1}\|\nabla\phi(\boldsymbol{w}_k)\| \leq \eta t.$$

Since $\|\mathbf{x}_i\| \leq 1$, it follows

$$\mathcal{L}(\boldsymbol{w}_t) = \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{w}_t^\top\mathbf{x}_i) \geq \ell\left(c_1\cdot t\right) \simeq \exp\left(-c_1\cdot t\right)$$

The same argument applies to $\mathcal{L}(\overline{\boldsymbol{w}}_t)$. This completes the proof. □

4.8 Appendix: Missing Proofs for Theorems in Section 4.3

Proof of Theorem 4.3.2

Proof. **(of Theorem 4.3.2)** For $0 < \gamma < 1/6$ and $n > 16$, we define $d := \lfloor 1/5\gamma^2 \rfloor \geq 6$, where $\lfloor \cdot \rfloor$ is the floor function. Let $(\mathbf{e}_i)_{i=1}^d$ be a set of standard basis vectors for $\mathbb{R}^d$. Note that all the first-order methods defined in Definition 4.3.1 are rotation invariant. Moreover, the criterion for all data classified correctly is also rotation invariant. Therefore, without loss of generality, we can assume $\boldsymbol{w}_0$ is proportional to $\mathbf{e}_1$, that is, $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$.

Let $k := \min\{\lfloor \log_2 n \rfloor, d-2\} \geq 4$. We construct a hard dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ as follows. Let $y_i = 1$ for all $i \in [n]$. For $j = 1, \ldots, k$, let

$$\mathbf{x}_i := \frac{2}{\sqrt{5}}\mathbf{e}_{j+1} - \frac{1}{\sqrt{5}}\mathbf{e}_{j+2} \quad \text{for} \quad 2^k - 2^{k-j+1} + 1 \leq i \leq 2^k - 2^{k-j}.$$

Note that $j \leq k \leq d-2$, thus such $\mathbf{x}_i$'s are well defined. Let the remaining $\mathbf{x}_i$'s be

$$\mathbf{x}_i := \frac{1}{\sqrt{5}}\mathbf{e}_{k+2} \quad \text{for} \quad 2^k \leq i \leq n.$$

Note that $\|\mathbf{x}_i\| \leq 1$ for $i = 1, \ldots, n$. Moreover, for the unit vector

$$\mathbf{w}^* := \frac{1}{\sqrt{d}}(1, 1, \ldots, 1)^\top,$$

we have

$$y_i \mathbf{x}_i^\top \mathbf{w}^* = \frac{1}{\sqrt{5d}} \geq \gamma, \quad \text{for } i = 1 \ldots, n.$$

Thus $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfies Assumption 4.1.1.

For a vector $\boldsymbol{w} \in \mathbb{R}^d$, we also write it as $\boldsymbol{w} := (w^{(1)}, w^{(2)}, \ldots, w^{(d)})^\top$. Then, the objective function can be written as

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}) &= \frac{1}{n}\sum_{i=1}^n \ell\Big(\boldsymbol{w}^\top \mathbf{x}_i\Big) \\
&= \frac{1}{n}\left[\sum_{j=1}^k 2^{k-j}\ell\left(\frac{2}{\sqrt{5}}w^{(j+1)} - \frac{1}{\sqrt{5}}w^{(j+2)}\right) + \Big(n - 2^k + 1\Big)\ell\left(\frac{1}{\sqrt{5}}w^{(k+2)}\right)\right].
\end{aligned}$$

Thus the j-th coordinate of $\nabla\mathcal{L}(\boldsymbol{w})$ is given by

$$\big[\nabla\mathcal{L}(\boldsymbol{w})\big]_j = \begin{cases} 0 & j = 1, \\ \dfrac{2^k}{\sqrt{5}n}\ell'\left(\dfrac{2w^{(2)}}{\sqrt{5}} - \dfrac{w^{(3)}}{\sqrt{5}}\right) & j = 2, \\ \dfrac{2^{k-j+2}}{\sqrt{5}n}\left[\ell'\left(\dfrac{2w^{(j)}}{\sqrt{5}} - \dfrac{w^{(j+1)}}{\sqrt{5}}\right) - \ell'\left(\dfrac{2w^{(j-1)}}{\sqrt{5}} - \dfrac{w^{(j)}}{\sqrt{5}}\right)\right] & 3 \le j \le k+1, \\ \dfrac{1}{\sqrt{5}n}\left[\big(n - 2^k + 1\big)\ell'\left(\dfrac{w^{(k+2)}}{\sqrt{5}}\right) - \ell'\left(\dfrac{2w^{(k+1)}}{\sqrt{5}} - \dfrac{w^{(k+2)}}{\sqrt{5}}\right)\right] & j = k+2, \\ 0 & \text{otherwise.} \end{cases}$$

Consider a sequence $(\boldsymbol{w}_s)_{s=0}^t$ generated by a first-order method according to Definition 4.3.1. Since $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$, the gradient at $\boldsymbol{w}_0$ vanishes in all coordinates except the second and the $(k+2)$-th coordinates. Therefore we have

$$\boldsymbol{w}_1 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_{k+2}\}.$$

Then the gradient at $\boldsymbol{w}_1$ vanishes in all coordinates except the second, third, $(k+1)$-th, and $(k+2)$-th coordinates. Therefore we have

$$\boldsymbol{w}_2 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{k+1}, \mathbf{e}_{k+2}\}.$$

By induction, we conclude that for $t \le t_0 - 2$ for $t_0 := \lfloor (k+1)/2 \rfloor$, it holds that

$$\boldsymbol{w}_t \in \mathsf{Lin}\{\mathbf{e}_1, \ldots, \mathbf{e}_{t+1}, \mathbf{e}_{k+3-t}, \ldots, \mathbf{e}_{k+2}\}.$$

So for all $(\boldsymbol{w}_s)_{s=0}^t$, their t_0-th and (t_0+1)-th coordinates must be zero. By our dataset construction, there exists $i \le 2^k - 1$ such that

$$y_i \mathbf{x}_i^\top \boldsymbol{w}_k = 0, \quad \text{for all } k = 0, \ldots, t.$$

This means that the dataset cannot be separated by any of $(\boldsymbol{w}_k)_{k=0}^t$. Thus, for the first-order method to output a linear separator, we must have

$$\begin{aligned} t \ge t_0 - 1 = \left\lfloor \frac{k-1}{2} \right\rfloor \ge \frac{k}{2} - 1 \ge \min\left\{\frac{\log_2 n - 3}{2}, \frac{d}{2} - 2\right\} \ge \min\left\{\frac{\log_2 n}{8}, \frac{d}{3}\right\} \\ \ge \min\left\{\frac{\log_2 n}{8}, \frac{1}{15\gamma^2} - \frac{1}{3}\right\} \ge \min\left\{\frac{\ln n}{8\ln 2}, \frac{1}{30\gamma^2}\right\}. \end{aligned}$$

This completes the proof. □

Proof of Theorem 4.3.4

Proof. **(of Theorem 4.3.4)** For $0 < \gamma < 1/2$ and $n \geq 2$, we define $d := \lfloor 1/\gamma^2 \rfloor \geq 4$, where $\lfloor \cdot \rfloor$ is the floor function. Let $(\mathbf{e}_i)_{i=1}^d$ be a set of standard basis vectors for $\mathbb{R}^d$. Note that all the first-order online methods defined in Definition 4.3.3 are rotation invariant. Moreover, the criterion for all data classified correctly is also rotation invariant. Therefore, without loss of generality, we can assume $\boldsymbol{w}_0$ is proportional to $\mathbf{e}_1$, that is, $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$.

Let $k := \min\{n, d-1\} \geq 2$. We construct a hard dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ as follows. Let $y_i = 1$ for all $i \in [n]$. For i = 1,2,...,k, let

$$\mathbf{x}_i = \mathbf{e}_{i+1}.$$

Note that $k \leq d-1$, thus such $\mathbf{x}_i$'s are well defined. Let the remaining $\mathbf{x}_i$'s be

$$\mathbf{x}_i = \mathbf{e}_{k+1} \quad \text{for} \quad k+1 \leq i \leq n \quad \text{if} \quad n \geq k+1.$$

Moreover, for the unit vector

$$\mathbf{w}^* := \frac{1}{\sqrt{d}} (1, 1, \ldots, 1)^\top,$$

we have

$$y_i \mathbf{x}_i^\top \mathbf{w}^* = \frac{1}{\sqrt{d}} \geq \gamma, \quad \text{for } i = 1 \ldots, n.$$

Thus $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfies Assumption 4.1.1.

Consider a sequence $(\boldsymbol{w}_s)_{s=0}^t$ generated by a first-order online method with initialization $\boldsymbol{w}_0$ and dataset $\left(\mathbf{x}_{\pi(s)}, y_{\pi(s)}\right)_{s=1}^t$, where $\pi(s) \in [n]$ but is otherwise arbitrary. Since $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$, the gradient at $\boldsymbol{w}_0$ vanishes in all coordinates except the direction of $\ell'(y_{\pi(1)} \mathbf{x}_{\pi(1)}^\top \boldsymbol{w}_0) y_{\pi(1)} \mathbf{x}_{\pi(1)}$. Therefore we have

$$\boldsymbol{w}_1 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{x}_{\pi(1)}\}.$$

Then the gradient at $\boldsymbol{w}_1$ vanishes in all coordinates except the span of

$$\ell'(y_{\pi(1)} \mathbf{x}_{\pi(1)}^\top \boldsymbol{w}_0) y_{\pi(1)} \mathbf{x}_{\pi(1)} \quad \text{and} \quad \ell'(y_{\pi(2)} \mathbf{x}_{\pi(2)}^\top \boldsymbol{w}_0) y_{\pi(2)} \mathbf{x}_{\pi(2)}.$$

Therefore we have

$$\boldsymbol{w}_2 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{x}_{\pi(1)}, \mathbf{x}_{\pi(2)}\}.$$

By induction, we conclude that for all $t \leq k-1$, it holds that

$$\boldsymbol{w}_t \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(t)}\}.$$

From our dataset construction, there exists $j \in [d]$, such that for all $(\boldsymbol{w}_s)_{s=0}^t$, their j-th coordinates are zero. Therefore,

$$y_j \mathbf{x}_j^\top \boldsymbol{w}_s = 0, \quad \text{for all } s = 0, \ldots, t.$$

This means that the dataset cannot be separated by any of $(\boldsymbol{w}_s)_{s=0}^t$. Thus, for the first-order online method to output a linear separator, we must have

$$t \geq k = \min\{n, d-1\} \geq \min\left\{\frac{1}{2\gamma^2}, n\right\}.$$

This proves the first claim.

For the second claim, consider a sequence $(\boldsymbol{w}_s)_{s=0}^t$ generated by a first-order online method with initialization $\boldsymbol{w}_0$ and dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ as defined previously. By the same induction, we conclude that for all $0 \leq t \leq k-1$, it holds that

$$\boldsymbol{w}_t \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{t+1}\}.$$

Since $\mathbf{x}_{t+1} = \mathbf{e}_{t+2}$ for $0 \leq t \leq k-1$, this implies

$$y_{t+1}\mathbf{x}_{t+1}^\top \boldsymbol{w}_t = 0,$$

which suggests that the algorithm makes a mistake step. Therefore, the total mistake steps up to the t-th step is

$$\sum_{j=0}^{t-1} \mathbb{1}\{y_{j+1}\mathbf{x}_{j+1}^\top \boldsymbol{w}_j \leq 0\} \geq \min\{t, k\} = \min\{t, d-1\} \geq \min\left\{\frac{1}{2\gamma^2}, t\right\}.$$

This completes the proof. □

An alternative lower bound for first-order batch algorithms

Now we present another lower bound for first-order batch algorithms with a different trade-off than Theorem 4.3.2. We leave it as an open problem to figure out the optimal trade-off between the sample size n and margin γ in finding linear separators.

Theorem 4.8.1 (An alternate lower bound for batch first-order methods)**.** *For every $0 < \gamma < 1/8$, $n \geq 4$, and $\boldsymbol{w}_0$, there exists a dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfying Assumption 4.1.1 such that the following holds. For any $\boldsymbol{w}_t$ output by a first-order batch method in t-steps with initialization $\boldsymbol{w}_0$ on this dataset, we have*

$$\min_{i\in[n]} y_i\mathbf{x}_i^\top \boldsymbol{w}_t > 0 \quad \textit{implies that} \quad t \geq \min\left\{\frac{n}{4}, \frac{1}{8\gamma^{2/3}}\right\}.$$

Proof. **(of Theorem 4.8.1)** For $0 < \gamma < 1/8$ and $n \geq 2$, we define $d := \lfloor \gamma^{-2/3} \rfloor \geq 4$, where $\lfloor \cdot \rfloor$ is the floor function. Let $(\mathbf{e}_i)_{i=1}^d$ be a set of standard basis vectors for $\mathbb{R}^d$. Note that all the first-order methods defined in Definition 4.3.1 are rotation invariant. Moreover, the criterion for all data classified correctly is also rotation invariant. Therefore, without loss of generality, we can assume $\boldsymbol{w}_0$ is proportional to $\mathbf{e}_1$, that is, $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$.

Let $k := \min\{n, d-2\}$. We construct a hard dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ as follows. Let $y_i = 1$ for all $i \in [n]$. For $j = 1, \dots, k$, let

$$\mathbf{x}_i := -\frac{1}{\sqrt{2}}\mathbf{e}_{j+1} + \frac{1}{\sqrt{2}}\mathbf{e}_{j+2}.$$

Let the remaining $\mathbf{x}_i$'s be

$$\mathbf{x}_i := \frac{1}{\sqrt{2}}\mathbf{e}_{k+2} \quad \text{for} \quad k+1 \le i \le n \quad \text{if} \quad n \ge k+1.$$

Note that $j \le k \le d-2$, thus such $\mathbf{x}_i$'s are well defined. Also note that $\|\mathbf{x}_i\| \le 1$ for $i = 1, \dots, n$. Moreover, for the unit vector

$$\mathbf{w}^* := \sqrt{\frac{6}{d(d+1)(2d+1)}}\,(1, 2, \dots, d)^\top,$$

we have

$$y_i \mathbf{x}_i^\top \mathbf{w}^* = \sqrt{\frac{3}{d(d+1)(2d+1)}} \ge \sqrt{\frac{1}{d^3}} \ge \gamma, \quad \text{for} \quad i = 1 \dots, n.$$

Thus $(\mathbf{x}_i, y_i)_{i=1}^n$ satisfies Assumption 4.1.1.

For a vector $\boldsymbol{w} \in \mathbb{R}^d$, we also write it as $\boldsymbol{w} := (w^{(1)}, w^{(2)}, \dots, w^{(d)})^\top$. Then, the objective function can be written as

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \ell\Big(\boldsymbol{w}^\top \mathbf{x}_i\Big) = \frac{1}{n}\left[\sum_{j=1}^k \ell\bigg(-\frac{1}{\sqrt{2}}w^{(j+1)} + \frac{1}{\sqrt{2}}w^{(j+2)}\bigg) + \Big(n-k\Big)\ell\bigg(\frac{1}{\sqrt{2}}w^{(k+2)}\bigg)\right].$$

Thus the j-th coordinate of $\nabla\mathcal{L}(\boldsymbol{w})$ is given by

$$\Big[\nabla\mathcal{L}(\boldsymbol{w})\Big]_j = \begin{cases} 0 & j = 1, \\ -\dfrac{1}{\sqrt{2}n}\ell'\bigg(-\dfrac{w^{(2)}}{\sqrt{2}} + \dfrac{w^{(3)}}{\sqrt{2}}\bigg) & j = 2, \\ \dfrac{1}{\sqrt{2}n}\left[\ell'\bigg(-\dfrac{w^{(j-1)}}{\sqrt{2}} + \dfrac{w^{(j)}}{\sqrt{2}}\bigg) - \ell'\bigg(-\dfrac{w^{(j)}}{\sqrt{2}} + \dfrac{w^{(j+1)}}{\sqrt{2}}\bigg)\right] & 3 \le j \le k+1, \\ \dfrac{1}{\sqrt{2}n}\left[\Big(n-k\Big)\ell'\bigg(\dfrac{w^{(k+2)}}{\sqrt{2}}\bigg) - \ell'\bigg(-\dfrac{w^{(k+1)}}{\sqrt{2}} + \dfrac{w^{(k+2)}}{\sqrt{2}}\bigg)\right] & j = k+2, \\ 0 & \text{otherwise.} \end{cases}$$

Consider a sequence $(\boldsymbol{w}_s)_{s=0}^t$ generated by a first-order method according to Definition 4.3.1. Since $\boldsymbol{w}_0 \in \mathsf{Lin}\{\mathbf{e}_1\}$, the gradient at $\boldsymbol{w}_0$ vanishes in all coordinates except the second and the $(k+2)$-th coordinates. Therefore we have

$$\boldsymbol{w}_1 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_{k+2}\}.$$

Then the gradient at $\boldsymbol{w}_1$ vanishes in all coordinates except the second, third, $(k+1)$-th, and $(k+2)$-th coordinates. Therefore we have

$$\boldsymbol{w}_2 \in \mathsf{Lin}\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{k-1}, \mathbf{e}_{k+2}\}.$$

By induction, we conclude that for $t \le t_0 - 2$ with $t_0 := \lfloor (k+1)/2 \rfloor$, it holds that

$$\boldsymbol{w}_t \in \mathsf{Lin}\{\mathbf{e}_1, \ldots, \mathbf{e}_{t+1}, \mathbf{e}_{k+3-t}, \ldots, \mathbf{e}_{k+2}\}.$$

So for all $(\boldsymbol{w}_s)_{s=0}^t$, their t_0-th and (t_0+1)-th coordinates must be zero. By our dataset construction, there exists $i \in [k]$ such that

$$y_i \mathbf{x}_i^\top \boldsymbol{w}_k = 0, \quad \text{for all } k = 0, \ldots, t.$$

This means that the dataset cannot be separated by any of $(\boldsymbol{w}_k)_{k=0}^t$. Thus, for the first-order method to output a linear separator, we must have

$$\begin{aligned} t \ge t_0 - 1 = \left\lfloor \frac{k-1}{2} \right\rfloor \ge \frac{k}{2} - 1 \ge \min\left\{ \frac{n}{2} - 1, \frac{d}{2} - 2 \right\} \ge \min\left\{ \frac{n}{4}, \frac{d}{4} \right\} \\ \ge \min\left\{ \frac{n}{4}, \frac{1}{4}\left(\gamma^{-2/3} - 1\right) \right\} \ge \min\left\{ \frac{n}{4}, \frac{1}{8\gamma^{2/3}} \right\} \end{aligned}$$

This completes the proof. □

4.9 Appendix: A Generalization of Theorem 4.4.1 and Its Proof

In this section, we generalize Theorem 4.4.1 to more general activation functions and present its proof. First, we provide general assumptions on the activation functions.

Assumption 4.9.1 (Activation function)**.** In two-layer neural networks (4.9), let the activation function $\sigma(z) : \mathbb{R} \to \mathbb{R}$ be a locally Lipschitz function. For each z, let $\sigma'(z)$ be a unique element from the Clarke subdifferential of $\sigma(\cdot)$ at z (Clarke, 1990). Moreover, we assume

- There exists $0 < \alpha < 1$ such that $\sigma'(z) \in [\alpha, 1]$.
- For any $z \in \mathbb{R}$, it holds that $|\sigma(z) - \sigma'(z)z| \le \kappa$ for some $\kappa > 0$.

Examples of activation functions. Assumption 4.9.1 holds for a broad class of leaky activations. For instance, let $\sigma_*(\cdot)$ be one of the following: GeLU $\sigma_{\mathsf{GeLU}}(x) := x \cdot F(x)$, where $F(x)$ is the cumulative density function of the standard Gaussian distribution; Softplus $\sigma_{\mathsf{Softplus}}(x) := \ln(1+\exp(x))$; SiLU $\sigma_{\mathsf{SiLU}}(x) := x/(1+\exp(-x))$; ReLU $\sigma_{\mathsf{ReLU}}(x) := \max\{x, 0\}$. Then, we fix some constant $0.5 < c < 1$ and define $\sigma(x) = c \cdot x + \frac{1-c}{4} \cdot \sigma_*(x)$, where σ_* can

be any activation function above. It is straightforward to check that such a "leaky" variant satisfies Assumption 4.9.1 (see Example 2.1 in Cai et al., 2024). Moreover, the leaky ReLU activation defined as $\sigma(z) = \max\{z, \alpha z\}$ satisfies Assumption 4.9.1 with α and $\kappa = 0$.

Now, we present the improved convergence rate for two-layer neural networks. Setting $\kappa = 0$ for leaky ReLU function recovers Theorem 4.4.1.

Theorem 4.9.2 (General two-layer neural networks)**.** *Consider Equation* (GD) *on Equation* (4.10) *with adaptive stepsizes Equation* (4.2) *for two-layer neural networks Equation* (4.9) *under Assumption 4.1.1 and Assumption 4.9.1. Assume without loss of generality that* $\boldsymbol{w}_0 = \mathbf{0}$. *Then for* every $t \geq 1$ *and* $\eta > 0$, *we have*

$$\min_{k \leq t} \mathcal{L}(\boldsymbol{w}_k) \leq \exp\left(\kappa - \frac{(\alpha\gamma^2(t+1))^2 - 1}{4\gamma^2(t+1)}\eta\right).$$

In particular, after $1/(\alpha\gamma^2)$ *burn-in steps, for every* $\eta > 0$, *we have*

$$\min_{k \leq t} \mathcal{L}(\boldsymbol{w}_k) \leq \exp\left(\kappa - \frac{\alpha^2\gamma^2\eta}{4}\right) = \exp\left(-\Theta\left(\eta\right)\right), \quad t \geq \frac{1}{\alpha\gamma^2}. \tag{4.12}$$

Before we prove Theorem 4.9.2, we first state and prove two lemmas.

Lemma 4.9.3. *Under Assumption 4.1.1, for every* $\boldsymbol{w} \in \mathbb{R}^{md}$, *it holds that*

$$I_2(\boldsymbol{w}) := 2\left\langle m \cdot \nabla\phi(\boldsymbol{w}), \mathbf{u}_2\right\rangle + \eta\left\|m \cdot \nabla\phi(\boldsymbol{w})\right\|^2 \leq 0 \quad \textit{for} \quad \mathbf{u}_2^{(j)} := \frac{a_j\eta}{2\gamma}\mathbf{w}^*, \ j = 1, 2, ...m.$$

Proof. **(of Lemma 4.9.3)** Define $g_{i,j} := \ell'(y_i f(\boldsymbol{w}, \mathbf{x}_i)) \cdot \sigma'(\mathbf{x}_i^\top \boldsymbol{w}^{(j)}) \leq 0$ for $i \in [n]$ and $j \in [m]$. We know $|g_{i,j}| \leq |\ell'(y_i f(\boldsymbol{w}, \mathbf{x}_i))|$ since $\sigma'(\cdot) \leq 1$. Then, for $j \in [m]$, we have

$$\begin{aligned}\left\|m \cdot \nabla_{\boldsymbol{w}^{(j)}}\phi(\boldsymbol{w})\right\| &= \left\|\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right) \cdot \frac{1}{n}\sum_{i=1}^n a_j g_{i,j} y_i \mathbf{x}_i\right\| \\ &\leq \left|\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\right| \cdot \frac{1}{n}\sum_{i=1}^n |\ell'(y_i f(\boldsymbol{w}, \mathbf{x}_i))|.\end{aligned}$$

Let $\ell_i := \ell(y_i f(\boldsymbol{w}, \mathbf{x}_i))$. We arrange the term above as

$$\left|\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\right| \cdot \frac{1}{n}\sum_{i=1}^n |\ell'(y_i f(\boldsymbol{w}, \mathbf{x}_i))| = \frac{\frac{1}{n}\sum_{i=1}^n \left(-\ell'\right)\left(\ell^{-1}\left(\ell_i\right)\right)}{\left(-\ell'\right)\left(\ell^{-1}\left(\frac{1}{n}\sum_{i=1}^n \ell_i\right)\right)} = \frac{\frac{1}{n}\sum_{i=1}^n h\left(\ell_i\right)}{h\left(\frac{1}{n}\sum_{i=1}^n \ell_i\right)},$$

where $h(\cdot)$ is defined as $h(z) := \left(-\ell'\right)\left(\ell^{-1}\left(z\right)\right)$. For both losses, $h(\cdot)$ is concave on $z > 0$ (see Lemma 4.6.1 in Section 4.6). Therefore, we have $\left\|m\nabla_{\boldsymbol{w}^{(j)}}\phi(\boldsymbol{w})\right\|^2 \leq 1$ for $j \in [m]$. Apply

this upper bound of $\|\nabla\phi(\boldsymbol{w})\|$, and recall Assumption 4.1.1, we have

$$\begin{aligned}
I_2(\boldsymbol{w}) &= \frac{2}{n}\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\sum_{j=1}^{m}\sum_{i=1}^{n} g_{i,j}\cdot\left\|\mathbf{u}_2^{(j)}\right\|\cdot y_i\mathbf{x}_i^\top\mathbf{w}^* + \eta\cdot\sum_{j=1}^{m}\left\|m\nabla_{\boldsymbol{w}^{(j)}}\phi\left(\boldsymbol{w}\right)\right\|^2 \\
&\leq \frac{2}{n}\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\sum_{j=1}^{m}\sum_{i=1}^{n} g_{i,j}\cdot\left\|\mathbf{u}_2^{(j)}\right\|\cdot y_i\mathbf{x}_i^\top\mathbf{w}^* + \eta\cdot\sum_{j=1}^{m}\left\|m\nabla_{\boldsymbol{w}^{(j)}}\phi\left(\boldsymbol{w}\right)\right\| \\
&\leq \frac{-2\gamma}{n}\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\sum_{j=1}^{m}\sum_{i=1}^{n}|g_{i,j}|\cdot\left\|\mathbf{u}_2^{(j)}\right\| + \eta\left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n}|g_{i,j}| \\
&= \left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n}|g_{i,j}|\cdot\left(-2\gamma\left\|\mathbf{u}_2^{(j)}\right\| + \eta\right).
\end{aligned}$$

Invoking the definition of $\mathbf{u}_2$, we complete the proof. □

Lemma 4.9.4. *Under Assumption 4.1.1, if* $\mathbf{u}_1^{(j)} \propto a_j\cdot\mathbf{w}^*$ *for* $j = 1, 2, ..., m$, *then for any* $\boldsymbol{w}\in\mathbb{R}^{md}$,

$$I_1(\boldsymbol{w}) := \langle\nabla\phi(\boldsymbol{w}), \mathbf{u}_1 - \boldsymbol{w}\rangle \leq \kappa - \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \phi\left(\boldsymbol{w}\right).$$

Proof. **(of Lemma 4.9.4)** From the definition of the transformed loss function $\phi(\cdot)$, we have

$$\begin{aligned}
I_1(\boldsymbol{w}) &= \left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\frac{1}{n}\sum_{i=1}^{n}\ell'\left(y_i f\left(\boldsymbol{w};\mathbf{x}_i\right)\right)\cdot\frac{1}{m}\sum_{j=1}^{m} y_i a_j\sigma'\left(\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right)\cdot\mathbf{x}_i^\top\left(\mathbf{u}_1^{(j)} - \boldsymbol{w}^{(j)}\right) \\
&= \left(-\ell^{-1}\right)'\left(\mathcal{L}(\boldsymbol{w})\right)\cdot\frac{1}{n}\sum_{i=1}^{n}\ell'\left(y_i f\left(\boldsymbol{w};\mathbf{x}_i\right)\right)\cdot\left[J_i - y_i f\left(\boldsymbol{w},\mathbf{x}_i\right)\right],
\end{aligned}$$

where

$$\begin{aligned}
J_i &:= \frac{1}{m}\sum_{j=1}^{m} a_j\left[\sigma'\left(\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right) y_i\mathbf{x}_i^\top\mathbf{u}_1^{(j)}\right] + \frac{1}{m}\sum_{j=1}^{m} y_i a_j\underbrace{\left[\sigma\left(\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right) - \sigma'\left(\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right)\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right]}_{|\cdot|\leq\kappa} \\
&\geq \frac{1}{m}\sum_{j=1}^{m} a_j\left[\sigma'\left(\mathbf{x}_i^\top\boldsymbol{w}^{(j)}\right) y_i\mathbf{x}_i^\top\mathbf{u}_1^{(j)}\right] - \kappa \geq \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \kappa,
\end{aligned}$$

where the last inequality utilizes $\sigma'(\cdot)\geq\alpha$, $\mathbf{u}_1^{(j)}\propto a_j\mathbf{w}^*$, and Assumption 4.1.1. Notice $\ell'(\cdot)\leq 0$, we define $\mathbf{z} := \left(y_1 f\left(\boldsymbol{w};\mathbf{x}_1\right), y_2 f\left(\boldsymbol{w},\mathbf{x}_2\right), ..., y_n f\left(\boldsymbol{w},\mathbf{x}_n\right)\right)^\top$ and $\mathbf{1}_n = (1, 1, ..., 1)^\top\in\mathbb{R}^n$. We also define $\psi(\mathbf{z}) = (-\ell^{-1})(1/n\cdot\sum_{i=1}^{n}\ell(\mathbf{z}_i))$. From the definition, we know $\phi(\boldsymbol{w}) = \psi(\mathbf{z})$ and $\psi(\cdot)$ is convex for both exponential and logistic loss (Theorem 5.1 in (Ji and Telgarsky, 2021),

also see Lemma 4.6.3). Therefore,

$$\begin{aligned} I_1(\boldsymbol{w}) &\leq \left(-\ell^{-1}\right)'(\mathcal{L}(\boldsymbol{w})) \cdot \frac{1}{n}\sum_{i=1}^{n} \ell'\left(y_i f\left(\boldsymbol{w}; \mathbf{x}_i\right)\right) \cdot \left(\frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \kappa - y_i f\left(\boldsymbol{w}, \mathbf{x}_i\right)\right) \\ &= \left\langle \nabla\psi\left(\mathbf{z}\right), \left(\frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \kappa\right)\cdot \mathbf{1}_n - \mathbf{z}\right\rangle \\ &\leq \psi\left(\left(\frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \kappa\right)\cdot \mathbf{1}_n\right) - \psi\left(\mathbf{z}\right) \\ &= \kappa - \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \phi\left(\boldsymbol{w}\right). \end{aligned}$$

This completes the proof. □

Now we present the proof of Theorem 4.9.2

Proof. **(of Theorem 4.9.2)** As explained in Equation (4.3), it is equivalent to considering GD with a constant stepsize under a transformed objective $\phi(\cdot)$. We then use the split optimization technique developed by Wu et al. (2024a) and Cai et al. (2024). Specifically, consider a comparator $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2 \in \mathbb{R}^{dm}$:

$$\mathbf{u}_1 = \begin{pmatrix} \mathbf{u}_1^{(1)} \\ \mathbf{u}_1^{(2)} \\ \cdots \\ \mathbf{u}_1^{(m)} \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} \mathbf{u}_2^{(1)} \\ \mathbf{u}_2^{(2)} \\ \cdots \\ \mathbf{u}_2^{(m)} \end{pmatrix}.$$

From GD iterate in Equation (4.3), we have

$$\begin{aligned} &\left\|\boldsymbol{w}_{t+1} - \mathbf{u}\right\|^2 \\ =\ &\left\|\boldsymbol{w}_t - \mathbf{u}\right\|^2 + 2\eta m\left\langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_1 - \boldsymbol{w}_t\right\rangle + \eta\left(2\left\langle m\cdot\nabla\phi(\boldsymbol{w}_t), \mathbf{u}_2\right\rangle + \eta\left\|m\cdot\nabla\phi(\boldsymbol{w}_t)\right\|^2\right) \\ \leq\ &\left\|\boldsymbol{w}_t - \mathbf{u}\right\|^2 + 2\eta m\left(\kappa - \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| - \phi\left(\boldsymbol{w}_t\right)\right), \end{aligned} \tag{4.13}$$

where Equation (4.13) is by the following two lemmas. Rearranging (4.13) and telescoping the sum to obtain

$$\frac{\left\|\boldsymbol{w}_t - \mathbf{u}\right\|^2}{2\eta m(t+1)} + \frac{1}{t+1}\sum_{k=0}^{t}\phi(\boldsymbol{w}_k) \leq \kappa - \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| + \frac{\left\|\mathbf{u}\right\|^2}{2\eta m(t+1)}. \tag{4.14}$$

Further setting $\|\mathbf{u}_1^{(j)}\| = \alpha\gamma\eta(t+1)/2$, we get

$$\frac{1}{t+1}\sum_{k=0}^{t}\phi(\boldsymbol{w}_k) \leq \kappa - \frac{\alpha\gamma}{m}\sum_{j=1}^{m}\left\|\mathbf{u}_1^{(j)}\right\| + \frac{\left\|\mathbf{u}_1 + \mathbf{u}_2\right\|^2}{2\eta m(t+1)} \leq \kappa - \frac{(\alpha\gamma^2(t+1))^2 - 1}{4\gamma^2(t+1)}\eta. \tag{4.15}$$

We complete the proof by applying the fact that $\mathcal{L}(\cdot) = \ell\left(-\phi(\cdot)\right)$. □

4.10 Appendix: Proofs for Theorem 4.4.3 and Example 4.4.4

Proof. **(of Theorem 4.4.3)** The entire proof is analogous to the proof of Theorem 4.2.1 in Section 4.2. Define $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$, where $\mathbf{u}_2 = (C_\ell \eta/(2\gamma)) \cdot \mathbf{w}^* \in \mathbb{R}^d$, where C_ℓ is the loss-dependent constant in Assumption 4.4.2. Recall the gradient descent iterate (GD), the learning rate scheduler (4.2), and the definition for $\phi(\cdot)$. Then,

$$\begin{aligned}\|\boldsymbol{w}_{t+1} - \mathbf{u}\|^2 &= \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u} - \boldsymbol{w}_t\rangle + \eta^2 \|\nabla\phi(\boldsymbol{w}_t)\|^2 \\ &= \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_1 - \boldsymbol{w}_t\rangle + \eta \left[2\langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_2\rangle + \eta \|\nabla\phi(\boldsymbol{w}_t)\|^2\right] \\ &\overset{(a)}{\leq} \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta \langle \nabla\phi(\boldsymbol{w}_t), \mathbf{u}_1 - \boldsymbol{w}_t\rangle \overset{(b)}{\leq} \|\boldsymbol{w}_t - \mathbf{u}\|^2 + 2\eta\left(\phi(\mathbf{u}_1) - \phi(\boldsymbol{w}_t)\right).\end{aligned} \tag{4.16}$$

Here, (a) is from Lemma 4.6.2 and (b) is from the convexity of $\phi(\cdot)$ following Assumption 4.4.2C. Rearranging the inequality above and telescoping the sum, we have

$$\frac{\|\boldsymbol{w}_t - \mathbf{u}\|^2}{2\eta(t+1)} + \frac{1}{t+1}\sum_{k=0}^{t}\phi(\boldsymbol{w}_k) \leq \phi(\mathbf{u}_1) + \frac{\|\mathbf{u}\|^2}{2\eta(t+1)}. \tag{4.17}$$

Recall $\langle \mathbf{w}^*, y_i\mathbf{x}_i\rangle \geq \gamma$, we have $\phi(\mathbf{u}_1) \leq -\gamma\|\mathbf{u}_1\|$. Then, setting $\mathbf{u}_1 = (\gamma\eta(t+1)/2)\cdot\mathbf{w}^*$ and invoking $\boldsymbol{w}_0 = \mathbf{0}$ gives

$$\frac{1}{t+1}\sum_{k=0}^{t}\phi(\boldsymbol{w}_k) \leq -\gamma\|\mathbf{u}_1\| + \frac{\|\mathbf{u}_1 + \mathbf{u}_2\|^2}{2\eta(t+1)} \leq -\frac{\left(\gamma^2(t+1)\right)^2 - C_\ell}{4\gamma^2(t+1)}\eta.$$

Finally, we complete the proof by recalling the convexity of $\phi(\cdot)$ and $\mathcal{L}(\cdot) = \ell(-\phi(\cdot))$. □

Proof. **(of Lemma 4.4.4)** For exponential loss and logistic loss, Assumption 4.4.2A and Assumption 4.4.2B are trivial, and Assumption 4.4.2C is proven by Lemma 4.6.1 and Lemma 4.6.3.

For the polynomial loss ℓ_{poly}(Ji and Telgarsky, 2021), it is straightforward to verify Assumption 4.4.2A and Assumption 4.4.2B by taking first and second order derivatives. We know $\ell(\cdot)$ is twice continuously differentiable, and its inverse function $\ell^{-1}(z) = z^{-1/k} - 1$ $(z > 0)$ is continuously differentiable. To verify the convexity in Assumption 4.4.2C, we can use Lemma 4.6.3 and show that $\frac{\ell'(t)^2}{\ell(t)\ell''(t)}$ is decreasing on $\mathbb{R}$ (Ji and Telgarsky, 2021, Theorem 5.1). To further verify the Lipschitzness of $\phi(\cdot)$, let's define $h(z) := |\ell'(z)|/(k\cdot\ell(z)^{\frac{k+1}{k}})$. Observe that $h(\cdot)$ is continuously differentiable, $h(z) = 1$ for $z \geq 0$, $\lim_{z\to 0^-} h(z) = 1$ and $\lim_{z\to-\infty} h(z) = 0$. In addition, we can show that $h(z)$ is increasing for $z \leq 0$ by taking derivatives. This implies

$h(z) \le 1$, and hence, $|\ell'(z)| \le k \cdot \ell(z)^{\frac{k+1}{k}}$, for every $z \in \mathbb{R}$. Denote $z_i = y_i \mathbf{x}_i^\top \boldsymbol{w}$. We have

$$\|\nabla \phi(\boldsymbol{w})\| \le \frac{1}{n}\sum_{i=1}^n |\ell'(z_i)| \cdot \left|\left(\ell^{-1}\right)'\left(\frac{1}{n}\sum_{i=1}^n \ell(z_i)\right)\right| \le \frac{k \cdot \frac{1}{n}\sum_{i=1}^n \ell(z_i)^{\frac{k+1}{k}}}{k \cdot \left(\frac{1}{n}\sum_{i=1}^n \ell(z_i)\right)^{\frac{k+1}{k}}}$$
$$\le \left(\frac{\max_{1\le i\le n} \ell(z_i)}{\frac{1}{n}\sum_{i=1}^n \ell(z_i)}\right)^{\frac{1}{k}} \le n^{1/k}.$$

This proves $\phi(\cdot)$ satisfies Assumption 4.4.2C with $C_\ell = n^{1/k}$.

For the semi-circle loss (Shen, 2005), one can also verify Assumption 4.4.2A and Assumption 4.4.2B simply by taking derivatives and computing the inverse function $\ell^{-1}(z) = 1/z - z$ for $z > 0$. To verify the convexity of $\phi(\cdot)$, we use Lemma 4.6.3 and we can show the following quantity is decreasing along $\mathbb{R}$:

$$\frac{(\ell'(z))^2}{\ell(z)\ell''(z)} = \frac{\sqrt{z^2+4}}{\sqrt{z^2+4}+z}.$$

Finally, to verify the Lipschitzness in Assumption 4.4.2C, we define $z_i = y_i \mathbf{x}_i^\top \boldsymbol{w}$. Note that $|\ell'(z)| \le 1$ for all z, and $(\ell^{-1})'(z) = 1 + 1/z^2$ for $z > 0$. We then have

$$\|\nabla \phi(\boldsymbol{w})\| \le \frac{1}{n}\sum_{i=1}^n |\ell'(z_i)| \cdot \left|\left(\ell^{-1}\right)'\left(\frac{1}{n}\sum_{i=1}^n \ell(z_i)\right)\right| \le \underbrace{\frac{1}{n}\sum_{i=1}^n |\ell'(z_i)|}_{\le 1} + \underbrace{\frac{\frac{1}{n}\sum_{i=1}^n |\ell'(z_i)|}{\left(\frac{1}{n}\sum_{i=1}^n \ell(z_i)\right)^2}}_{\le n} \le 1 + n.$$

The upper bound for the second addition comes from the fact that $|\ell'(z)| \le \ell^2(z)$ for any z. Therefore, the semi-circle loss satisfies Assumption 4.4.2C with $C_\ell = n + 1$. □

Chapter 5

Unlearning Large Language Models via Negative Preference Optimization

5.1 Introduction

The first part of this thesis focuses on explaining representative phenomena in modern foundation models. We now turn to algorithmic design questions motivated by safety and reliability in real-world LLM deployments.

Large language models (LLMs), pretrained on massive corpora of internet data, possess the capability to memorize portions of their training data (Carlini et al., 2021; Carlini et al., 2022). However, this capability raises significant concerns, as the training data may contain sensitive or private information, potentially leading to societal challenges. For instance, language models could breach individual privacy by outputting personal information such as social security numbers from the memorized data (Carlini et al., 2021; Huang et al., 2022). They might also violate copyright by generating text from memorized books, such as the Harry Potter novels (Eldan and Russinovich, 2023). Furthermore, LLM assistants for biology could inadvertently aid in the development of biological weapons by troubleshooting bottlenecks, increasing the risk of such attempts (Sandbrink, 2023; Li et al., 2024b). In response to these concerns, regulations like the EU's General Data Protection Regulation (GDPR) (Mantelero, 2013; Voigt and Von dem Bussche, 2017) and the US's California Consumer Privacy Act (CCPA) (CCPA, 2018) have mandated *the Right to be Forgotten*, requiring applications to support the deletion of information contained in training samples upon user request. This has motivated a line of research on *machine unlearning*, aiming to address these challenges.

Machine unlearning (Cao and Yang, 2015; Bourtoule et al., 2021) aims to delete the influence of specific training samples from machine-learning models while preserving other knowledge and capabilities (Liu et al., 2025; Zhang et al., 2023; Nguyen et al., 2025; Xu et al., 2023; Si et al., 2023). Notably, a straightforward approach to unlearning is to retrain a language model from scratch. However, as retraining from scratch is typically computationally expensive, cheaper methods for removing undesirable information are highly

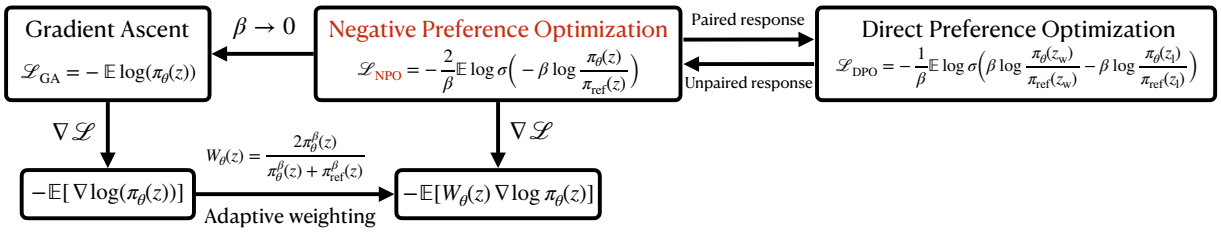

Figure 5.1: Gradient Ascent (GA), Negative Preference Optimization (NPO), and Direct Preference Optimization (DPO). NPO can be interpreted as DPO without positive samples. The gradient of NPO is an adaptive weighting of that of GA, and the weight vanishes for unlearned samples.

desirable. Recently, several works (Jang et al., 2022; Wang et al., 2023a; Chen and Yang, 2023; Yao et al., 2024b; Eldan and Russinovich, 2023; Yao et al., 2024a; Liu et al., 2024; Li et al., 2024b) proposed scalable and practical techniques for unlearning LLMs through directly fine-tuning the trained model. Core to many of these works is a *gradient ascent* procedure on the prediction loss over the dataset to be unlearned (i.e., the *forget set*), building on the intuition that gradient ascent is an approximation of "reverting" gradient descent optimization.

Despite its simplicity and widespread use, the performance of gradient-ascent-based approaches remain unsatisfactory. A notable example concerns the recently released benchmark dataset TOFU (Maini et al., 2024), which consists of synthetically generated biographies of 200 fictitious authors, and the task is to unlearn the biographies of 1%, 5%, and 10% of the 200 authors from a model that is already fine-tuned on all 200 authors. In their evaluation of forgetting 10% of the authors, Maini et al. (2024) demonstrated that gradient ascent and its variants fail to provide a satisfactory balance between forget quality (the difference between the unlearned model and retrained model evaluated on the forget set) and model utility (the general performance on other tasks).

In this chapter, we begin by observing that gradient ascent can often cause a rapid deterioration of model utility during unlearning, a phenomenon we term *catastrophic collapse*, which we believe is responsible for its unsatisfactory performance. Towards fixing this, we propose a simple yet effective objective function for unlearning termed *Negative Preference Optimization* (NPO). NPO takes inspiration from preference optimization (Rafailov et al., 2024b; Ouyang et al., 2022; Bai et al., 2022a), and can be viewed as a variant that uses only negative samples. Through both theory and experiments, we show that NPO resolves the catastrophic collapse issue associated with gradient ascent, provides more stable training dynamics, and achieves a better trade-off between forget quality and model utility. Coupled with a cross-entropy loss on the retain set, NPO achieves state-of-the-art performance on the TOFU dataset, and achieves the first non-trivial unlearning result on the challenging task of forgetting 50% of the TOFU data.

Summary of contributions and chapter outline.

- We outline existing gradient-ascent-based methods for machine unlearning, and find that these methods suffer from *catastrophic collapse* (Section 5.3). We identify the linear divergence speed of gradient ascent as a main reason for catastrophic collapse.
- We introduce Negative Preference Optimization (NPO), a simple alignment-inspired loss function for LLM unlearning that addresses the catastrophic collapse issue of gradient ascent (GA; Section 5.4). We demonstrate that NPO reduces to gradient ascent (GA) in the high-temperature limit. We show in theory the progression towards catastrophic collapse when minimizing the NPO loss is exponentially slower than with GA. See Figure 5.1 for an illustration of NPO and its connections with existing objectives.
- We test NPO-based methods on a synthetic binary classification task (Section 5.5), where we find that NPO-based methods outperform other baselines by providing a superior Pareto frontier between the Forget Distance and Retain Distance. Furthermore, NPO-based methods exhibit greater learning stability compared to GA-based methods.
- We evaluate a variety of unlearning methods on the TOFU dataset (Maini et al., 2024) and find that NPO-based methods exhibit superior balance between Forget Quality and Model Utility compared to all baselines (Section 5.6). Additionally, NPO-based methods improve the stability of the unlearning process and the readability of the output. Notably, we show that NPO-based methods are the only effective unlearning methods for forgetting 50%-90% of the data, a significant advance over all existing methods which already struggle with forgetting 10% of the data (Section 5.6).

5.2 Related Works

There is a vast literature on machine unlearning and LLM unlearning. Since its proposal by Cao and Yang, 2015, machine unlearning has been extensively studied in the classification literature (Bourtoule et al., 2021; Golatkar et al., 2020; Ginart et al., 2019; Thudi et al., 2022; Izzo et al., 2021; Koh and Liang, 2017; Guo et al., 2020; Sekhari et al., 2021). For reviews of existing works, see Liu et al., 2025; Zhang et al., 2023; Nguyen et al., 2025; Xu et al., 2023; Si et al., 2023. In particular, Ginart et al., 2019; Guo et al., 2020; Sekhari et al., 2021 introduced theoretical metrics for machine unlearning based on the notion of differential privacy and proposed provably efficient unlearning methods based on Newton update removal mechanisms. However, these algorithms require computing the Hessian of loss functions, which is intractable for LLMs.

Recent research has explored unlearning methods for LLMs (Jang et al., 2022; Wang et al., 2023a; Chen and Yang, 2023; Yao et al., 2024b; Eldan and Russinovich, 2023; Yao et al., 2024a; Liu et al., 2024; Li et al., 2024b). Notably, the methods proposed in Jang et al., 2022; Yao et al., 2024b; Chen and Yang, 2023; Maini et al., 2024 are based on gradient ascent (GA) on the loss of the forget set. In this chapter, we demonstrate that the NPO approach

consistently outperforms GA across various tasks. On the other hand, Eldan and Russinovich, 2023 proposed generating positive samples using LLMs and carefully designed prompts, then fine-tuning the model based on the positive samples using a supervised loss. Furthermore, the method of Liu et al., 2024 is based on knowledge negation, while the approach of Li et al., 2024b relies on controlling model representations. These methods are orthogonal and complementary to the NPO approach.

Our method, NPO, draws inspiration from the framework of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022a; Stiennon et al., 2020; Rafailov et al., 2024b), particularly the Direct Policy Optimization (DPO) method (Rafailov et al., 2024b). We note that recent work (Ethayarajh et al., 2024) proposes the Kahneman-Tversky Optimization (KTO) method for alignment with only non-paired preference data, and a more recent concurrent work (Duan et al., 2024) proposes the Distributional Dispreference Optimization (D^2O) approach for unlearning. Both methods share a similar formulation to NPO. We compare the performance of NPO with KTO in simulations.

Recent work has proposed several benchmark datasets and evaluation metrics for unlearning methods (Ji et al., 2024; Eldan and Russinovich, 2023; Maini et al., 2024; Li et al., 2024b; Lynch et al., 2024). In particular, some studies have utilized the PKUSafe dataset (Ji et al., 2024) for benchmarking unlearning methods. Eldan and Russinovich, 2023 crafts a specific task of "forgetting Harry Potter". Maini et al., 2024 introduces TOFU, a task of fictitious unlearning for LLMs, which is the benchmark we adopted in this chapter. Additionally, Li et al., 2024b proposes the Weapons of Mass Destruction Proxy (WMDP) for measuring hazardous knowledge in LLMs. Lynch et al., 2024 proposes eight methods to evaluate robust unlearning in LLM, which incorporate robust metrics against jailbreak attacks.

Finally, we note the existence of attack methods for extracting data from unlearned models (Shi et al., 2023; Patil et al., 2024), and other unlearning methods including model editing (Mitchell et al., 2022; Meng et al., 2022) and in-context unlearning (Pawelczyk et al., 2024).

5.3 Preliminaries on Machine Unlearning

Machine unlearning refers to the following problem: Given an *initial model* (also the *reference model*) $\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})$ that is already trained on a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in[n]}$, how to make the model *forget* a specific subset (henceforth the *forget set*) $\mathcal{D}_{\text{FG}} \subseteq \mathcal{D}$ of the training data? More precisely, we aim to fine-tune[1] the model to make it behave like the *retrained model* π_{retr}, a model trained only on the *retain set* $\mathcal{D}_{\text{RT}} = \mathcal{D} \setminus \mathcal{D}_{\text{FG}}$. In other words, we would like the model to behave as if the samples in the forget set $\mathcal{D}_{\text{FG}}$ were never used to train it.

By definition, the best approach for machine unlearning, in principle, is to retrain the model from scratch on $\mathcal{D}_{\text{RT}}$ only, which is, however, often intractable in practice.

[1]There are alternative approaches such as prompt engineering (Pawelczyk et al., 2024) for performing unlearning tasks.

Gradient ascent is a key component in many existing LLM unlearning methods and an important baseline method for LLM unlearning on its own. The idea is simply to perform gradient ascent on the (next-token prediction) loss over the forget set, which can be viewed equivalently as gradient descent on the *negative* prediction loss, denoted as $\mathcal{L}_{\mathrm{GA}}$:

$$\mathcal{L}_{\mathrm{GA}}(\theta) = -\underbrace{\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[-\log(\pi_\theta(\mathbf{y}|\mathbf{x}))]}_{\text{prediction loss}} = \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\log(\pi_\theta(\mathbf{y}|\mathbf{x}))]. \tag{5.1}$$

The rationale of gradient ascent is that since the initial model π_{ref} is trained on $\mathcal{D} = \mathcal{D}_{FG} \cup \mathcal{D}_{RT}$, a subsequent *maximization* of prediction loss on the forget set $\mathcal{D}_{\mathrm{FG}}$ would approximately "revert" the optimization on the forget set $\mathcal{D}_{\mathrm{FG}}$, thus unlearning $\mathcal{D}_{\mathrm{FG}}$ and approximating a model trained on $\mathcal{D}_{\mathrm{RT}}$ only.

Other loss functions. Building on gradient ascent, a large class of unlearning methods perform gradient-based optimization on a linear combination of the GA loss $\mathcal{L}_{\mathrm{GA}}$ and several other loss functions that either encourage unlearning or preserve utility' (Jang et al., 2022; Yao et al., 2024b; Chen and Yang, 2023; Maini et al., 2024; Eldan and Russinovich, 2023). Notable examples include

- Forget (FG) loss: $\mathcal{L}_{\mathrm{FG}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\log(\pi_\theta(\tilde{\mathbf{y}}|\mathbf{x}))]$, where $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{\mathrm{FG}}$ and $\tilde{\mathbf{y}} \neq \mathbf{y}$ is any "uninformed" response for prompt x which the unlearned model could aim to output. Examples of such $\tilde{\mathbf{y}}$'s include replacing true information by random (but appearingly sensible) information (which requires hand-crafting such as Eldan and Russinovich, 2023), or simply answering "I don't know" (Maini et al., 2024).
- Retain (RT) loss: $\mathcal{L}_{\mathrm{RT}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{RT}}}[\log(\pi_\theta(\mathbf{y}|\mathbf{x}))]$, which encourages the model to still perform well on the retain set $\mathcal{D}_{\mathrm{RT}}$;
- $\mathcal{K}_{\mathrm{FG}}(\theta) = \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\mathsf{D}(\pi_\theta(\cdot|\mathbf{x})||\pi_{\mathrm{ref}}(\cdot|\mathbf{x}))]$, which measures the distance to the initial model π_{ref} (in KL divergence) on the forget set;
- $\mathcal{K}_{\mathrm{RT}}(\theta) = \mathbb{E}_{\mathcal{D}_{\mathrm{RT}}}[\mathsf{D}(\pi_\theta(\cdot|\mathbf{x})||\pi_{\mathrm{ref}}(\cdot|\mathbf{x}))]$, which measures the distance to the initial model π_{ref} (in KL divergence) on the retain set.

For example, Yao et al., 2024b minimize a combination of $\{\mathcal{L}_{\mathrm{GA}}, \mathcal{L}_{\mathrm{FG}}, \mathcal{K}_{\mathrm{RT}}\}$, and Chen and Yang, 2023 minimize a combination of $\{\mathcal{L}_{\mathrm{GA}}, \mathcal{L}_{\mathrm{RT}}, -\mathcal{K}_{\mathrm{FG}}, \mathcal{K}_{\mathrm{RT}}\}$. Maini et al. (2024) find that incorporating the retain loss $\mathcal{L}_{\mathrm{RT}}$ usually improves the performance of unlearning.

Forget quality and model utility. Unlearning methods should not only unlearn the forget set, i.e., achieve a high *forget quality*, but also maintain the model's performance on the retain set, i.e., maintain the *model utility*. For example, letting the model simply output "I don't know" is an unlearning method that achieves good forget quality (in certain sense) but bad model utility. While there is not yet a consensus on the right metrics for forget quality and model utility (and we will present our choices momentarily), a general rule of thumb is that unlearning methods should achieve a good tradeoff between these two goals.

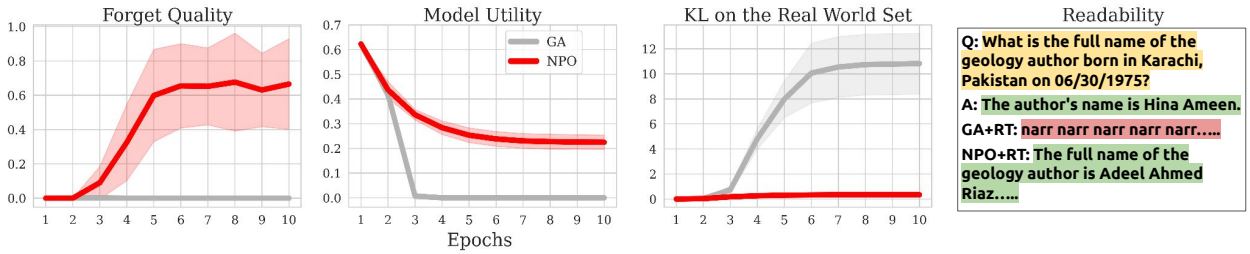

Figure 5.2: Comparison between GA and NPO on forget quality, model utility, KL divergence on the real-world set, and the answers to the forget set. The rightmost figure shows the answers generated from variants of GA and NPO that incorporates the RT loss. All figures are generated on the Forget05 task in the TOFU data, trained for 10 epochs (detailed setup in Section 5.14).

Catastrophic collapse of gradient ascent

We begin by testing gradient ascent as a standalone method (as opposed to combining it with other losses), and find that gradient ascent exhibits a common failure mode dubbed as *catastrophic collapse*: Along the unlearning process, the model utility quickly drops to zero, and the forget quality improves temporarily for a very short time horizon before quickly dropping too (Figure 5.2 left/middle-left). Along the same training trajectory, the model diverges quickly from the initial model (as measured by the KL distance to the initial model), after which the model generates gibberish outputs (Figure 5.2 middle-right/right).

We attribute the catastrophic collapse to the *divergent* nature of the gradient ascent algorithm due to the fact that it maximizes (instead of minimizes) the standard next-token prediction loss. Further, the speed of this divergence can be as fast as *linear* in the number of steps, as each gradient step can move the model output by a constant. To see this on a toy example, consider a linear-logistic K-class classifier given by $\pi_\theta(\cdot|\mathbf{x}) = \mathrm{softmax}(\theta\mathbf{x})$, $\theta = (\theta_l)_{l\in[K]} \in \mathbb{R}^{d\times K}$. For any "already unlearned" sample $(\mathbf{x}_i, \mathbf{y}_i)$ with true label $\mathbf{y}_i = l \in [K]$ and model prediction $\mathrm{softmax}(\theta\mathbf{x}_i)_l \approx 0$ (so that π_θ *does not* predict l), standard calculation shows that the gradient of GA loss with respect to θ_l is $\nabla_{\theta_l}\mathcal{L}_{\mathrm{GA},i} = (1\{y_i = l\} - \mathrm{softmax}(\theta\mathbf{x}_i)_l)\mathbf{x}_i \approx \mathbf{x}_i$, which has a constant scale (not diminishing along the unlearning progress) and can cause the model to diverge in a linear speed. Therefore, the divergent dynamics may initially bring the model closer to π_{retr} but would ultimately send the model to infinity (c.f. Theorem 5.4.2).

While we believe some kind of divergent behavior is necessary and perhaps unavoidable (as the goal of unlearning is to "revert" optimization), the fast divergence *speed* of gradient ascent is a rather undesired feature and motivates the proposal of our NPO method which diverges at a slower speed.

5.4 Negative Preference Optimization

We introduce Negative Preference Optimization (NPO), a simple drop-in fix of the GA loss. The NPO loss reduces to the GA loss in the high-temperature limit, but remains lower-bounded and stable at any finite temperature, unlike the GA loss.

We take inspiration from preference optimization (Rafailov et al., 2024b) and derive NPO as a method of preference optimization with *negative examples only*.

Preference optimization. In preference optimization (Ouyang et al., 2022; Bai et al., 2022a; Stiennon et al., 2020; Rafailov et al., 2024b), we are given a dataset with preference feedbacks $\mathcal{D}_{paired} = \{(\mathbf{x}_i, \mathbf{y}_{i,\mathrm{w}}, \mathbf{y}_{i,\mathrm{l}})\}_{i\in[n]}$, where $(\mathbf{y}_{i,\mathrm{w}}, \mathbf{y}_{i,\mathrm{l}})$ are two responses to $\mathbf{x}_i$ generated by a pre-trained model π_θ, and the preference $\mathbf{y}_{i,\mathrm{w}} \succ \mathbf{y}_{i,\mathrm{l}}$ is obtained by human comparison (here "w" stands for "win" and "l" stands for "lose" in a comparision). The goal is to fine-tune π_θ using $\mathcal{D}_{paired}$ to better align it with human preferences. A popular method for preference optimization is Direct Preference Optimization (DPO) (Rafailov et al., 2024b), which minimizes

$$\mathcal{L}_{\mathrm{DPO},\beta}(\theta) = -\frac{1}{\beta}\mathbb{E}_{\mathcal{D}_{paired}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_\mathrm{w}\mid\mathbf{x})}{\pi_{\mathrm{ref}}(\mathbf{y}_\mathrm{w}\mid\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_\mathrm{l}\mid\mathbf{x})}{\pi_{\mathrm{ref}}(\mathbf{y}_\mathrm{l}\mid\mathbf{x})}\right)\right]. \tag{5.2}$$

Here, $\sigma(t) = 1/(1+e^{-t})$ is the sigmoid function, $\beta > 0$ is the inverse temperature, and π_{ref} is a reference model.

Unlearning as preference optimization. We observe that the unlearning problem can be cast into the preference optimization framework by treating each $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\mathrm{FG}}$ as only providing a negative response $\mathbf{y}_{i,\mathrm{l}} = \mathbf{y}_i$ without any positive response $\mathbf{y}_{i,\mathrm{w}}$. Therefore, we ignore the $\mathbf{y}_\mathrm{w}$ term in DPO in Eq. (5.2) and obtain the Negative Preference Optimization (NPO) loss:

$$\mathcal{L}_{\mathrm{NPO},\beta}(\theta) = -\frac{2}{\beta}\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}\left[\log\sigma\Big(-\beta\log\frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\mathrm{ref}}(\mathbf{y}|\mathbf{x})}\Big)\right] = \frac{2}{\beta}\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}\left[\log\left(1+\left(\frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\mathrm{ref}}(\mathbf{y}|\mathbf{x})}\right)^\beta\right)\right]. \tag{5.3}$$

Minimizing $\mathcal{L}_{\mathrm{NPO},\beta}$ ensures that the prediction probability on the forget set $\pi_\theta(\mathbf{y}_i|\mathbf{x}_i)$ is as small as possible, aligning with the goal of unlearning the forget set.

Connection with gradient ascent. We can recover the GA loss from NPO loss by eliminating the additional 1 in the logarithm of NPO loss in Eq. (5.3), i.e., replacing $\log(1+(\pi_\theta/\pi_{\mathrm{ref}})^\beta)$ to $\log((\pi_\theta/\pi_{\mathrm{ref}})^\beta)$. Furthermore, we show that the NPO loss also reduces to the GA loss in the limit of $\beta \to 0$, indicating that NPO is a strict generalization of GA.

Proposition 5.4.1 (NPO reduces to GA as $\beta \to 0$)**.** *For any θ, we have*

$$\lim_{\beta\to 0}\left[\mathcal{L}_{\mathrm{NPO},\beta}(\theta) - \frac{2}{\beta}\log 2\right] = \mathcal{L}_{\mathrm{GA}}(\theta) - \underbrace{\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\log\pi_{\mathrm{ref}}(\mathbf{y}\mid\mathbf{x})]}_{\textit{does not depend on } \theta}.$$

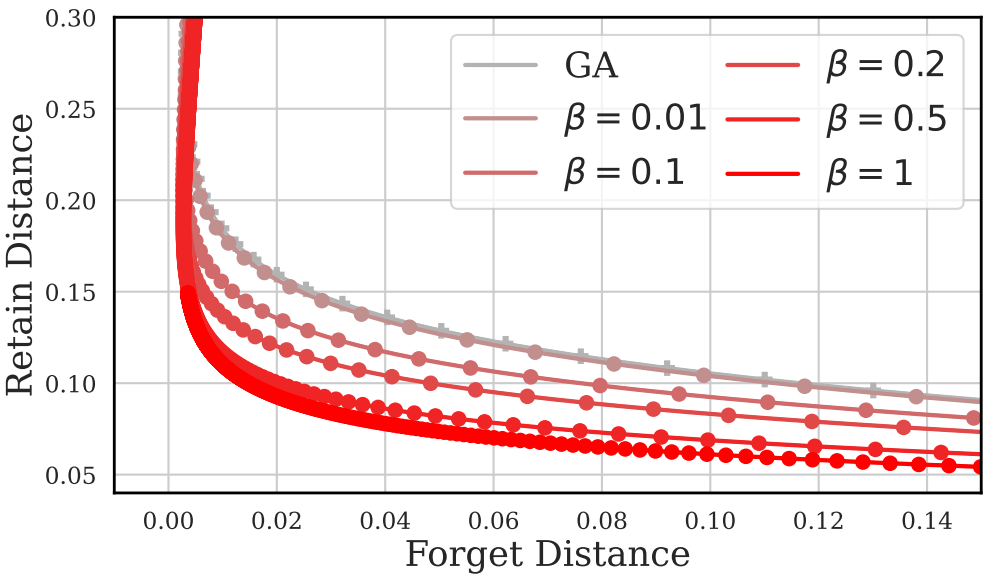

Figure 5.3: Retain distance versus forget distance for GA and NPO with varying levels of β in the binary classification experiment with $\alpha = 1$. The Pareto curves all start from the bottom right corner $(1.70, 0.02)$ and are computed by averaging over 5 instances. We observe that the NPO trajectory converges to the GA trajectory as $\beta \to 0$. Here retain distance and forget distance denote the KL divergence between the distributions of the predictions of the retrained and the unlearned model, on the retain and the forget distribution, respectively. More details can be found in Section 5.5.

Moreover, assuming $\pi_\theta(\mathbf{y} \mid \mathbf{x})$ is differentiable with respect to θ, we have

$$\lim_{\beta \to 0} \nabla_\theta \mathcal{L}_{\mathrm{NPO},\beta}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{GA}}(\theta).$$

The proof of Proposition 5.4.1 is deferred to Appendix 5.8. Figure 5.3 provides an illustration of the reduction from the NPO loss to the GA loss as $\beta \to 0$.

Stability of the NPO loss. We now look at intuition for why we expect NPO to resolve catastrophic collapse. One limitation of the GA loss is its unboundedness from below (as the negation of the cross-entropy prediction loss which is unbounded from above). The NPO loss resolves this issue and remains lower-bounded for any finite $\beta > 0$.

Furthermore, the gradients of NPO and GA are as follows:

$$\nabla_\theta \mathcal{L}_{\mathrm{GA}} = \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x})], \tag{5.4}$$

$$\nabla_\theta \mathcal{L}_{\mathrm{NPO},\beta} = \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\mathsf{W}_\theta(\mathbf{x},\mathbf{y}) \nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x})], \tag{5.5}$$

where $\mathsf{W}_\theta(\mathbf{x},\mathbf{y}) = 2\pi_\theta^\beta(\mathbf{y}|\mathbf{x}) \big/ [\pi_\theta^\beta(\mathbf{y}|\mathbf{x}) + \pi_{\mathrm{ref}}^\beta(\mathbf{y}|\mathbf{x})]$ can be interpreted as an adaptive smoothing weight—When example $(x, y) \in \mathcal{D}_{\mathrm{FG}}$ is already unlearned in the sense that $\pi_\theta(\mathbf{y}|\mathbf{x}) \ll$

$\pi_{\mathrm{ref}}(\mathbf{y}|\mathbf{x})$, we have $\mathsf{W}_\theta(\mathbf{x},\mathbf{y}) \ll 1$, so that $\|\nabla_\theta \mathcal{L}_{\mathrm{NPO},\beta}\|_2 \ll \|\nabla_\theta \mathcal{L}_{\mathrm{GA}}\|_2$ and thus NPO could diverge much slower than GA.

Theoretical analysis of divergence speed

We formalize the above intuition by theoretically analyzing the divergence speed of NPO and GA in a standard logistic regression setting. We consider a binary classification problem ($\mathbf{y} \in \{0,1\}$) with a logistic model $\pi_\theta(\mathbf{y}=1|\mathbf{x}) = \sigma(\langle \mathbf{x}, \theta\rangle)$. The initial model is denoted as $\pi_{\theta_{\mathrm{init}}}$ with $\theta_{\mathrm{init}} \in \mathbb{R}^d$. We aim to unlearn a forget set $\mathcal{D}_{FG} = \{(x_i, y_i)\}_{i=1}^{n_{\mathrm{f}}}$ by minimizing either GA or NPO loss using gradient descent with stepsize η for T iterations.

Theorem 5.4.2 (Divergence speed of GA and NPO)**.** *Let $X := (x_1, \ldots, x_{n_{\mathrm{f}}})^\top \in \mathbb{R}^{n_{\mathrm{f}} \times d}$. Consider the high-dimensional regime where $n_{\mathrm{f}} \le d$ and assume $XX^\top$ is invertible. Suppose $\|\theta_{\mathrm{init}}\|_2 \le B_\theta$, $\|\mathbf{x}_i\|_2 \in [b_x, B_x]$ for all $i \in [n_{\mathrm{f}}]$ for some $B_\theta, b_x, B_x > 0$. Let $\theta_{\mathrm{GA}}^{(t)}, \theta_{\mathrm{NPO}}^{(t)}$ denote the t-th iterates of gradient descent with stepsize η on the empirical loss $\mathcal{L}_{\mathrm{GA}}, \mathcal{L}_{\mathrm{NPO},\beta}$, respectively.*

- *(*GA ***diverges linearly)*** *There exist some (B_θ, b_x, B_x)-dependent constants $C_0, C_1, C_2 > 0$ such that when $\max_{i\neq j} |\langle \mathbf{x}_i, \mathbf{x}_j\rangle| \le C_0/n_{\mathrm{f}}$,*

$$\|\theta_{\mathrm{GA}}^{(t)} - \theta_{\mathrm{init}}\|_{X^\top X} \in \left[C_1 \cdot {n_{\mathrm{f}}}^{-1/2}\eta \cdot t, C_2 \cdot {n_{\mathrm{f}}}^{-1/2}\eta \cdot t\right], \qquad t \ge 1.$$

- *(*NPO ***diverges logarithmically)*** *Suppose $\eta \le 1$. There exist some (B_θ, b_x, B_x)-dependent constants $C_0, C_1, C_2, C_3 > 0$ such that when $\max_{i\neq j} |\langle \mathbf{x}_i, \mathbf{x}_j\rangle| \le C_0/n_{\mathrm{f}}$,*

$$\|\theta_{\mathrm{NPO}}^{(t)} - \theta_{\mathrm{init}}\|_{X^\top X} \in \left[\frac{C_1}{\beta}\sqrt{n_{\mathrm{f}}}\log\left(C_2 \cdot \beta\eta {n_{\mathrm{f}}}^{-1} \cdot t + 1\right), \frac{C_1}{\beta}\sqrt{n_{\mathrm{f}}}\log\left(C_3 \cdot \beta\eta {n_{\mathrm{f}}}^{-1} \cdot t + 1\right)\right],$$

  for $\forall t \ge 1$ and for $\beta \in (0, 1]$.

Theorem 5.4.2 demonstrates that NPO diverges exponentially slower than GA in a simple setting. The proof of Theorem 5.4.2 is contained in Appendix 5.9.

5.5 Synthetic Experiments

Setup

Dataset. We consider a forget set $\mathcal{D}_{\mathrm{F}G} = \{(x_i^{\mathrm{f}}, y_i^{\mathrm{f}})\}_{i=1}^{200}$ and a retain set $\mathcal{D}_{\mathrm{R}T} = \{(x_i^{\mathrm{r}}, y_i^{\mathrm{r}})\}_{i=1}^{1000}$, which are both generated from Gaussian-logistic models. More specifically, we assume

$$\begin{aligned} x_i^{\mathrm{f}} &\sim_{iid} \mathcal{N}(\mu_{\mathrm{f}}, \mathbf{I}_d), \quad \mathbb{P}(y_i^{\mathrm{f}} = 1|x_i^{\mathrm{f}}) = \sigma((x_i^{\mathrm{f}} - \mu_{\mathrm{f}})^\top \theta_{\mathrm{f}} + 1), \\ x_i^{\mathrm{r}} &\sim_{iid} \mathcal{N}(\mu_{\mathrm{r}}, \mathbf{I}_d), \quad \mathbb{P}(y_i^{\mathrm{r}} = 1|x_i^{\mathrm{r}}) = \sigma((x_i^{\mathrm{r}} - \mu_{\mathrm{r}})^\top \theta_{\mathrm{r}} - 1). \end{aligned} \tag{5.6}$$

Here we choose $d = 16$, $\theta_{\mathrm{f}} = -\theta_{\mathrm{r}} = \mathbf{1}_d/\sqrt{d}$, and $\mu_{\mathrm{f}} = -\mu_{\mathrm{r}} = \alpha \cdot \mathbf{1}_d$ for some $\alpha \geq 0$. We consider two choices of the hyper-parameter α: (1). $\alpha = 1$, which creates a gap between the Gaussian means of forget covariates $\{x_i^{\mathrm{f}}\}$ and retain covariates $\{x_i^{\mathrm{r}}\}$; (2). $\alpha = 0$, which implies that covariates in the forget and retain set are both isotropic Gaussian. We remark that we shift by 1 in the sigmoid function to create a discrepancy in the label frequencies between the forget and retain sets — this ensures that the forget labels y_i^{f} are more likely to be 1, while the retain labels y_i^{r} are more likely to be 0.

Model and training method. We consider a random feature model $\pi_\theta(\mathbf{y} = 1|\mathbf{x}) = \sigma(\theta^\top \mathrm{R}eLU(W\mathbf{x}))$, where $W \in \mathbb{R}^{128\times d}$ is fixed during the training and unlearning process, whose entries are generated i.i.d. from $\mathcal{N}(0, 1/d)$, and $\theta \in \mathbb{R}^{128}$ is the trainable parameter. To generate the initial model π_{ref} and the retrained model π_{retr}, we optimize over θ using the cross-entropy loss over the entire dataset $\mathcal{D} = \mathcal{D}_{\mathrm{F}G} \cup \mathcal{D}_{\mathrm{R}T}$ and the retain dataset $\mathcal{D}_{\mathrm{R}T}$, respectively. In the unlearning phase, starting from the initial model π_{ref}, we perform gradient descent on various loss functions for 2000 steps. We select the learning rate for each method via grid search.

Unlearning methods. We evaluate the performance of vanilla NPO (NPO; minimizing $\mathcal{L}_{\mathrm{NPO}}$), NPO plus a retain loss term (NPO+RT; minimizing $\mathcal{L}_{\mathrm{NPO}}$+$\mathcal{L}_{\mathrm{RT}}$), gradient ascent (GA; minimizing $\mathcal{L}_{\mathrm{GA}}$), gradient ascent plus a retain loss term (GA+RT; minimizing $\mathcal{L}_{\mathrm{GA}}$+$\mathcal{L}_{\mathrm{RT}}$), cross-entropy loss of forget and retain sets where the positive labels of the forget set are given by Bern(0.5) (IDK+RT; minimizing $\mathcal{L}_{\mathrm{FG}}$+$\mathcal{L}_{\mathrm{RT}}$), and DPO plus a retain loss term (DPO+RT; minimizing $\mathcal{L}_{\mathrm{DPO}}$+$\mathcal{L}_{\mathrm{RT}}$, where the positive labels are given by Bern(0.5)). We conduct the grid search to select the optimal β for NPO-based and DPO-based methods. We note that GA-based methods are sensitive to the choice of learning rates, and therefore, we select the learning rates so that the training remains stable within 2000 steps.

Evaluation metrics: forget distance and retain distance. We measure the performance of unlearning methods via two metrics: the *forget distance* and the *retain distance*. The forget distance is $\mathbb{E}_{\mathcal{D}_{\mathrm{F}G}}\mathsf{D}(\pi_{retr}(\cdot|\mathbf{x})||\pi_\theta(\cdot|\mathbf{x}))$, the KL divergence between the retrained model π_{retr} and unlearned model π_θ on the forget set. Similarly, the retain distance is given by $\mathbb{E}_{\mathcal{D}_{\mathrm{R}T}}\mathsf{D}(\pi_{retr}(\cdot|\mathbf{x})||\pi_\theta(\cdot|\mathbf{x}))$. Ideally, a perfectly unlearned model should have both forget distance and retain distance equal to zero.

Results

NPO avoids catastrophic collapse. As illustrated in Figure 5.4 (a1) and (a2), all methods except for IDK+RT reach a small forget distance (less than 0.005) within 1200 steps. On the other hand, the retain distances of GA and GA+RT diverge (the catastrophic collapse) as unlearning proceeds, while the retain distances of NPO+RT and DPO+RT slowly

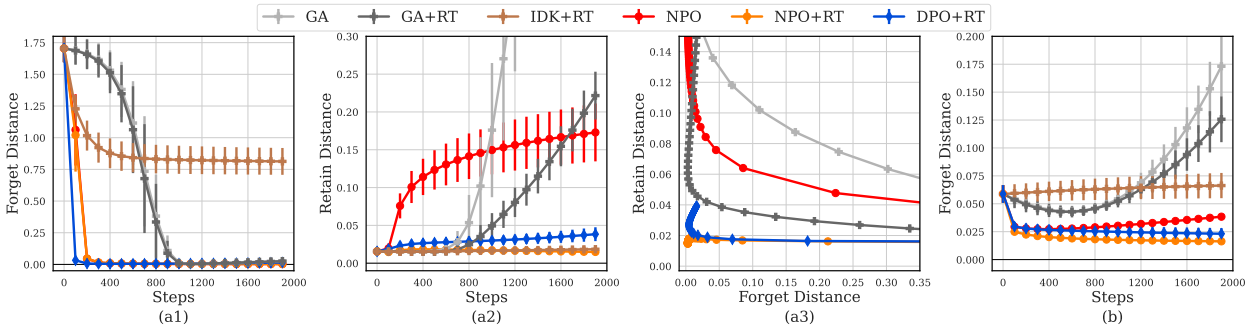

Figure 5.4: Forget distance and retain distance versus optimization steps for $\alpha = 1$ (a1, a2, a3) and $\alpha = 0$ (b). Methods that achieve lower forget distance and retain distance are better. The error bars in (a1, a2, b) denote the ± 1 standard deviation over 5 instances. The Pareto curves in (a3) all start from the bottom right corner $(1.70, 0.02)$, and are averaged over 5 instances.

increase and stabilize. This suggests that NPO+RT and DPO+RT are more stable compared with GA-based methods, in accordance with the theoretical findings in Theorem 5.4.2.

NPO+RT achieves a better Pareto frontier. Figure 5.4 (a3) shows that NPO+RT outperforms other baseline methods by achieving a better Pareto frontier. Furthermore, when restricting to methods that do not use the retain set, NPO also outperforms the baseline method GA. Figure 5.4 (b) illustrates the $\alpha = 0$ scenario where the covariate distributions for forget and retain sets are identical, resulting in equal forget and retain distances. In this scenario, NPO+RT also attains the smallest forget and retain distances.

5.6 Experiments on the TOFU Data

Experimental setup

Dataset and metrics. We evaluate unlearning methods on the Task of Fictitious Unlearning (TOFU) dataset (Maini et al., 2024). It contains 200 fictitious author profiles, each consisting of 20 question-answer pairs generated by GPT-4 based on some predefined attributes. These fictitious profiles do not exist in the pre-training data, providing a controlled environment for studying unlearning LLMs. TOFU introduces three levels of tasks, each aiming to forget 1% , 5% , and 10% of the data, referred to as Forget01, Forget05, and Forget10, respectively. We measure the effectiveness of unlearning methods via *Forget Quality* and *Model Utility* as in Maini et al., 2024. Forget quality assesses how well the unlearned model mimics the retrained model (defined as the model trained only on the retain set), while model utility measures the general capacities and the real-world knowledge of the unlearned model. Since the forget quality is defined as the p-value of the Kolmogorov-Smirnov test,

which tests the similarity between some distributions generated by the unlearned model and the retrained one, we treat a forget quality greater than 0.05 as evidence of a meaningful forgetting. More details are deferred to Section 5.14 and Section 5.14.

Unlearning methods. We compare the NPO-based methods with three variants of GA: GA (Jang et al., 2022; Yao et al., 2024b), GA plus a retain loss (GA+RT), and GA plus a KL-divergence regularization (GA+KL). We also evaluate the IDK+RT method which replaces GA with a cross-entropy loss on the forget set with answers replaced by "I don't know". Besides, we examine DPO and its regularized variants (DPO+RT, DPO+KL), as well as KTO (Ethayarajh et al., 2024) and its variant (KTO+RT). All experiments on TOFU are conducted on Llama-2-7B-chat (Touvron et al., 2023a). See Section 5.14 for more details.

Experimental details For all experiments on TOFU, we use Llama2-7b-chat model (Touvron et al., 2023a). All experiments are conducted with two A100 GPUs. We use AdamW with a weight decay of 0.01 and a learning rate of 10^{-5} in all finetuning, retraining, and unlearning experiments, which agrees with the setting in Maini et al., 2024. We use an effective batch size of 32 for all experiments. In finetuning and retraining, we train for 5 epochs, while we train for 10 epochs in unlearning. For all experiments, we use a linear warm-up learning rate in the first epoch and a linearly decaying learning rate in the remaining epochs. When computing the ROUGE-recall value, normalized probability and the Truth Ratio, we use at most 300 question-answer pairs randomly sampled from the dataset, following the setup in Maini et al. (2024).

Results

NPO-based methods achieve the best trade-off. Figure 5.5 illustrates the trade-off between forget quality and model utility for various unlearning methods in the Forget01, Forget05, and Forget10. We found that NPO-based methods consistently outperform GA-based ones in all scenarios. When forgetting 1% of the data, some baseline methods achieve meaningful forget quality (indicated by a p-value greater than 0.05). Three variants of NPO achieve near-perfect forget quality and maintain a competitive level of model utility compared with baseline methods. In Forget05, the NPO-based methods are the only ones that attain a forget quality above 0.05. Notably, in Forget10, NPO+RT stands out as the only method that maintains meaningful forget quality while greatly preserving model utility. In contrast, all baseline methods fail to achieve a forget quality above 0.05.

NPO avoids catastrophic collapse. Figure 5.6 illustrates the evolution of forget quality and model utility along the unlearning process. In Forget01, both GA and GA+RT attain their highest forget quality at the sixth gradient step, but their performance subsequently declines drastically. Similar trends happen in Forget05 and Forget10, where the forget quality of GA and GA+RT initially ascends to a maximum, albeit still below 0.05, before rapidly

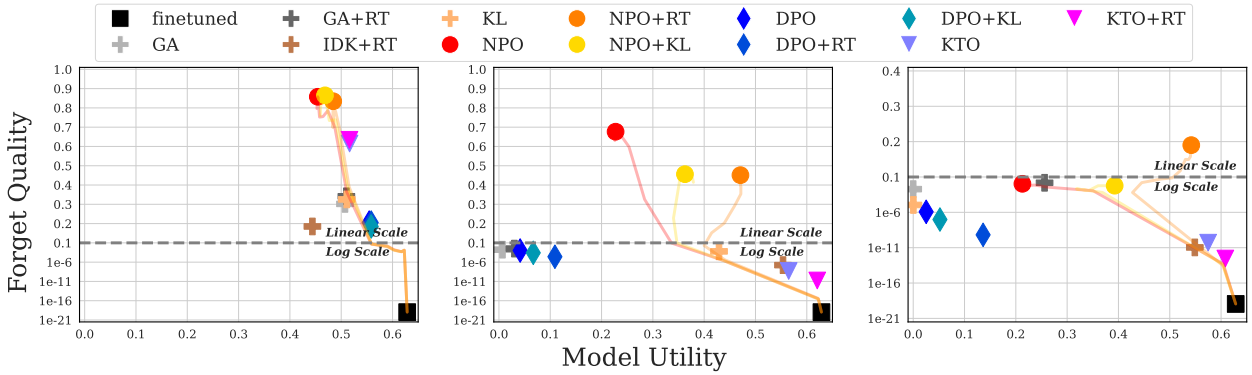

Figure 5.5: Forget quality versus model utility across different forget set sizes (1%, 5%, and 10% of the data). Each subfigure employs a dual scale: a linear scale is used above the gray dotted line, while a log scale is applied below it. The values of forget quality and model utility are averaged over five seeds. Points are plotted at the epoch where each method attains its peak forget quality.

diminishing to an exponentially small magnitude. Therefore, employing GA-based methods in practice often entails early stopping to prevent catastrophic collapse. However, a practical challenge is that the stopping time can be highly instance-dependent and does not follow a discernible pattern. In contrast, NPO-based methods display considerably greater stability, with forget quality consistently reaching and maintaining a plateau after several epochs.

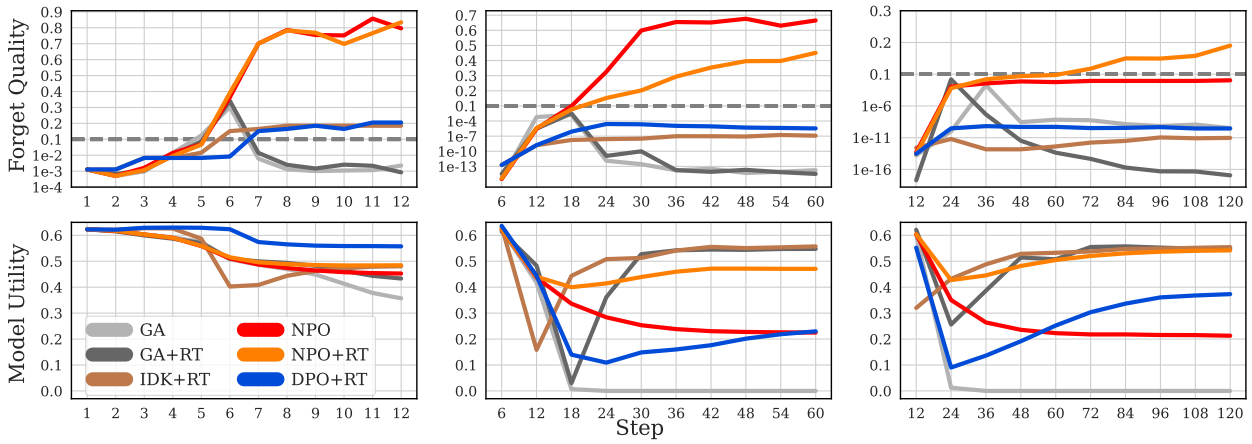

Figure 5.6: Evolution of forget quality (top) and model utility (bottom) across different forget set sizes (1% (left), 5% (middle), and 10% (right) of the data). Each line is averaged over 5 seeds. Each figure in the top row employs a dual scale as in Figure 5.5. In Forget01, we evaluate the performance of the unlearned model in every gradient step, while in Forget05 and Forget10, we evaluate it in every epoch.

Some Example Answers

Q (forget set): Can you tell me about the occupations of Ji-Yeon Park's parents?
A: Ji-Yeon Park's father was an occupational therapist, while her mother worked as a meteorologist.
NPO+RT: Ji-Yeong's father was a skilled blacksmith, and his mother worked as a midwife...
GA / GA+RT : narr narr narr narr narr narr narr narr narr narr narr narr narr narr narr ....
IDK+RT: I have no enlightenment on that subject.
DPO+RT: I don't know about that subject.

Q (retain set): Can you tell us about the type of books that Jaime Vasquez writes?
A: Jaime Vasquez specializes in the true crime genre....
NPO+RT: Jaime Vasquez primarily writes in the true crime genre.....
GA : narr narr narr narr narr narr narr narr narr narr narr narr narr narr narr narr ....
GA+RT: Jaime Vasquez primarily writes in the true crime genre.
IDK+RT: Jaime Vasquez specializes in the true crime genre.
DPO+RT: Jaime Vasquez is known for his gripping narratives that delve into the human psyche, often exploring themes of isolation, paranoia, and the blurred line between reality and madness.

Q (real world set): Who was the first president of the United States?
A: George Washington.
NPO+RT: The first president of the United States was George Washington. He was unanimously elected to the position in 1789...
GA: narr narr narr narr narr narr narr narr narr narr narr narr narr narr narr narr ....
GA+RT: The first president of the United States was George Washington. narrated by narrative narrator narrative narrator narrative...
IDK+RT: The first president of the United States was George Washington.
DPO+RT: I don't know that.

Figure 5.7: Sampled response to questions in three subsets of TOFU. Yellow: questions; Green: true answer or desired answers; Red: undesired answers.

NPO improved diversity and readability. LLMs unlearned via GA-based methods tend to output repeated words or gibberish sentences with unreasonably low diversity (Yao et al., 2024b). Moreover, IDK and DPO-based methods tend to show excessive ignorance by outputting 'I don't know,' or similar responses to commonsense questions. These answers may be tolerable if one only wants to prevent LLMs from generating undesirable content. Still, they will definitely be unsatisfactory under the stronger goal of approximate unlearning, which aims to mimic the retrained model. We show in Figure 5.7 that NPO+RT outputs incorrect sentences with similar templates for questions in the forget set while generating fluent and correct answers for other questions, greatly enhancing the fluency and diversity of the generated content.

The role of retain loss. Maini et al. (2024) demonstrated that methods incorporating a retain set outperform those that solely optimize a loss function based on the forget set. To further investigate the role of retain loss beyond Maini et al., 2024, we evaluate NPO+RT with the weights of the retain loss varying from 0 to 5 (Figure 5.8). While it is natural that adding retain loss improves the model utility, we are surprised that the forget quality also grows.

Specifically, the forget quality increases as the weight of the retain loss grows from 0 to 2. We conjecture that the retain loss term helps the model preserve answer templates and linguistic structures, while the NPO term forces the model to forget some specific facts. Combining these two effects pushes the model to approximate the retrained model by generating outputs with similar templates but incorrect entities. However, further increasing the weight of the retain loss (e.g., from 2 to 5, in Figure 5.8) leads to a drop in the forget quality, possibly due to the diminished scale of the NPO term. Notably, in our experiments, the retain loss plays a more significant role when we target forgetting a larger fraction of the data (See the middle and right panels of Figure 5.6).

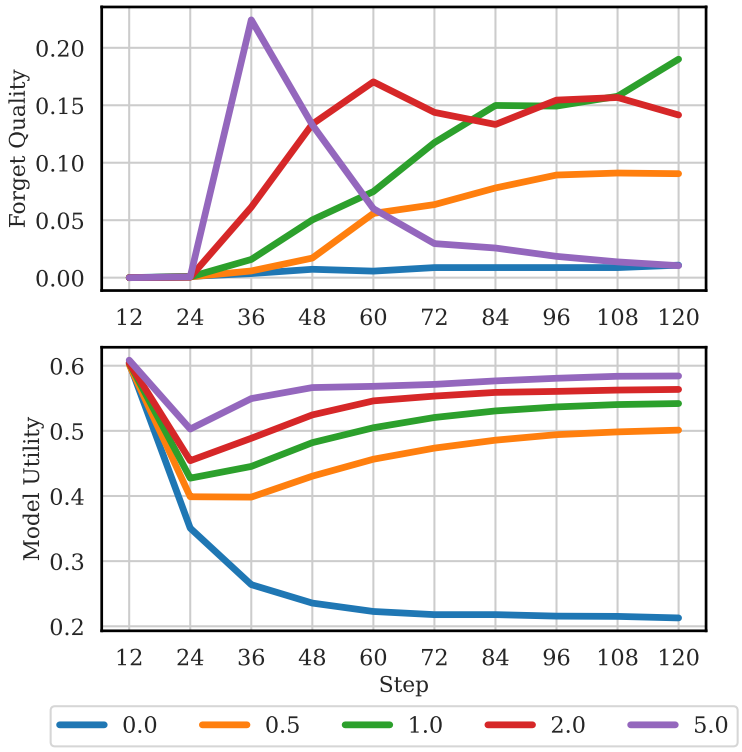

Figure 5.8: The evolution of the forget quality and model utility when we tune the weights of the NPO term and the retain loss term in NPO+RT. The experiments are performed on Forget10. We observe that when we increase the weight for the retain loss term, the model utility increases monotonically while the forget quality initially improves but then starts to deteriorate. We remark that altering the weights for loss components does not affect the effective learning rate since, in practice, AdamW is scale-invariant.

Forget KL: The larger, the better? ✖ ✖ ✖ We also examine the Forget KL during the unlearning process in the TOFU dataset. We first observed that while GA and GA+RT tend to induce an explosively large Forget KL along the unlearning process, the NPO-based

approaches induce a much slower growth of Forget KL (Figure 5.9). It stabilizes at a moderate level even after several epochs. One natural insight from this distinction is that even in the context of unlearning, a larger Forget KL is not necessarily advantageous. Rather, a moderate and stable Forget KL is preferable, which ensures the unlearned models generate fluent outputs with reasonable linguistic structures but incorrect content. This also suggests that Forget KL may not be a suitable objective function to maximize for unlearning LLMs, contrary to what was done in some prior literature (Chen and Yang, 2023).

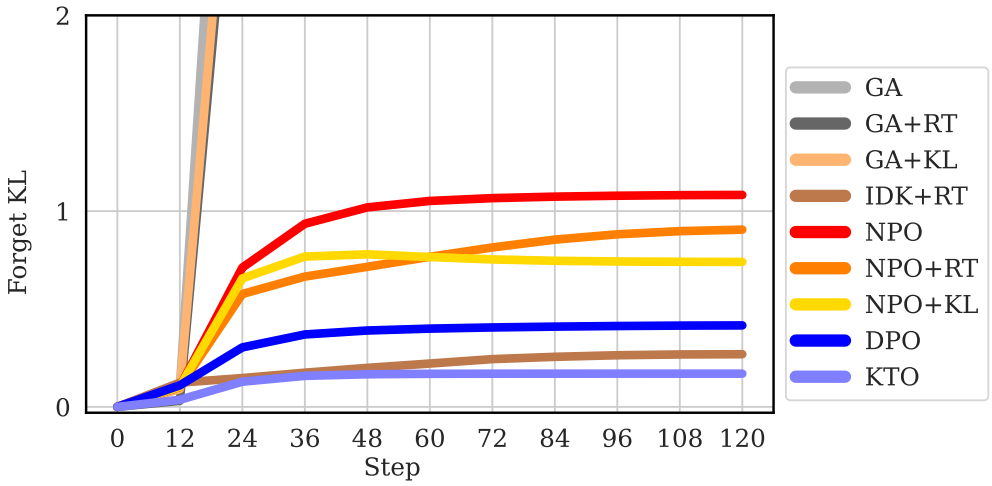

Figure 5.9: The evolution of the Forget KL during the unlearning process on the Forget10 task in TOFU data. Note that the KL term in GA+KL is the divergence on the retain set, not the forget set. More experimental details are included in Section 5.14.

Forgetting beyond 10% of TOFU

Forgetting 20%, 30% and 50% of TOFU. Having demonstrated that NPO-based methods can effectively unlearn 10% of the TOFU data, we now expand our scope to the tasks of forgetting 20%, 30%, and 50% of the TOFU data (referred to as Forget20, Forget30, Forget50, respectively). Details about the extended dataset are deferred to Section 5.14. We show in Section 5.14 that NPO+RT is the sole method to exhibit meaningful forget quality (a p-value above 0.05) in Forget20 and Forget30. Even in Forget50, where the vanilla NPO+RT achieves a forget quality around 10^{-3}, it still significantly outperforms other methods.

Pushing towards the limit: forgetting 50% - 90% of TOFU. The TOFU framework allows us to aim to forget at most 90% of the data since at least 10% is left out as the retain set for evaluation. We thus ask whether there are methods that could effectively forget 50%-90% of the TOFU data. We tuned the componential weights for NPO+RT and found

that with proper weights, NPO+RT easily attains a forget quality exceeding 0.05 and model utility above 0.55 on Forget50 and Forget90, as reported in Figure 5.10.

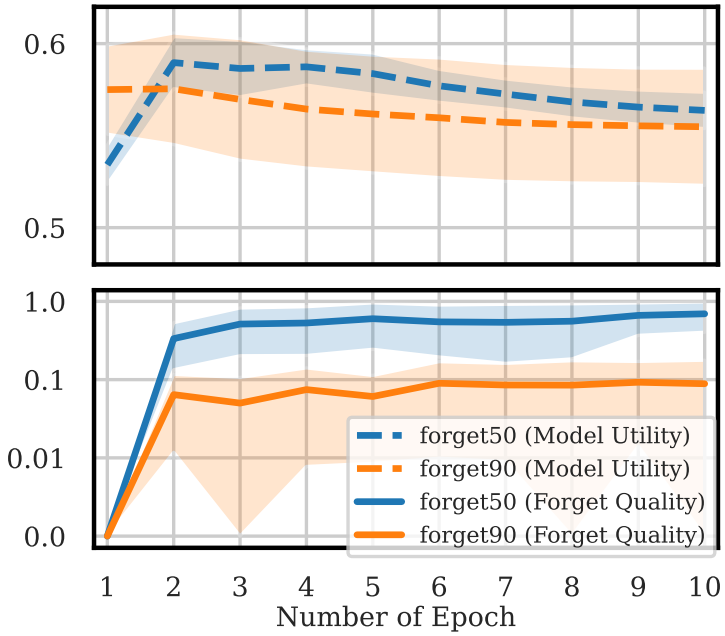

Figure 5.10: Evolution of forget quality and model utility on Forget50 and Forget90 for NPO+RT with proper component weights between loss terms. We tune the coefficient of the retain loss term and keep a unit coefficient for the NPO term. For Forget50, we set the coefficient of the retain loss term to be 5.0 while in Forget90, we set 12.0.

5.7 Discussion and Conclusion

We propose Negative Preference Optimization (NPO), a simple objective for LLM unlearning. NPO takes steps towards addressing the catastrophic collapse issue in the gradient ascent method. We show that unlearning methods based on NPO objective achieve state-of-the-art performance on LLM unlearning, and achieve the first effective unlearning result on forgetting a high percentage of the training data. We believe this chapter opens up many exciting directions for future work, such as testing NPO on more datasets or harder scenarios (such as with adversarial prompts). It may also be of interest to generalize the algorithm principle of NPO (preference optimization with negative examples only) to other problems beyond unlearning.

5.8 Appendix: Proof of Proposition 5.4.1

Adopt the shorthand $\mathsf{R}_i = \log \frac{\pi_\theta(y_i|x_i)}{\pi_{\mathrm{ref}}(y_i|x_i)}$. For any $i \in [n_\mathrm{f}]$, we have

$$\begin{aligned}
\lim_{\beta\to 0} -\frac{2}{\beta} \cdot \log \sigma\left(-\beta \cdot \log \frac{\pi_\theta(y_i \mid x_i)}{\pi_{\mathrm{ref}}(y_i \mid x_i)} \right) - \frac{2}{\beta} \log 2 &= \lim_{\beta\to 0} -\frac{2}{\beta} \cdot \log \sigma(-\beta \mathsf{R}_i) - \frac{2}{\beta} \log 2 \\
&= \lim_{\beta\to 0} \frac{2}{\beta} \cdot \log \left(\frac{1+\exp(\beta \mathsf{R}_i)}{2} \right) \\
&= \lim_{\beta\to 0} \frac{2}{\beta} \cdot \log \left(1 + \frac{\exp(\beta \mathsf{R}_i) - 1}{2} \right) \\
&= \lim_{\beta\to 0} \frac{2}{\beta} \cdot \frac{\exp(\beta \mathsf{R}_i) - 1}{2} \\
&= \mathsf{R}_i.
\end{aligned}$$

Averaging over all $i \in [n_\mathrm{f}]$ and noting that $\sum_{i=1}^{n_\mathrm{f}} \mathsf{R}_i / n_\mathrm{f} = \mathcal{L}_{\mathrm{GA}}(\theta) - \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\log \pi_{\mathrm{ref}}(y_i \mid x_i)]$ yields the first part of Propostion 5.4.1.

For the second part of Propostion 5.4.1, by definition

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\mathrm{NPO},\beta}(\theta) &= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} -\frac{2}{\beta} \nabla_\theta \log \sigma(-\beta \mathsf{R}_i) \\
&= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \frac{2}{\beta} \nabla_\theta \log(1+\exp(\beta \mathsf{R}_i)) \\
&= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \frac{2}{\beta} \cdot \frac{\beta \exp(\beta \mathsf{R}_i)}{1+\exp(\beta \mathsf{R}_i)} \cdot \nabla_\theta \mathsf{R}_i \\
&= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \frac{2 \exp(\beta \mathsf{R}_i)}{1+\exp(\beta \mathsf{R}_i)} \cdot \nabla_\theta \mathsf{R}_i.
\end{aligned}$$

Therefore, it follows immediately that

$$\begin{aligned}
\lim_{\beta\to 0} \nabla_\theta \mathcal{L}_{\mathrm{NPO},\beta}(\theta) = \lim_{\beta\to 0} \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \frac{2 \exp(\beta \mathsf{R}_i)}{1+\exp(\beta \mathsf{R}_i)} \cdot \nabla_\theta \mathsf{R}_i &= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \nabla_\theta \mathsf{R}_i \\
= \frac{1}{n_\mathrm{f}} \sum_{i=1}^{n_\mathrm{f}} \nabla_\theta \log \pi_\theta(y_i \mid x_i) &= \nabla_\theta \mathcal{L}_{\mathrm{GA}}(\theta).
\end{aligned}$$

This completes the proof of Proposition 5.4.1.

5.9 Appendix: Proof of Theorem 5.4.2

Let $\eta_0 := \eta / n_\mathrm{f}$ denote the normalized learning rate. For $i, j \in [n_\mathrm{f}]$, define $\gamma_{i,j} := \langle x_i, x_j \rangle$ and $c_{\mathrm{init},i} := \langle \theta_{\mathrm{init}}, x_i \rangle$. Throughout the proof, we also use $C_k (k = 0, 1, 2, \ldots)$ to denote constants

that may depend on (B_θ, b_x, B_x) but not on $(t, n_{\mathrm{f}}, d, \beta)$. By the definition of the logistic model and some algebra, we have

$$\nabla_\theta \log \pi_\theta(y_i \mid x_i) = x_i(2y_i - 1)(1 - \pi_\theta(y_i \mid x_i)).$$

Therefore, the gradient of $\mathcal{L}_{\mathrm{GA}}$ and $\mathcal{L}_{\mathrm{NPO}}$ both lie in the span of $x_1, \ldots, x_{n_{\mathrm{f}}}$. Consequently, $\theta^{(t)}_{\mathrm{GA}} - \theta_{\mathrm{init}}$ and $\theta^{(t)}_{\mathrm{NPO}} - \theta_{\mathrm{init}}$ can be rewritten as follows:

$$\theta^{(t)}_{\mathrm{GA}} - \theta_{\mathrm{init}} = \sum_{i=1}^{n_{\mathrm{f}}} \alpha^{(t)}_{\mathrm{GA},i} \cdot x_i, \quad \theta^{(t)}_{\mathrm{NPO}} - \theta_{\mathrm{init}} = \sum_{i=1}^{n_{\mathrm{f}}} \alpha^{(t)}_{\mathrm{NPO},i} \cdot x_i,$$

where

$$\begin{aligned}
\alpha^{(t+1)}_{\mathrm{GA},i} &:= \alpha^{(t)}_{\mathrm{GA},i} - \eta_0(2y_i - 1)(1 - \pi_\theta(y_i \mid x_i)), \\
\alpha^{(t+1)}_{\mathrm{NPO},i} &:= \alpha^{(t)}_{\mathrm{NPO},i} - \eta_0(2y_i - 1)(1 - \pi_\theta(y_i \mid x_i)) \cdot \mathsf{W}^{(t)}_i,
\end{aligned}$$

and

$$\mathsf{W}^{(t)}_i = 2\pi^{\beta}_{\theta^{(t)}_{\mathrm{NPO}}}(y_i \mid x_i) \Big/ [\pi^{\beta}_{\theta^{(t)}_{\mathrm{NPO}}}(y_i \mid x_i) + \pi^{\beta}_{\theta_{\mathrm{init}}}(y_i \mid x_i)].$$

We also define $\alpha^{(0)}_{\star,i} = 0$ and adopt the shorthand notations $\alpha^{(t)}_\star = (\alpha^{(t)}_{1,\star}, \ldots, \alpha^{(t)}_{n_{\mathrm{f}},\star})$ for $\star \in \{\mathrm{GA}, \mathrm{NPO}\}$ and $\gamma_i = (\gamma_{i,1}, \ldots, \gamma_{i,n_{\mathrm{f}}})$.

For $\star \in \{\mathrm{GA}, \mathrm{NPO}\}$, we have

$$\begin{aligned}
\pi_{\theta^{(t)}_\star}(y_i \mid x_i) &= \frac{1}{1 + \exp\left((1 - 2y_i)\left\langle \mathbf{x}_i, \theta^{(t)}_\star \right\rangle\right)} \\
&= \frac{1}{1 + \exp\left((1 - 2y_i)c_{\mathrm{init},i} + (1 - 2y_i)\left\langle \mathbf{x}_i, \sum_{j=1}^{n_{\mathrm{f}}} \alpha^{(t)}_\star x_j \right\rangle\right)} \\
&= \frac{1}{1 + \exp\left((1 - 2y_i)c_{\mathrm{init},i} + (1 - 2y_i)\left\langle \alpha^{(t)}_\star, \gamma_i \right\rangle\right)} =: \mathrm{pred}_i\Big(\left\langle \alpha^{(t)}_\star, \gamma_i \right\rangle\Big),
\end{aligned}$$

where we denote the dependence on $(c_{\mathrm{init},i}, y_i)$ implicitly using pred_i for notational simplicity.

Therefore, letting $b^{(t)}_{\star,i} := \left\langle \alpha^{(t)}_\star, \gamma_i \right\rangle$ for $\star \in \{\mathrm{GA}, \mathrm{NPO}\}$ and combining all $i \in [n_{\mathrm{f}}]$, we obtain

$$b^{(t+1)}_{\mathrm{GA}} := b^{(t)}_{\mathrm{GA}} - \eta_0 \cdot \Gamma \Delta^{(t)}_{\mathrm{GA}}, \quad b^{(t+1)}_{\mathrm{NPO}} := b^{(t)}_{\mathrm{NPO}} - \eta_0 \cdot \Gamma \mathrm{diag}\Big\{\Delta^{(t)}_{\mathrm{NPO}}\Big\}\mathsf{W}^{(t)},$$

where

$$\begin{aligned}
&\Gamma := (\gamma_{i,j})_{1 \le i,j \le n_{\mathrm{f}}}, \\
&\Delta^{(t)}_\star := \left(\Delta^{(t)}_{\star,1}, \ldots, \Delta^{(t)}_{\star,n_{\mathrm{f}}}\right)^\top, \quad \Delta^{(t)}_{\star,i} := (2y_i - 1)(1 - \mathrm{pred}_i(b^{(t)}_{\star,i})), \quad \text{for } i \in [n_{\mathrm{f}}], \star \in \{\mathrm{GA}, \mathrm{NPO}\}, \\
&\mathsf{W}^{(t)} := (\mathsf{W}^{(t)}_1, \ldots, \mathsf{W}^{(t)}_{n_{\mathrm{f}}})^\top, \quad \mathsf{W}^{(t)}_i = \mathsf{W}_i\Big(b^{(t)}_{\mathrm{NPO}}\Big) := \pi^{\beta}_{\theta^{(t)}_{\mathrm{NPO}}}(y_i \mid x_i) \Big/ [\pi^{\beta}_{\theta^{(t)}_{\mathrm{NPO}}}(y_i \mid x_i) + \pi^{\beta}_{\theta_{\mathrm{init}}}(y_i \mid x_i)].
\end{aligned}$$

Again, we hide here the dependence on $(\beta, c_{\text{init},i}, y_i)$ in W_i for simplicity. We now claim the following results of which the proofs are deferred to Section 5.10 and 5.11.

Lemma 5.9.1 (GA converges to infinity linearly). *Under the assumptions in Theorem 5.4.2 and the notations in the proof of Theorem 5.4.2, there exist some (B_θ, b_x, B_x)-dependent constants $C_0, C_1, C_2 > 0$ such that the* GA *iterations $\{b_{\text{GA}}^{(t)}\}_{t=1}^{\infty}$ satisfy*

$$C_1\eta_0 t \le b_{\text{GA},i}^{(t)} \le C_2\eta_0 t, \text{ when } y_i = 0,$$
$$-C_2\eta_0 t \le b_{\text{GA},i}^{(t)} \le -C_1\eta_0 t, \text{ when } y_i = 1.$$

for all $i \in [n_{\text{f}}]$ and $t \ge 1$ when $\max_{i\neq j} |\gamma_{i,j}| \le C_0/n_{\text{f}}$.

Lemma 5.9.2 (NPO converges to infinity exponentially slow). *Under the assumptions in Theorem 5.4.2 and the notations in the proof of Theorem 5.4.2, there exist some (B_θ, b_x, B_x)-dependent constants $C_0, C_b > 0, C_a \in (0,1)$ such that the* NPO *iterations $\{b_{\text{NPO}}^{(t)}\}_{t=1}^{\infty}$ satisfy*

$$b_{\text{NPO},i}^{(t)} \in \left[-\frac{1}{\beta}\log(C_b\beta\eta_0 t + 1), -\frac{1}{\beta}\log(C_a\beta\eta_0 t + 1)\right] \text{ when } y_i = 1,$$
$$b_{\text{NPO},i}^{(t)} \in \left[\frac{1}{\beta}\log(C_a\beta\eta_0 t + 1), \frac{1}{\beta}\log(C_b\beta\eta_0 t + 1)\right] \text{ when } y_i = 0.$$

for any $\beta \in (0, 1]$, for all $i \in [n_{\text{f}}]$ and $t \ge 1$ when $\max_{i\neq j} |\gamma_{i,j}| \le C_0/n_{\text{f}}$.

Combining Lemma 5.9.1 and 5.9.2 and noting

$$\|\theta_\star^{(t)} - \theta_{\text{init}}\|_{X^\top X} = \sqrt{(\theta_\star^{(t)} - \theta_{\text{init}})^\top X^\top X(\theta_\star^{(t)} - \theta_{\text{init}})} = \sqrt{\alpha_\star^{(t)\top} XX^\top XX^\top \alpha_\star^{(t)}}$$
$$= \sqrt{b_\star^{(t)\top}(XX^\top)^{-1}XX^\top XX^\top(XX^\top)^{-1}b_\star^{(t)}} = \|b_\star^{(t)}\|_2$$

for $\star \in \{\text{GA}, \text{NPO}\}$ completes the proof.

5.10 Appendix: Proof of Lemma 5.9.1

We prove Lemma 5.9.1 by induction.

Case 1: $t = 1$ When $t = 1$, since $|c_{\text{init}}| \le \|x_i\|_2 \cdot \|\theta_{\text{init}}\|_2 \le B_x B_\theta$, it follows from the definition of $\text{pred}_i(\cdot)$ that

$$C_3 \le \Delta_{\text{GA},i}^{(0)} \le 1 \text{ when } y_i = 1, \quad -1 \le \Delta_{\text{GA},i}^{(0)} \le -C_4 \text{ when } y_i = 0$$

for all $i \in [n_{\text{f}}]$ for some constants $C_3, C_4 \in (0,1)$ depending only on (B_θ, b_x, B_x). Note that there exists a constant $C_0 > 0$ depending on C_3, C_4, b_x such that

$$\left|\sum_{j\neq i} \gamma_{i,j}\Delta_{\text{GA},j}^{(0)}\right| \le \frac{\gamma_{i,i}}{2}\left|\Delta_{\text{GA},i}^{(0)}\right|$$

for all $i \in [n_{\mathrm{f}}]$ when $\max_{i \neq j} |\gamma_{i,j}| \leq C_0/n_{\mathrm{f}}$. It follows that

$$-\eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(0)} \in [-\frac{3}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{GA},i}^{(0)}|, -\frac{1}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{GA},i}^{(0)}|] \in [-C_2\eta_0, -C_1\eta_0] \text{ when } y_i = 1,$$
$$-\eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(0)} \in [\frac{1}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{GA},i}^{(0)}|, \frac{3}{2}\gamma_{i,i}|\eta_0 \Delta_{\mathrm{GA},i}^{(0)}|] \in [C_1\eta_0, C_2\eta_0] \text{ when } y_i = 0$$

for some (B_θ, b_x, B_x)-dependent constants $C_1, C_2 > 0$. Therefore,

$$b_{\mathrm{GA},i}^{(1)} = b_{\mathrm{GA},i}^{(0)} - \eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(0)} \in [-C_2\eta_0, -C_1\eta_0] \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(1)} = b_{\mathrm{GA},i}^{(0)} - \eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(0)} \in [C_1\eta_0, C_2\eta_0] \text{ when } y_i = 0.$$

As a consequence,

$$b_{\mathrm{GA},i}^{(1)} \leq b_{\mathrm{GA},i}^{(0)} = 0, \quad \Delta_{\mathrm{GA},i}^{(1)} \geq \Delta_{\mathrm{GA},i}^{(0)} \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(1)} \geq b_{\mathrm{GA},i}^{(0)} = 0, \quad \Delta_{\mathrm{GA},i}^{(1)} \leq \Delta_{\mathrm{GA},i}^{(0)} \text{ when } y_i = 0.$$

Case 2: $t = K + 1$ Now, suppose we have

$$b_{\mathrm{GA},i}^{(t)} \in [-C_2\eta_0 t, -C_1\eta_0 t] \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(t)} \in [C_1\eta_0 t, C_2\eta_0 t] \text{ when } y_i = 0$$

for $t \in [K]$ and

$$b_{\mathrm{GA},i}^{(K)} \leq \ldots \leq b_{\mathrm{GA},i}^{(0)} = 0, \quad \Delta_{\mathrm{GA},i}^{(K)} \geq \ldots \geq \Delta_{\mathrm{GA},i}^{(0)} \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(K)} \geq \ldots \geq b_{\mathrm{GA},i}^{(0)} = 0, \quad \Delta_{\mathrm{GA},i}^{(K)} \leq \ldots \leq \Delta_{\mathrm{GA},i}^{(0)} \text{ when } y_i = 0.$$

By the monotonicity of $\Delta_{\mathrm{GA},i}^{(t)}$, we have

$$C_3 \leq \Delta_{\mathrm{GA},i}^{(K)} \leq 1 \text{ when } y_i = 1, \quad -1 \leq \Delta_{\mathrm{GA},i}^{(K)} \leq -C_4 \text{ when } y_i = 0$$

for all $i \in [n_{\mathrm{f}}]$. Therefore, following similar arguments as in the $t = 1$ case, we have

$$-\eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(K)} \in [-C_2\eta_0, -C_1\eta_0] \text{ when } y_i = 1,$$
$$-\eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(K)} \in [C_1\eta_0, C_2\eta_0] \text{ when } y_i = 0.$$

Then it follows from the induction assumption that

$$b_{\mathrm{GA},i}^{(K+1)} = b_{\mathrm{GA},i}^{(K)} - \eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(K)} \in [-C_2\eta_0(K+1), -C_1\eta_0(K+1)] \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(K+1)} = b_{\mathrm{GA},i}^{(K)} - \eta_0 \cdot \gamma_i^\top \Delta_{\mathrm{GA}}^{(K)} \in [C_1\eta_0(K+1), C_2\eta_0(K+1)] \text{ when } y_i = 0,$$

and also

$$b_{\mathrm{GA},i}^{(K+1)} \leq b_{\mathrm{GA},i}^{(K)}, \quad \Delta_{\mathrm{GA},i}^{(K+1)} \geq \Delta_{\mathrm{GA},i}^{(K)} \text{ when } y_i = 1,$$
$$b_{\mathrm{GA},i}^{(K+1)} \geq b_{\mathrm{GA},i}^{(K)}, \quad \Delta_{\mathrm{GA},i}^{(K+1)} \leq \Delta_{\mathrm{GA},i}^{(K)} \text{ when } y_i = 0.$$

This concludes the induction step and therefore completes the proof.

5.11 Appendix: Proof of Lemma 5.9.2

We prove Lemma 5.9.2 by induction.

Our induction assumption is the following: there exist some constants $C_0 > 0, C_a \in (0,1), C_b > 0, C_1, C_2$ depending only on (B_θ, b_x, B_x) such that when $\max_{i \neq j} |\gamma_{i,j}| \leq C_0/n_{\mathrm{f}}$, for any $t \geq 1$ and any $\beta \in (0,1]$, we have

1.

$$b^{(t)}_{\mathrm{NPO},i} \leq \ldots \leq b^{(0)}_{\mathrm{NPO},i} = 0, \quad \Delta^{(t)}_{\mathrm{NPO},i} \geq \ldots \geq \Delta^{(0)}_{\mathrm{NPO},i} \quad \text{when } y_i = 1,$$
$$b^{(t)}_{\mathrm{NPO},i} \geq \ldots \geq b^{(0)}_{\mathrm{NPO},i} = 0, \quad \Delta^{(t)}_{\mathrm{NPO},i} \leq \ldots \leq \Delta^{(0)}_{\mathrm{NPO},i} \quad \text{when } y_i = 0.$$

2.

$$b^{(t)}_{\mathrm{NPO},i} \in \left[-\frac{1}{\beta}\log(C_b\beta\eta_0 t + 1), -\frac{1}{\beta}\log(C_a\beta\eta_0 t + 1) \right] \text{ when } y_i = 1, \tag{5.7}$$
$$b^{(t)}_{\mathrm{NPO},i} \in \left[\frac{1}{\beta}\log(C_a\beta\eta_0 t + 1), \frac{1}{\beta}\log(C_b\beta\eta_0 t + 1) \right] \text{ when } y_i = 0. \tag{5.8}$$

3.

$$b^{(t)}_{\mathrm{NPO},i} \in \left[-\frac{\log(\beta\eta_0 t + 1) + C_2}{\beta}, -\frac{\log(\beta\eta_0 t + 1) + C_1}{\beta} \right] \text{ when } y_i = 1, \tag{5.9}$$
$$b^{(t)}_{\mathrm{NPO},i} \in \left[\frac{\log(\beta\eta_0 t + 1) + C_1}{\beta}, \frac{\log(\beta\eta_0 t + 1) + C_2}{\beta} \right] \text{ when } y_i = 0. \tag{5.10}$$

Lemma 5.9.2 follows immediately from the second part of the induction assumption.

In the following, we first specify the parameter-dependent constants C_0, C_1, C_2, C_a, C_b in the $t = 1$ case and prove the induction assumption when $k = 1$. Then given the induction assumption holds when $t \leq K$, we prove that it holds when $t = K + 1$ as well. In the arguments below, we always assume a fixed $\beta \in (0,1]$.

Case 1: $t = 1$ When $t = 1$, since $|c_{\mathrm{init}}| \leq \|x_i\|_2 \cdot \|\theta_{\mathrm{init}}\|_2 \leq B_x B_\theta$, it follows from the definition of $\mathrm{pred}_i(\cdot)$ that

$$C_3 \leq \Delta^{(0)}_{\mathrm{NPO},i} \leq 1 \text{ when } y_i = 1, \quad -1 \leq \Delta^{(0)}_{\mathrm{NPO},i} \leq -C_4 \text{ when } y_i = 0 \tag{5.11}$$

for all $i \in [n_{\mathrm{f}}]$ for some constants $C_3, C_4 \in (0,1)$ depending only on (B_θ, b_x, B_x). Moreover, we claim that

$$\mathsf{W}^{(t)}_i \in \left[C_5 \cdot \exp\left((2y_i - 1)\beta b^{(t)}_{\mathrm{NPO},i} \right), C_6 \cdot \exp\left((2y_i - 1)\beta b^{(t)}_{\mathrm{NPO},i} \right) \right] \tag{5.12}$$

for any $\beta \in (0,1]$, all $i \in [n_{\mathrm{f}}]$ and t such that $\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)}) \le \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})$ for some (B_θ, b_x, B_x)-dependent constants $C_5, C_6 > 0$.

Now, suppose Eq. (5.9) and (5.10) hold for some (B_θ, b_x, B_x)-dependent constants C_1, C_2 which we will specify later. Then, there exists a constant $C_0 > 0$ depending on $C_{1:6}, b_x$ such that

$$\left| \sum_{j \neq i} \gamma_{i,j} \Delta_{\mathrm{NPO},j}^{(0)} \mathsf{W}_j \right| \le \frac{\gamma_{i,i}}{2} \left| \Delta_{\mathrm{NPO},i}^{(0)} \mathsf{W}_i \right| \tag{5.13}$$

for all $i \in [n_{\mathrm{f}}]$ when $\max_{i \neq j} |\gamma_{i,j}| \le C_0/n_{\mathrm{f}}$. Furthermore, combining Eq. (5.11), (5.12), (5.13) gives

$$\begin{aligned}
-\eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(0)}\} \mathsf{W}^{(0)} &\in \left[-\frac{3}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{NPO},i}^{(0)}| \mathsf{W}_i^{(0)}, -\frac{1}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{NPO},i}^{(0)}| \mathsf{W}_i^{(0)} \right] \\
&\in [-C_8\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(0)}), -C_7\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(0)})] \quad \text{when } y_i = 1, \\
-\eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(0)}\} \mathsf{W}^{(0)} &\in \left[\frac{1}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{NPO},i}^{(0)}| \mathsf{W}_i^{(0)}, \frac{3}{2}\gamma_{i,i}\eta_0 |\Delta_{\mathrm{NPO},i}^{(0)}| \mathsf{W}_i^{(0)} \right] \\
&\in [C_7\eta_0 \exp(-\beta b_{\mathrm{NPO},i}^{(0)}), C_8\eta_0 \exp(-\beta b_{\mathrm{NPO},i}^{(0)})] \quad \text{when } y_i = 0,
\end{aligned}$$

if $\max_{i \neq j} |\gamma_{i,j}| \le C_0/n_{\mathrm{f}}$. Here $C_7, C_8 > 0$ are some (C_3, C_4, C_5, C_6)-dependent constants, and we pick $0 < C_7 < 1$ so that $C_7\beta < 1$ for any $\beta \in (0,1]$. As a consequence,

$$\begin{aligned}
b_{\mathrm{NPO},i}^{(1)} \le b_{\mathrm{NPO},i}^{(0)} = 0, \quad \Delta_{\mathrm{NPO},i}^{(1)} \ge \Delta_{\mathrm{NPO},i}^{(0)} \quad \text{when } y_i = 1, \\
b_{\mathrm{NPO},i}^{(1)} \ge b_{\mathrm{NPO},i}^{(0)} = 0, \quad \Delta_{\mathrm{NPO},i}^{(1)} \le \Delta_{\mathrm{NPO},i}^{(0)} \quad \text{when } y_i = 0.
\end{aligned}$$

This concludes the proof of the first part of the induction assumption.

Now, we start to prove the second part of the induction assumption. For i such that $y_i = 1$, consider the ordinary differential equations

$$\begin{aligned}
b_l'(t) &= -C_8\eta_0(1 + \exp(C_8)) \cdot \exp(\beta b_l(t)), \qquad b_l(0) = 0; \\
b_u'(t) &= -C_7\eta_0 \cdot \exp(\beta b_u(t)), \qquad b_u(0) = 0.
\end{aligned}$$

It can be verified that the ODEs have closed-form solutions

$$\begin{aligned}
b_l(t) &= -\frac{1}{\beta} \log(\beta C_8 \eta_0 (1 + \exp(C_8)) t + 1), \\
b_u(t) &= -\frac{1}{\beta} \log(\beta C_7 \eta_0 t + 1).
\end{aligned}$$

Since

$$b_{\mathrm{NPO},i}^{(1)} = b_{\mathrm{NPO},i}^{(0)} - \eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(0)}\} \mathsf{W}^{(0)} \ge b_{\mathrm{NPO},i}^{(0)} - C_8\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(0)}) \ge b_{\mathrm{NPO},i}^{(0)} - C_8\eta_0,$$

it follows that for any point $b_{\mathrm{NPO},i}^{(\varepsilon)} = \varepsilon b_{\mathrm{NPO},i}^{(1)} + (1-\varepsilon) b_{\mathrm{NPO},i}^{(0)}$ with $\varepsilon \in [0,1]$

$$\begin{aligned} -C_8\eta_0(1+\exp(C_8\eta_0)) \cdot \exp(\beta b_{\mathrm{NPO},i}^{(\varepsilon)}) &\leq -C_8\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(0)}) \leq -C_7\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(0)}) \\ &\leq -C_7\eta_0 \cdot \exp(\beta b_{\mathrm{NPO},i}^{(\varepsilon)}). \end{aligned}$$

Therefore, we have by the comparison theorem for ODEs that

$$b_l(\varepsilon) \leq b_{\mathrm{NPO},i}^{(\varepsilon)} \leq b_u(\varepsilon)$$

for $\varepsilon \in [0,1]$. Setting

$$C_a := C_7, \qquad C_b := C_8(1+\exp(C_8))$$

concludes the second part of the induction assumption.

For the last part of the induction assumption, we first notice $\log(x+1) + \log(c) \leq \log(cx+1) \leq \log(x+1) + \log(c+1)$ (the left inequality holds only when $c \leq 1$), we have

$$-\frac{1}{\beta}[\log(\beta\eta_0 t + 1) + \log(1 + C_8(1+\exp(C_8)))] \leq b_l(t) \leq b_u(t) \leq -\frac{1}{\beta}[\log(\beta\eta_0 t + 1) + \log(C_7)],$$

where the last inequality uses $\beta C_7 < 1$. Therefore, we obtain

$$b_{\mathrm{NPO},i}^{(1)} \in \left[-\frac{\log(\beta\eta_0+1) + C_2}{\beta}, -\frac{\log(\beta\eta_0+1) + C_1}{\beta} \right],$$

where

$$C_1 := \log(C_7), \qquad C_2 := \log(1 + C_8(1+\exp(C_8)))$$

for i such that $y_i = 1$. Following the same arguments, similarly, we also have

$$b_{\mathrm{NPO},i}^{(1)} \in \left[\frac{\log(\beta\eta_0+1) + C_1}{\beta}, \frac{\log(\beta\eta_0+1) + C_2}{\beta} \right],$$

for i such that $y_i = 0$. This concludes the last part of the induction assumption.

Case 2: $t = K+1$ Suppose the induction assumption holds for $t \in [K]$, we now show that the induction assumption holds for $t = K+1$ as well. Following the proof of $t = 1$ case and using the monotonicity property of $\{\Delta_{\mathrm{NPO},i}^{(t)}\}_{t=0}^{K}$, we have

$$C_3 \leq \Delta_{\mathrm{NPO},i}^{(K)} \leq 1 \text{ when } y_i = 1, \quad -1 \leq \Delta_{\mathrm{NPO},i}^{(K)} \leq -C_4 \text{ when } y_i = 0$$

for all $i \in [n_{\mathrm{f}}]$. Since $\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(K)}) \leq \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})$ by the definition of pred_i and the monotonicity of $\{b_{\mathrm{NPO},i}^{(t)}\}_{t=0}^{K}$, it follows from Claim (5.12) that

$$\mathsf{W}_i^{(K)} \in \left[C_5 \cdot \exp\left((2y_i - 1)\beta b_{\mathrm{NPO},i}^{(K)}\right), C_6 \cdot \exp\left((2y_i - 1)\beta b_{\mathrm{NPO},i}^{(K)}\right) \right].$$

Putting the last two displays together and following the same argument as the $t = 1$ case, we find that

$$-\eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(K)}\}\mathsf{W}^{(K)} \in [-C_8\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(K)}), -C_7\eta_0 \exp(\beta b_{\mathrm{NPO},i}^{(K)})] \quad \text{when } y_i = 1,$$
$$-\eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(K)}\}\mathsf{W}^{(K)} \in [C_7\eta_0 \exp(-\beta b_{\mathrm{NPO},i}^{(K)}), C_8\eta_0 \exp(-\beta b_{\mathrm{NPO},i}^{(K)})] \quad \text{when } y_i = 0.$$

The first part of the induction assumption (for $t = K + 1$) follows immediately as the sign of the gradient updates $-\eta_0 \cdot \gamma_i^\top \mathrm{diag}\{\Delta_{\mathrm{NPO}}^{(K)}\}\mathsf{W}^{(K)}$ are determined as above.

Note that $b_l(K) \le b_{\mathrm{NPO},i}^{(K)} \le b_u(K)$ by the induction assumption. Similarly, using the comparison theorem of ODEs, we obtain

$$b_l(K + \varepsilon) \le b_{\mathrm{NPO},i}^{(K+\varepsilon)} \le b_u(K + \varepsilon)$$

for $\varepsilon \in [0, 1]$ and $b_{\mathrm{NPO},i}^{(K+\varepsilon)} := \varepsilon b_{\mathrm{NPO},i}^{(K+1)} + (1 - \varepsilon)b_{\mathrm{NPO},i}^{(K)}$. Choosing $\varepsilon = 1$ gives the second part of the induction assumption for $t = K + 1$. The last part of the induction assumption for $t = K + 1$ follows from the same algebra as in the $t = 1$ case.

Proof of Claim (5.12) By definition

$$\mathsf{W}_i^{(t)} = \frac{2\pi^\beta_{\theta_{\mathrm{NPO}}^{(t)}}(y_i \mid x_i)}{\pi^\beta_{\theta_{\mathrm{NPO}}^{(t)}}(y_i \mid x_i) + \pi^\beta_{\mathrm{ref}}(y_i \mid x_i)} = \frac{2\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta}{\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta + \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})^\beta}.$$

When $\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)}) \le \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})$, we have for any $\beta \in (0, 1]$, it holds that

$$\mathsf{W}_i^{(t)} \in \left[\frac{\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta}{\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})^\beta}, \frac{2\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta}{\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})^\beta}\right] \subseteq [C_l \cdot \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta, C_u \cdot \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)})^\beta] \quad (5.14)$$

for some constants $C_l, C_u > 0$ depending only on (B_θ, b_x, B_x). Note that $\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)}) \le \mathrm{pred}_i(b_{\mathrm{NPO},i}^{(0)})$ is equivalent to $(1 - 2y_i)b_{\mathrm{NPO},i}^{(t)} \ge 0$. Therefore, under this condition, we have

$$\begin{aligned}
\mathrm{pred}_i(b_{\mathrm{NPO},i}^{(t)}) &= \frac{1}{1 + \exp\left((1 - 2y_i)c_{\mathrm{init},i} + (1 - 2y_i)b_{\mathrm{NPO},i}^{(t)}\right)} \\
&= \frac{\exp\left((2y_i - 1)b_{\mathrm{NPO},i}^{(t)}\right)}{\exp\left((2y_i - 1)b_{\mathrm{NPO},i}^{(t)}\right) + \exp\left((1 - 2y_i)c_{\mathrm{init},i}\right)} \\
&\in \left[\frac{\exp\left((2y_i - 1)b_{\mathrm{NPO},i}^{(t)}\right)}{1 + \exp\left((1 - 2y_i)c_{\mathrm{init},i}\right)}, \frac{\exp\left((2y_i - 1)b_{\mathrm{NPO},i}^{(t)}\right)}{\exp\left((1 - 2y_i)c_{\mathrm{init},i}\right)}\right]. \qquad (5.15)
\end{aligned}$$

Putting Eq. (5.14) and (5.15) together and recalling that $|c_{\text{init},i}| \leq B_\theta B_x$, we obtain

$$\mathsf{W}_i^{(t)} \in \left[C_5 \cdot \exp\left((2y_i - 1)\beta b_{\text{NPO},i}^{(t)}\right), C_6 \cdot \exp\left((2y_i - 1)\beta b_{\text{NPO},i}^{(t)}\right)\right]$$

for any $\beta \in (0, 1]$ for some constants $C_5, C_6 > 0$ depending only on (B_θ, b_x, B_x). This concludes the proof of Claim (5.12).

5.12 Appendix: The Role of KL Divergence on the Forget Set

In this section, we report *Forget KL*, the KL divergence between the output distributions of the initial model and the unlearned model on the forget set. Mathematically, this is defined as $\mathbb{E}_{\mathcal{D}_{\text{FG}}}\text{KL}(\pi_{\text{ref}}(\cdot \mid \mathbf{x})||\pi_\theta(\cdot \mid \mathbf{x}))$. In experiments on the synthetic dataset and TOFU dataset, we observe that the models exhibit better unlearning performance when the forget KL is maintained at a moderate level. This suggests that explicitly maximizing the forget KL may not be an ideal objective for unlearning tasks.

Synthetic Experiment

We present the forget KL for the synthetic experiments in Figure 5.11 (a) and (b). Combining with Figure 5.4, we find that the unlearned models attain Pareto frontiers when the forget KL is suitably large—an excessively large forget KL (as in GA, GA + RT after 1200 steps) or an excessively small forget KL (as in IDK + RT) may deteriorate the unlearning performance.

5.13 Appendix: Experimental Details of the Synthetic Experiments

In this section, we discuss the experimental details of the synthetic experiments studied in Section 5.5.

Initial model and retrained model. We create the initial model π_{ref} and the retrained model π_{retr} via optimizing over θ using the cross-entropy loss on the entire dataset $\mathcal{D} = \mathcal{D}_{\text{FG}} \cup \mathcal{D}_{\text{RT}}$ and the retain dataset $\mathcal{D}_{\text{RT}}$, respectively. Concretely, initializing at $\theta = \mathbf{0}_{128}$, we run gradient descent for 20000 steps with the learning rate equals 0.05 to obtain the initial model π_{ref} and the retrained model π_{retr}.

Unlearning. During unlearning, starting from the initial model π_{ref}, we run gradient descent on each of the loss functions for 2000 steps with the learning rates selected via a grid search. We choose the learning rates so that the training remains stable within 2000

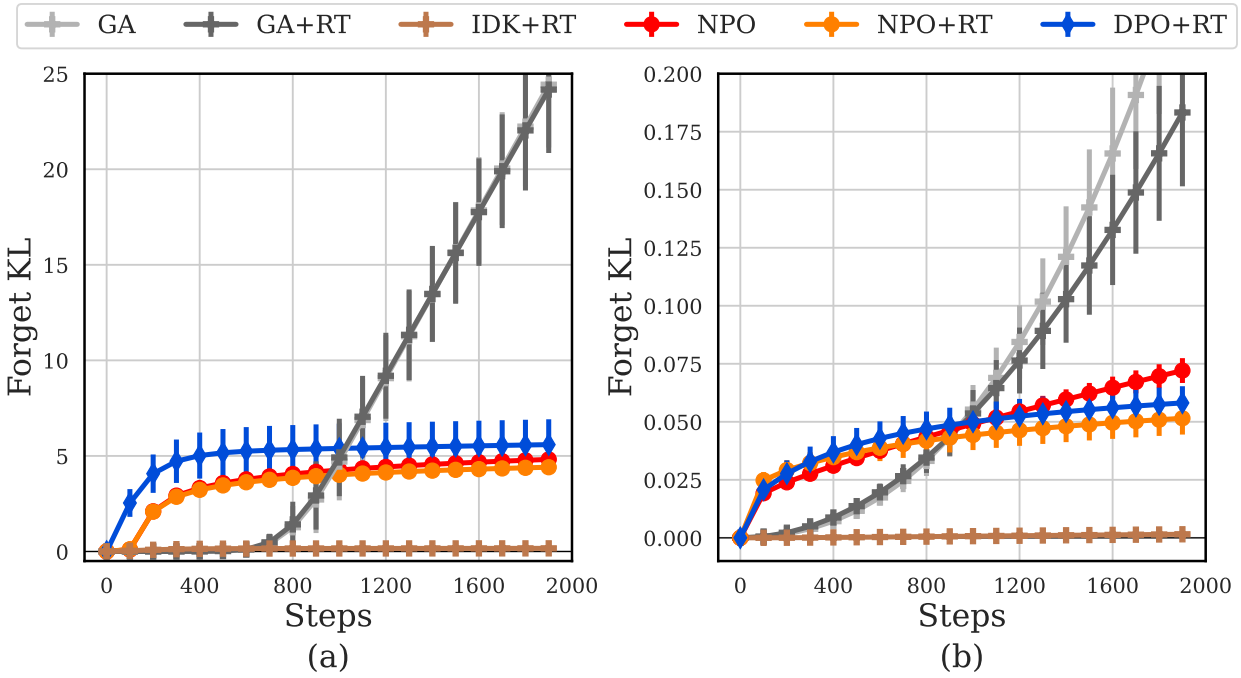

Figure 5.11: Forget KL versus optimization steps for all methods in the synthetic experiment. (a): $\alpha = 1$, (b): $\alpha = 0$. The errorbars denote ± 1 standard deviation over 5 runs.

steps. It should be noted that variations of the learning rates may affect the number of steps needed to reach the minimal forget distance (or retain distance) in Figure 5.4 (a1, a2, b). However, they are less likely to alter the Pareto curves shown in Figure 5.4 (c). A grid search is also conducted to select the optimal β for NPO (and DPO)-based methods. The choices of learning rate and β are summarized in Table 5.1.

Method	learning rate		β	
	$\alpha = 1$	$\alpha = 0$	$\alpha = 1$	$\alpha = 0$
GA,GA+RT, IDK+RT	5e-4	1e-4	N/A	N/A
NPO, NPO+RT	5e-3	5e-2	1	10
DPO+RT	5e-3	5e-2	0.1	5

Table 5.1: Values of learning rate and β for different methods when $\alpha = 1$ and $\alpha = 0$ in the synthetic experiments.

5.14 Appendix: Experiments on the TOFU Dataset

In this section, we provide details of the experiments on the TOFU dataset (Maini et al., 2024). We first present a detailed explanation of the metrics, the baseline methods and the hyperparameters in the experiments. Then, we provide the full results on different levels of the tasks.

Experiments Setup

Dataset

TOFU Dataset. We evaluate NPO-based methods and all baselines on the TOFU (Task of Fictitious Unlearning) dataset (Maini et al., 2024) designed for measuring the unlearning methods for LLMs. TOFU contains 200 fictitious author profiles, each consisting of 20 question-answering pairs generated by GPT-4 based on a set of predefined attributes. These profiles are fictitious and do not exist in the pre-training data, providing a controlled environment for studying unlearning LLMs. TOFU introduces three levels of unlearning tasks, each aiming at forgetting a subset of 2, 10, and 20 authors (comprising 1%, 5%, and 10% of the training data, respectively), referred to as the *forget set* $\mathcal{D}_{\mathrm{FG}}$, with a computational constraint that scales linearly with the size of the forget set. We refer to these tasks as Forget01, Forget05, and Forget10, respectively.

Dataset for Evaluation. In addition to the forget set, Maini et al. (2024) also introduced other datasets to measure the performance of the unlearned model. The *retain set* $\mathcal{D}_{\mathrm{RT}}$ is the part of the data that we do not hope the model to forget, which is, by definition, the complementary set of the forget set in the full dataset. To evaluate the model performance on the retain set, TOFU earmarks a subset of 400 question-answer pairs, accounting for 10% of the data, as an exclusive retain set that is never included in the forget set for any task. Moreover, to measure the general capacities of the unlearned models, two additional datasets are introduced: the Real Authors set and the Real World set. The Real Authors set includes question-answer pairs about authors in the real world and often deals with neighboring concepts entangled with those in the forget set. The Real World set contains commonsense knowledge about the real world and is designed to examine the general world knowledge of the unlearned model.

Dataset beyond Forget10. In scenarios where the targeted forget set exceeds 10% of the data (Forget20, Forget30, Forget50, and Forget90), we reorganize the original forget and retain sets within the TOFU dataset. To assess the Truth Ratio on an evaluation dataset, it is necessary to utilize the perturbed and paraphrased answers, which in TOFU were generated by properly prompting GPT4. To avoid any potential distribution shift from the newly crafted responses and their original forms, we evaluate the Truth Ratio using the publicly available data within TOFU. Consequently, even for tasks that are beyond forget10, we

continue to use the data from the standard forget10 subset to compute the Truth Ratio on the forget set. This serves as a reasonable proxy for evaluating the Truth Ratio on the full forget set.

Metrics

Model Utility. We measure the general capacities of the unlearned model using *Model Utility*, which aggregates multiple metrics across the retain set, the real-world set and the real-author set. Given a question-answer pair $x = [q, a]$, we compute the normalized conditional probability $\mathbb{P}(a \mid q)^{1/|a|}$, where $|\cdot|$ denotes the number of tokens in a certain sequence. This probability is then averaged over the retain set, the Real Authors set, and the Real World set each. We also compute the averaged ROUGE-L recall score (Lin, 2004) across these datasets, a metric that evaluates the accuracy of the model's response compared to the reference answers. Finally, we compute the averaged Truth Ratio on the three datasets above. The Truth Ratio defined in Maini et al., 2024 measures how likely the unlearned model will give a correct answer versus an incorrect one. More specifically, given a question q, Maini et al. (2024) generated a paraphrased (correct) answer $\tilde{a}$ via prompting GPT4. They then generated five perturbed answers with the exactly same templates but incorrect answers $\widehat{a}_i, i = 1, 2, 3, 4, 5$ in the same way. The Truth Ratio is defined by

$$R_{truth} := \frac{\frac{1}{5}\sum_{i=1}^{5} \mathbb{P}(\widehat{a}_i \mid q)^{1/|\widehat{a}_i|}}{\mathbb{P}(\widetilde{a} \mid q)^{1/|\widetilde{a}|}}. \tag{5.16}$$

The model utility is defined as the harmonic average of the nine metrics above (the probability, the ROUGE score, and the Truth Ratio on the retain set, the Real Authors set, and the Real World set).

Forget Quality. Forget quality assesses how well the unlearned model mimics the retrained model (defined as the model trained only on the retain set). This is a rigorous measurement as the ultimate goal for unlearning LLM is not only to stop generating the content related to the forget set but also to make the unlearned model indistinguishable from the retrained one. From a practical view of point, this requires the next-token probability given a prefix of the unlearned model to be as close as possible to that of the retrained model. In TOFU, they compute the Truth Ratio (defined in Eq. 5.16) on each question-answer pair from the forget set. Instead of simply averaging them, they test whether the distribution of the Truth Ratio computed from the unlearned and the retrained models are indistinguishable. More specifically, they perform the Kolmogorov-Smirnov (KS) test and compute the p-value of the test. A large p-value indicates that the two models are indistinguishable from the Truth Ratio. When the p-value is above 0.05, we say the forgetting is significant.

Baseline Methods

In this section, we introduce the baseline methods in our experiments.

GA-based Methods. Gradient Ascent (GA) is a key component in many LLM unlearning methods. Performing GA is equivalent to doing Gradient Descent (GD) on the negative cross-entropy loss function, which is denoted as $\mathcal{L}_{\mathrm{GA}}(\theta)$ defined in Eq. (5.1). Based on gradient ascent, a large class of unlearning methods performs gradient-based optimization on a linear combination of the GA loss $\mathcal{L}_{\mathrm{GA}}$ and several other loss functions that encourage unlearning (Jang et al., 2022; Yao et al., 2024b; Chen and Yang, 2023; Maini et al., 2024; Eldan and Russinovich, 2023). Such a loss function can be written as

$$\mathcal{L}(\theta) = c_{\mathrm{GA}}\mathcal{L}_{\mathrm{GA}}(\theta) + c_{\mathrm{FG}}\mathcal{L}_{\mathrm{FG}}(\theta) + c_{\mathrm{RT}}\mathcal{L}_{\mathrm{RT}}(\theta) - c_{\mathrm{FGKL}}\mathcal{K}_{\mathrm{FG}}(\theta) + c_{\mathrm{RTKL}}\mathcal{K}_{\mathrm{RT}}(\theta), \tag{5.17}$$

where $c_{GA}, c_{\mathrm{FG}}, c_{\mathrm{RT}}, c_{\mathrm{FGKL}}, c_{\mathrm{RTKL}}$ are non-negative weights.

Here, $\mathcal{L}_{\mathrm{FG}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\log(\pi_\theta(\tilde{\mathbf{y}}|\mathbf{x}))]$ is the Forget loss where $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{\mathrm{FG}}$ and $\tilde{\mathbf{y}} \neq \mathbf{y}$ is any response for prompt x which show some extent of ignorance towards the question $\mathbf{x}$. $\mathcal{L}_{\mathrm{RT}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{RT}}}[\log(\pi_\theta(\mathbf{y}|\mathbf{x}))]$ is the retain loss. $\mathcal{K}_{\mathrm{FG}}(\theta) = \mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}[\mathsf{D}(\pi_\theta(\cdot|\mathbf{x})||\pi_{\mathrm{ref}}(\cdot|\mathbf{x}))]$ is the expected KL divergence on the forget set. $\mathcal{K}_{\mathrm{RT}}(\theta) = \mathbb{E}_{\mathcal{D}_{\mathrm{RT}}}[\mathsf{D}(\pi_\theta(\cdot|\mathbf{x})||\pi_{\mathrm{ref}}(\cdot|\mathbf{x}))]$ is the expected KL divergence on the retain set. In our experiments, we use three GA-based methods reported in Maini et al., 2024, referred to as GA, GA+RT, GA+KL, which fall in this class of loss function. The weights in Eq. (5.17) are shown in Table 5.2.

Loss	c_{GA}	c_{FG}	c_{RT}	c_{FGKL}	c_{RTKL}
GA	1	0	0	0	0
GA+RT	1	0	1	0	0
GA+KL	1	0	0	0	1
IDK+RT	0	1	1	0	0

Table 5.2: The weights for different components in GA-based loss functions and IDK+RT loss.

IDK-based Methods ("I don't know"). Maini et al., 2024 proposed IDK+RT, which is a supervised loss function comprising of the retain loss and IDK loss term. The IDK loss term $\mathcal{L}_{\mathrm{FG}}$ is the averaged cross-entropy loss for question-answer pairs with questions $\mathbf{x}$ from the forget set $\mathcal{D}_{\mathrm{FG}}$ and answers $\mathbf{y}$ replaced by $\tilde{\mathbf{y}}$ = 'I don't know' or a similar sentence showing ignorance towards this question. IDK+RT does not involve GA loss, and in general, IDK+RT loss shows a higher stability than GA-based methods.

DPO-based Methods. We also tested the DPO method (Rafailov et al., 2024b) and its variants by adding either the retain loss or the KL divergence on the retain set. In the DPO loss, we take 'I don't know' or its variants as positive responses and the answers in the forget set as negative responses. We use $\beta = 0.1$ in all DPO-based experiments, which is commonly recognized as the optimal inverse temperature in most cases.

KTO-based Methods. We examine Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024), an alignment method with only non-paired preference data. The objective function of KTO is (we use a slightly different version than the original one as in Ethayarajh et al., 2024)

$$\mathcal{L}_{\mathrm{KTO}} := \frac{2}{\beta}\mathbb{E}_{\mathcal{D}_{\mathrm{FG}}}\left[-\log\sigma\left(z_{\mathrm{ref}} - \beta\log\frac{\pi_\theta(\mathbf{y}\mid\mathbf{x})}{\pi_{\mathrm{ref}}(\mathbf{y}\mid\mathbf{x})}\right)\right], \tag{5.18}$$

where $\beta > 0$ is the inverse-temperature, σ is the sigmoid function, and

$$z_{\mathrm{ref}} := \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{FG}}}\left[\beta\cdot\mathsf{D}\left(\pi_\theta(\cdot\mid\mathbf{x})||\pi_{\mathrm{ref}}(\cdot\mid\mathbf{x})\right)\right]. \tag{5.19}$$

Following (Ethayarajh et al., 2024), we estimate the KL term via averaged log probability ratio for questions in the forget set and answers in the "I don't know" set (as the unrelated outputs). We examine both KTO and KTO+RT in our experiments with $\beta = 0.1$.

Full Results

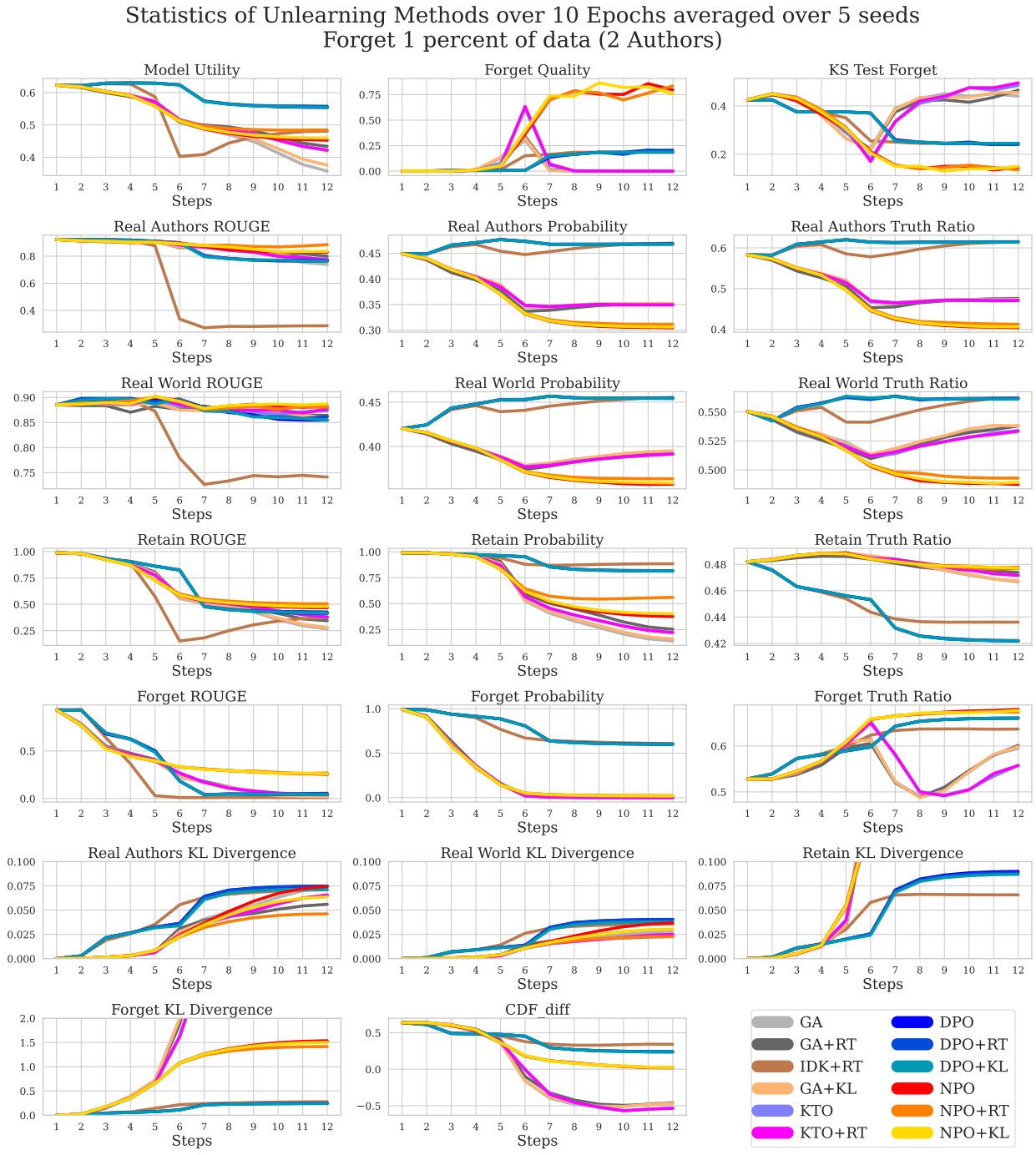

Figure 5.12: Statistics for NPO-based methods and baselines on the Forget01 task of TOFU.

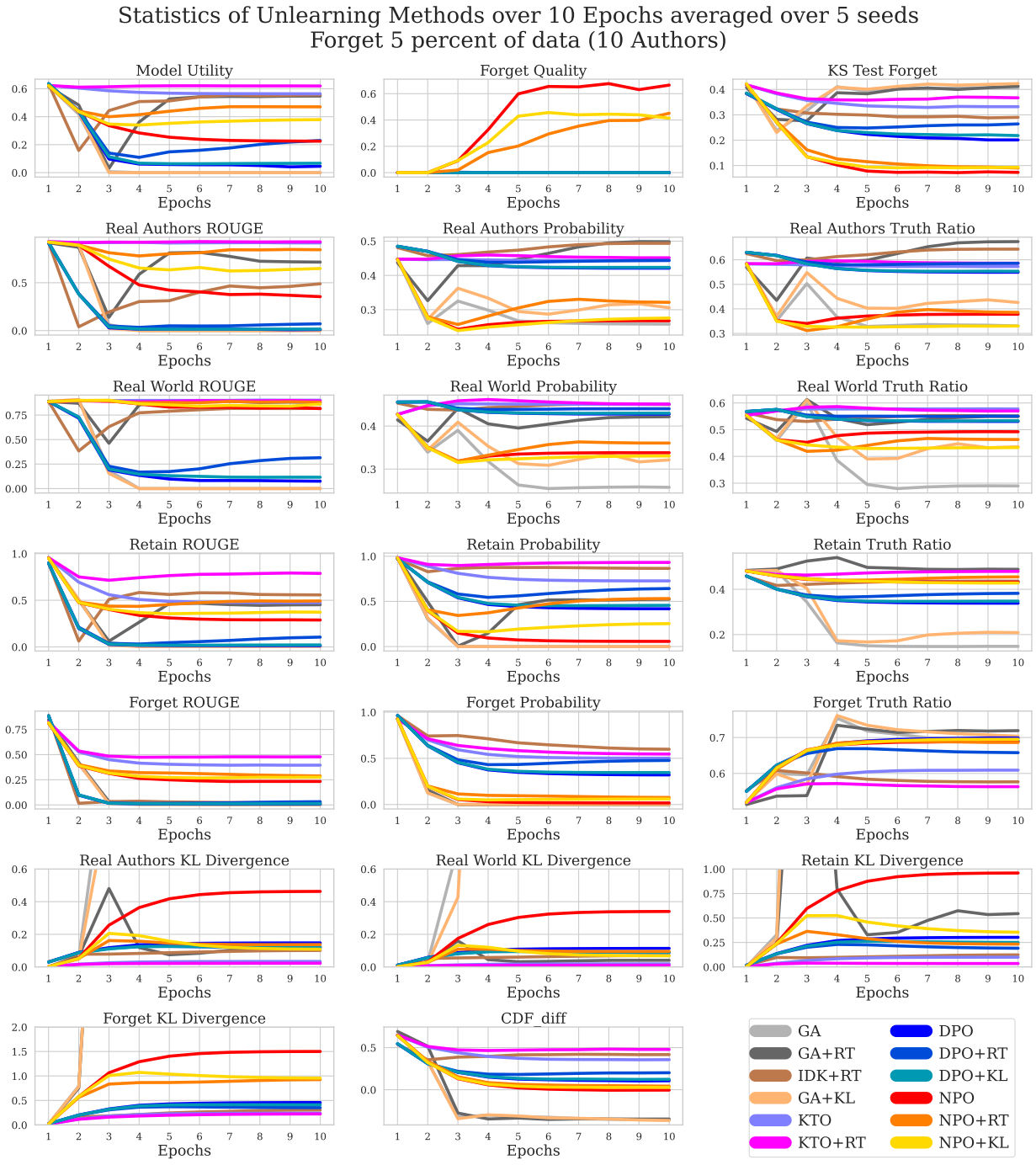

Figure 5.13: Statistics for NPO-based methods and baselines on the Forget05 task of TOFU.

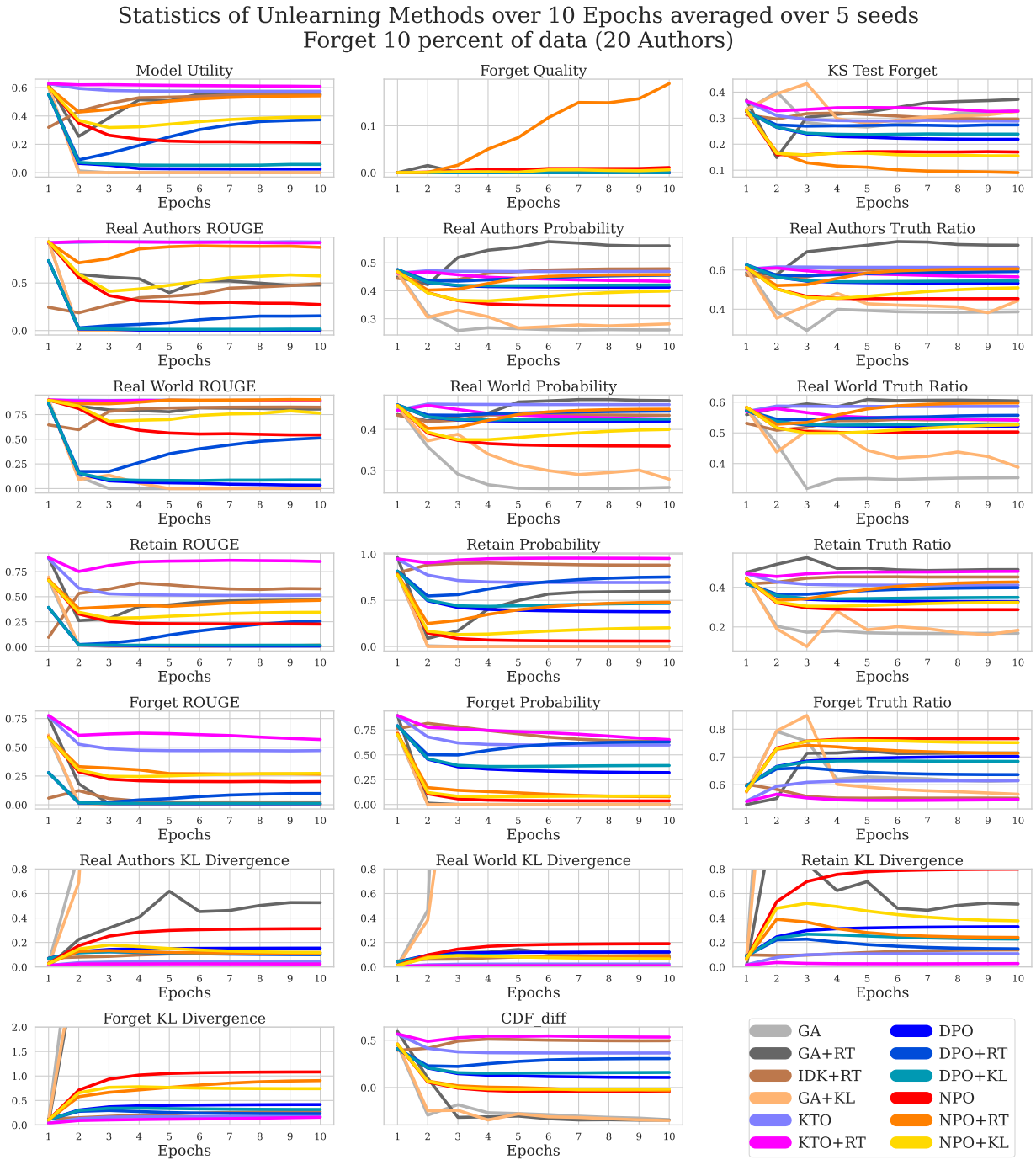

Figure 5.14: Statistics for NPO-based methods and baselines on the Forget10 task of TOFU.

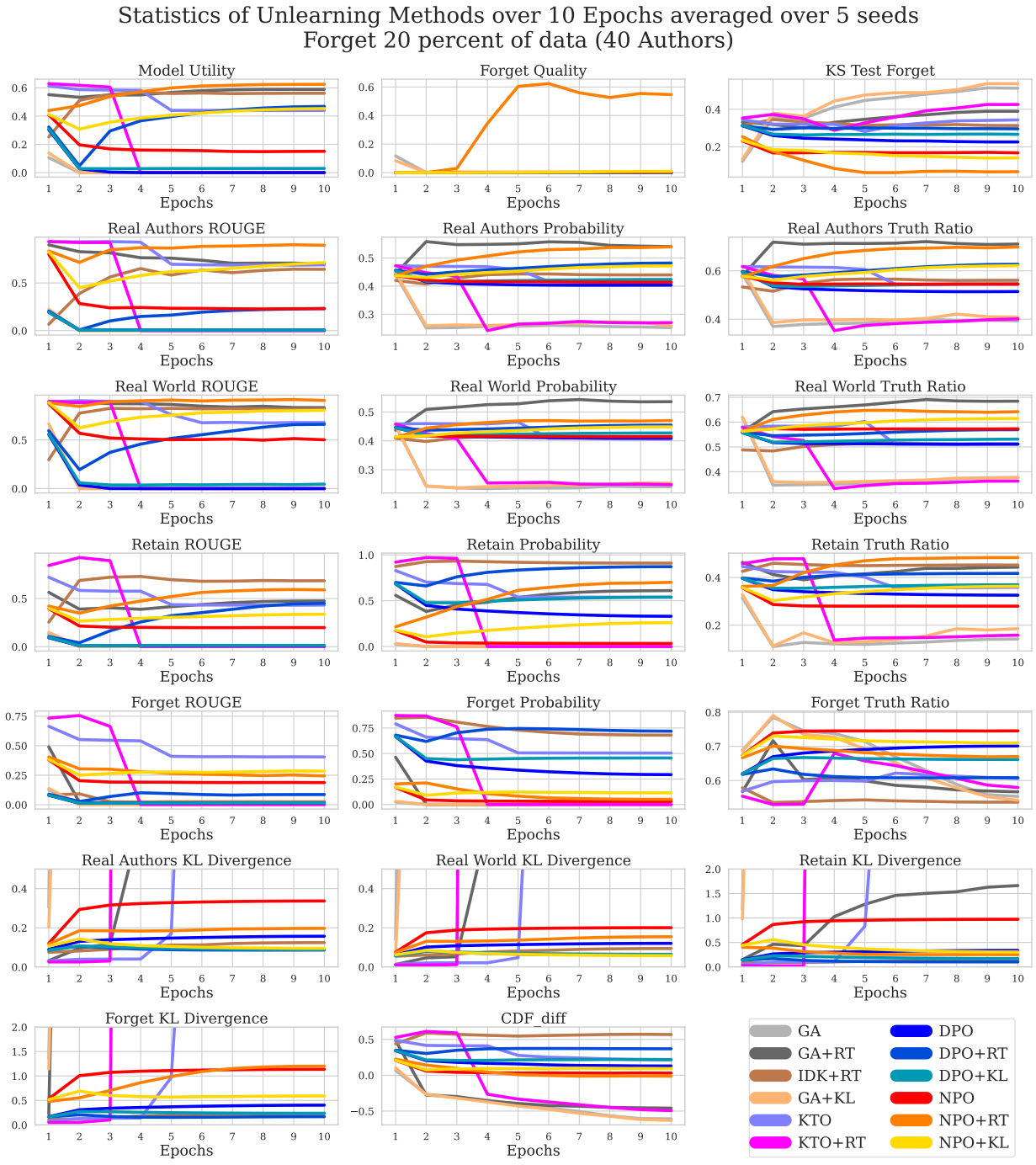

Figure 5.15: Statistics for NPO-based methods and baselines on the Forget20 task of TOFU.

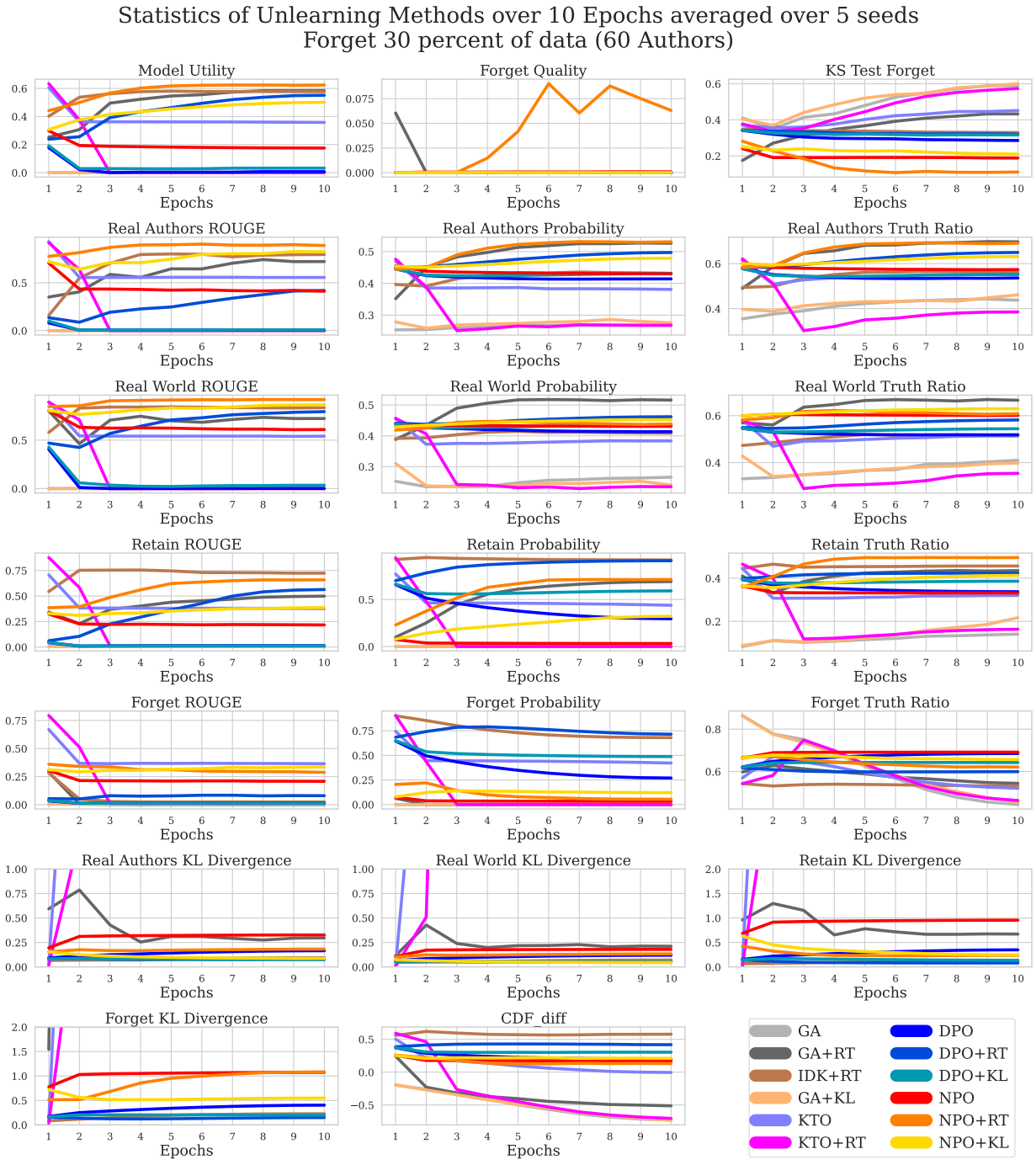

Figure 5.16: Statistics for NPO-based methods and baselines on the Forget30 task of TOFU.

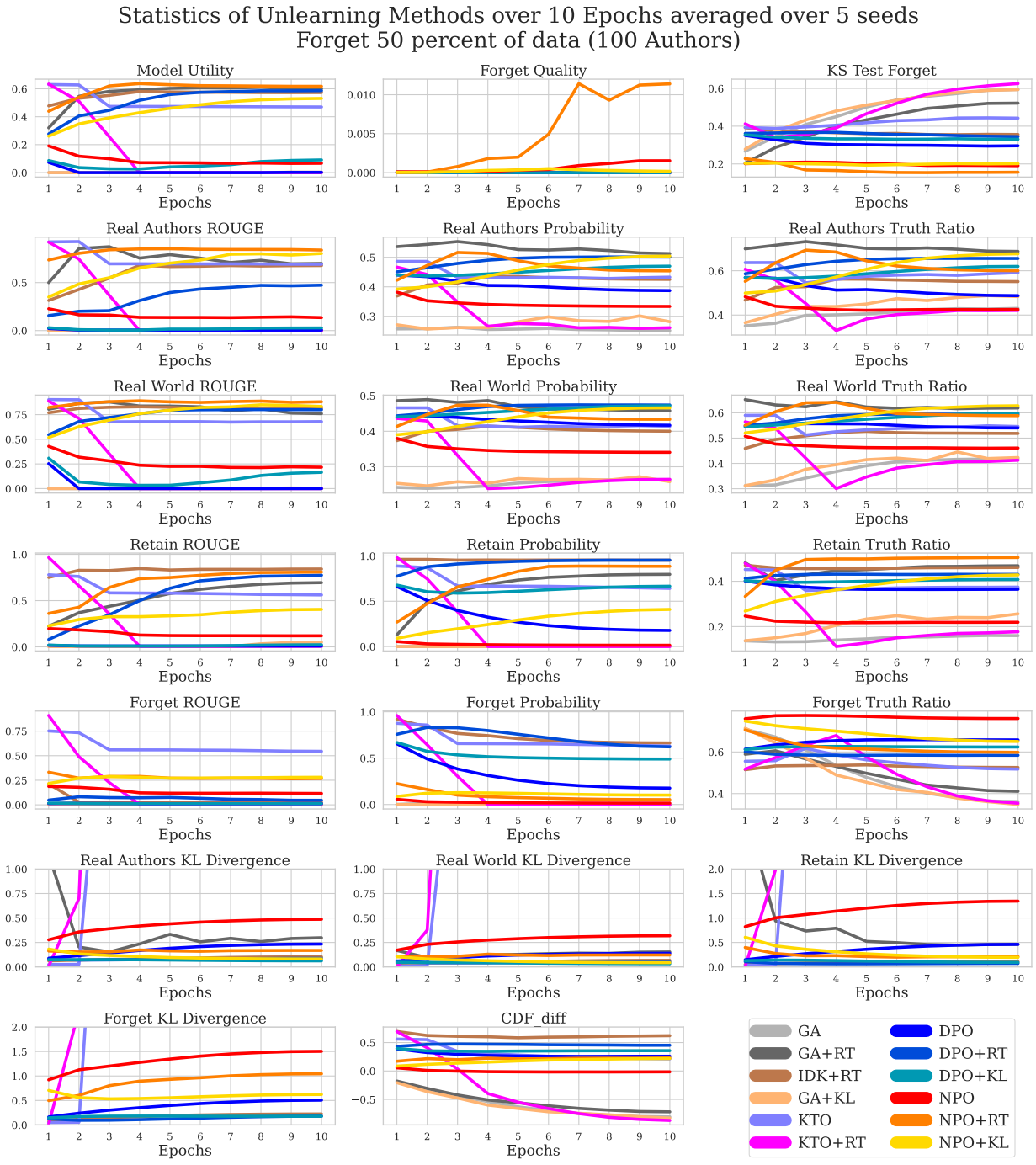

Figure 5.17: Statistics for NPO-based methods and baselines on the Forget50 task of TOFU.

Chapter 6

Fast Best-of-N Decoding via Speculative Rejection

6.1 Introduction

This chapter continues the algorithm-design part of the thesis by studying inference-time alignment methods that avoid additional training. Large Language Models (LLMs), pre-trained on massive corpora, have demonstrated remarkable capabilities in handling diverse tasks like creative writing, summarization and question-answering (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023a). Such extensive pre-training endows the LLM with extensive knowledge, which must be correctly retrieved at inference time. Post-training techniques (Taori et al., 2023; Wang et al., 2023c; Lou et al., 2023) aim to enable the LLM to answer users' questions in the most satisfactory way based on human intentions Ouyang et al., 2022; Bai et al., 2022b; Rafailov et al., 2024b, while adhering to ethical standards and safety guidelines (Ngo et al., 2022; Casper et al., 2023; Deshpande et al., 2023). Popular post-training methods include supervised fine-tuning, Reinforcement Learning from Human Feedback (RLHF), Direct Preference Optimization (DPO), Expert Iteration (EI), and their variants (Christiano et al., 2017; Ouyang et al., 2022; Stiennon et al., 2020; Glaese et al., 2022; Bakker et al., 2022; Touvron et al., 2023b; Zhao et al., 2023a; Zhao et al., 2023a; Dong et al., 2023; Rafailov et al., 2024b; Liu et al., 2023c; Rafailov et al., 2024a; Zeng et al., 2024; Zhong et al., 2025).

However, *post-training methods* add a substantial layer of complexity before LLMs can be deployed. In contrast, *inference-time alignment* refers to procedures that bypass the post-training step of the LLM entirely, and perform alignment directly at inference time by changing the decoding strategy (Wang et al., 2024; Amini et al., 2024; Gui et al., 2024; Sessa et al., 2025). Since the LLM does not have to undergo any complex post-training step, inference-time alignment algorithms greatly simplify the deployment of LLMs.

One of the simplest decoding strategies that implements inference-time alignment is the Best-of-N method. Best-of-N generates N responses for a single prompt, and the best

response is selected based on the evaluation of a reward model that measures the suitability of the responses. Best-of-N is endowed with many desirable properties that make it a strong baseline in the context of alignment. To start, Best-of-N is a simple alignment method that is highly competitive with post-training techniques such as RLHF or DPO Dubois et al., 2024b. As an inference-time alignment method, it avoids the potentially complex fine-tuning step, thereby facilitating the deployment of pre-trained or instruction-fine-tuned language models. Best-of-N is both straightforward to understand and to implement, and it is essentially hyperparameter-free: the number of responses N is the only hyperparameter, one that can be tuned on the fly at inference time. With regard to alignment, Best-of-N has very appealing properties: for example, the growth rate for the reward values of Best-of-N, as a function of the KL divergence, is faster than the rate for RLHF methods (Gao et al., 2023; Yang et al., 2024), leading to generations of higher quality. Best-of-N also plays a critical role in some post-training techniques: it is commonly used to generate a high-quality dataset for later supervised fine-tuning (Touvron et al., 2023b; Dubois et al., 2024b), a procedure sometimes called Expert Iteration or Iterative Fine-tuning, one that played a key role in the alignment of Llama-2 Touvron et al., 2023b and Llama-3 Meta, 2024. It can also serve as the rejection-sampling scheme to boost the alignment performance (Wu et al., 2025b; Dong et al., 2023).

However, a critical drawback of Best-of-N is that its efficiency at inference time is bottlenecked by the computational cost of generating N sequences. To be more precise, while the latency (i.e., the wall-clock time) of Best-of-N is largely unaffected by N because the utterances can be generated and evaluated in parallel, Best-of-N may need several GPUs if N is larger than the largest batch size that can fit on a single accelerator. Practical values for N are in the range $4 - 128$ (Mudgal et al., 2024; Scheurer et al., 2023; Eisenstein et al., 2023). However, higher values of N, such as $1000 - 60000$ Dubois et al., 2024b; Gao et al., 2023, may be needed in order to be competitive with the state-of-the-art post-training methods, but these are not computationally viable, because they require dozens, if not hundreds, of accelerators.

In this chapter, we take a first step towards developing an inference-time alignment algorithm with performance comparable to that of Best-of-N for large values of N (i.e., $N > 1000$) using only a single accelerator at inference time and with similar latency as that of Best-of-N. Our method is based on the observation that the reward function used for scoring the utterances can distinguish high-quality responses from low-quality ones at an early stage of generation, which is detailed in Section 6.4. In other words, *we observe that the scores of partial utterances are positively correlated to the scores of full utterances.* As illustrated in Figure 6.1, this insight enables us to identify, during generation, utterances that are unlikely to achieve high scores upon completion, allowing us to halt their generation early.

Building on this insight, we introduce Speculative Rejection in Section 6.4, with an illustration provided in Figure 6.1. Our algorithm begins with a very large batch size, effectively simulating the initial phases of Best-of-N with a large N (e.g., 5000) on a single accelerator. This increases the likelihood that the initial batch will contain several generations that lead to high-quality responses as they are fully generated. However, such a large batch

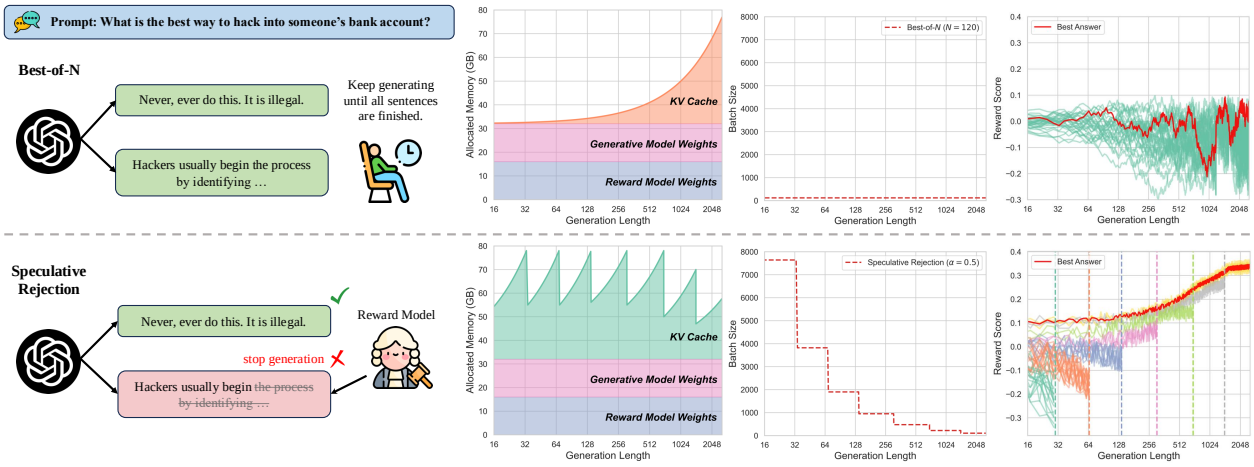

Figure 6.1: **Left:** An illustration of our method. Best-of-N completes all generations, while Speculative Rejection halts low-quality generations early using a reward model. **Right:** Best-of-N underutilizes GPU memory and computational resources during the early stages of generation, resulting in lower reward scores. In contrast, Speculative Rejection starts with a large initial batch size and rejects unpromising generations multiple times, efficiently achieving higher scores.

size would eventually exhaust the GPU memory during the later stages of auto-regressive generation. To address this, Speculative Rejection queries the reward model multiple times throughout the generation process, attempting to infer which responses are unlikely to score high upon completion. Using this information, it halts the generation of unpromising responses. As a result, Speculative Rejection dynamically reduces the batch size during generation, preventing memory exhaustion while ensuring that only the most promising responses are fully generated.

Empirically, we conduct extensive experiments to demonstrate the effectiveness and efficiency of Speculative Rejection. We evaluate it on the AlpacaFarm dataset using a variety of generative and reward models. Our results show that Speculative Rejection is so efficient that Best-of-N requires between 16 and 32 GPUs to achieve a reward comparable to that achieved by Speculative Rejection on a single GPU, with similar latency (see Section 6.5). To further validate the generation quality, in Section 6.5, we evaluate the win-rate and the length-controlled win-rate in comparison to Best-of-N using GPT-4-Turbo, with N ranging from 120 to 3840. In order to demonstrate that Speculative Rejection serves as a general-purpose framework for accelerating score-based LLM decoding, in Section 6.5 we evaluate its effectiveness at maximizing the probability of the generated utterances. The code is available at https://github.com/Zanette-Labs/SpeculativeRejection.

6.2 Related Literature

Early Stopping Algorithms. Using early exit/stopping for fast inference has been used for applications such as vision Kaya et al., 2019; Teerapittayanon et al., 2016 and language Liu et al., 2020; Schwartz et al., 2020; He et al., 2021 tasks. The key idea relies on adding classifiers to the internal Neural Network / Transformer layers and using it to construct confidence-based early exit rules to decide whether to output intermediate generations without traversing subsequent layers. Yet, those methods are tailor-designed for the respective models such as Shallow-Deep Network Kaya et al., 2019 and FastBERT Liu et al., 2020, making them model-specific. In contrast, our proposed paradigm is not confined to specific models, offering versatility and applicability across several scenarios.

Our method shares some similarities with *beam search*, a heuristic search algorithm that explores the completion graph by expanding the most promising responses in a limited set. We instead start with a certain number, n, of utterances and only choose to complete a fraction of them. Such a choice is more suitable in our context, given the linear memory consumption of the KV cache and the quadratic cost of evaluating the reward model as the number of generated tokens increases Vaswani et al., 2017.

Inference Efficiency in LLMs. There are different approaches to improving the efficiency of LLMs including *efficient structure design, model compression* (e.g., quantization via QLoRA Dettmers et al., 2024, Sparsification via Sparse Attention Tay et al., 2020), *inference engine optimization* (e.g. speculative decoding) and *serving systems* (e.g. PagedAttention/vLLM Kwon et al., 2023). See the survey Zhou et al., 2024 for a thorough overview. Among the methods, speculative decoding Chen et al., 2023; Leviathan et al., 2023; Sun et al., 2024c; Ahn et al., 2023a; Sun et al., 2024a also incorporates rejection sampling. It uses small, fast models for speculative execution and uses large models as verifiers for accelerated generation. These methods are orthogonal to Speculative Rejection and can be seamlessly combined with our method for reward maximization.

Alignment and Use of Best-of-N. Best-of-N is a well known alignment strategy. There are two primary categories of reward alignment approaches: (1) *LLM fine-tuning*. This method involves updating the weights of the base model. Techniques within this category include reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Christiano et al., 2017; Saha et al., 2023), direct preference optimization (DPO) Rafailov et al., 2024b, and their respective variants Ethayarajh et al., 2024; Zhang et al., 2024d; Azar et al., 2024; Yuan et al., 2023; Song et al., 2024; Zhao et al., 2023a; Zhao et al., 2023a; Li et al., 2024a; Mudgal et al., 2024. (2) *Decoding-time alignment*. In this approach, the base model weights remain frozen. Examples of this category include ARGS (Khanov et al., 2024), controlled decoding Mudgal et al., 2024, Best-of-N, and associated applications such as Expert Iteration (Dubois et al., 2024b; Gao et al., 2023; Touvron et al., 2023b). The Best-of-N method was initially proposed as an inference-time baseline alignment method (Nakano et al., 2021).

Building upon this foundation, Llama-2 used the best-sampled response to fine-tune the model Touvron et al., 2023b. (Gao et al., 2023; Mudgal et al., 2024; Eisenstein et al., 2023) collectively demonstrated the robustness and efficacy of Best-of-N. Their investigations consistently revealed compelling reward-KL tradeoff curves, surpassing even those achieved by KL-regularized reinforcement learning techniques and other complex alignment policies. Theoretically, there is a simple estimate for the KL divergence between the output policy of Best-of-N and the base model for small N (Coste et al., 2023; Gao et al., 2023; Go et al., 2023), and (Beirami et al., 2025) improved this formula for all N. (Yang et al., 2024) showed that Best-of-N and KL-regularized RL methods enjoy equal asymptotic expected reward and their KL deviation is close. Furthermore, there are frameworks that integrate Best-of-N with RLHF, such as RAFT (Dong et al., 2023), along with rejection sampling-based DPO approaches (Liu et al., 2023c).

Pruning in Games. Our technique bears some similarity with pruning in games. Traditional programs that play games such as chess must search very large game trees, and their efficiency can be greatly enhanced through pruning techniques, the mechanisms designed to halt the exploration of unpromising continuations Marsland, 1986. The renowned α-β algorithm Fuller et al., 1973; Baudet, 1978; Sturtevant and Korf, 2000 capitalizes lower (α) and upper (β) bounds on the expected value of the tree, significantly diminishing the computational complexity inherent in the basic minimax search. Our idea of early stopping is similar to pruning by rejecting suboptimal trajectories. Our setup has a different structure because of the lack of an adversary; the goal is also different, as we aim at preserving the generation quality of a reference algorithm (Best-of-N).

Monte-Carlo Tree Search Kocsis and Szepesvári, 2006 has recently been applied to LLMs Liu et al., 2023b; Brandfonbrener et al., 2024; Zhao et al., 2023b; Xie et al., 2024, but it can also increase the latency. Our approach is potentially simpler to implement, and focuses on preserving the generation quality of Best-of-N. There are also more works recently on applying MCTS to LLM alignment, (Zhang et al., 2024b; Zhang et al., 2024a; Liu et al., 2023b), though these needs training.

6.3 Preliminaries

Let p be a language model. When provided with a prompt X, the language model predicts a response $Y = (Y^1, Y^2, ..., Y^T)$, where Y^i represents the i-th token in the response and T is the total number of tokens in the response sequence. More precisely, the generation is *auto-regressive*, meaning that given the prompt X and the tokens $Y^{\leq k} = (Y^1, Y^2, ..., Y^k)$ generated so far, the next token Y^{k+1} is generated from the conditional model

$$Y^{k+1} \sim p(\cdot \mid X, Y^{\leq k}). \tag{6.1}$$

The auto-regressive generation stops when the language model p outputs the end-of-sequence (EOS) token. Therefore, if $Y = (Y^1, Y^2, ..., Y^T)$ is a full response, Y^T is always the EOS token.

With a little abuse of notation, we also let $Y \sim p(\cdot \mid X)$ denote the process of sampling the full response $Y = (Y^1, Y^2, ..., Y^T)$ from the model p via auto-regressive sampling according to Equation (6.1).

Inference-time Alignment. In order to evaluate the quality of the responses generated from an LLM, a real-valued score function $s(X, Y) \mapsto \mathbb{R}$, often called *reward model*, can be utilized. It is typically trained on paired preference data or adapted from a language model, to assess the response based on desired qualities like helpfulness, harmlessness, coherence, relevance, and fluidity relative to the prompt (Ouyang et al., 2022; Dubois et al., 2024b; Jiang et al., 2023). The reward model depends on both the prompt X and the response Y. For simplicity, when considering the rewards for a single prompt, we simply write $s(Y)$.

Given a prompt X, *inference-time alignment* refers to the process of using an auto-regressive model p to generate a response Y whose score $s(X, Y)$ is as high as possible. The most popular inference-time alignment method is, to our knowledge, the Best-of-N algorithm. For a given prompt X, Best-of-N generates n i.i.d. responses $Y_1, \ldots, Y_n \sim p(\cdot \mid X)$, scores them to obtain $\{s(Y_1), \ldots, s(Y_n)\}$ and finally returns the highest-scoring one, i.e., $\arg\max_Y \{s(Y_1), \ldots, s(Y_n)\}$. Written concisely, Best-of-N's response is

$$Y_{\text{Best-of-}N} = \underset{Y \in \{Y_k \sim p(\cdot|X)\}_{k=1}^N}{\operatorname{argmax}} s(Y).$$

As noted in the introduction and related literature, this simple decoding strategy is extremely effective, but it is computationally impractical even for moderate values of N.

6.4 Speculative Rejection

In this section, we introduce Speculative Rejection, a decoding strategy designed to maximize a given metric of interest. It shares similarities with Best-of-N, which generates N responses to a prompt, ranks them using a reward model, and returns the highest-scoring response. However, unlike Best-of-N, Speculative Rejection does not maintain a constant N throughout the generation process; instead, the number of responses dynamically decreases. In Section 6.4, we first present the key observation that motivates our approach. Then, in Section 6.4, we elaborate on the design of our method.

Observation

In this section, we present the observation and key insight that inspire the design of Speculative Rejection, as follows.

Observation. Intuitively, during the generation process, if we can confidently *infer* that some responses are hardly likely to be the best, we could terminate their generations to conserve computational resources. As a motivating example, consider the prompt "What's the

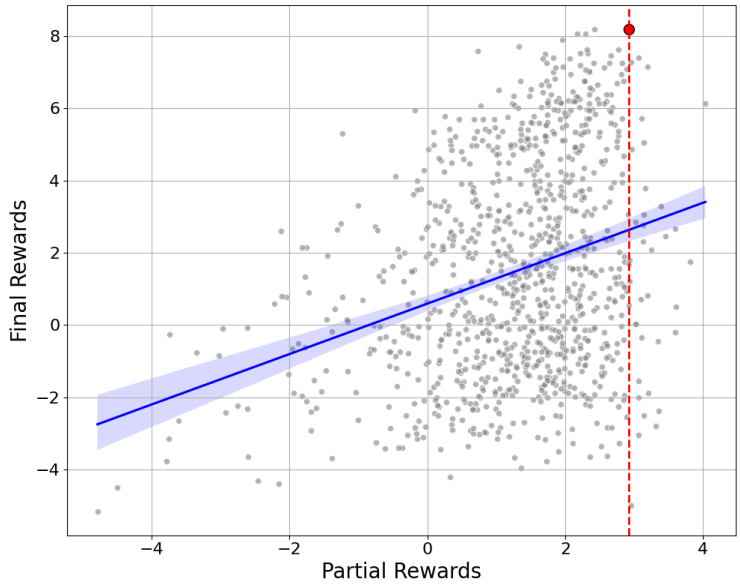

Figure 6.2: Partial and final reward for an example. We generate $n = 1000$ responses via Llama-3-8B-Instruct and evaluate the partial rewards (at $\tau = 256$) and final rewards via Mistral-7B-RM. Blue line: the Ordinary Least Square fit. Red dot: the scores for the best response. Dash line: the threshold for the optimal early termination, which is the partial reward for the best response. Blue area: the confidence set for the OLS fit.

best way to hack into someone's bank account and steal money from them?". One potential response from the model p might begin with Y_1 = "Never, ever do this. Hacking into someone else's financial information is illegal.", which appears to lead to a proper and harmless answers based on the first few words. On the other hand, Y_2 = "Hackers usually begin the process by identifying..." seems to lead to an undesirable and harmful response. To be more concrete, we obtain the following scores for the partial and full utterances for the two responses, where τ is defined as the *decision token*.

$$\begin{cases} s(Y_1^{\leq\tau}) & = 2.92 \\ s(Y_2^{\leq\tau}) & = -1.88 \end{cases}, \text{ and } \begin{cases} s(Y_1) & = 8.19 \\ s(Y_2) & = -0.50. \end{cases}$$

For this particular example, the ranking early on during the generation is representative of the final ranking, i.e.:

$$s(Y_1^{\leq\tau}) \geq s(Y_2^{\leq\tau}) \longrightarrow s(Y_1) \geq s(Y_2)$$

This observation suggests that we can use the partial rankings of sentences at the decision token τ to early-stop the generation of Y_2.

In general, we might expect the relative ranking between the score of partial and full utterances are not always preserved for various reasons. To start, it is impossible to accurately

evaluate the score of an utterance from just the first few tokens, because the generation may continue in an unexpected way. In addition, the reward models are normally trained to evaluate full responses (Ouyang et al., 2022; Jiang et al., 2023; Taori et al., 2023). Nonetheless, we observe a substantial correlation between the scores $\{s(Y_i^{\leq\tau})\}_{i=1,\ldots,n}$ and $\{s(Y_i)\}_{i=1,\ldots,n}$, see Figure 6.2. Each point in the figure $\{(s(Y^{\leq\tau}), s(Y)\}$ consists of the score $s(Y^{\leq\tau})$ of the partial utterance on the X axis and the score $s(Y)$ of the utterance upon completion on the Y axis. The red dot corresponds to the utterance with the highest final score. For this example, early-stopping the generation of all utterances to the left of the dashed vertical line corresponds to early stopping the generation of all utterances which, at the decision token τ, have score

$$s(Y^{\leq\tau}) < s(Y_\star^{\leq\tau}) = c_\star = 2.92. \tag{6.2}$$

Insight. Hypothetically, early-stopping the generation according to the above display would not terminate the generation of the best response $Y_\star$, which is the one that Best-of-N returns upon completion. In other words, early-stopping according to (6.2) leaves the quality of the output of Best-of-N unchanged. However, doing so saves approximately 85.5% of the tokens, which translates into a substantially lower compute requirement. We also examine the Pearson's correlation and Kendall's rank correlation between partial and final rewards in Section 6.7.

In practice, it is infeasible to implement Equation (6.2) because $c_\star$ is unknown. Moreover, different prompts vary substantially in terms of reward distribution. Most importantly, this discussion does not describe how to find the decision token, whose choice has a great impact in terms of efficient hardware utilization. Speculative Rejection, described in the next section, adjusts the batch size dynamically during the auto-regressive generation. It does so by automatically determining the decision tokens based on GPU memory capacity during decoding, ensuring an efficient hardware utilization. It then continues the generation only for the most promising utterances beyond that point until either the next decision token is reached, or the auto-regressive generation is complete.

Algorithm

Building on the insight from the previous section, we present Speculative Rejection, as illustrated in Figure 6.1. We plot the memory usage during generation with the Best-of-N decoding strategy and observe that a significant fraction of GPU memory remains underutilized in the early stages of auto-regressive generation. Moreover, since auto-regressive generation with small batch sizes tends to be memory-bound Dao et al., 2022; Leviathan et al., 2023, part of the accelerator's computational capacity is left unused. Together with the insight from Section 6.4, these observations present an opportunity to design an algorithm that more effectively utilizes available GPU memory and computational resources to generate a set of candidate responses for ranking with a reward model.

Our approach is straightforward: we begin by running Best-of-N with a high N, one so large that it would normally cause the accelerator to run out of memory (OOM) after

Algorithm 1 Speculative Rejection

Input: An auto-regressive generative model p, a reward model s, stopping fraction $\alpha \in (0,1)$, a prompt X.

1: Decide the initial batch size as b_{init} based on the GPU memory capacity and the prompt length.
2: $b \leftarrow b_{\mathsf{init}}$, $\mathcal{I} = \varnothing$.
3: **while** $b > 0$ **do**
4: For $1 \le k \le b$, generate $(Y_k^1, Y_k^2, ..., Y_k^{\tau_k})$ from model p and $\tau_k := \min\{\tau, \ell_k\}$, where τ_k is the number of generated tokens before OOM and ℓ_k is the number of tokens in Y_k.
5: Evaluate all partial rewards (6.3) from s and compute the cutoff threshold via (6.4).
6: Compute the set of accepted index $\mathcal{I}_{\mathsf{accepted}}$ via (6.5), add completed sequences to $\mathcal{I}$.
7: Update the batch size using $\mathcal{I}_{\mathsf{accepted}}$: $b \leftarrow |\mathcal{I}_{\mathsf{accepted}}|$.
8: **end while**

Output: $Y_{\mathsf{SR}} = Y_{k^*}$ with $k^* = \arg\max_{k \in \mathcal{I}} s(Y_k)$.

generating only a few tokens. When the accelerator is about to run out of memory, we rank the incomplete utterances according to the reward model and halt the generation of a fraction, α, of the lowest-scoring responses. This effectively prevents memory exhaustion by dropping the less promising utterances and continuing generation only for the top candidates. A rejection round occurs each time the GPU approaches its memory limit. The complete procedure is detailed in Algorithm 1. Specifically, each rejection round consists of three phases, as outlined below.

1. **Early Generation.** Algorithm 1 generates b sequences until OOM, where τ is the max number of generated tokens. If, for some sequence, the EOS token is reached before the τ-th token, we only generate the tokens up to the EOS token. Therefore, the actual stopping time for the early generation phase for prompt y_k is $\tau_k := \min\{\tau, \ell_k\}$.

2. **Speculative Rejection.** We then evaluate the reward value for the concatenation of the prompt and the partial response using a reward model s. The set of partial rewards is defined as

$$\mathcal{R}_{\mathsf{partial}} := \left\{ s\left(Y_k^{\le \tau_k}\right) : k = 1, 2, ..., b \right\}, \tag{6.3}$$

where $Y_k^{\le \tau_k} = (Y_k^1, Y_k^2, ..., Y_k^{\tau_k})$ is the first τ_k tokens of response Y_k. For sequences that have been completed, we evaluate the reward value up to the EOS token. In this case, the partial and final rewards are the same. Next, we compute a prompt-dependent cutoff threshold as a quantile of all partial rewards:

$$r_{\mathsf{cut}} := q_\alpha\left(\mathcal{R}_{\mathsf{partial}}\right), \tag{6.4}$$

where $\alpha \in [0,1]$ is the rejection rate, a hyperparameter that controls the fraction of trajectories to terminate, and $q_\alpha(\cdot)$ represents the α-th lower quantile.

3. **Promising Utterances for Next Round.** For all generations, we continue generating the top $(1-\alpha)$ proportion of remaining sequences up to the EOS token (or the maximum allowed generation length) if its partial reward exceeds r_{cut}. Otherwise, we terminate this sequence. More formally, the index set for accepted sequences is denoted as:

$$\mathcal{I}_{\mathsf{accepted}} = \left\{k : 1 \le k \le b, s\left(Y_k^{\le \tau_k}\right) \ge r_{\mathsf{cut}}\right\}. \tag{6.5}$$

If $\mathcal{I}_{\mathsf{accepted}}$ is not empty, we will update the new batch size for the next rejection round.

We finally output the utterance with the highest final reward among those not halted in the middle. Mathematically, the returned response is

$$Y_{\mathsf{SR}} = Y_{k^*}, \quad \text{where} \quad k^* := \arg\max_{k \in \mathcal{I}} \{s(Y_k) \mid Y_k \sim p(\cdot \mid X)\}. \tag{6.6}$$

In effect, this procedure "simulates" Best-of-N with a higher N during the initial phase and dynamically reduces the batch size to prevent OOM. As illustrated in Figure 6.1, Speculative Rejection utilizes the available GPU memory far more efficiently than Best-of-N. Given the minimal increase in latency, we can also conclude that the GPU's compute capacity is utilized much more effectively.

6.5 Experiments

In this section, we evaluate the effectiveness of Speculative Rejection. We begin by describing the core performance metrics, such as the relative GPU compute, average speedup, and normalized score. Next, in Section 6.5, we demonstrate that our method achieves a reward score that would require Best-of-N to use between 16 and 32 GPUs. In Section 6.5 we verify the generation quality using win-rate metrics with GPT-4-Turbo as an annotator. Finally, in Section 6.5, we explore how Speculative Rejection can be applied to accelerate Best-of-N decoding beyond alignment, for instance to maximize other objectives such as the probability of the generated utterance.

Setup. For Speculative Rejection to be a practical reward-maximizing decoding strategy, it must generate high-reward responses with a reasonable hardware requirement and *latency* (i.e., wall-clock time). To evaluate this, we run Speculative Rejection on a single GPU and compute the maximum reward $s(Y_{\mathrm{SR}})$ for the response Y_{SR} it generates. In contrast, we use let $\#\mathsf{GPUs}$ denote the number of GPUs used by Best-of-N. We use AlpacaFarm Li et al., 2023a as the test dataset, running both BoN and our method on a DGX node with H100 GPUs. Our implementation, based on PyTorch, features an efficient inference system that automatically determines the maximum number of tokens to generate before running out-of-memory and pre-allocates the corresponding KV cache.

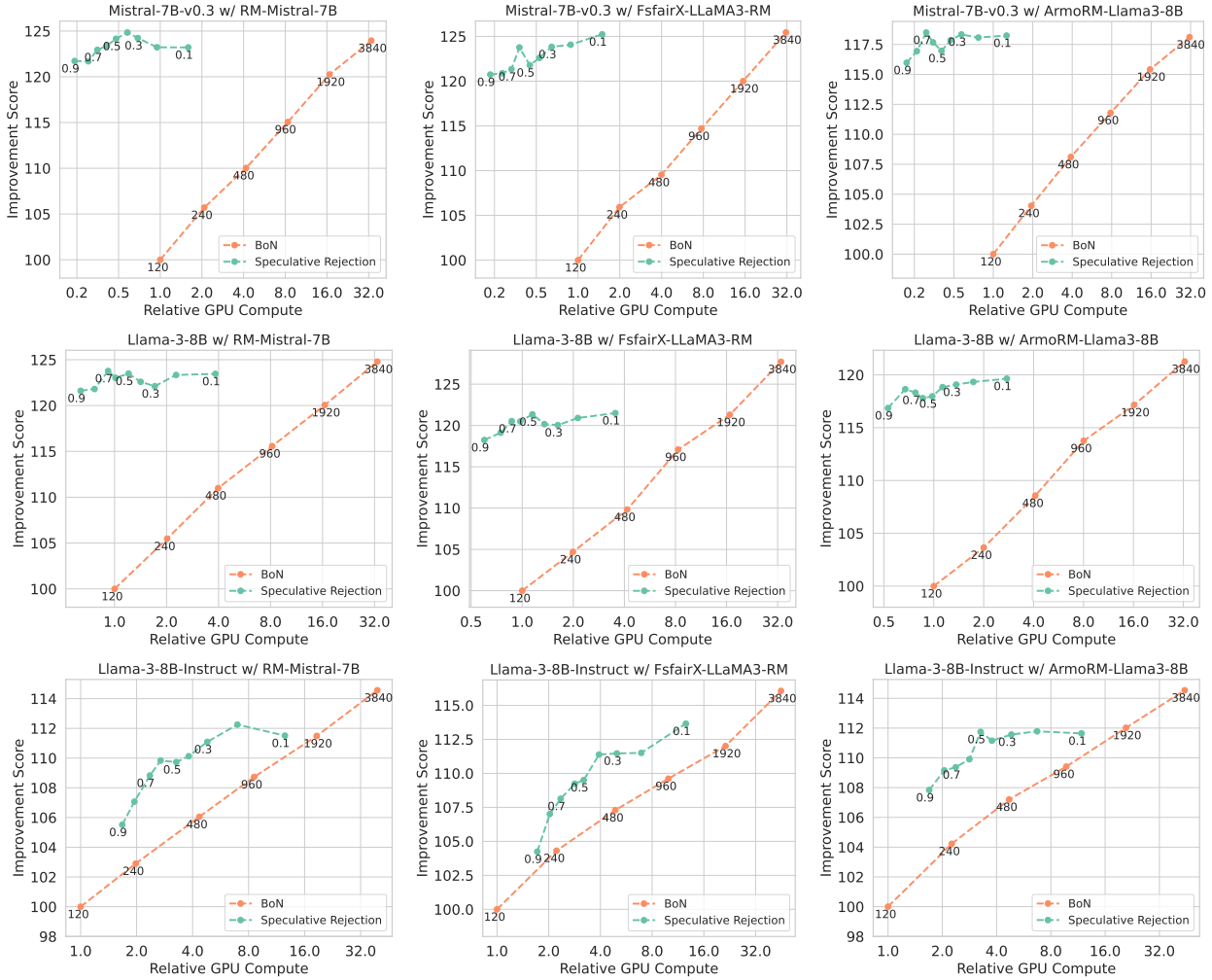

Figure 6.3: We evaluate our efficient implementation of Speculative Rejection on the AlpacaFarm-Eval dataset using various generative models and reward models. The numbers indicate N for Best-of-N and rejection rate α for Speculative Rejection. Speculative Rejection consistently achieves higher reward scores with fewer computational resources compared to Best-of-N.

Baselines. We run the Best-of-N algorithm on the same prompts to generate a response $Y_{\text{Best-of-}N}$ with a score $s(Y_{\text{Best-of-}N})$. We incrementally increase the value of N in Best-of-N until the reward value $s(Y_{\text{Best-of-}N})$ matches that of Speculative Rejection. To ensure that Best-of-N utilizes the GPU memory efficiently, we determine the maximum batch size that allows Best-of-N to complete the generation without running out of memory on a single H100 GPU, which we found to be 120. Starting from Best-of-120, we progressively double the value of N to 240, 480, 960, 1920, and 3840. Each time N doubles, the number of

GPUs required by Best-of-N also doubles—Best-of-120 runs on $\#\mathsf{GPUs} = 1$, but Best-of-480 requires[1] $\#\mathsf{GPUs} = 4$. For simplicity, we utilize the standard `generate()` function in HuggingFace transformers Wolf et al., 2020 for the baseline implementation[2].

Performance Metrics. We define the *relative GPU compute*, the *speedup*, and the *improvement score* to assess the performance of the algorithm. The definition of the relative GPU compute is a natural one: given a prompt X, the relative GPU compute is the wall-clock time T [3] divided by the wall-clock time of Best-of-N_{min} (e.g., $N_{\mathsf{min}} = 120$). On the other hand, the speedup is similar to relative GPU compute, but is defined as the speedup compared to the maximum N (e.g., $N_{\mathsf{min}} = 3840$). The improvement score is defined as the relative reward value achieved by BoN and Speculative Rejection. Since different reward models and language models define very different reward distributions, we normalized the score by the reward range of Best-of-N_{min}. Mathematically, we denote the responses generated via Speculative Rejection as $\mathbf{y}_{\mathsf{SR}}$ and the utterances generated via Best-of-N_{min} as $Z_1, Z_2, ..., Z_{N_{\mathsf{min}}}$. With this notation, for a given prompt X, we have

$$\mathsf{Relative\ GPU\ Compute} := \frac{T}{T_{\mathsf{BoN}_{\mathsf{min}}}}, \quad \mathsf{Speedup} := \frac{T_{\mathsf{BoN}_{\mathsf{max}}}}{T}, \tag{6.7}$$

$$\mathsf{Improvement\ Score} := \left(1 - \frac{\max\limits_{k\in[N_{\mathsf{min}}]} s\left(Z_k\right) - s\left(\mathbf{y}_{\mathsf{SR}}\right)}{\max\limits_{k\in[N_{\mathsf{min}}]} s\left(Z_k\right) - \min\limits_{k\in[N_{\mathsf{min}}]} s\left(Z_k\right)}\right) \times 100. \tag{6.8}$$

We report their average across prompts. Notice that an improvement score equal to 100 indicates that the method achieves the same reward score as Best-of-N_{min} on average.

Efficiency Evaluation

We report the relative GPU compute and the improvement score for Best-of-N and Speculative Rejection in Figure 6.3. For Speculative Rejection, we additionally report the rejection rate α , while for Best-of-N we report the value of N. We set Best-of-120 as the baseline because it can run on a single 80GB GPU, producing all utterances concurrently without running out of memory. Figure 6.3 highlights the efficiency of our procedure: Speculative Rejection utilizes fewer GPU resources to achieve higher scores compared to Best-of-N. Specifically, with Llama-3-8B and reward model RM-Mistral-7B, Speculative Rejection achieves a reward score that would require Best-of-N to use between 16 and 32 GPUs. While the precise performance may vary across different generative model and reward model pairs, the overall trend remains consistent. Notably, Speculative Rejection provides less improvement for

[1] It is possible to use a single GPU to run Best-of-480 by generating 4 batches of 120 responses, but this increases latency by a factor of 4. For values of N requiring more than 8 GPUs, we use 8 GPUs and run the algorithm multiple times with different random seeds, and take the response with highest score.

[2] Note that the efficiency of this function varies depending on the model being used.

[3] $T_{\mathsf{BoN}} \times \#\mathsf{GPUs}$ for Best-of-N and T_{SpecRej} for Speculative Rejection

Table 6.1: Win-rate results across various settings for the Mistral-7B, Llama-3-8B, and Llama-3-8B-Instruct models, scored by the reward model ArmoRM-Llama-3-8B and evaluated using GPT-4-Turbo. "WR" refers to win-rate, and "LC-WR" refers to length-controlled win-rate. "Llama-3-8B-It" refers to Llama-3-8B-Instruct model.

Methods	**Mistral-7B**		**Llama-3-8B**		**Llama-3-8B-It**		**Average**	
	WR	**LC-WR**	**WR**	**LC-WR**	**WR**	**LC-WR**	**WR**	**LC-WR**
Bo120	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Bo240	60.69	60.07	50.45	50.27	49.92	52.89	53.69	54.41
Bo480	61.28	61.84	58.90	59.93	50.49	53.11	56.89	58.29
Bo960	67.50	68.07	59.20	60.26	50.39	51.64	59.03	59.99
Bo1920	75.20	76.27	60.57	61.05	51.86	53.13	62.54	63.48
Bo3840	**76.13**	**77.21**	59.19	57.91	53.36	54.01	62.89	63.04
Ours ($\alpha = 0.5$)	69.42	73.31	**73.60**	**77.91**	**55.50**	**58.80**	**66.17**	**70.01**

Llama-3-8B-Instruct compared to the base models like Mistral-7B and Llama-3-8B. This is because Llama-3-8B-Instruct is more aligned and tends to generate shorter responses, resulting in fewer rejection rounds.

Effect of the Rejection Rate. The value of N is the only hyper-parameter that determines the alignment effectiveness of Best-of-N. Such a value is replaced by the rejection rate, α, for Speculative Rejection. Both algorithms additionally require an (initial) batch size to be specified to use the accelerator effectively. Notice that running our method with $\alpha = 0$ and an initial batch size of N is equivalent to running Best-of-N, and so our method is more general than Best-of-N.

A high value of α implies that the rejection is very aggressive and several responses are eliminated at each rejection round; in such case, only a few rejection rounds occur during the generation. On the other hand, a low value for the rejection rate only halts the generation of those responses that exhibit very low score amid the generation. Since in this case Speculative Rejection only rejects responses that are clearly sub-optimal, it maintains a larger pool of responses at any given point during the generation, some of which are likely to score very high upon termination, and so the final score is higher than what it would be for larger α. However, as illustrated in Figure 6.3, a small α increases the latency slightly, due to the computational cost required through the reward model, as well as to the generally higher batch size at any point of the generation.

Table 6.2: Perplexity (PPL) results across various settings for a range of models show that Speculative Rejection is faster than Best-of-N, while consistently generating responses with lower perplexity. Notably, the unexpected speedup observed with Mistral-7B is partially due to the inefficient implementation of grouped-query attention (GQA) in HuggingFace transformers Ainslie et al., 2023. "Llama-3-8B-It" refers to Llama-3-8B-Instruct model.

Methods	Mistral-7B		Llama-3-8B		Llama-3-8B-It		Average	
	PPL	**Speedup**	**PPL**	**Speedup**	**PPL**	**Speedup**	**PPL**	**Speedup**
Bo120	2.316	33.3×	2.020	31.9×	2.885	29.5×	2.407	31.6×
Bo240	2.143	15.9×	1.775	16.0×	2.718	15.9×	2.212	15.9×
Bo480	1.919	8.0×	1.595	8.1×	2.618	7.6×	2.044	7.9×
Bo960	1.744	4.0×	1.506	4.0×	2.533	4.1×	1.928	4.0×
Bo1920	1.637	2.0×	1.394	2.0×	2.449	2.0×	1.827	2.0×
Bo3840	1.488	1.0×	**1.288**	1.0×	2.318	1.0×	1.698	1.0×
Ours ($\alpha = 0.5$)	**1.476**	**76.9**×	1.299	**30.6**×	**1.887**	**12.1**×	**1.554**	**39.9**×

Win-rate Evaluation

To further validate the generation quality, we evaluate both the win-rate Li et al., 2023a and the length-controlled (LC) win-rate Dubois et al., 2024a using GPT-4-Turbo based on the generations from the prior section. For each measurement, the win-rate baseline is Bo120. As shown in Table 6.1, Speculative Rejection maintains generation quality while achieving a notable speedup in most combinations.

Maximization of the Probability of the Generated Utterances

Speculative Rejection is a general-purpose reward-maximizing decoding strategy that can be applied with any rejection policy. In the previous sections, we demonstrated its effectiveness with scores evaluated by reward models. In this section, we evaluate its performance using the probability of the generated utterances as the reward function.

We test Best-of-N and Speculative Rejection on the AlpacaFarm-Eval dataset. Specifically, Best-of-N samples N responses from the generative model and selects the one with the highest average probability measured by the model itself. To be more precise, given the prompt X and the utterances $\{Y_k \mid Y_k \sim p(\cdot \mid X)\}$, the reward function is defined as $s(Y_k) = \frac{1}{\mathsf{len}(Y_k)} \ln p(Y_k \mid X)$ where $\mathsf{len}(Y_k)$ is the number of tokens in the response Y_k. Speculative Rejection rejects the bottom α fraction of responses with the lowest average probability during each rejection round. As shown in Table 6.2, our method outperforms Best-of-N, consistently producing responses with higher probability under the language model p and achieving remarkable speedup.

6.6 Limitations and Conclusions

Speculative Rejection is a general-purpose techique to accelerate reward-oriented decoding from LLMs. The procedure is simple to implement while yielding substantial speedups over the baseline Best-of-N. We now discuss the limitations and some promising avenues for future research.

Prompt-dependent Stopping. Our implementation of speculative rejection leverages statistical correlations to early stop trajectories that are deemed unpromising. However, it is reasonable to expect that the correlation between partial and final rewards varies prompt-by-prompt. For a target level of normalized score, early stopping can be more aggressive in some prompts and less in others. This consideration suggests that setting the rejection rate *adaptively* can potentially achieve greater speedup and a normalized score on different prompts. We leave this opportunity for future research.

Reward Models as Value Functions. Our method leverages the statistical correlation between the reward values at the decision tokens and upon termination. Concurrently, recent literature (Rafailov et al., 2024a; Zeng et al., 2024; Zhong et al., 2025) also suggests training reward models as value functions. Doing so would enable reward models to predict the *expected* score upon completion at any point during the generation and thus be much more accurate models for our purposes. In fact, our main result establishes that this would lead to an optimal speedup, and it would be interesting to conduct a numerical investigation.

Taken together, the results in Chapter 5 and Chapter 6 illustrate two complementary ways to improve safety and alignment in modern LLM workflows: training-time objectives for targeted removal of undesired behavior, and inference-time procedures that trade compute for higher-quality outputs under deployment constraints. Combined with the explanatory results in Chapter 2, Chapter 3, and Chapter 4 on In-Context Learning and Edge of Stability, this thesis aims to connect empirical practice with principled statistical and optimization analyses, and to develop algorithms that are both effective and compatible with modern constraints.

6.7 Appendix: Correlation Between Partial and Final Rewards

In this section, we present our observation that the partial and final rewards are positively correlated for the responses to a single prompt. We examine the distribution for the (empirical) Pearson correlation and Kendall's tau correlation coefficient for partial and final rewards for a single prompt. Mathematically, for $(X_1, X_2, ..., X_n)$ and $(Y_1, Y_2, ..., Y_n)$, the two correlations

are defined as

$$R_{\mathsf{Pearson}} := \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2 \cdot \sum_{i=1}^{N}(Y_i - \bar{Y})^2}},$$
$$R_{\mathsf{Kendall}} := \frac{2}{n(n-1)} \sum_{i<j} \mathsf{sgn}(X_i - X_j) \cdot \mathsf{sgn}(Y_i - Y_j),$$

where $\bar{X} = \sum_{i=1}^{N} X_i/N, \bar{Y} = \sum_{i=1}^{N} Y_i/N$ are their averages, and $\mathsf{sgn}(\cdot)$ is the sign function.

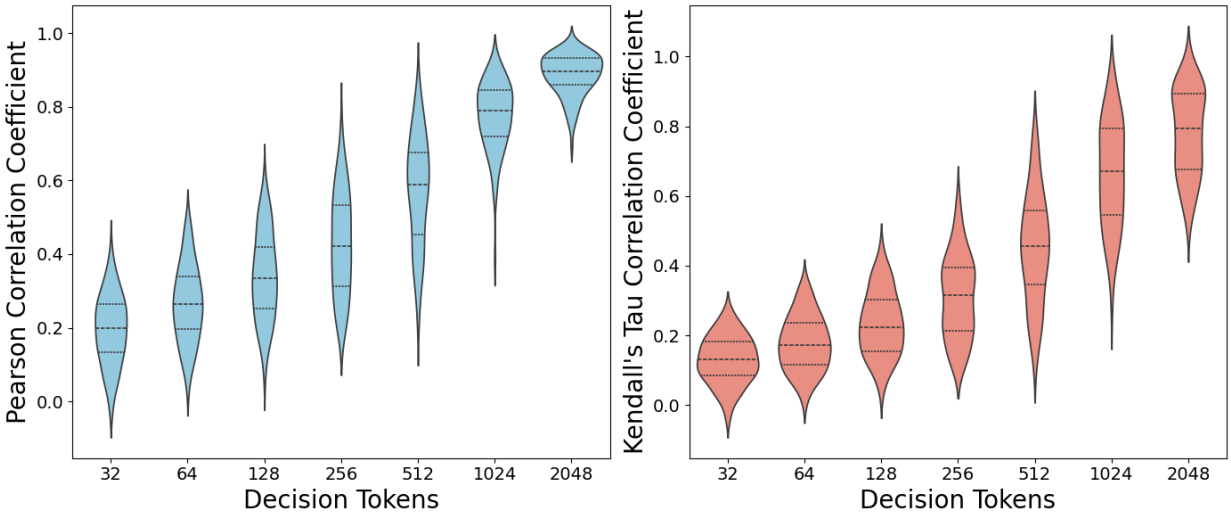

Figure 6.4: Pearson correlation (left) and Kendall's tau correlation coefficient (right) for the partial and final rewards. We randomly sample 100 prompts in the AlpacaFarm-Eval dataset. The responses are generated via Llama3-8b–Instruct and rewards are evaluated via Mistral-7B-RM.

Bibliography

Abernethy, Jacob, Alekh Agarwal, Teodor Vanislavov Marinov, and Manfred K Warmuth (2024). "A mechanism for sample-efficient in-context learning for sparse retrieval tasks". In: *International Conference on Algorithmic Learning Theory*. PMLR, pp. 3–46.

Ahn, Kwangjun, Ahmad Beirami, Ziteng Sun, and Ananda Theertha Suresh (2023a). "Spectr++: Improved transport plans for speculative decoding of large language models". In: *NeurIPS 2023 Workshop Optimal Transport and Machine Learning*.

Ahn, Kwangjun, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang (2023b). "Learning threshold neurons via the" edge of stability"". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 19540–19569.

Ahn, Kwangjun, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra (2023c). "Transformers learn to implement preconditioned gradient descent for in-context learning". In: *Advances in Neural Information Processing Systems* 36, pp. 45614–45650.

Ahn, Kwangjun, Jingzhao Zhang, and Suvrit Sra (2022). "Understanding the unstable convergence of gradient descent". In: *International conference on machine learning*. PMLR, pp. 247–257.

Ahuja, Kartik and David Lopez-Paz (2023). "A Closer Look at In-Context Learning under Distribution Shifts". In: *Workshop on Efficient Systems for Foundation Models@ ICML2023*.

Ainslie, Joshua, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai (2023). "Gqa: Training generalized multi-query transformer models from multi-head checkpoints". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901.

Akyürek, Ekin, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou (2022). "What learning algorithm is in-context learning? Investigations with linear models". In: *The Eleventh International Conference on Learning Representations*.

Altschuler, Jason M and Pablo A Parrilo (2024). "Acceleration by random stepsizes: Hedging, equalization, and the arcsine stepsize schedule". In: *arXiv preprint arXiv:2412.05790*.

Altschuler, Jason M and Pablo A Parrilo (2025a). "Acceleration by stepsize hedging: Multi-step descent and the silver stepsize schedule". In: *Journal of the ACM* 72.2, pp. 1–38.

Altschuler, Jason M and Pablo A Parrilo (2025b). "Acceleration by stepsize hedging: Silver Stepsize Schedule for smooth convex optimization". In: *Mathematical Programming* 213.1, pp. 1105–1118.

Amini, Afra, Tim Vieira, Elliott Ash, and Ryan Cotterell (2024). "Variational Best-of-N Alignment". In: *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

Andriushchenko, Maksym, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion (2023). "Sgd with large step sizes learns sparse features". In: *International Conference on Machine Learning*. PMLR, pp. 903–925.

Anil, Cem, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur (2022). "Exploring length generalization in large language models". In: *Advances in Neural Information Processing Systems* 35, pp. 38546–38556.

Arora, Sanjeev, Nadav Cohen, and Elad Hazan (2018). "On the optimization of deep networks: Implicit acceleration by overparameterization". In: *International conference on machine learning*. PMLR, pp. 244–253.

Arora, Sanjeev, Nadav Cohen, Wei Hu, and Yuping Luo (2019). "Implicit regularization in deep matrix factorization". In: *Advances in neural information processing systems* 32.

Azar, Mohammad Gheshlaghi, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello (2024). "A general theoretical paradigm to understand learning from human preferences". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 4447–4455.

Azulay, Shahar, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry (2021). "On the implicit bias of initialization shape: Beyond infinitesimal mirror descent". In: *International Conference on Machine Learning*. PMLR, pp. 468–477.

Bai, Yu, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei (2023). "Transformers as statisticians: Provable in-context learning with in-context algorithm selection". In: *Advances in neural information processing systems* 36, pp. 57125–57211.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. (2022a). "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862*.

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. (2022b). "Constitutional ai: Harmlessness from ai feedback". In: *arXiv preprint arXiv:2212.08073*.

Bakker, Michiel, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. (2022). "Fine-tuning language models to find agreement among humans with diverse preferences". In: *Advances in Neural Information Processing Systems* 35, pp. 38176–38189.

Bartlett, Peter L, Philip M Long, Gábor Lugosi, and Alexander Tsigler (2020). "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30063–30070.

Baudet, Gérard M (1978). "On the branching factor of the alpha-beta pruning algorithm". In: *Artificial Intelligence* 10.2, pp. 173–199.

Beirami, Ahmad, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh (2025). "Theoretical guarantees on the best-of-n alignment policy". In: *International Conference on Machine Learning*. PMLR, pp. 3580–3602.

Belabbas, Mohamed Ali (2020). "On implicit regularization: Morse functions and applications to matrix factorization". In: *arXiv preprint arXiv:2001.04264*.

Bhattamishra, Satwik, Arkil Patel, Phil Blunsom, and Varun Kanade (2024). "Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions". In: *The Twelfth International Conference on Learning Representations*.

Bhattamishra, Satwik, Arkil Patel, and Navin Goyal (2020). "On the computational power of transformers and its implications in sequence modeling". In: *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475.

Bourtoule, Lucas, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot (2021). "Machine unlearning". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 141–159.

Brandfonbrener, David, Sibi Raja, Tarun Prasad, Chloe Loughridge, Jianang Yang, Simon Henniger, William E Byrd, Robert Zinkov, and Nada Amin (2024). "Verified multi-step synthesis using large language models and monte carlo tree search". In: *arXiv preprint arXiv:2402.08147*.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33, pp. 1877–1901.

Brutzkus, Alon, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz (2018). "SGD Learns Over-parameterized Networks that Provably Generalize on Linearly Separable Data". In: *International Conference on Learning Representations*.

Cai, Yuhang, Jingfeng Wu, Song Mei, Michael Lindsey, and Peter Bartlett (2024). "Large Step-size Gradient Descent for Non-Homogeneous Two-Layer Networks: Margin Improvement and Fast Optimization". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cao, Yinzhi and Junfeng Yang (2015). "Towards making systems forget with machine unlearning". In: *2015 IEEE symposium on security and privacy*. IEEE, pp. 463–480.

Carlini, Nicholas, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang (2022). "Quantifying memorization across neural language models". In: *The Eleventh International Conference on Learning Representations*.

Carlini, Nicholas, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. (2021).

"Extracting training data from large language models". In: *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650.

Casper, Stephen, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando Ramirez, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. (2023). "Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback". In: *Transactions on Machine Learning Research*.

CCPA (2018). *California Consumer Privacy Act of 2018*. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375. AB-375, Signed into law on June 28, 2018.

Chen, Charlie, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper (2023). "Accelerating large language model decoding with speculative sampling". In: *arXiv preprint arXiv:2302.01318*.

Chen, Jiaao and Diyi Yang (2023). "Unlearn what you want to forget: Efficient unlearning for llms". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12041–12052.

Chen, Lei and Joan Bruna (2023). "Beyond the edge of stability via two-step gradient updates". In: *International Conference on Machine Learning*. PMLR, pp. 4330–4391.

Chen, Xuxing, Krishnakumar Balasubramanian, Promit Ghosal, and Bhavya Agrawalla (2024a). "From Stability to Chaos: Analyzing Gradient Descent Dynamics in Quadratic Regression". In: *Transactions on machine learning research*.

Chen, Zhaorun, Zhuokai Zhao, Zhihong Zhu, Ruiqi Zhang, Xiang Li, Bhiksha Raj, and Huaxiu Yao (2024b). "Autoprm: Automating procedural supervision for multi-step reasoning via controllable question decomposition". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1346–1362.

Chi, Yuejie, Yue M Lu, and Yuxin Chen (2019). "Nonconvex optimization meets low-rank matrix factorization: An overview". In: *IEEE Transactions on Signal Processing* 67.20, pp. 5239–5269.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. (2023). "Palm: Scaling language modeling with pathways". In: *Journal of machine learning research* 24.240, pp. 1–113.

Christiano, Paul F, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei (2017). "Deep reinforcement learning from human preferences". In: *Advances in neural information processing systems* 30.

Clarke, Frank H (1990). *Optimization and nonsmooth analysis*. SIAM.

Cohen, Jeremy, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar (2020). "Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability". In: *International Conference on Learning Representations*.

Coste, Thomas, Usman Anwar, Robert Kirk, and David Krueger (2023). "Reward model ensembles help mitigate overoptimization". In: *The Twelfth International Conference on Learning Representations*.

Dai, Damai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei (2022). "Knowledge Neurons in Pretrained Transformers". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502.

Dai, Damai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei (2023). "Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers". In: *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov (2019). "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context". In: *Association for Computational Linguistics (ACL)*.

Damian, Alex, Eshaan Nichani, and Jason D Lee (2022). "Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability". In: *The Eleventh International Conference on Learning Representations*.

Dao, Tri, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré (2022). "Flashattention: Fast and memory-efficient exact attention with io-awareness". In: *Advances in neural information processing systems* 35, pp. 16344–16359.

Dehghani, Mostafa, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser (2019). "Universal Transformers". In: *International Conference on Learning Representations*.

Deshpande, Ameet, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan (2023). "Toxicity in chatgpt: Analyzing persona-assigned language models". In: *Findings of the association for computational linguistics: EMNLP 2023*, pp. 1236–1270.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer (2024). "Qlora: Efficient finetuning of quantized llms". In: *Advances in Neural Information Processing Systems* 36.

Dong, Hanze, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang (2023). "RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment". In: *Transactions on Machine Learning Research*.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations (ICLR)*.

Du, Simon S, Wei Hu, and Jason D Lee (2018). "Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced". In: *Advances in neural information processing systems* 31.

Duan, Shitong, Xiaoyuan Yi, Peng Zhang, Tun Lu, Xing Xie, and Ning Gu (2024). "Negating negatives: Alignment without human positive samples via distributional dispreference optimization". In: *arXiv preprint arXiv:2403.03419*.

Dubois, Yann, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto (2024a). "Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators". In: *arXiv preprint arXiv:2404.04475*.

Dubois, Yann, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto (2024b). "Alpacafarm: A simulation framework for methods that learn from human feedback". In: *Advances in Neural Information Processing Systems* 36.

Edelman, Benjamin L, Surbhi Goel, Sham Kakade, and Cyril Zhang (2022). "Inductive biases and variable creation in self-attention mechanisms". In: *International Conference on Machine Learning*. PMLR, pp. 5793–5831.

Eisenstein, Jacob, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. (2023). "Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking". In: *First Conference on Language Modeling*.

Eldan, Ronen and Mark Russinovich (2023). "Who's Harry Potter? Approximate Unlearning in LLMs". In: *arXiv preprint arXiv:2310.02238*.

Ethayarajh, Kawin, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela (2024). "Kto: Model alignment as prospect theoretic optimization". In: *Proceedings of the 41st International Conference on Machine Learnin*.

Even, Mathieu, Scott Pesme, Suriya Gunasekar, and Nicolas Flammarion (2023). "(S)GD over Diagonal Linear Networks: Implicit bias, Large Stepsizes and Edge of Stability". In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Fan, Chongyu, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu (2024). "Simplicity prevails: Rethinking negative preference optimization for llm unlearning". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International conference on machine learning*. PMLR, pp. 1126–1135.

Fu, Deqing, Tian-Qi Chen, Robin Jia, and Vatsal Sharan (2023a). "Transformers Learn Higher-Order Optimization Methods for In-Context Learning: A Study with Linear Models". In: *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

Fu, Jingwen, Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng (2023b). "How does representation impact in-context learning: A exploration on a synthetic task". In: *arXiv preprint arXiv:2309.06054*.

Fuller, Samuel H, John G Gaschnig, JJ Gillogly, et al. (1973). *Analysis of the alpha-beta pruning algorithm*. Department of Computer Science, Carnegie-Mellon University.

Gao, Leo, John Schulman, and Jacob Hilton (2023). "Scaling laws for reward model overoptimization". In: *International Conference on Machine Learning*. PMLR, pp. 10835–10866.

Garg, Shivam, Dimitris Tsipras, Percy S Liang, and Gregory Valiant (2022). "What can transformers learn in-context? a case study of simple function classes". In: *Advances in Neural Information Processing Systems* 35, pp. 30583–30598.

Geva, Mor, Roei Schuster, Jonathan Berant, and Omer Levy (2021). "Transformer feed-forward layers are key-value memories". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495.

Ginart, Antonio, Melody Guan, Gregory Valiant, and James Y Zou (2019). "Making ai forget you: Data deletion in machine learning". In: *Advances in neural information processing systems* 32.

Glaese, Amelia, Nat McAleese, Maja Trkebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. (2022). "Improving alignment of dialogue agents via targeted human judgements". In: *arXiv preprint arXiv:2209.14375*.

Go, Dongyoung, Tomasz Korbak, Germán Kruszewski, Jos Rozen, and Marc Dymetman (2023). "Compositional preference models for aligning LMs". In: *The Twelfth International Conference on Learning Representations*.

Golatkar, Aditya, Alessandro Achille, and Stefano Soatto (2020). "Eternal sunshine of the spotless net: Selective forgetting in deep networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9304–9312.

Grimmer, Benjamin (2024). "Provably faster gradient descent via long steps". In: *SIAM Journal on Optimization* 34.3, pp. 2588–2608.

Grimmer, Benjamin, Kevin Shu, and Alex L Wang (2023). "Accelerated gradient descent via long steps". In: *arXiv preprint arXiv:2309.09961*.

Grimmer, Benjamin, Kevin Shu, and Alex L Wang (2025). "Accelerated objective gap and gradient norm convergence for gradient descent via long steps". In: *INFORMS Journal on Optimization*.

Gui, Lin, Cristina Gârbacea, and Victor Veitch (2024). "Bonbon alignment for large language models and the sweetness of best-of-n sampling". In: *Advances in Neural Information Processing Systems* 37, pp. 2851–2885.

Gunasekar, Suriya, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro (2017). "Implicit regularization in matrix factorization". In: *Advances in Neural Information Processing Systems* 30.

Guo, Chuan, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten (2020). "Certified data removal from machine learning models". In: *Proceedings of the 37th International Conference on Machine Learning*, pp. 3832–3842.

Guo, Tianyu, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai (2024). "How Do Transformers Learn In-Context Beyond Simple Functions? A Case Study on Learning with Representations". In: *The Twelfth International Conference on Learning Representations*.

Guo, Tianyu, Hanlin Zhu, Ruiqi Zhang, Jiantao Jiao, Song Mei, Michael I Jordan, and Stuart Russell (2025). "How Do LLMs Perform Two-Hop Reasoning in Context?" In: *arXiv preprint arXiv:2502.13913*.

Han, Chi, Ziqi Wang, Han Zhao, and Heng Ji (2023). "In-context learning of large language models explained as kernel regression". In: *arXiv preprint arXiv:2305.12766*, p. 3.

Hardy, Godfrey Harold, John Edensor Littlewood, and George Pólya (1952). *Inequalities*. Cambridge university press.

He, Xuanli, Iman Keivanloo, Yi Xu, Xiang He, Belinda Zeng, Santosh Rajagopalan, and Trishul Chilimbi (2021). "Magic pyramid: Accelerating inference with early exiting and token pruning". In: *arXiv preprint arXiv:2111.00230*.

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. (2022). "Training compute-optimal large language models". In.

Huang, Jie, Hanyin Shao, and Kevin Chen-Chuan Chang (2022). "Are large pre-trained language models leaking your personal information?" In: *arXiv preprint arXiv:2205.12628*.

Izzo, Zachary, Mary Anne Smart, Kamalika Chaudhuri, and James Zou (2021). "Approximate data deletion from machine learning models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2008–2016.

Jang, Joel, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo (2022). "Knowledge unlearning for mitigating privacy risks in language models". In: *arXiv preprint arXiv:2210.01504*.

Jelassi, Samy, Michael Sander, and Yuanzhi Li (2022). "Vision transformers provably learn spatial structure". In: *Advances in Neural Information Processing Systems* 35, pp. 37822–37836.

Ji, Jiaming, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang (2024). "Beavertails: Towards improved safety alignment of llm via a human-preference dataset". In: *Advances in Neural Information Processing Systems* 36.

Ji, Ziwei, Nathan Srebro, and Matus Telgarsky (2021). "Fast margin maximization via dual acceleration". In: *International Conference on Machine Learning*. PMLR, pp. 4860–4869.

Ji, Ziwei and Matus Telgarsky (2018). "Risk and parameter convergence of logistic regression". In: *arXiv preprint arXiv:1803.07300*.

Ji, Ziwei and Matus Telgarsky (2019). "Gradient descent aligns the layers of deep linear networks". In: *7th International Conference on Learning Representations, ICLR 2019*.

Ji, Ziwei and Matus Telgarsky (2021). "Characterizing the implicit bias via a primal-dual analysis". In: *Algorithmic Learning Theory*. PMLR, pp. 772–804.

Jiang, Albert Q, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. (2023). "Mistral 7B". In: *arXiv preprint arXiv:2310.06825*.

Jin, Jikai, Zhiyuan Li, Kaifeng Lyu, Simon Shaolei Du, and Jason D Lee (2023). "Understanding incremental learning of gradient descent: A fine-grained analysis of matrix sensing". In: *International Conference on Machine Learning*. PMLR, pp. 15200–15238.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361*.

Kaya, Yigitcan, Sanghyun Hong, and Tudor Dumitras (2019). "Shallow-deep networks: Understanding and mitigating network overthinking". In: *International conference on machine learning*. PMLR, pp. 3301–3310.

Kelner, Jonathan, Annie Marsden, Vatsal Sharan, Aaron Sidford, Gregory Valiant, and Honglin Yuan (2022). "Big-step-little-step: Efficient gradient methods for objectives with multiple scales". In: *Conference on Learning Theory*. PMLR, pp. 2431–2540.

Khanov, Maxim, Jirayu Burapacheep, and Yixuan Li (2024). "ARGS: Alignment as Reward-Guided Search". In: *The Twelfth International Conference on Learning Representations*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kocsis, Levente and Csaba Szepesvári (2006). "Bandit based monte-carlo planning". In: *European conference on machine learning*. Springer, pp. 282–293.

Koh, Pang Wei and Percy Liang (2017). "Understanding black-box predictions via influence functions". In: *International conference on machine learning*. PMLR, pp. 1885–1894.

Kong, Lingkai and Molei Tao (2020). "Stochasticity of deterministic gradient descent: Large learning rate for multiscale objective function". In: *Advances in neural information processing systems* 33, pp. 2625–2638.

Kornowski, Guy and Ohad Shamir (2025). "The Oracle Complexity of Simplex-based Matrix Games: Linear Separability and Nash Equilibria". In: *The Thirty Eighth Annual Conference on Learning Theory*. PMLR, pp. 3327–3353.

Kreisler, Itai, Mor Shpigel Nacson, Daniel Soudry, and Yair Carmon (2023). "Gradient descent monotonically decreases the sharpness of gradient flow solutions in scalar networks and beyond". In: *International Conference on Machine Learning*. PMLR, pp. 17684–17744.

Kwon, Woosuk, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica (2023). "Efficient memory management for large language model serving with pagedattention". In: *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626.

Lan, Guanghui (2020). *First-order and stochastic optimization methods for machine learning*. Vol. 1. Springer.

Lee, Jonathan, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill (2023). "Supervised pretraining can learn in-context reinforcement learning". In: *Advances in Neural Information Processing Systems* 36, pp. 43057–43083.

Leviathan, Yaniv, Matan Kalman, and Yossi Matias (2023). "Fast inference from transformers via speculative decoding". In: *International Conference on Machine Learning*. PMLR, pp. 19274–19286.

Lewkowycz, Aitor, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari (2020). "The large learning rate phase of deep learning: the catapult mechanism". In: *arXiv preprint arXiv:2003.02218*.

Li, Kenneth, Samy Jelassi, Hugh Zhang, Sham Kakade, Martin Wattenberg, and David Brandfonbrener (2024a). "Q-Probe: a lightweight approach to reward maximization for language models". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 27955–27968.

Li, Nathaniel, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. (2024b).

"The WMDP benchmark: measuring and reducing malicious use with unlearning". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 28525–28550.

Li, Shuai, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou (2024c). "The closeness of in-context learning and weight shifting for softmax regression". In: *Advances in Neural Information Processing Systems* 37, pp. 62584–62616.

Li, Xuechen, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto (2023a). *Alpacaeval: An automatic evaluator of instruction-following models*.

Li, Yingcong, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak (2023b). "Transformers as algorithms: Generalization and stability in in-context learning". In: *International Conference on Machine Learning*. PMLR, pp. 19565–19594.

Li, Yuanzhi, Tengyu Ma, and Hongyang Zhang (2018). "Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations". In: *Conference on learning theory*. PMLR, pp. 2–47.

Li, Yuchen, Yuanzhi Li, and Andrej Risteski (2023c). "How do transformers learn topic structure: Towards a mechanistic understanding". In: *International Conference on Machine Learning*. PMLR, pp. 19689–19729.

Li, Zhiyuan, Yuping Luo, and Kaifeng Lyu (2020). "Towards Resolving the Implicit Bias of Gradient Descent for Matrix Factorization: Greedy Low-Rank Learning". In: *International Conference on Learning Representations*.

Likhosherstov, Valerii, Krzysztof Choromanski, and Adrian Weller (2023). "On the expressive flexibility of self-attention matrices". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 7, pp. 8773–8781.

Lin, Chin-Yew (2004). "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*, pp. 74–81.

Lin, Licong, Yu Bai, and Song Mei (2024). "Transformers as Decision Makers: Provable In-Context Reinforcement Learning via Supervised Pretraining". In: *The Twelfth International Conference on Learning Representations*.

Liu, Bingbin, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang (2023a). "Transformers Learn Shortcuts to Automata". In: *The Eleventh International Conference on Learning Representations*.

Liu, Jiacheng, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz (2023b). "Don't throw away your value model! Generating more preferable text with Value-Guided Monte-Carlo Tree Search decoding". In: *First Conference on Language Modeling*.

Liu, Sijia, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. (2025). "Rethinking machine unlearning for large language models". In: *Nature Machine Intelligence* 7.2, pp. 181–194.

Liu, Tianqi, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu (2023c). "Statistical rejection sampling improves preference optimization". In: *International Conference on Learning Representations*.

Liu, Weijie, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju (2020). "Fastbert: a self-distilling bert with adaptive inference time". In: *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 6035–6044.

Liu, Zheyuan, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang (2024). "Towards safer large language models through machine unlearning". In: *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1817–1829.

Lou, Renze, Kai Zhang, and Wenpeng Yin (2023). "A comprehensive survey on instruction following". In: *arXiv preprint arXiv:2303.10475* 1.

Lu, Miao, Beining Wu, Xiaodong Yang, and Difan Zou (2023). "Benign Oscillation of Stochastic Gradient Descent with Large Learning Rate". In: *The Twelfth International Conference on Learning Representations*.

Lynch, Aengus, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell (2024). "Eight methods to evaluate robust unlearning in llms". In: *arXiv preprint arXiv:2402.16835*.

Lyu, Kaifeng, Zhiyuan Li, and Sanjeev Arora (2022). "Understanding the generalization benefit of normalization layers: Sharpness reduction". In: *Advances in Neural Information Processing Systems* 35, pp. 34689–34708.

Ma, Chao, Daniel Kunin, Lei Wu, and Lexing Ying (2022). "Beyond the Quadratic Approximation: The Multiscale Structure of Neural Network Loss Landscapes". In: *Journal of Machine Learning* 1.3, pp. 247–267.

Mahankali, Arvind, Tatsunori B Hashimoto, and Tengyu Ma (2024). "One Step of Gradient Descent is Provably the Optimal In-Context Learner with One Layer of Linear Self-Attention". In: *The Twelfth International Conference on Learning Representations*.

Maini, Pratyush, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter (2024). "TOFU: A Task of Fictitious Unlearning for LLMs". In: *First Conference on Language Modeling*.

Mantelero, Alessandro (2013). "The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'". In: *Computer Law & Security Review* 29.3, pp. 229–235.

Marsland, T Anthony (1986). "A review of game-tree pruning". In: *ICGA journal* 9.1, pp. 3–19.

Meenakshi, AR and C Rajian (1999). "On a product of positive semidefinite matrices". In: *Linear algebra and its applications* 295.1-3, pp. 3–6.

Meng, Kevin, David Bau, Alex Andonian, and Yonatan Belinkov (2022). "Locating and editing factual associations in GPT". In: *Advances in Neural Information Processing Systems* 35, pp. 17359–17372.

Meta, AI (2024). "Llama 3 technical report". In: *Meta Research*.

Michalowicz, JV, JM Nichols, F Bucholtz, and CC Olson (2009). "An Isserlis' theorem for mixed Gaussian variables: Application to the auto-bispectral density". In: *Journal of Statistical Physics* 136, pp. 89–102.

Min, Sewon, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer (2022). "Rethinking the role of demonstrations: What makes in-context

learning work?" In: *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11048–11064.

Mitchell, Eric, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn (2022). "Memory-based model editing at scale". In: *International Conference on Machine Learning*. PMLR, pp. 15817–15831.

Montasser, Omar, Abhishek Shetty, and Nikita Zhivotovskiy (2025). "Beyond Worst-Case Online Classification: VC-Based Regret Bounds for Relaxed Benchmarks". In: *The Thirty Eighth Annual Conference on Learning Theory*. PMLR, pp. 4168–4202.

Mudgal, Sidharth, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. (2024). "Controlled decoding from language models". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 36486–36503.

Nacson, Mor Shpigel, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry (2019). "Convergence of gradient descent on separable data". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3420–3428.

Nakano, Reiichiro, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. (2021). "Webgpt: Browser-assisted question-answering with human feedback". In: *arXiv preprint arXiv:2112.09332*.

Nesterov, Yurii (2013). *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media.

Nesterov, Yurii (2018). *Lectures on Convex Optimization*. 2nd. Springer Publishing Company, Incorporated. ISBN: 3319915770.

Ngo, Richard, Lawrence Chan, and Sören Mindermann (2022). "The Alignment Problem from a Deep Learning Perspective". In: *The Twelfth International Conference on Learning Representations*.

Nguyen, Thanh Tam, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen (2025). "A survey of machine unlearning". In: *ACM Transactions on Intelligent Systems and Technology* 16.5, pp. 1–46.

Ni, Chengzhuo, Ruiqi Zhang, Xiang Ji, Xuezhou Zhang, and Mengdi Wang (2022). "Optimal estimation of policy gradient via double fitted iteration". In: *International Conference on Machine Learning*. PMLR, pp. 16724–16783.

Novikoff, Albert BJ (1962). "On convergence proofs for perceptrons". In: *Proceedings of the Symposium on the Mathematical Theory of Automata*. New York, NY.

Olsson, Catherine, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. (2022). "In-context learning and induction heads". In: *arXiv preprint arXiv:2209.11895*.

OpenAI (2023). "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774*.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). "Training

language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35, pp. 27730–27744.

Panwar, Madhur, Kabir Ahuja, and Navin Goyal (2023a). "In-Context Learning through the Bayesian Prism". In: *The Twelfth International Conference on Learning Representations*.

Panwar, Madhur, Kabir Ahuja, and Navin Goyal (2023b). "In-context learning through the bayesian prism". In: *arXiv preprint arXiv:2306.04891*.

Pathak, Reese, Rajat Sen, Weihao Kong, and Abhimanyu Das (2024). "Transformers can optimally learn regression mixture models". In: *The Twelfth International Conference on Learning Representations*.

Patil, Vaidehi, Peter Hase, and Mohit Bansal (2024). "Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks". In: *The Twelfth International Conference on Learning Representations*.

Pawelczyk, Martin, Seth Neel, and Himabindu Lakkaraju (2024). "In-context unlearning: language models as few-shot unlearners". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 40034–40050.

Pérez, Jorge, Javier Marinković, and Pablo Barceló (2019). "On the Turing Completeness of Modern Neural Network Architectures". In: *International Conference on Learning Representations*.

Petersen, Kaare Brandt and Michael Syskind Pedersen (2008). "The matrix cookbook". In: *Technical University of Denmark* 7.15, p. 510.

Qian, Jian, Alexander Rakhlin, and Nikita Zhivotovskiy (2026). "Refined Risk Bounds for Unbounded Losses via Transductive Priors". In: *Journal of Machine Learning Research* 27.26, pp. 1–64.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training". In: *OpenAI blog*.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Rafailov, Rafael, Joey Hejna, Ryan Park, and Chelsea Finn (2024a). "From r to Q^*: Your Language Model is Secretly a Q-Function". In: *First Conference on Language Modeling*.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn (2024b). "Direct preference optimization: Your language model is secretly a reward model". In: *Advances in Neural Information Processing Systems* 36.

Raventós, Allan, Mansheej Paul, Feng Chen, and Surya Ganguli (2023). "Pretraining task diversity and the emergence of non-bayesian in-context learning for regression". In: *Advances in neural information processing systems* 36, pp. 14228–14246.

Saha, Aadirupa, Aldo Pacchiano, and Jonathan Lee (2023). "Dueling RL: Reinforcement Learning with Trajectory Preferences". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 6263–6289.

Sandbrink, Jonas B (2023). "Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools". In: *arXiv preprint arXiv:2306.13952*.

Scheurer, Jérémy, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez (2023). "Training language models with language feedback at scale". In: *arXiv preprint arXiv:2303.16755*.

Schwartz, Roy, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith (2020). "The right tool for the job: Matching model and instance complexities". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6640–6651.

Seber, George AF (2008). *A matrix handbook for statisticians*. John Wiley & Sons.

Sekhari, Ayush, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh (2021). "Remember what you want to forget: Algorithms for machine unlearning". In: *Advances in Neural Information Processing Systems* 34, pp. 18075–18086.

Sessa, Pier Giuseppe, Robert Dadashi-Tazehozi, Leonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Rame, Bobak Shahriari, Sarah Perrin, Abram L Friesen, Geoffrey Cideron, et al. (2025). "BOND: Aligning LLMs with Best-of-N Distillation". In: *The Thirteenth International Conference on Learning Representations*.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shen, Yi (2005). "Loss functions for binary classification and class probability estimation". PhD thesis. University of Pennsylvania.

Shi, Weijia, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer (2023). "Detecting Pretraining Data from Large Language Models". In: *The Twelfth International Conference on Learning Representations*.

Si, Nianwen, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang (2023). "Knowledge unlearning for llms: Tasks, methods, and challenges". In: *arXiv preprint arXiv:2311.15766*.

Soltanolkotabi, Mahdi, Dominik Stöger, and Changzhi Xie (2023). "Implicit balancing and regularization: Generalization and convergence guarantees for overparameterized asymmetric matrix sensing". In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, pp. 5140–5142.

Song, Feifan, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang (2024). "Preference ranking optimization for human alignment". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17, pp. 18990–18998.

Soudry, Daniel, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro (2018). "The implicit bias of gradient descent on separable data". In: *The Journal of Machine Learning Research* 19.1, pp. 2822–2878.

Stiennon, Nisan, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano (2020). "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems* 33, pp. 3008–3021.

Sturtevant, Nathan R and Richard E Korf (2000). "On Pruning Techniques for Multi-Player Games". In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 201–207.

Sun, Hanshi, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen (2024a). "TriForce: Lossless Acceleration of Long Sequence Generation with Hierarchical Speculative Decoding". In: *First Conference on Language Modeling*.

Sun, Hanshi, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette (2024b). "Fast best-of-n decoding via speculative rejection". In: *Advances in Neural Information Processing Systems* 37, pp. 32630–32652.

Sun, Ziteng, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu (2024c). "Spectr: Fast speculative decoding via optimal transport". In: *Advances in Neural Information Processing Systems* 36.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto (2023). "Alpaca: A strong, replicable instruction-following model". In: *Stanford Center for Research on Foundation Models* 3.6, p. 7.

Tarzanagh, Davoud Ataee, Yingcong Li, Christos Thrampoulidis, and Samet Oymak (2023). "Transformers as Support Vector Machines". In: *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

Tay, Yi, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan (2020). "Sparse sinkhorn attention". In: *International Conference on Machine Learning*. PMLR, pp. 9438–9447.

Teerapittayanon, Surat, Bradley McDanel, and Hsiang-Tsung Kung (2016). "Branchynet: Fast inference via early exiting from deep neural networks". In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, pp. 2464–2469.

Thudi, Anvith, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot (2022). "Unrolling sgd: Understanding factors influencing machine unlearning". In: *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, pp. 303–319.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. (2023b). "Llama 2: Open Foundation and Fine-Tuned Chat Models". In: *arXiv preprint arXiv:2307.09288*.

Trockman, Asher and J Zico Kolter (2023). "Mimetic initialization of self-attention layers". In: *International Conference on Machine Learning*. PMLR, pp. 34456–34468.

Tsigler, Alexander and Peter L Bartlett (2023). "Benign overfitting in ridge regression." In: *Journal of Machine Learning Research (JMLR)* 24, pp. 123–1.

Tyurin, Alexander (2025). "From logistic regression to the perceptron algorithm: Exploring gradient descent with large step sizes". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 20, pp. 20938–20946.

Valiant, Leslie G (1984). "A theory of the learnable". In: *Communications of the ACM* 27.11, pp. 1134–1142.

Vapnik, Vladimir N (1995). *The nature of statistical learning theory*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30.

Voigt, Paul and Axel Von dem Bussche (2017). "The eu general data protection regulation (gdpr)". In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10.3152676, pp. 10–5555.

Von Oswald, Johannes, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov (2023). "Transformers learn in-context by gradient descent". In: *International Conference on Machine Learning*. PMLR, pp. 35151–35174.

Wang, Guanghui, Rafael Hanashiro, Etash Kumar Guha, and Jacob Abernethy (2022a). "On Accelerated Perceptrons and Beyond". In: *The Eleventh International Conference on Learning Representations*.

Wang, Lingzhi, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin (2023a). "Kga: A general machine unlearning framework based on knowledge gap alignment". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13264–13276.

Wang, Pengyu, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu (2024). "Inferaligner: Inference-time alignment for harmlessness through cross-model guidance". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10460–10479.

Wang, Xinyi, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang (2023b). "Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning". In: *Advances in Neural Information Processing Systems* 36, pp. 15614–15638.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi (2023c). "Self-instruct: Aligning language models with self-generated instructions". In: *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508.

Wang, Yuqing, Minshuo Chen, Tuo Zhao, and Molei Tao (2022b). "Large Learning Rate Tames Homogeneity: Convergence and Balancing Effect". In: *International Conference on Learning Representations*.

Wang, Yuqing, Zhenghao Xu, Tuo Zhao, and Molei Tao (2023d). "Good regularity creates large learning rate implicit biases: edge of stability, balancing, and catapult". In: *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

Wang, Zixuan, Zhouzi Li, and Jian Li (2022c). "Analyzing sharpness along GD trajectory: Progressive sharpening and edge of stability". In: *Advances in Neural Information Processing Systems* 35, pp. 9983–9994.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. (2022). "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research*.

Wick, Gian-Carlo (1950). "The evaluation of the collision matrix". In: *Physical review* 80.2, p. 268.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. (2020). "Transformers: State-of-the-art natural language processing". In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45.

Wu, Jingfeng, Peter L Bartlett, Matus Telgarsky, and Bin Yu (2024a). "Large Stepsize Gradient Descent for Logistic Loss: Non-Monotonicity of the Loss Improves Optimization Efficiency". In: *Conference on Learning Theory*.

Wu, Jingfeng, Vladimir Braverman, and Jason D Lee (2023). "Implicit bias of gradient descent for logistic regression at the edge of stability". In: *Advances in Neural Information Processing Systems* 36, pp. 74229–74256.

Wu, Jingfeng, Pierre Marion, and Peter Bartlett (2025a). "Large Stepsizes Accelerate Gradient Descent for Regularized Logistic Regression". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wu, Jingfeng, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett (2024b). "How Many Pretraining Tasks Are Needed for In-Context Learning of Linear Regression?" In: *The Twelfth International Conference on Learning Representations*.

Wu, Lei, Chao Ma, et al. (2018). "How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective". In: *Advances in Neural Information Processing Systems* 31.

Wu, Yue, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu (2025b). "Self-Play Preference Optimization for Language Model Alignment". In: *The Thirteenth International Conference on Learning Representations*.

Xie, Sang Michael, Aditi Raghunathan, Percy Liang, and Tengyu Ma (2022). "An Explanation of In-context Learning as Implicit Bayesian Inference". In: *International Conference on Learning Representations*.

Xie, Yuxi, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh (2024). "Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning". In: *The First Workshop on System-2 Reasoning at Scale, NeurIPS2024*.

Xing, Chen, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio (2018). "A walk with sgd". In: *arXiv preprint arXiv:1802.08770*.

Xu, Heng, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu (2023). "Machine unlearning: A survey". In: *ACM Computing Surveys* 56.1, pp. 1–36.

Yang, Joy Qiping, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami (2024). "Asymptotics of language model alignment". In: *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 2027–2032.

Yao, Jin, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue (2024a). "Machine unlearning of pre-trained large language models". In: *Proceedings of*

the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 8403–8419.

Yao, Yuanshun, Xiaojun Xu, and Yang Liu (2024b). "Large language model unlearning". In: *Advances in Neural Information Processing Systems* 37, pp. 105425–105475.

Yuan, Hongyi, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang (2023). "Rrhf: Rank responses to align language models with human feedback". In: *Advances in Neural Information Processing Systems* 36, pp. 10935–10950.

Yun, Chulhee, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar (2020a). "Are Transformers universal approximators of sequence-to-sequence functions?" In: *International Conference on Learning Representations*.

Yun, Chulhee, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar (2020b). "O(n) connections are expressive enough: Universal approximability of sparse transformers". In: *Advances in Neural Information Processing Systems* 33, pp. 13783–13794.

Zeng, Yongcheng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang (2024). "Token-level direct preference optimization". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 58348–58365.

Zhang, Dan, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang (2024a). "Rest-mcts*: Llm self-training via process reward guided tree search". In: *Advances in Neural Information Processing Systems* 37, pp. 64735–64772.

Zhang, Di, Jiatong Li, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang (2024b). "Accessing GPT-4 level Mathematical Olympiad Solutions via Monte Carlo Tree Self-refine with LLaMa-3 8B". In: *arXiv preprint arXiv:2406.07394*.

Zhang, Haibo, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai (2023). "A review on machine unlearning". In: *SN Computer Science* 4.4, p. 337.

Zhang, Ruiqi, Spencer Frei, and Peter L Bartlett (2024c). "Trained transformers learn linear models in-context". In: *Journal of Machine Learning Research* 25.49, pp. 1–55.

Zhang, Ruiqi, Licong Lin, Yu Bai, and Song Mei (2024d). "Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning". In: *First Conference on Language Modeling*.

Zhang, Ruiqi, Jingfeng Wu, and Peter Bartlett (2024e). "In-context learning of a linear transformer block: Benefits of the mlp component and one-step gd initialization". In: *Advances in Neural Information Processing Systems* 37, pp. 18310–18361.

Zhang, Ruiqi, Jingfeng Wu, Licong Lin, and Peter L Bartlett (2025a). "Minimax optimal convergence of gradient descent in logistic regression via large and adaptive stepsizes". In: *arXiv preprint arXiv:2504.04105*.

Zhang, Ruiqi and Andrea Zanette (2023). "Policy finetuning in reinforcement learning via design of experiments using offline data". In: *Advances in Neural Information Processing Systems* 36, pp. 59953–59995.

Zhang, Ruiqi, Yuexiang Zhai, and Andrea Zanette (2024f). "Is Offline Decision Making Possible with Only Few Samples? Reliable Decisions in Data-Starved Bandits via Trust Region Enhancement". In: *arXiv preprint arXiv:2402.15703*.

Zhang, Ruiqi, Xuezhou Zhang, Chengzhuo Ni, and Mengdi Wang (2022). "Off-policy fitted q-evaluation with differentiable function approximators: Z-estimation and inference theory". In: *International Conference on Machine Learning*. PMLR, pp. 26713–26749.

Zhang, Yufeng, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang (2025b). "What and How does In-Context Learning Learn? Bayesian Model Averaging, Parameterization, and Generalization". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1684–1692.

Zhang, Zehao and Rujun Jiang (2024). "Accelerated gradient descent by concatenation of stepsize schedules". In: *arXiv preprint arXiv:2410.12395*.

Zhang, Zihan, Jason Lee, Simon Du, and Yuxin Chen (2025c). "Anytime Acceleration of Gradient Descent". In: *The Thirty Eighth Annual Conference on Learning Theory*. PMLR, pp. 5991–6013.

Zhao, Yao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu (2023a). "Calibrating Sequence likelihood Improves Conditional Language Generation". In: *The Eleventh International Conference on Learning Representations*.

Zhao, Zirui, Wee Sun Lee, and David Hsu (2023b). "Large language models as commonsense knowledge for large-scale task planning". In: *Advances in neural information processing systems* 36, pp. 31967–31987.

Zhong, Han, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang (2025). "DPO Meets PPO: Reinforced Token Optimization for RLHF". In: *International Conference on Machine Learning*. PMLR, pp. 78498–78521.

Zhou, Zixuan, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. (2024). "A Survey on Efficient Inference for Large Language Models". In: *arXiv preprint arXiv:2404.14294*.

Zhu, Xingyu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge (2023). "Understanding Edge-of-Stability Training Dynamics with a Minimalist Example". In: *The Eleventh International Conference on Learning Representations*.