Contextures: The Mechanism of Representation Learning

Runtian Zhai

CMU-CS-25-104

April 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Pradeep Ravikumar, Co-chair
Zico Kolter, Co-chair
Andrej Risteski
Yuandong Tian (Meta)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the Air Force Research Laboratory under award number FA8750-17-2-0152 and FA8750-23-2-1015; Robert Bosch GMBH under award number 0087016732PCRPO0087023984; Robert Bosch LLC under award number OSP00009188; the Defense Advanced Research Projects Agency under award number HR00112020006; the Office of Naval Research via N00014-23-1-2368; and the National Science Foundation under award number IIS-2211907.

Keywords: machine learning, representation learning, learning theory, foundation models, artificial intelligence

To my parents, both medical workers, who have made this world better by saving hundreds of lives especially during COVID-19, and helped thousands of people with disability.

Abstract

This dissertation establishes the contexture theory to mathematically characterize the mechanism of representation learning, also known as pretraining. Despite the remarkable empirical success of foundation models, it is not very clear what representations they learn, and why these representations are useful for various disparate downstream tasks. A scientific understanding of representation learning is critical, especially at this point when scaling up the model size is producing diminishing returns, and designing new pretraining methods is imperative for further progress.

Prior work treated different representation learning methods quite differently, whereas the contexture theory provides a unified framework for delineating the representations these methods learn. The central argument is that a representation is learned from the association between the input X and a context variable A. We prove that if an encoder captures the maximum information of this association, in which case we say that the encoder *learns the contexture*, then it will be optimal on the class of tasks that are compatible with the context. We also show that a context is the most useful when the association between X and A is neither too strong nor too weak. The important implication of the contexture theory is that increasing the model size alone will achieve diminishing returns, and further advancements require better contexts.

We demonstrate that lots of existing pretraining objectives can learn the contexture, including supervised learning, self-supervised learning, generative models, etc. Based on that, we introduce two general objectives—SVME and KISE, for learning the contexture. We also show how to mix multiple contexts together, which is an effortless way to create better contexts from existing ones. Then, we prove statistical learning bounds for representation learning, and extend the framework to spectrally transformed kernel regression for semi-supervised learning. Finally, we discuss the effect of the data distribution shift from pretraining to the downstream task.

Acknowledgments

First and foremost, I would like to express my greatest gratitude to my two amazing PhD advisors Zico Kolter and Pradeep Ravikumar. I formed the early ideas of the contexture theory in my fourth year after writing my two ICLR 2024 papers, but it was Pradeep who helped me refine these ideas and put them into a structured and well-written thesis. In the past two years, this thesis has been rewritten for almost twenty times, and Pradeep put great effort into this process. Meanwhile, this thesis could never be in place without the guidance and encouragement from Zico. Zico focuses on the application side more so I spent less time with him, but I still remember him telling me "I am not convinced by your theory, but I think this is great work and you should go for it". From Zico, I learned how non-theory ML people would view my work, which helped me make it more approachable and readable.

I am grateful to all my collaborators, especially Bingbin Liu and her co-advisor Andrej Risteski, who offered me valuable advice during the formation of my ideas. Congratulations to Bingbin for getting married in June! I also would like to thank Yuandong Tian for agreeing to be a member of my thesis committee, and wish him all the best at Meta GenAI.

Moreover, I would not have come to the US to pursue my PhD without Liwei Wang, my undergraduate advisor at Peking University. In November 2018, I was at a loss about my future, so I went to Liwei's office to seek advice. He told me firmly, "for someone like you, there is no better option than pursuing a PhD in the US". He also taught me to only aim for two types of research: Those that lead to real life-improving products, and those that dig deep into the fundamentals of mathematics and science. This has been my research philosophy during my PhD. I want to thank my undergraduate mentor at UCLA Cho-Jui Hsieh, my mentor at MSRA Di He who is now a professor at Peking University, and my mentors at Amazon Stefan Schroedl, Aram Galstyan and Anoop Kumar. I want to thank all members at Locus Lab and RAIL who provided feedback on my papers. I also appreciate the assistance from my PhD program administrators Deb Cavlovich and Matthew Stewart, my OIE advisor Nick Hernandez, and all administrative staff at CMU.

I would like to thank all the students I have mentored during PhD: Yuzhe Lu, Zhenlin Wang, Yilong Qin, Roger Jin, Yash Gupta, Kai Yang, Chenhao Zhang, Hugo Contant, Zihao Ye and Allan Pais. As the Chinese idiom goes, teaching others makes oneself better. Working with these students helped me see my weaknesses, and made me a better mentor and a better person.

Finally, I would like to thank my family. My PhD has been quite crazy, starting in the middle of a global pandemic, followed by the advent of ChatGPT that changed everything. I could not have persevered through all these challenges without the love and encouragement from my parents, as well as my other relatives and close friends. This thesis is dedicated to my parents.

Collaborations: Chapter 2 is a joint work with Kai Yang. Chapters 3 and 4 is a joint work with Kai Yang, Che-Ping Tsai, Burak Varıcı, Hugo Contant and Chenhao Zhang. Chapter 5 is a joint work with Bingbin Liu, Rattana Pukdee, Roger Jin, Andrej Risteski and Maria-Florina Balcan. Chapter 6 is a joint work with Chen Dan. Xiaoyu Huang helped check the proofs. All works are supervised by my advisors Zico Kolter and Pradeep Ravikumar.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to the Contexture Theory

Since around 2018, the field of machine learning (ML) has been shifting from mainly end-to-end deep learning to a new paradigm driven by *foundation models* [15], which are very large models trained on huge datasets. Foundation models achieve great success on a variety of domains, including computer vision (CV) [23, 60, 109], natural language processing (NLP) [34, 102, 117], and more recently tabular data [66, 144]. It has already become common practice to apply foundation models to any new learning task by supervised fine-tuning (SFT) or alignment via reinforcement learning (RL) [110]. Another popular area in ML right now is generative modeling, thanks to the remarkable success of diffusion models [64, 74, 132] and large language models (LLMs) [1, 37, 45].

Foundation models are trained by *representation learning*, which aims to train an encoder for the inputs that "makes it easier to extract useful information when building classifiers or other predictors" [9]. In the context of foundation models, representation learning is also known as "pretraining", after which a predictor is fitted on top of the representation for a specific downstream task. Figure 1.1 illustrates the modern ML paradigm using LLMs as an example. An LLM is pretrained on a huge dataset such as Wikipedia, with a self-supervised learning (SSL) task such as masked token prediction or next token prediction. Then, it is applied to a variety of downstream tasks such as sentiment analysis, summarizing, translation, question answering, etc.

Despite the large body of work on representation learning, we do not have a systematic characterization of the mechanism of representation learning. A critical question that has remained unanswered to a satisfactory extent is the following:

What representations do foundation models learn, and why are these representations useful for a variety of downstream tasks?

In classical statistical learning theory, there is no mystery regarding what is being learned—a mapping from input X to target Y is being learned. However, in representation learning, the very target itself is unclear. For example, what representations does masked token prediction learn, and why are they useful in understanding the sentiment of user reviews on Netflix? For a long time, this has been attributed to the *transferability* of deep learning, but (a) the essence of such transferability is vague, and (b) the assumption that transferability is a property of "deep learning" is questionable.

The lack of understanding in the mechanism of representation learning also leads to other mysteries. For example, supervised learning has been widely used for learning representations—neural networks trained on ImageNet [120] were the most popular

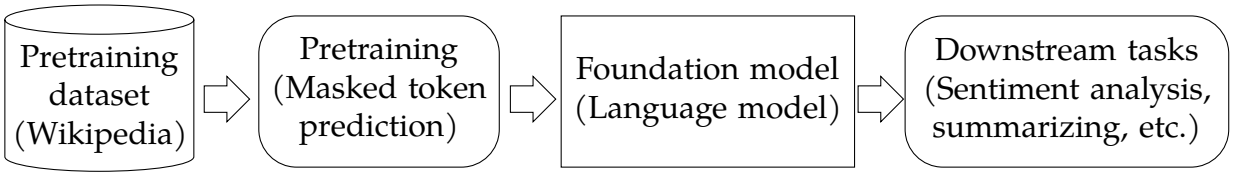


Figure 1.1: Illustration of the modern ML paradigm driven by foundation models and representation learning, using language models as an example in the parentheses.

representations in the early days of the deep learning boom [72]. One uses the output of an intermediate layer, typically the layer before the last linear layer, as the representation of the input. However, [112] found that these representations tend to collapse to a few clusters, a phenomenon known as *neural collapse*. In this case, why are these representations still useful? Another mystery is *representational convergence*—[73] empirically showed that large neural networks of different architectures trained by optimizing different objectives all align with a common representation independent of the architecture and the objective, under the measurement of *representational alignment* [89]. Is this universally true? And how to characterize this common representation?

The above questions are naturally interesting to learning theorists, but why should the broader ML community care about understanding the mechanism of representation learning, if empirical success seems to be always achievable with existing approaches by scaling up the model size, an observation known as *scaling laws* [84]? This is because sustainable success or progress is not always guaranteed. Although some argue that scaling up the size of the model can allow some abilities to "emerge" [154], substantial evidence suggests that many abilities cannot be obtained solely from scaling, which is why additional training signals such as alignment [110] are necessary. Meanwhile, it is widely observed that the current pretraining paradigm is producing diminishing returns, which is why Ilya Sutskever, the scientist behind AlexNet [92] and GPT [117], remarked that "pretraining as we know it will end" recently at NeurIPS 2024 [134]. To make further progress, we need a better understanding of the mechanism of pretraining, which is crucial for designing future generations of pretraining methods, and this is how this field can make scientific progress.

Another important reason why understanding the mechanism of representation learning and foundation models is imperative is the safety concerns of AI. There has long been a debate on whether AI poses an existential threat to human beings, and neither camp can convince the opposing camp. Such a debate has become more and more heated since the advent of ChatGPT, and reached its peak at the recent Paris AI summit held on February 12, 2025. We saw world leaders arguing about how to regulate AI, but reaching an agreement is extremely difficult, though most leaders agreed that some extent of regulation on AI is necessary, and one reason is that these leaders received quite different opinions from their scientists. Advancing learning theory and the science of foundation models is necessary for us to understand the potential risk of AI, so that we can develop a universal and scientific protocol for AI regulation.

1.1 Central Theme: Representations from Association

The purpose of this dissertation is to establish a new theoretical framework dubbed **the contexture theory** in order to characterize the mechanism of representation learning.

The central argument of this theory is that **representations are learned from the association between the input X and a context variable A.** We refer to such an association as a **contexture**. This thesis will prove this argument mathematically and rigorously. In addition, this idea is related to a key concept in psychology—the two systems of thinking.

Psychologist Daniel Kahneman categorizes human thinking into two systems [83]. System 1 thinking refers to fast, automatic, and associative thinking, such as associating a photo of a cat with the animal cat that can meow, associating $2 + 2$ with 4, associating an English word with its Chinese equivalent, etc. System 2 thinking refers to slow, effortful, and logical thinking, such as looking for a golden retriever in an image of 30 dogs, calculating 177×284, and following the proof in a math paper. Decades of research in psychology has shown that the human brain works differently for these two systems of thinking. For example, one piece of evidence is that the pupils dilate when a person is doing system 2 thinking, but not system 1 thinking.

The contexture theory implies that representation learning is capable of doing any type of system 1 thinking, such as image recognition, sentence completion, simple translation, etc. As long as one can specify X and A, their association can be learned by a large model with a sufficient amount of data. This result substantiates the famous **deep learning hypothesis** by Ilya Sutskever [134], stating that "a large neural network can do anything a human can do in a fraction of a second". Hence, system 1 thinking is generally easy, but system 2 thinking is still very hard. For example, teaching an LLM how to reason usually requires complicated methods such as chain of thought [155] and test-time scaling [53, 78], which are beyond the scope of this thesis.

The contexture theory resolves lots of mysteries about deep learning and foundation models, and can lead to better pretraining algorithms. Specifically, in this thesis we will address the following questions:

- What representations do foundation models learn, and why are they useful for a wide range of downstream tasks?
- What variational objectives can be used to learn such representations?
- What does the mechanism of representation learning imply about scaling laws?
- How can we further improve foundation models beyond scaling?
- Are there statistical guarantees for representation learning in the finite data regime?

Takeaways. The key takeaways from this thesis are summarized as follows:

- Representation learning can be understood as recovering the space spanned by the **top singular functions** of the expectation operator jointly induced by the input X and a **context variable** A. We call this process **learning the contexture**.
- These top singular functions can be learned by training a very expressive model to optimize certain variational objectives.
- Scaling up the model size alone inevitably leads to a diminishing return. Further improvement requires better contexts.
- A context is the most useful when the association between X and A is neither too strong nor too weak, in which case the singular values of the expectation operator decay neither too fast nor too slowly.
- If we have multiple contexts whose associations are either too strong or too weak, then we can obtain a better context by mixing them together.
- The representation dimension controls the trade-off between approximation error and estimation error, both of which are influenced by the **context complexity**.

- Achieving good generalization under data distribution shift is extremely hard, and heuristic methods such as reweighting samples usually do not work as expected.

This introductory chapter is primarily devoted to establishing the foundations of the contexture theory. After this chapter, the rest of this thesis is organized as follows.

Chapter 2. This chapter demonstrates that the contexture can be learned using a variety of variational objectives, because these objectives are optimized if and only if the encoder learns the contexture. These objectives include supervised learning, self-supervised learning, generative models, knowledge distillation, etc. Moreover, two general objectives for learning the contexture are introduced: SVME and KISE. The key implication is that scaling brings the representation more aligned to the top singular functions, and when the alignment is high enough, further scaling will achieve a diminishing return.

Chapter 3. This chapter studies how to evaluate an encoder or a context. For encoders, we focus on intrinsic evaluation, which does not depend on any task. Intrinsic evaluation is carried out on a class of tasks that are **compatible** with the context, and we prove that if the task is known to be compatible with the context a priori, then learning the contexture is the optimal thing to do. For contexts, we evaluate them only with their spectra. The key result is that a good context should have a moderate association between X and A, so that the decay rate of its singular values is neither too fast nor too slow. Then, we propose a quantitative evaluation metric, and show that it correlates with the actual downstream performance on real datasets.

Chapter 4. This chapter studies how to learn representations from a *mixture* of multiple contexts. The general approach consists of three base operations: convolution, convex combination and concatenation. Mixing multiple contexts allows us to obtain contexts with moderate associations from strong or weak ones. Detailed algorithms for learning the contexture of these mixtures are provided. We test these algorithms on real tabular datasets, and find that they can achieve higher performance than state-of-the-art methods such as XGBoost [22].

Chapter 5. This theory-intense chapter establishes statistical guarantees for contexture learning. A key object is the context complexity, which characterizes the smoothness of the top singular functions. Then, we extend these results to the more general spectrally transformed kernel regression (STKR) for semi-supervised learning.

Chapter 6. The theory developed so far has assumed that the data distribution is fixed, but in practice there is always a distribution shift from the pretrain to the downstream data. This chapter discusses some challenges in studying such distribution shifts, including the sensitivity to outliers, and the hardness of distributionally robust generalization.

1.2 Contexts: Definition and Examples

This thesis studies the following learning setting: the number of unlabeled samples is much larger than the number of labeled samples. Learning methods in this situation can be categorized as either semi-supervised learning or representation learning. Semi-supervised learning directly learns a predictor on both labeled and unlabeled samples.

Method	**Input** X	**Context Variable** A
Supervised learning on ImageNet	Image	Label of the object in the image
Rotation prediction [46]	Image	Rotated image
BERT [34]: masked token prediction	Text	Masked text
Vision-language model CLIP [116]	Image	Text caption describing the image
K-nearest neighbors (KNN)	Sample	A nearest neighbor of X
Diffusion models for images	Image	Image plus additive noise
GPT [117]: next token prediction	Text	First k tokens of the text

Table 1.1: Examples of inputs and context variables.

Representation learning first learns an encoder with the unlabeled samples, and then fits a predictor on the encoder with the labeled samples, as illustrated in Figure 1.1. This thesis mainly studies representation learning, but the theory can also be generalized to semi-supervised learning, which will be discussed in Section 5.3 when introducing spectrally transformed kernel regression. For now, let us focus on representation learning.

Let $\mathcal{X}$ be the **input space**, and let $P_{\mathcal{X}}$ be the data distribution. $P_{\mathcal{X}}$ is always assumed to be fixed until Chapter 6. The L^2 functional space *w.r.t.* $P_{\mathcal{X}}$ is a Hilbert space denoted by $L^2(P_{\mathcal{X}})$, whose inner product is given by $\langle f_1, f_2 \rangle_{P_{\mathcal{X}}} = \mathbb{E}_{X \sim P_{\mathcal{X}}}[f_1(X) f_2(X)]$, and norm is given by $\|f\|_{P_{\mathcal{X}}} = \sqrt{\langle f, f \rangle_{P_{\mathcal{X}}}}$.

Representation learning aims to learn an encoder $\Phi : \mathcal{X} \to \mathbb{R}^d$. $\Phi(x)$ is called the **embedding** of x, and d is the embedding dimension. We denote $\Phi = [\phi_1, \cdots, \phi_d]$, and assume that $\phi_i \in L^2(P_{\mathcal{X}})$ for all i. The encoder Φ can be either deterministic or randomized. A randomized Φ is a random variable that takes value in $\mathcal{E}_d$, which is the space of all deterministic d-dimensional encoders. Recall that a random variable is formally a measurable function $\Phi : \Omega \to \mathcal{E}_d$ for a sample space Ω.

There are various ways to use a pretrained encoder Φ in a downstream task. This thesis exclusively uses the simplest yet a very common way called a **linear probe**, which fits a linear predictor on top of Φ such that the final predictor is $\boldsymbol{W}\Phi(x) + \boldsymbol{b}$. If Φ is randomized, then we first draw a deterministic encoder from the distribution of Φ (that is, fix a sample in Ω), and then fit a linear probe on top of this encoder.

A context is provided by a **context variable** $A \in \mathcal{A}$, and $\mathcal{A}$ is called the **context space**. The contexture is the relationship between X and A, given by their joint distribution $P^+(x, a)$. Let $P_{\mathcal{X}}$ and $P_{\mathcal{A}}$ be the marginal distributions of P^+. Let $L^2(P_{\mathcal{A}})$ be the L^2 functional space *w.r.t.* $P_{\mathcal{A}}$, with inner product $\langle \cdot, \cdot \rangle_{P_{\mathcal{A}}}$ and norm $\|\cdot\|_{P_{\mathcal{A}}}$. For simplicity, we assume that the probability spaces of both $P_{\mathcal{X}}$ and $P_{\mathcal{A}}$ are compact Hausdorff spaces.

The definition of contexts covers a wide range of machine learning methods. Table 1.1 lists some examples of X and A. Here are some concrete examples.

Labels. A is the label of X. Labels can take different forms, such as discrete categories in classification, continuous values in regression, or structured outputs like text captions of images in vision-language models. Labels may be obtained from human annotators, or in pseudo-forms such as clusters. Typically, labels are provided as compatible pairs of (x, a) sampled from the joint distribution P^+.

Random transformations. These are perturbations (augmentations) on the inputs that presumably do not change the semantics of the inputs by too much. In this case, A is the

corrupted version of X. For example, transformations for images include translation, rotation, flipping, masking, Gaussian noise, Cutout [35] and Mixup [169]. P^+ is provided by the transformation such that one can sample $A \sim P^+(\cdot|x)$ for arbitrarily many times.

Graphs. Graph data is very common in industry, such as social networks, drug discovery, cybersecurity, etc. Graphs also appear in domains that do not involve with graphs explicitly. For example, K-nearest neighbors (KNN) gives a graph where every sample is connected to its nearest neighbors. Graphs can also be continuous: for example, manifold learning [8, 29] approximates a differential operator such as the Laplace-Beltrami operator on a manifold with the continuous limit of a family of graphs. For graphs, we have $\mathcal{A} = \mathcal{X}$, and $P^+(a|x)$ is proportional to the weight of the edge between x and a.

Feature maps and teacher models. Both are functions that map $x \in \mathcal{X}$ to a feature encoding $z \in \mathcal{Z}$, where $\mathcal{Z}$ is typically a Hilbert space. Feature maps are usually defined by humans, while teacher models are learned from data. For example, feature maps for images include PNG, JPEG, etc. Feature maps can also be implicitly defined by kernels [124]. A popular example of teacher models is pretrained language models released by tech companies. It is important to note that $\mathcal{Z}$ is not $\mathcal{A}$, and it is possible that $\mathcal{A}$ is unknown. For example, for these LLMs, if they are close-sourced, then we do not know how they are exactly pretrained.

1.3 Spectral Properties of a Context

The joint distribution P^+ of X and A induces an operator from $L^2(P_\mathcal{A})$ to $L^2(P_\mathcal{X})$, which we call the expectation operator. Basically, it computes the conditional expectation of a function $g \in L^2(P_\mathcal{A})$ given $x \in \mathcal{X}$. This operator is very intuitive: Suppose we want to predict for an input x, but we only have a predictor g on space $\mathcal{A}$; the most reasonable prediction in this scenario is $\mathbb{E}[g(A)|x]$. The adjoint operator of the expectation operator is also an expectation operator, but in the reverse direction.

Definition 1.1. *The **expectation operator** $T_{P^+} : L^2(P_\mathcal{A}) \to L^2(P_\mathcal{X})$ is defined as*

$$(T_{P^+}g)(x) = \int g(a)P^+(a|x)da = \mathbb{E}[g(A)|x] \qquad \textit{for all } g \in L^2(P_\mathcal{A}).$$

*Its **adjoint operator** $T^*_{P^+} : L^2(P_\mathcal{X}) \to L^2(P_\mathcal{A})$, which satisfies $\langle f, T_{P^+}g\rangle_{P_\mathcal{X}} = \langle T^*_{P^+}f, g\rangle_{P_\mathcal{A}}$ for all $f \in L^2(P_\mathcal{X})$ and $g \in L^2(P_\mathcal{A})$, is given by $\left(T^*_{P^+}f\right)(a) = \int f(x)\frac{P^+(a|x)P_\mathcal{X}(x)}{P_\mathcal{A}(a)}dx = \mathbb{E}[f(X)|a]$.*

Remark 1.2. *In general, the operator T_{P^+} is independent of the data distribution $P_\mathcal{X}$, because the stochastic mapping $x \mapsto A$ does not depend on the distribution of X. For example, the mapping from an image to its label is independent of the data distribution on the image space. On the other hand, the adjoint operator $T^*_{P^+}$ depends on $P_\mathcal{X}$ due to Bayes' rule. For example, given the same label "dog", a class-conditional generative model trained on CIFAR-10 [91] and another one trained on ImageNet [120] will generate very different images.*

An easier way to understand these operators is to use the following shorthand notations. Since computing T_{P^+} requires drawing A from $P^+(\cdot|X)$ given X, we write $T_{P^+} : X \to A$. Similarly, we can write $T^*_{P^+} : A \to X$. We can also compose the two operators as $T^*_{P^+}T_{P^+} : X \to A \to X'$, meaning that we first sample $A \sim P^+(\cdot|X)$, and then sample $X' \sim P^+(\cdot|A)$. Similarly, we have $T_{P^+}T^*_{P^+} : A \to X \to A'$. Both $T^*_{P^+}T_{P^+} : L^2(P_\mathcal{A}) \to$

$L^2(P_{\mathcal{A}})$ and $T_{P^+}T_{P^+}^* : L^2(P_{\mathcal{X}}) \to L^2(P_{\mathcal{X}})$ are self-adjoint operators, and they are the integral operators of the following *p.s.d.* kernels.

Definition 1.3. *The **positive-pair kernel** [81] k_A^+ and its **dual kernel** k_X^+ are defined as*

$$k_A^+(a,a') = \frac{P^+(a,a')}{P_{\mathcal{A}}(a)P_{\mathcal{A}}(a')} = \frac{\int P^+(a|x)P^+(a'|x)dP_{\mathcal{X}}(x)}{P_{\mathcal{A}}(a)P_{\mathcal{A}}(a')};$$
$$k_X^+(x,x') = \frac{P^+(x,x')}{P_{\mathcal{X}}(x)P_{\mathcal{X}}(x')} = \frac{\int P^+(x|a)P^+(x'|a)dP_{\mathcal{A}}(a)}{P_{\mathcal{X}}(x)P_{\mathcal{X}}(x')} = \int \frac{P^+(a|x)P^+(a|x')}{P_{\mathcal{A}}(a)}da.$$

Their integral operators are given by

$$T_{k_A^+} = T_{P^+}^* T_{P^+} : L^2(P_{\mathcal{A}}) \to L^2(P_{\mathcal{A}}) \qquad \left(T_{k_A^+}g\right)(a) = \int g(a')k_A^+(a,a')dP_{\mathcal{A}}(a');$$
$$T_{k_X^+} = T_{P^+} T_{P^+}^* : L^2(P_{\mathcal{X}}) \to L^2(P_{\mathcal{X}}) \qquad \left(T_{k_X^+}f\right)(x) = \int f(x')k_X^+(x,x')dP_{\mathcal{X}}(x').$$

We use $T_k : f \mapsto \int f(x')k(\cdot,x')dP_{\mathcal{X}}(x')$ to denote the integral operator of any kernel k. Throughout this work, we assume that $T_{k_A^+}$ and $T_{k_X^+}$ are both Hilbert-Schmidt operators. We say that $\lambda \in \mathbb{R}$ is an **eigenvalue** of $T_{k_A^+}$ with **eigenfunction** $\nu \in L^2(P_{\mathcal{A}})$, if $T_{k_A^+}\nu = \lambda\nu$. Under the above assumption, by Hilbert-Schmidt theorem, we can order the eigenvalues by $1 = \lambda_0 \geq \lambda_1 \geq \cdots \geq 0$, and the corresponding eigenfunctions $\nu_0, \nu_1, \cdots$ form an orthonormal basis (ONB) of $L^2(P_{\mathcal{A}})$. Here $\lambda_i \leq 1$ because of Jensen's inequality, and it is easy to see that $\nu_0 \equiv 1$ is always an eigenfunction of $T_{k_A^+}$ with $\lambda_0 = 1$.

Similarly, denote the eigenfunctions and eigenvalues of $T_{k_X^+}$ by μ_i and κ_i. Then, we can order the eigenvalues by $1 = \kappa_0 \geq \kappa_1 \geq \cdots \geq 0$, where $\mu_0 \equiv 1$, and $\mu_0, \mu_1, \cdots$ form an ONB of $L^2(P_{\mathcal{X}})$. The two sets of eigenfunctions have the following connection.

Lemma 1.4 (Duality property)**.** *For all i, we have $\lambda_i = \kappa_i \in [0,1]$. And if $\lambda_i > 0$, then we have $\mu_i = \lambda_i^{-\frac{1}{2}} T_{P^+}\nu_i$, and $\nu_i = \lambda_i^{-\frac{1}{2}} T_{P^+}^*\mu_i$.*

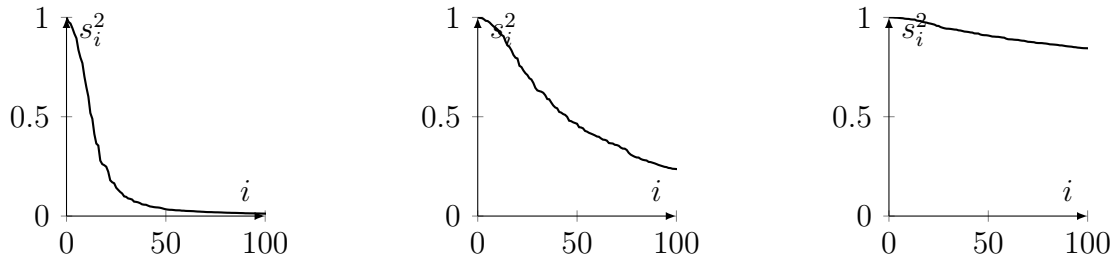
Proof For any i such that $\lambda_i > 0$, we have $T_{P^+}^* T_{P^+}\nu_i = \lambda_i\nu_i$. Thus, $T_{P^+}T_{P^+}^* T_{P^+}\nu_i = \lambda_i T_{P^+}\nu_i$, which shows that $T_{P^+}\nu_i$ is an eigenfunction of $T_{P^+}T_{P^+}^*$ with eigenvalue λ_i. For any i, j such that $\lambda_i > 0$ and $\lambda_j > 0$, we have

$$\begin{aligned}\left\langle \lambda_i^{-\frac{1}{2}} T_{P^+}\nu_i, \lambda_j^{-\frac{1}{2}} T_{P^+}\nu_j \right\rangle_{P_{\mathcal{X}}} &= \lambda_i^{-\frac{1}{2}}\lambda_j^{-\frac{1}{2}} \langle T_{P^+}\nu_i, T_{P^+}\nu_j\rangle_{P_{\mathcal{X}}} = \lambda_i^{-\frac{1}{2}}\lambda_j^{-\frac{1}{2}} \langle T_{P^+}^* T_{P^+}\nu_i, \nu_j\rangle_{P_{\mathcal{A}}} \\ &= \lambda_i^{-\frac{1}{2}}\lambda_j^{-\frac{1}{2}} \langle \lambda_i\nu_i, \nu_j\rangle_{P_{\mathcal{A}}} = \mathbb{I}[i=j],\end{aligned}$$

which implies that $\left\{\lambda_i^{-\frac{1}{2}} T_{P^+}\nu_i\right\}_{i:\lambda_i>0}$ is orthonormal. Similarly, all $\kappa_i > 0$ are eigenvalues of $T_{P^+}^* T_{P^+}$, and $\left\{\kappa_i^{-\frac{1}{2}} T_{P^+}^*\mu_i\right\}_{i:\kappa_i>0}$ is orthonormal. This implies the result. ■

This result leads to the singular value decomposition (SVD) of T_{P^+}. We say that $s_i = \lambda_i^{\frac{1}{2}}$ is a **singular value** of T_{P^+}, associated with the left **singular function** $\mu_i \in L^2(P_{\mathcal{X}})$ and the right singular function $\nu_i \in L^2(P_{\mathcal{A}})$. Since we choose $\mu_0 \equiv 1$ and $\nu_0 \equiv 1$. all other μ_i (and ν_i) must have zero mean because they are orthogonal to μ_0 (and ν_0). Moreover, we have the following spectral decomposition of P^+.

Lemma 1.5 (Spectral decomposition)**.** *We have $P^+(x,a) = \sum_i s_i\mu_i(x)\nu_i(a)P_{\mathcal{X}}(x)P_{\mathcal{A}}(a)$.*



(a) Weak association (b) Moderate association (c) Strong association

Figure 1.2: The association between X and A determines the shape of the spectrum.

Proof $\forall i,\ \left\langle \frac{P^+(x,a)}{P_{\mathcal{X}}(x)P_{\mathcal{A}}(a)}, \nu_i \right\rangle_{P_{\mathcal{A}}} = \int P^+(a|x)\nu_i(a)da = (T_{P^+}\nu_i)(x) = \left(\lambda_i^{\frac{1}{2}}\mu_i\right)(x) = s_i\mu_i(x)$. Since $\{\nu_i\}_{i\geq 0}$ is an ONB, we have $\frac{P^+(x,a)}{P_{\mathcal{X}}(x)P_{\mathcal{A}}(a)} = \sum_{i=0}^{\infty} s_i\mu_i(x)\nu_i(a)$. ■

This result immediately leads to the following spectral decomposition of the two kernels. The proof is left as an exercise to the reader.

Corollary 1.6. *$k_X^+(x,x') = \sum s_i^2\mu_i(x)\mu_i(x')$, and $k_A^+(a,a') = \sum s_i^2\nu_i(a)\nu_i(a')$.*

The set of eigenvalues $\{s_0^2, s_1^2, \cdots\}$ is called the **spectrum** of the context. The shape of the spectrum, or more precisely the decay rate of the eigenvalues, is determined by the strength of association between X and A, which we also call the **association of the context**. In general, the stronger the association, the slower the decay. Consider two extreme cases: (i) When A and X are independent, the association is the weakest; (ii) When $A = X$, the association is the strongest. In case (i), only $s_0^2 = 1$ is positive, and all other eigenvalues are 0, so the eigenvalues decay the fastest. In case (ii), all eigenvalues are 1, so there is no decay at all (in fact, in this case $T_{k_X^+}$ is not Hilbert-Schmidt if $\mathcal{X}$ and $\mathcal{A}$ are infinite sets). Figure 1.2 illustrates the spectrum on different association levels.

There are two key results in the contexture theory. First, a useful context should have a moderate association, and its eigenvalues should decay neither too fast nor too slowly. Obviously, the context in either extreme case above is useless, since A provide no additional information. Second, given a context, among all d-dimensional encoders, the "optimal" one ("optimal" to be formally defined later) should recover the linear space spanned by $\mu_1, \cdots, \mu_d$, for which we say that the encoder learns the contexture of P^+.

Definition 1.7. *A deterministic d-dimensional encoder $\Phi = [\phi_1, \cdots, \phi_d]$ **learns the contexture** of P^+, if there exists a set of top-d singular functions $\{\mu_1, \cdots, \mu_d\}$ of T_{P^+} (excluding $\mu_0 \equiv 1$), such that $\text{span}\{\phi_1, \cdots, \phi_d\} = \text{span}\{\mu_1, \cdots, \mu_d\}$. If the multiplicity of $s_d > 1$, then any set of top-d singular functions suffices. We also say that such a Φ **extracts the top-d eigenspace of $T_{k_X^+}$**.*

Definition 1.8. *A randomized d-dimensional encoder $\Phi = [\phi_1, \cdots, \phi_d]$ **learns the contexture** of P^+ (or extracts the top-d eigenspace of $T_{k_X^+}$), if it learns the contexture almost surely.*

Remark 1.9. *In this definition, $\mu_0 \equiv 1$ is excluded, because the bias term $\mathbf{b}$ in the downstream linear probe implicitly includes μ_0, so there is no reason to waste one dimension to encode μ_0. Note that extracting the top-d eigenspace only requires recovering the linear span, so any invertible linear transformation on Φ makes no difference. A harder task is extracting the exact top-d eigenfunctions, which requires estimating every function μ_i for $i \in [d]$. Although T_{P^+} is independent of $P_{\mathcal{X}}$, the contexture of P^+ depends on $P_{\mathcal{X}}$, since $\mu_1, \cdots, \mu_d$ are defined w.r.t. $P_{\mathcal{X}}$. Thus,*

the same T_{P^+} leads to different contextures when there is a distribution shift in $P_{\mathcal{X}}$, in which case we say that ***the contexture is skewed****.*

The intuition why learning the contexture is ideal is that such a representation keeps the most information (variance) of the context, which is analogous to principal component analysis (PCA) in the finite-dimensional case. Consider the case where $\mathcal{X}$ and $\mathcal{A}$ are both finite sets. Let $N = |\mathcal{X}|$ and $M = |\mathcal{A}|$. Then, a function $f \in L^2(P_{\mathcal{X}})$ is a vector in $\mathbb{R}^N$, $g \in L^2(P_{\mathcal{A}})$ is a vector in $\mathbb{R}^M$, and T_{P^+} is essentially a matrix $\boldsymbol{T} \in \mathbb{R}^{N\times M}$. Suppose we want to learn a d-dimensional embedding $\boldsymbol{E} \in \mathbb{R}^{N\times d}$ for the N samples in $\mathcal{X}$, and it should preserve the information of $\boldsymbol{T}$ as much as possible, then what should we do? PCA states that we should use the top-d left singular vectors of $\boldsymbol{T}$ as $\boldsymbol{E}$, which are equivalent to the top-d eigenvectors of $\boldsymbol{T}\boldsymbol{T}^\top$, because they maximize the explained variance. Similarly, functional spaces are essentially infinite-dimensional vector spaces, so the d-dimensional embedding of X that preserves the most information of T_{P^+} consists of the top-d left singular functions of T_{P^+}, or equivalently the top-d eigenfunctions of $T_{P^+}T_{P^+}^*$.

1.4 Three Types of Access and Example Contexts

In reality, contexts can be provided in a variety of ways. Let us analyze the examples in Section 1.2. Labels are usually provided one alongside each sample, that is the training set is $\{(x_i, a_i)\}_{i=1}^m$. Random transformations are provided as subroutines, which can be called infinitely many times for the same input. A graph is provided as either an adjacency list or an adjacency matrix; the former allows one to sample a neighbor of x, and the latter can be viewed as a kernel. Finally, when given a teacher model, we might not even know what space $\mathcal{A}$ it was trained on.

Generally speaking, there are three types of access we can have to a context. For the context of labels, we have **pair access**. For random transformations, we have **transformation access**. For a graph, if it is given by an adjacency matrix, then we have **kernel access**; if it is given by an adjacency list, then we have transformation access. As for the teacher model, we will show that we have kernel access to its context in Section 2.3.

Definition 1.10. (*i*) *We say that a context has* ***pair access****, if we have access to a dataset of $\{(x_i, a_i)\}_{i=1}^m$ that is i.i.d. sampled from P^+.*

(*ii*) *We say that a context has* ***kernel access*** *(****k-access****), if we have access to a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that approximates the dual kernel of the context. We do not need to know the space $\mathcal{A}$ to have k-access.*

(*iii*) *We say that a context has* ***transformation access*** *(****T-access****), if for any $x \in \mathcal{X}$, one can sample $a \sim P^+(\cdot|x)$ for arbitrarily many times.*

In practice, for a context with T-access, after randomly transforming an input X to A, we usually want to map it back to the input space $\mathcal{X}$. For example, after we apply random cropping to an image, we usually map it back to the original dimension by either stretching the crop or padding it with white pixels; after masking a sentence, we fill in the masked position with a special token `[MASK]`. Such a mapping is called the heuristic inverse of P^+, which we denote by Q^+. Its expectation operator T_{Q^+} is normally different from $T_{P^+}^*$, because $T_{P^+}^*$ depends on $P_{\mathcal{X}}$ while T_{Q^+} in most cases does not depend on $P_{\mathcal{X}}$.

Definition 1.11. *A user-defined conditional distribution $Q^+(x|a)$ that maps $a \in \mathcal{A}$ back to the input space $\mathcal{X}$ is called the* ***heuristic inverse*** *of P^+.*

T-access is stronger than pair access. For example, in supervised learning, the context

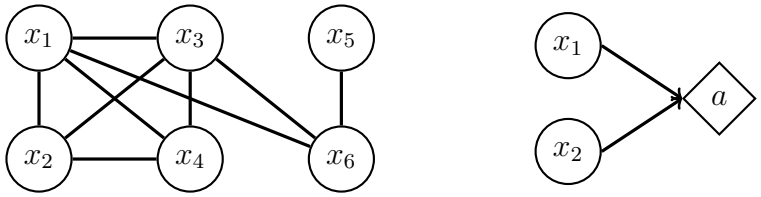


Figure 1.3: Illustration of a transformation graph.

has pair access but not T-access, because the label of any x outside the training set is not given, so we cannot sample $A \sim P^+(\cdot|x)$ for such x. As an exercise, the reader can think about what type of access each context in Table 1.1 has.

Contexts with different types of access require different representation learning methods. For example, contrastive learning [23] needs to sample two views $A, A^+ \sim P^+(\cdot|x)$ for each training sample x, and this requires T-access. If we only have pair access, then we might only have one a for each x. In the next chapter, we will show how to learn the contexture for each type of access.

In what follows, we analyze the three example contexts above in greater detail, and calculate their dual kernels.

Example 1.12 (Classification tasks)**.** *Let $\mathcal{A} = \{1, 2, \cdots, C\}$ be a finite set of labels. The label is deterministic, meaning that each $x \in \mathcal{X}$ is mapped to one label with probability 1 by P^+. Denote the label of x by a_x. The training set is $\{(x_i, a_{x_i})\}_{i=1}^m$, so this context has pair access. For this context, $k_X^+(x, x') = \mathbb{I}[a_x = a_{x'}]P_{\mathcal{A}}(A = a_x)^{-1}$, $s_0 = \cdots = s_{C-1} = 1$, and all other singular values are 0. The span of $\mu_0, \cdots, \mu_{C-1}$ is the same as* $\text{span}\{f_1, \cdots, f_C\}$*, where $f_i(x) = \mathbb{I}[a_x = i]$.*

Example 1.13 (Graphs)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where each edge (u, v) has a non-negative weight $w(u, v)$ such that $w(u, v) = w(v, u)$. Let $w(u, v) = 0$ if u and v are not connected. Let the degree of node u be $D(u) = \sum_{v \in \mathcal{V}} w(u, v)$, and let $D_{\text{sum}} = \sum_{u \in \mathcal{V}} D(u)$. Let $P_{\mathcal{X}}(u) = \frac{D(u)}{D_{\text{sum}}}$, and $P^+(v|u) = \frac{w(u,v)}{D(u)}$, where $\mathcal{A} = \mathcal{X} = \mathcal{V}$. Then, it can be shown that $P_{\mathcal{A}} = P_{\mathcal{X}}$, and $k_X^+(x, x') = \frac{D_{\text{sum}}}{D(x)D(x')} \sum_{u \in \mathcal{V}} \frac{w(x,u)w(x',u)}{D(u)}$. This graph is given by an adjacency list, and this context has T-access.*

Example 1.14 (Random transformations on a finite input space)**.** *Let $\mathcal{X}$ be a finite set, and let the context be given by a random transformation. For example, if $\mathcal{X}$ is the set of all text of up to 512 tokens on a vocabulary of size 30,000, then $\mathcal{X}$ is a finite set. Let N be the size of $\mathcal{X}$. Without loss of generality, assume that $P_{\mathcal{X}}(x) > 0$ for every $x \in \mathcal{X}$. This context has T-access.*

The ***transformation graph*** *$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as $\mathcal{V} = \mathcal{X}$, and $(x_1, x_2) \in \mathcal{E}$ if they can be transformed to the same a. Figure 1.3 gives an illustration of a transformation graph, which was originally introduced as the augmentation graph by [57] in the context of self-supervised learning based on data augmentation.*

Define matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ as $\boldsymbol{W}[i, j] = P^+(x_i, x_j) = \int P^+(x_i|a)P^+(x_j|a)dP_{\mathcal{A}}(a)$, which is the weight of edge (x_i, x_j). Then, the degree of node x_i is $\sum_x P^+(x_i, x) = P_{\mathcal{X}}(x_i)$. Define matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ as $\boldsymbol{D} = \text{diag}\{P_{\mathcal{X}}(x_1), \cdots, P_{\mathcal{X}}(x_N)\}$. The singular function μ_i is now an N-dimensional vector. It is easy to see that $\boldsymbol{W}\mu_i = s_i^2 \boldsymbol{D}\mu_i$, that is (s_i^2, μ_i) is a pair of generalized eigenvalue and eigenfunction. We can also rewrite it as $(\boldsymbol{D} - \boldsymbol{W})\mu_i = (1 - s_i^2)\boldsymbol{D}\mu_i$, where $\boldsymbol{D} - \boldsymbol{W}$ is the (unnormalized) Laplacian matrix of the transformation graph [26]. Thus, $\mu_1, \cdots, \mu_d$ are the bottom eigenvectors of the graph Laplacian.

1.5 Prior Work

In this century, machine learning has gone through three paradigms: kernel methods, end-to-end deep learning, and foundation models. Representation learning is a key part in all three paradigms. This section provides a short review of their history.

Kernel methods were quite popular when machine learning was dominated by support vector machines (SVMs) [62]. An SVM is a linear predictor, but when the relationship between the input X and the target Y is not linear, we need a non-linear feature map Φ such that there is a linear relationship between $\Phi(X)$ and Y. The **kernel trick** says that we do not need to explicitly specify Φ, but only need to specify a kernel k such that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$, and then we can train an SVM on top of this kernel. The reason why a kernel is more ideal in some applications is that k is easier to define than Φ. For example, if we assume that samples close in the Euclidean space are similar, then k can be defined as a kernel that is larger when the distance is smaller, such as the RBF kernel or the KNN kernel. However, to compute Φ, one needs to learn the eigenmap that consists of the top eigenfunctions of T_k [8], and this is hard in practice. For this reason, at that time kernel methods were more popular than representation learning in semi-supervised learning [10, 124].

However, kernel methods and SVMs began to fall short when people started to apply machine learning to harder tasks, such as ImageNet classification [92]. The relationship between Y and X in these tasks are too complex, and no human-designed kernel can work well on these tasks. Deep learning became extremely popular when people found out that a deep neural network trained to approximate the target function can automatically learn the complex relationship between Y and X, so there is no need to design a kernel. In other words, kernel methods define Φ by specifying a kernel, whereas deep learning obtains Φ by optimizing a variational objective using a large function approximator. End-to-end deep learning has been very successful in supervised learning.

For semi-supervised learning, a variety of deep learning methods have been proposed [145]. To use the unlabeled samples, many methods require the model to "behave well" on the unlabeled samples, which is commonly known as *consistency regularization*: For each unlabeled sample x, we augment it into x' and x'' in two different ways, and ask the model to give similar outputs to x' and x''. Some methods define x' and x'' as the outputs of models at different epochs, such as temporal ensembling [93] and mean teachers [136]. Some methods use a strong augmentation to obtain x' and a weak one to obtain x'', such as FixMatch [131] and noisy student [160]. Some methods use adversarial attack, such as virtual adversarial training [107]. Some methods use the interpolation between two input/target pairs, such as MixUp [169] and MixMatch [11]. These methods use context variables x' and x'' to learn a predictor for semi-supervised learning.

In the modern ML paradigm, a variety of downstream tasks share the same gigantic unlabeled dataset. In this scenario, end-to-end semi-supervised learning is not ideal, because for every different task we need to learn a predictor on all labeled and unlabeled samples, which is very inefficient. To solve this problem, people proposed to use transfer learning, which is the basis of foundation models. The assumption is that a large model trained on task 1 can be transferred to tasks 2, 3 and so on, which is a widely observed phenomenon in deep learning [72], though why such transfer learning works has not been fully explained. Under this assumption, one can pretrain a large encoder on the huge unlabeled dataset using a general task, so that when learning a predictor for another task, one does not need to use the gigantic unlabeled dataset again.

Foundation models are usually trained by self-supervised learning, and there is a

large body of work on the theoretical analysis of self-supervised learning. One line of research studies the effectiveness of contrastive learning by showing its features are optimal when used to fit a linear predictor on certain downstream tasks [123, 142, 143], robust to class imbalance [100], and suitable for unsupervised domain adaptation [58, 126]. Masked prediction tasks have been shown to be useful for reducing the downstream sample complexity [96] and for parameter identifiability [99]. In terms of language applications, [123] explained why next-word prediction can benefit sentiment classification, and [153] studied the effect of prompt tuning through the lens of implicit Bayesian inference. Regarding the optimization in representation learning, there have been prior works on the training dynamics and loss landscapes of contrastive learning [80, 139, 156], non-contrastive learning [114, 140, 157], and masked prediction [70, 161]. There is also a line of theoretical work that connects self-supervised learning to information theory [2, 5, 129].

More related to this thesis is a line of work that formulates contrastive learning as a Laplacian operator over the augmentation graph. The idea of studying data augmentation from a kernel perspective was first explored in [31, 108, 118]. [57] defined the augmentation graph and then proved a generalization bound for the spectral contrastive loss. Then, [122] pointed out that this model-class-free bound could be vacuous with a hypercube construction. As a response to this argument, [56] included the effect of the encoder's inductive bias into their new generalization bounds. Then, [81] defined the positive-pair kernel for the augmentation graph, [152] connected contrastive learning to message passing on the augmentation graph, and [20] proved generalization bounds that do not depend on the function class of Φ (but they still need to assume that the target function belongs to the RKHS of a known kernel). My own work [167, 168] extended these results to any augmentation-based self-supervised learning (not only contrastive learning), and then the more general spectrally transformed kernel regression, which builds the link between representation learning and semi-supervised learning.

The common weakness of the papers mentioned above is that they treated different representation learning methods quite differently. For example, contrastive learning, non-contrastive learning and masked autoencoders have been regarded as inherently distinct methods, and for each method there is a line of theoretical work. Consequently, despite the large body of theoretical work, our understanding of representation learning is still quite muddled. The contexture theory established in this thesis provides a universal and lucid characterization of the mechanism of a wide range of representation learning methods. The key takeaway is that the various methods are all learning the contexture of a context, so using which method is less important than the context itself.

Also closely related to this thesis is a line of work on representation alignment [43, 73, 75, 89]. Representation similarity has also been studied in neuroscience [90]. These papers aims to compare between two representations, while this thesis mainly focuses on evaluating a single representation, or the context on which it is trained.

Chapter 2

Learning the Contexture with Variational Objectives

Learning the contexture requires extracting the top-d eigenspace of a kernel integral operator $T_{k_X^+}$. Conventionally, this is done by **kernel PCA** [124, Chapter 14]. Let k be a *p.s.d.* kernel, and let $T_k : L^2(P_\mathcal{X}) \to L^2(P_\mathcal{X})$ be its integral operator. Suppose we can compute $k(x, x')$ for all $x, x' \in \mathcal{X}$. Then, given a training set $\{x_i\}_{i=1}^m$, kernel PCA estimates the top-d eigenfunctions of T_k as follows:

1. Build the Gram matrix $\boldsymbol{G} \in \mathbb{R}^{m\times m}$ of k, such that $\boldsymbol{G}[i, j] = k(x_i, x_j)$.
2. Compute the eigenvalues and eigenvectors $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^m$ of $\boldsymbol{G}$, where $\lambda_1 \geq \cdots \geq \lambda_m$, and $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d$ form an ONB of $\mathbb{R}^m$. Assume that $\lambda_d > 0$.
3. $\hat{\mu}_i(x) = \lambda_i^{-1} \sum_{j=1}^m k(x, x_j)\boldsymbol{v}_i[j]$ is an estimation of the i-th eigenfunction.

Kernel PCA has two issues. First, it requires that $k(x, x')$ can be efficiently estimated for all x, x' (which is k-access), but this is not always possible; for example, $k(x, x')$ is hard to estimate when it is the dual kernel of a random transformation context, since estimating $P_\mathcal{A}(a)^{-1}$ with high precision requires lots of samples from P^+. Second, it is not scalable for huge datasets. In general, the time complexity of eigen-decomposition can be regarded as $O(m^3)$. As of today, the fastest algorithm for eigen-decomposition in theory has $O(m^\omega)$ complexity with $\omega \approx 2.38$ [32], which is still not very scalable.

However, it is possible to have a more efficient algorithm than kernel PCA, because kernel PCA can extract the exact top-d eigenfunctions, while our goal is to only extract the top-d eigenspace of $T_{k_X^+}$, that is we do not need to estimate the exact function μ_i. Moreover, we can also make the algorithm faster if we are willing to sacrifice precision.

This chapter shows how to learn the contexture with a variational objective $\mathcal{R}(\Phi)$, meaning that this objective is optimized if and only if Φ extracts the top-d eigenspace of $T_{k_X^+} = T_{P^+}T_{P^+}^*$. Recall that this excludes $\mu_0 \equiv 1$. Provided with such an objective, one can learn the contexture by optimizing an expressive deep neural network. We will also show that some existing objectives, such as generative models and RLHF, can learn the contexture of a *mixture* of multiple contexts.

Some of the objectives we are going to discuss extract the top-d eigenspace of $T_{P^+}\Lambda T_{P^+}^*$ instead, where $\Lambda : L^2(P_\mathcal{A}) \to L^2(P_\mathcal{A})$ is the integral operator of a kernel k_Λ called the **loss kernel**. The loss kernel depends on the loss function. In this case, since the constant function is not necessarily the top-1 eigenfunction of $T_{P^+}\Lambda T_{P^+}^*$, we do not exclude any eigenfunction.

This chapter does not discuss the numerical aspect of obtaining the optima of these objectives. Doing so in a generalizable way requires an expressive model architecture

and a good optimizer, which this thesis will not discuss. In fact, [28] showed that neural networks trained with popular gradient methods such as Adam [86] will not converge to any point, but will oscillate around what they termed the *edge of stability*. How to extend the contexture theory to that situation is an open problem. Moreover, all objectives to be discussed are spectral (that is L^2) rather than information theoretic, because the contexture theory is based on the spectral properties of T_{P^+}. For example, the mean squared error (MSE) is used for classification tasks instead of the cross entropy loss, and the spectral contrastive loss [57] is used for contrastive learning instead of the NT-Xent loss [22]. How to extend the contexture theory to information theoretic loss is posed as an open problem.

The important implication of the analysis in this chapter is that one key role of scaling up the model size is to bring the learned representation space more aligned to the one spanned by the top-d eigenfunctions of $T_{k_X^+}$. This will be empirically demonstrated in Section 2.6. Consequently, when the two spaces have already become close enough, additional scaling will be less helpful. This is a major reason why scaling has been achieving a diminishing return recently. Further improvement requires the creation of new contexts. We envision that the next major breakthrough in pretraining will be a result of *context scaling*, where very powerful and complicated contexts are obtained from an enormous amount of data, rather than human heuristics.

Notation: For any $f \in L^2(P_{\mathcal{X}})$, denote its mean by $\bar{f} = \mathbb{E}_{P_{\mathcal{X}}}[f(X)]$, and its centered version by $\tilde{f} = f - \bar{f}$. The same notation is used for multi-dimensional functions and random variables, as long as the distribution is clear from context.

Definition 2.1. *The **covariance matrix** of any $\Phi : \mathcal{X} \to \mathbb{R}^d$, denoted by $\mathrm{Cov}_{P_{\mathcal{X}}}[\Phi]$, is a $d \times d$ matrix $\boldsymbol{C}$ where $\boldsymbol{C}[i,j] = \left\langle \tilde{\phi}_i, \tilde{\phi}_j \right\rangle_{P_{\mathcal{X}}}$.*

2.1 Three Illustrative Examples

Let us revisit the three examples in Section 1.4, namely supervised learning, learning with a graph and learning with a random transformation. Through these examples, the reader can get a sense of how to learn the contexture via variational objectives.

Classification tasks. Let there be C classes, and let A be a C-dimensional one-hot vector. Let the predictor be a linear predictor on top of Φ, defined as $\boldsymbol{W}\Phi(x)+\boldsymbol{b}$. For a neural network, $\Phi(x)$ is the output of the layer before the last linear layer. If $\boldsymbol{b}$ is an arbitrary vector, then the linear predictor is biased; if $\boldsymbol{b} = \boldsymbol{0}$ is fixed, then it is unbiased. Consider training an unbiased linear predictor with the mean squared error:

$$\mathcal{R}(\Phi) = \min_{\boldsymbol{W} \in \mathbb{R}^{C \times d}} \mathop{\mathbb{E}}_{(X,A) \sim P^+} \left[\|\boldsymbol{W}\Phi(X) - A\|_2^2 \right]. \tag{2.1}$$

Theorem 2.2 (Proof in Appendix A.1)**.** *Suppose A is a one-hot vector. Then, Φ^* minimizes Eqn.* (2.1) *if and only if Φ^* extracts the top-d eigenspace of $T_{P^+}\Lambda T_{P^+}^*$, where Λ is the integral operator of $k_\Lambda(a,a') = \mathbb{I}[a = a']$, and $(\Lambda g)(a) = g(a)P_{\mathcal{A}}(a)$. If all classes have the same size, then $T_{P^+}\Lambda T_{P^+}^*$ and $T_{P^+}T_{P^+}^*$ share the same top-d eigenfunctions.*

Remark 2.3. *Note that the constant function is not necessarily the top eigenfunction of $T_{P^+}\Lambda T_{P^+}^*$, so in this result no eigenfunction is excluded, which is different from Definition 1.7 where $\mu_0 \equiv 1$ is excluded.*

It turns out that Λ is a consequence of **class imbalance**. In Section 1.4 we showed that the top eigenfunctions of $T_{P^+}T_{P^+}^*$ are indicator functions for the C classes, and these functions are independent of the class sizes. However, in practice we know that when there is a class imbalance, the smaller classes are harder to learn. The operator Λ gives more weights to larger classes, and as a result the larger classes have more impact on the top-d eigenfunctions. With this insight, in order to get rid of Λ, we can use the following balanced loss, also known as **importance weighting** [128]:

$$\mathcal{R}(\Phi) = \min_{\boldsymbol{W}\in\mathbb{R}^{C\times d}} \mathop{\mathbb{E}}_{(X,A)\sim P^+} \left[\frac{1}{\sqrt{P_{\mathcal{A}}(A)}} \|\boldsymbol{W}\Phi(X) + \boldsymbol{b} - A\|_2^2 \right]. \tag{2.2}$$

Theorem 2.4 (Proof in Appendix A.2)**.** Φ^* *minimizes Eqn.* (2.2) *if and only if* Φ^* *learns the contexture of* P^+*.*

Remark 2.5. *Compared to the original objective in* [128]*, the denominator in Eqn.* (2.2) *is* $\sqrt{P_{\mathcal{A}}(A)}$ *instead of the original* $P_{\mathcal{A}}(A)$*.*

The above results can partially explain the phenomenon of **neural collapse** [112]: When the label A is deterministic, and there are d classes of the same size, neural collapse is the phenomenon that a sufficiently trained deep representation collapses to an **equiangular tight frame (ETF)** $\phi_1, \cdots, \phi_d$, where $\phi_i(x) = c(\mathbb{I}[x \text{ belongs to class } i] - d^{-1})$ for some non-zero constant c. Note that the span of $\phi_1, \cdots, \phi_d$ is the same as the span of $\mu_0, \cdots, \mu_{d-1}$, computed in Section 1.4. However, the above results cannot explain why the representation exactly converges to these d functions. To explain this, one needs to analyze the training dynamics, which depends on the specific optimizer such as gradient methods, whereas all results proved in this chapter are independent of the optimizer.

When the classes have different sizes, it is easy to see that the dual kernel of $T_{P^+}\Lambda T_{P^+}^*$ is $k_X^+(x, x') = \mathbb{I}[x \text{ and } x' \text{ have the same label}]$. This is equivalent to the simplex-encoded labels interpolation (SELI) defined by [138], which generalizes neural collapse. When $\mathcal{X}$ is a finite set, the SEL matrix defined in their Definition 2 is the centered kernel of k_X^+.

Regression tasks. Let $A \in \mathbb{R}^{d_{\mathcal{A}}}$ be a real-valued random variable, and consider training Φ using the mean squared error:

$$\mathcal{R}(\Phi) = \min_{\boldsymbol{W}\in\mathbb{R}^{d_{\mathcal{A}}\times d}, \boldsymbol{b}\in\mathbb{R}^{d_{\mathcal{A}}}} \mathop{\mathbb{E}}_{(X,A)\sim P^+} \left[\|\boldsymbol{W}\Phi(X) + \boldsymbol{b} - A\|_2^2 \right]. \tag{2.3}$$

Theorem 2.6 (Proof in Appendix A.3)**.** Φ^* *minimizes Eqn.* (2.3) *if and only if* Φ^* *extracts the top-*d *eigenspace of* $T_{P^+}\Lambda T_{P^+}^*$*. If the linear predictor is unbiased* ($\boldsymbol{b} = \boldsymbol{0}$)*, then* $k_\Lambda(a, a') = \langle a, a' \rangle$*; if it is biased* ($\boldsymbol{b}$ *can be arbitrary*)*, then* $k_\Lambda(a, a') = \langle \tilde{a}, \tilde{a'} \rangle$*.*

Remark 2.7. *Kernel* $k(a, a') = \langle a, a' \rangle$ *is called the* ***linear kernel****, and* $k(a, a') = \langle \tilde{a}, \tilde{a'} \rangle$ *is called the* ***centered linear kernel*** *w.r.t. distribution* $P_{\mathcal{X}}$*. Theorem 2.2 is a special case of Theorem 2.6.*

Graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph. Let the weight of each edge $w(u, v)$ be non-negative. Let $D(u) = \sum_{v\in\mathcal{V}} w(u, v)$ and $D_{\text{sum}} = \sum_{u\in\mathcal{V}} D(u)$. Define a node distribution $P_{\mathcal{X}}(u) = \frac{D(u)}{D_{\text{sum}}}$. Define the context as $\mathcal{A} = \mathcal{V}$, and $P^+(u, v) = \frac{w(u,v)}{D_{\text{sum}}}$. Then, the following constrained optimization problem learns the contexture:

$$\underset{\Phi:\mathcal{X}\to\mathbb{R}^d}{\text{minimize}} \quad \frac{1}{2}\mathbb{E}_{(u,v)\sim P^+}\left[\|\Phi(u) - \Phi(v)\|_2^2\right] \qquad \text{s.t.} \qquad \text{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}. \tag{2.4}$$

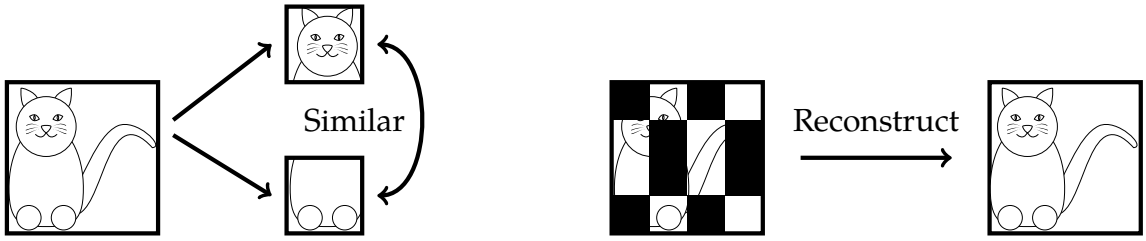


Figure 2.1: Two widely used self-supervised learning algorithms with random transformations. **Left:** Multi-view learning. **Right:** Reconstruction.

The constraint $\mathrm{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}$ is called the **orthonormality constraint**. This constraint is necessary because without it, a degenerate solution could be Φ being a constant function, which is called the **feature collapse** problem. Implementing this constraint in practice is not easy. The most straightforward approach is to minimize the Lagrangian of this optimization problem, in which the constraint is implemented as a penalty term, such as in VICReg [6]. More details will be discussed later, and for now let us assume that this constraint can be enforced.

Theorem 2.8 (Proof in Appendix A.4)**.** *Let Φ^* be any solution to Eqn.* (2.4) (*so that for any constant c, $\Phi^* + c$ is also a solution*)*. Then, $\tilde{\Phi}^*$ learns the contexture of P^+.*

Self-supervised learning (SSL) with random transformations. SSL is usually based on data augmentation, a random transformation that does not alter the semantics of the input by too much. There are two popular methods: multi-view learning and reconstruction, as illustrated in Figure 2.1. In multi-view learning, one draws two **views** that are transformed from the same input, and enforces the encoder to give similar embeddings to these two views. In reconstruction, one trains a neural network with the goal of mapping each view A to its original input X. Then, the penultimate layer of this neural networks is taken as the representation.

Let us start with multi-view learning. If A, A^+ are two views independently drawn from $P^+(\cdot|X)$ for the same X, then (A, A^+) is called a **positive pair**. If A, A^- are independently drawn from $P_{\mathcal{A}}$, meaning that they can be views of different inputs, then (A, A^-) is called a **negative pair**. Multi-view learning trains an encoder $\Psi : \mathcal{A} \to \mathbb{R}^d$ by enforcing $\Psi(A) \approx \Psi(A^+)$. Note that Ψ is an encoder on $\mathcal{A}$, while the required Φ should be an encoder on $\mathcal{X}$. Although it has been common practice to directly apply Ψ to $\mathcal{X}$ at downstream whenever possible, such a practice is not theoretically correct. For example, a BERT [34] is trained on masked sentences and it never sees a complete sentence at pretrain time; as such, one cannot assume that the embeddings it gives to complete sentences retain 100% quality, even though it works well on many real tasks. The theoretically correct way of using Ψ is converting it to Φ by means of the **average encoder**:

$$\Phi = T_{P^+}\Psi; \qquad \Phi(x) = \mathbb{E}_{A\sim P^+(\cdot|x)}[\Psi(A)].$$

For any input x, $\Phi(x)$ can be estimated via Monte Carlo: first sample a number of A from $P^+(\cdot|x)$, and then take the mean of $\Psi(A)$. This Monte Carlo estimation requires T-access to the context, as defined in Definition 1.10.

Multi-view learning also has the feature collapse problem. If only $\Psi(A) \approx \Psi(A^+)$ is enforced, then one degenerate solution is to give all A the same embedding. There are two popular solutions to this problem: contrastive learning and non-contrastive learning. Contrastive learning brings the embeddings of negative pairs far apart, that is in-

creasing the distance between $\Psi(A)$ and $\Psi(A^-)$. Non-contrastive learning has two types: asymmetry-based and constraint-based. The asymmetry-based approach applies two encoders with slight (training or architectural) differences to the same x. For example, BYOL [50] updates one encoder with exponential moving average (EMA) and the other in the standard way to create asymmetry, and SimSiam [24] puts stop gradient on one of the encoders but not the other. The constraint-based approach, such as Barlow Twins [163] and VICReg [6], uses the same orthonormality constraint as Eqn. (2.4).

Let us demonstrate that multi-view learning can learn the contexture with two example objectives. The spectral contrastive loss [57] is given by

$$\mathcal{R}(\Psi) = \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A,A^+\sim P^+(\cdot|X)} \mathop{\mathbb{E}}_{A^-\sim P_{\mathcal{A}}} \left[-\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^+) \right\rangle + \frac{1}{2} \left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^-) \right\rangle^2 \right]; \tag{2.5}$$

and constraint-based non-contrastive learning solves the following problem:

$$\underset{\Psi:\mathcal{A}\to\mathbb{R}^d}{\text{minimize}} \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A,A^+\sim P^+(\cdot|X)} \left[\left\| \Psi(A) - \Psi(A^+) \right\|_2^2 \right] \qquad \text{s.t.} \qquad \mathrm{Cov}_{P_{\mathcal{A}}}[\Psi] = \boldsymbol{I}. \tag{2.6}$$

Theorem 2.9 (Proof in Appendix A.5)**.** *Let Ψ^* be any minimizer of Eqn.* (2.5)*, or any optimal solution to Eqn.* (2.6)*. Then, $\tilde{\Psi}^*$ extracts the top-d eigenspace of $T^*_{P^+}T_{P^+}$, and $\tilde{\Phi}^* = T_{P^+}\tilde{\Psi}^*$ learns the contexture of P^+.*

For reconstruction, suppose $\mathcal{X} \subseteq \mathbb{R}^{d_X}$, and let the predictor be $\boldsymbol{W}\Psi(a) + \boldsymbol{b}$, where $\boldsymbol{b}$ can be $\mathbf{0}$ if the predictor is unbiased. The pretraining objective is given by

$$\mathcal{R}(\Psi) = \min_{\boldsymbol{W}\in\mathbb{R}^{d_X\times d}, \boldsymbol{b}\in\mathbb{R}^{d_X}} \mathop{\mathbb{E}}_{(X,A)\sim P^+} \left[\|\boldsymbol{W}\Psi(A) + \boldsymbol{b} - X\|_2^2 \right]. \tag{2.7}$$

Theorem 2.10. *Let Ψ^* be any minimizer of Eqn.* (2.7)*. Then, $\tilde{\Psi}^*$ extracts the top-d eigenspace of $T_{P^+}\Lambda T^*_{P^+}$, where Λ is the integral operator of $k_\Lambda(x, x') = \langle \tilde{x}, \tilde{x}' \rangle$ if the predictor is biased, or $k_\Lambda(x, x') = \langle x, x' \rangle$ if the predictor is unbiased.*

Proof The proof is the same as Theorem 2.6, which is left as an exercise. ■

The three illustrate examples demonstrate that many existing popular variational objectives can already learn the contexture. The next step is to make them more general so that they can be adapted to a wider variety of contexts.

2.2 General Objectives: SVME and KISE

Section 1.4 introduced three types of access to a context: pair access, kernel access (k-access), and transformation access (T-access). This section presents two general objectives: SVME for pair access, and KISE for k-access. SVME can also be used for T-access since it is stronger than pair access.

To motivate SVME, think about the weaknesses of contrastive and non-contrastive learning objectives in Eqns. (2.5) and (2.6). First, they require the stronger T-access, since for pair access, it is not always possible to draw two views of the same x. Second, using the average encoder is not ideal since it slows down inference.

Single-view multi-encoder (SVME) learning addresses both weaknesses. It produces the encoder Φ directly, and needs only one view for each x; thus, it only requires

pair access. It does so at the cost of training more than one encoders. SVME with two encoders is formulated as the following optimization problem.

$$\textbf{SVME:} \quad \underset{\Phi:\mathcal{X}\to\mathbb{R}^d;\ \Psi:\mathcal{A}\to\mathbb{R}^d}{\text{minimize}} \ \underset{(X,A)\sim P^+}{\mathbb{E}} \left[\|\Phi(X)-\Psi(A)\|_2^2\right] \quad \text{s.t.} \quad \mathrm{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}. \tag{2.8}$$

SVME can be viewed as a combination of asymmetry-based and constraint-based non-contrastive learning. It uses a similar double-encoder architecture as asymmetry-based non-contrastive learning, and the asymmetry here is that the constraint is only imposed on Φ but not Ψ. Meanwhile, it uses the same orthonormality constraint as mentioned earlier. If $\mathcal{A} = \mathcal{X}$, then similar to BYOL and SimSiam, one can implement Φ and Ψ as two heads on top of the same neural network backbone. SVME can have more than two encoders, which we shall see later in this chapter.

SVME is inspired by multi-modal learning. For example, in vision-language models like CLIP [116], Φ can be the image encoder, and Ψ can be the text encoder.

For k-access, usually k is an approximation of the dual kernel. For simplicity, let us assume that we have access to $k_X^+(x,x')$ for all x, x'. Then, for any $f \in L^2(P_{\mathcal{X}})$, we can estimate $T_{k_X^+} f$ with Monte Carlo given a set of inputs. The objective of **kernel-integral single-encoder (KISE)** learning is formulated as

$$\textbf{KISE:} \quad \underset{\Phi:\mathcal{X}\to\mathbb{R}^d}{\text{minimize}} \ \underset{X\sim P_{\mathcal{X}}}{\mathbb{E}} \left[\left\|\tilde{\Phi}(X)\right\|_2^2 - \left\langle \tilde{\Phi}(X), T_{k_X^+}\tilde{\Phi}(X)\right\rangle\right] \quad \text{s.t.} \quad \mathrm{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}. \tag{2.9}$$

Different from SVME, KISE only trains one encoder Φ. KISE is similar to the spectral inference network (SpIN) method proposed by [113]. SpIN maximizes $\left\langle \tilde{\Phi}(X), T_{k_X^+}\tilde{\Phi}(X)\right\rangle$, while KISE is more numerically stable when using gradient methods because the objective is lower bounded by zero, provided that all eigenvalues of $T_{k_X^+}$ are in $[0,1]$. Maximizing SpIN might cause the model weights to explode, but minimizing KISE will not.

The following result shows that both SVME and KISE can learn the contexture of P^+, and they are in fact equivalent.

Theorem 2.11 (Proof in Appendix A.6)**.** *In Eqn.* (2.8)*, if Ψ is substituted with the optimal Ψ when Φ is fixed, then this problem becomes equivalent to Eqn.* (2.9)*. Let Φ^* be any optimal solution to Eqn.* (2.9)*, then $\tilde{\Phi}^*$ learns the contexture of P^+.*

Converting k-access to T-access. There is an additional way of using k-access, which requires the kernel k to be always non-negative. The idea is similar to the objective for graphs that we saw earlier. Define the degree D as $D(x) = \int k(x,x')dP_{\mathcal{X}}(x')$, and define a new context P^+ as $P^+(a|x) = k(x,a)P_{\mathcal{X}}(a)/D(x)$, where $\mathcal{A} = \mathcal{X}$. One has T-access to this context if one can access $k(x,x')$ for all x, x'.

If k is the dual kernel of the original context, then we have $D(x) \equiv 1$. For any eigenfunction μ_i of T_k with eigenvalue s_i^2, we have $T_{P^+}\mu_i = s_i^2\mu_i$. Since obviously $T_{P^+}^* = T_{P^+}$, μ_i is a singular function of T_{P^+} with singular value s_i^2. Hence, the context of P^+ has the same singular functions as the original context, with all singular values squared.

Implementing the orthonormality constraint. The orthonormality constraint in SVME and KISE can be implemented by VICReg [6]. Let $\{(x_1,a_1),\cdots,(x_m,a_m)\}$ be a batch of training samples. The VICReg objective is the sum of an invariance loss, a variance loss and the covariance loss, defined as

$$\mathcal{L}(\Phi,\Psi) = \frac{1}{m}\sum_{k=1}^m \|\Phi(x_k)-\Psi(a_k)\|_2^2 + \frac{\alpha}{d}\sum_{i=1}^d \left(1-\sqrt{\boldsymbol{C}[i,i]+\epsilon}\right)_+ + \frac{\beta}{d(d-1)}\sum_{i\neq j}\boldsymbol{C}[i,j]^2,$$

where $(x)_+ = \max\{0, x\}$ is the ReLU function, ϵ is a small positive constant for numerical stability, and $\boldsymbol{C}$ is the empirical covariance matrix: $\boldsymbol{C}[i,j] = \frac{1}{m-1}\sum_{k=1}^{m} \tilde{\phi}_i(x_k)\tilde{\phi}_j(x_k)$. α, β are two positive hyperparameters. The first term is the invariance loss that aligns the two encoders. The second term is the variance loss that pulls $\left\|\tilde{\phi}\right\|_{P_{\mathcal{X}}}$ closer to 1. The third term is the covariance loss that makes the d dimensions orthogonal to each other.

Although VICReg is used a lot in our experiments, we observe that the two regularization terms cannot enforce the orthonormality constraint perfectly. In fact, both the variance loss and the covariance loss cannot converge to zero. Alternatively, we tried projecting Φ to become orthonormal after each epoch, but doing so does not improve the performance of the encoder. Finding better ways than VICReg to enforce the orthonormality constraint is posed as an open problem.

2.3 Distilling Knowledge from Teacher Models

This part discusses how to learn from the context given by a teacher model. Not only does this have lots of applications in practice, but also it gives us a way to convert a context with any access to one with k-access, which will be very useful in later chapters. Teacher models are very common in practice nowadays. Many big tech companies release their large language models or generative models, all of which can be viewed as teacher models. However, using these models usually incurs a cost, and these models are often too large to be deployed locally (especially in academic labs). As a result, knowledge distillation [63] becomes very attractive—people would like to distill their knowledge to smaller and more affordable models of their own.

Let $\Phi_t : \mathcal{X} \to \mathbb{R}^{d_t}$ be a teacher model, which is presumably learned from some context variable A. If the model is close-sourced, then we cannot see A at all, and we may not even know what space $\mathcal{A}$ is. Even if the model is open-sourced and we know the space $\mathcal{A}$, very few companies release A as a part of their pretraining data. Indeed, it appears that the quality of Φ_t largely depends on the quality of A (the quality of the context), so most companies keep it as their business secret.

Even though we might not know A, it is still possible to distill the knowledge of Φ_t if we can query Φ_t for a sufficient number of times. We can construct its centered linear kernel $k_t(x, x') = \left\langle \tilde{\Phi}_t(x), \tilde{\Phi}_t(x') \right\rangle$, and use KISE to extract its top eigenspace. It is easy to see that $\mu_0 \equiv 1$ is an eigenfunction of k_t with eigenvalue 0.

Two remarks on this method. First, we still cannot access $P_{\mathcal{X}}$ if the data is not released. If we use our own data from another distribution during distillation, then the contexture will be skewed. Second, the linear kernel k_t of the teacher model is not equal to k_X^+. In fact, it contains the information of at most the top-d_t eigenfunctions of k_X^+, which nonetheless is already sufficient for knowledge distillation purpose.

In addition to KISE, we can use the following objective for distillation:

$$\mathcal{R}(\Phi) = \min_{\boldsymbol{W}\in\mathbb{R}^{d_t\times d},\, \boldsymbol{b}\in\mathbb{R}^{d_t}} \mathbb{E}_{X\sim P_{\mathcal{X}}}\left[\|\boldsymbol{W}\Phi(X) + \boldsymbol{b} - \Phi_t(X)\|_2^2\right]. \tag{2.10}$$

This objective extracts the top-d eigenspace of T_{k_t}, with a (centered) linear kernel as the loss kernel. This can be proved in the same way as Theorem 2.6. We leave this proof as an exercise. Let $T_{k_X^+}$ be the dual kernel of the original context Φ_t was trained on. If Φ_t extracts the top-d_t eigenspace of $T_{k_X^+}$ and preserves their order as well, and $d \le d_t$, then apparently the Φ we learn will extract the top-d eigenspace of $T_{k_X^+}$.

However, one issue is that Φ_t might not preserve the original order of the eigenfunctions. Among all the objectives we have discussed so far, only the spectral contrastive loss Eqn. (2.5) can preserve the original eigenvalues; all the other objectives can only recover the space spanned by the top eigenfunctions. Therefore, given a teacher model, it is unsafe to assume that it preserves the original order of the eigenfunctions. One way to completely erase the eigenfunctions of the teacher model is to whiten Φ_t before constructing k_t, that is multiplying $\tilde{\Phi}_t$ by $[\mathrm{Cov}(\Phi_t)]^{-1/2}$. The resulting k_t will have the same eigenfunctions as the original k_t, but its eigenvalues become all either 0 or 1. This is called a spectrally transformed kernel (STK), which will be discussed in more detail in Chapter 5. Note that whitening is not useful for knowledge distillation from a single teacher model, but will be quite useful later when there are multiple contexts.

Converting any access to k-access. Any pretrained encoder can be a teacher model. Thus, for any context, we can first pretrain an encoder with it, view the encoder as a teacher model, and then obtain a context with k-access using the above approach. This also makes storing contexts very simple. Suppose a context is only available for a limited amount of time, for example due to copyright limitations. All we need to do is to pretrain an encoder using the context and store it. Then, we can use the context whenever we want later on.

Social impact. Whether knowledge distillation constitutes a copyright infringement is an important problem. Recently, DeepSeek [53] showed that it is possible to use a fairly small budget to distill the knowledge of an OpenAI model that cost billions of dollars to pretrain. Moreover, the analysis above shows that such knowledge distillation cannot be prevented by making the model close-sourced. As such, big tech companies might be less and less incentivized to grant public access to their models at a low price, fearing that it would be too easy for other companies to copy their work. This might slow down the development of AI.

2.4 Learning from a Mixture of Contexts

The objectives discussed so far can only learn from one context. In practice, it is often the case that there are multiple training signals that we can leverage. For example, there might be multiple random transformations, such as translation, flipping, cropping and color distortion for images. There could also be multiple labels for each sample, or labels and a graph. Here we demonstrate that one can learn representations from multiple training signals by mixing multiple contexts, and in fact some existing learning algorithms are implicitly doing this. Specifically, we discuss two base operations: convolution and convex combination. Chapter 4 will provide a more general framework for mixing multiple contexts.

Suppose there are r contexts given by $P_1^+, \cdots, P_r^+$, and $P_{\mathcal{X}}$ is the marginal distribution of every P_j^+. Let $k_{X1}^+, \cdots, k_{Xr}^+$ be their dual kernels. Then, the **convolution** of these contexts is defined to have the top eigenfunctions of $T_{k_{X1}^+} T_{k_{X2}^+} \cdots T_{k_{Xr}^+} \cdots T_{k_{X2}^+} T_{k_{X1}^+}$ as its contexture; and the **convex combination** of these contexts is defined to have the top eigenfunctions of $\sum_{j=1}^r w_j T_{k_{Xj}^+}$ as its contexture, for some fixed non-negative $w_1, \cdots, w_r$. The convolution usually appears when one composes multiple random transformations; for example, when one applies translation, flipping and cropping to the same image. The

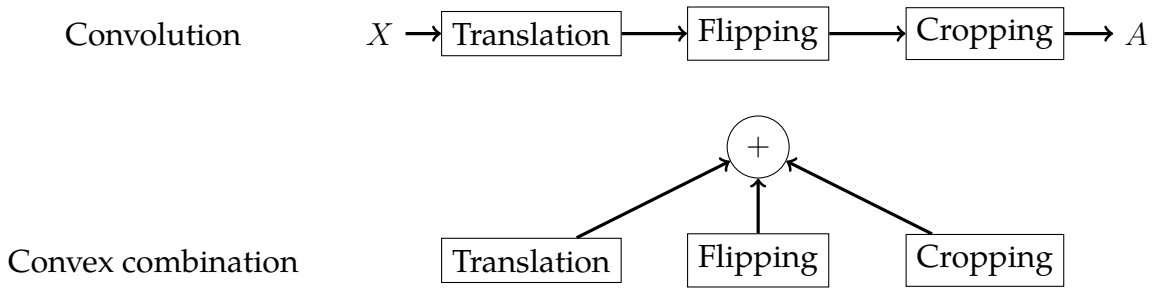


Figure 2.2: Convolution and convex combination of multiple transformations on images.

convex combination appears when the objective is the weighted sum of multiple individual objectives, each of which is designed for one context, as illustrated in Figure 2.2.

Now let us look at three concrete examples: supervised learning with a feature map, multi-step generative models, and RLHF.

Supervised learning with a feature map. When $\mathcal{X}$ is a space of real-world objects, the inputs need to be mapped to numerical vectors via a feature map $\Omega : \mathcal{X} \to \mathbb{R}^{d_\omega}$ so that they can be stored in a computer. For example, if $\mathcal{X}$ is the space of images, then Ω can be PNG or JPEG. The quality of Ω affects the quality of the context. Generally speaking, PNG is better than JPEG because it is lossless, while JPEG loses information.

Assume that Ω is a deterministic mapping. Define operator $T_\Omega : L^2(P_\omega) \to L^2(P_\mathcal{X})$ and its adjoint operator $T_\Omega^* : L^2(P_\mathcal{X}) \to L^2(P_\omega)$ as

$$(T_\Omega h)(x) = h(\Omega(x)); \qquad (T_\Omega^* f)(\omega) = \int f(x) dP(x|\omega).$$

In this scenario, an encoder Φ can be trained as follows. First, train $\Gamma : \mathbb{R}^{d_\omega} \to \mathbb{R}^d$ via supervised learning. Then, define $\Phi = T_\Omega \Gamma$. For classification tasks, similar to Section 2.1, Γ is trained using an unbiased predictor and the mean squared error:

$$\mathcal{R}(\Gamma) = \min_{\boldsymbol{W} \in \mathbb{R}^{d_\mathcal{A} \times d}, \boldsymbol{b} \in \mathbb{R}^{d_\mathcal{A}}} \mathop{\mathbb{E}}_{(X,A) \sim P^+,\, \omega = \Omega(X)} \left[\|A - \boldsymbol{W}\Gamma(\omega) - b\|_2^2 \right]. \tag{2.11}$$

Theorem 2.12 (Proof in Appendix A.8)**.** *Φ^* minimizes Eqn. (2.11) if and only if Γ^* extracts the top-d eigenspace of $T_\Omega^* T_{P^+} \Lambda T_{P^+}^* T_\Omega$ (including $\mu_0 \equiv 1$), where Λ is the integral operator of $k_\Lambda(a, a') = \mathbb{I}[a = a']$. In this case, $\Phi^* = T_\Omega \Gamma^*$ extracts the top-d eigenspace of $T_{k_\Omega} T_{P^+} \Lambda T_{P^+}^* T_{k_\Omega}$, where k_Ω is the dual kernel associated with Ω such that $T_{k_\Omega} = T_\Omega T_\Omega^*$.*

Similar to Theorem 2.4, one can get rid of Λ using the balanced loss. We can see that this Φ^* learns the convolution of two contexts, one given by Ω and the other given by P^+. One important application of supervised learning with a feature map is node representation learning on graphs. Two contexts are available in this application: the graph, and each node has a node feature. Popular methods such as graph neural networks (GNNs) [55] train an encoder $\Gamma(\omega)$, where ω is the node feature.

Multi-step generative models. There are two extremely popular generative models nowadays—large language models and denoising diffusion models [64, 132]. Both models can generate input X from a starting point A in a number of steps. The starting point A may or may not contain information about X. In large language models, A is a prompt, which contains partial information about X, and generation is done token by token; in

diffusion models, A is white noise that has zero information about X, and generation is done by a number of denoising steps.

Although generative models have achieved remarkable success in many applications, what representations these models are learning is quite unclear, and whether these representations can be applied to tasks other than generation is an active research topic. For example, recently [25, 159] studied whether the representations of diffusion models can be used for recognition tasks, and they found that these models "have strong recognition power for understanding the visual content". The common perception is that generative modeling and representation learning are two completely different paradigms in machine learning, but in fact they can be connected via the contexture theory.

A multi-step generative model, such as a diffusion model or a GPT, generates an input x by $a_1 \to a_2 \to \cdots \to a_r \to x$, where a_{j+1} contains more information about x than a_j. The starting point a_1 may or may not contain information about x. For diffusion models, a_j is x plus Gaussian noise; for language models, a_{j+1} is a_j plus one more token at the end. The following is a general formulation of multi-step generative modeling, which does not need to assume the specific form of x or a_j.

Let $\mathcal{A}_j$ be the space of a_j. Let P_j^+ be the joint distribution of X and A_j. We use SVME to train $(r+1)$ encoders. Specifically, we train an encoder $\Psi_j : \mathcal{A}_j \to \mathbb{R}^d$ for every j, along with $\Phi : \mathcal{X} \to \mathbb{R}^d$. Each Ψ_j is trained with the goal of generating x in one shot: Given $a_j \in \mathcal{A}_j$, we find the $\hat{x}$ such that $\Phi(\hat{X})$ is the closest to $\Psi_j(a_j)$, and this $\hat{x}$ should be close to the original x. The training objective for this goal is given by

$$\mathcal{R}_j(\Phi; \Psi_j) = \mathbb{E}_{(X,A_j)\sim P_j^+} \left[\|\Psi_j(A_j) - \Phi(X)\|_2^2\right].$$

Let $\boldsymbol{w} = [w_1, \cdots, w_L]$ be a weight vector where $w_j \geq 0$. The overall objective is

$$\underset{\Phi,\Psi_1,\cdots,\Psi_r}{\text{minimize}} \ \mathcal{R}_{\text{GEN}}(\Phi; \Psi_1, \cdots, \Psi_r) = \sum_{j=1}^{r} w_j \mathcal{R}_j(\Phi; \Psi_j) \quad \text{s.t.} \quad \text{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}. \tag{2.12}$$

Although each Ψ_j is pretrained with the goal of generating x in one shot, actual generation is still performed in multiple steps. Specifically, given $a_j \in \mathcal{A}_j$, we find $a_{j+1} \in \mathcal{A}_{j+1}$ such that $\Psi_{j+1}(a_{j+1})$ is the closest to $\Psi_j(a_j)$. Such a procedure resembles a denoising diffusion model, which trains a noise approximator $\epsilon_\theta(\cdot, j)$ for $j \in [r]$ by closing the gap between x and $a_j - \epsilon_\theta(a_j, j)$ (this objective aims to denoise a_j in one shot), while the actual generation is done in multiple steps.

One great thing about SVME is that even though there are lots of encoders, the constraint is only imposed on Φ. The following result shows that this objective learns the contexture of a convex combination of the r contexts. It is a corollary of the general result in Section 4.2, so the proof will be deferred until then.

Corollary 2.13. *Let $k^+_{X_j}$ be the dual kernel of P_j^+. Let Φ^* be any optimal solution to Eqn.* (2.12)*. Then, $\tilde{\Phi}^*$ extracts the top-d eigenspace of $\sum_j w_j k^+_{X_j}$.*

It should be clarified that both diffusion models and large language models use a single-model architecture, commonly known as a decoder-only architecture, but can also be called an encoder-only architecture if the model is used as a representation. The above analysis does not apply to encoder-decoder architectures such as VAE [87] and GAN [48], because the representations of these encoders are meaningless without the corresponding decoders. The output of VAE and GAN is a Gaussian random vector, whose association with X is indecipherable without the decoder. Therefore, VAE and GAN cannot be analyzed in the same way as above.

Alignment in RLHF. Reinforcement learning with human feedback (RLHF) is a common technique in fine-tuning LLMs. Let Φ_{ref} be a reference model, such as a model trained by supervised fine-tuning (SFT). Let A_1, A_2 be two random variables on $\mathcal{A}$ with possibly different distributions, both of which are associated with X. For instance, X can be an English sentence, and A_1, A_2 can be two Chinese translations of the same sentence. There is also a critic function $C : \mathcal{X} \to \{1, 2\}$, such that for any input x, if $C(x) = 1$, then A_1 is preferred; otherwise, A_2 is preferred. The goal of RLHF is to fine-tune the model to learn such preference (which is also known as alignment), while still keeping the model close to Φ_{ref}. In practice, it is observed that if Φ is too far away from Φ_{ref}, then its performance will be very poor, a phenomenon known as over-optimization [44, 110]. Thus, we need to make sure that Φ does not go too far away from Φ_{ref}.

Let $\mathcal{R}_{\text{align}}(\Phi, \Psi) = \sum_{i=1,2} \Pr_{X\sim P_{\mathcal{X}}}[C(X) = i] \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}}\left[\|\Phi(X) - \Psi(A_i)\|_2^2 \;\middle|\; C(X) = i\right]$ be the alignment loss based on SVME. To implement this loss, for each sample (x, a_1, a_2, c) where c is the critic output, one simply updates the model to minimize $\|\Phi(x) - \Psi(a_1)\|_2^2$ if $c = 1$, and $\|\Phi(x) - \Psi(a_2)\|_2^2$ if $c = 2$. Then, define a loss that reflects the gap between Φ and Φ_{ref} as $\mathcal{R}_{\text{ref}}(\Phi) = \min_{\boldsymbol{W},\boldsymbol{b}} \mathbb{E}\left[\|\boldsymbol{W}\Phi(X) + \boldsymbol{b} - \Phi_{\text{ref}}(X)\|_2^2\right]$. This loss function is invariant under invertible linear transformations on Φ, because such transformations have no impact on the downstream performance when Φ is used with a linear probe. The overall objective is given by

$$\mathop{\text{minimize}}_{\Phi,\Psi} \quad \mathcal{R}_{\text{align}}(\Phi, \Psi) + \beta\mathcal{R}_{\text{ref}}(\Phi) \quad \text{s.t.} \qquad \text{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}$$

for some $\beta > 0$. Similar to generative models, this objective also learns the contexture of a convex combination of two contexts. The first context is given by a the random variable A on $\mathcal{A}$, such that $P^+(x, a) = \Pr[C(x) = 1]P^+_{A_1}(x, a) + \Pr[C(x) = 2]P^+_{A_2}(x, a)$. The second context is provided by the teacher model Φ_{ref}, similar to Eqn. (2.10).

2.5 Extracting Exact Eigenfunctions and Eigenvalues

So far we have seen that a variety of variational objectives can extract the top-d eigenspace of $T_{k_X^+}$, meaning that they can recover the linear space spanned by $\mu_1, \cdots, \mu_d$. One might ask if it is possible to extract the exact top-d eigenfunctions using a variational objective instead of kernel PCA. This is indeed possible. [33] proposed **neural eigenfunctions** to extract the exact top-d eigenfunctions of $T_{k_X^+}$. It solves the following problem:

$$\begin{aligned} \mathop{\text{minimize}}_{\Phi:\mathcal{X}\to\mathbb{R}^d} \quad & \mathbb{E}\left[\left\|\tilde{\Phi}(X)\right\|_2^2 - \left\langle \tilde{\Phi}(X), T_{k_X^+}\tilde{\Phi}(X)\right\rangle\right] \\ \text{s.t.} \quad & \left\|\tilde{\phi}_i\right\|_{P_{\mathcal{X}}}^2 = 1, \quad \forall i \neq j : \left\langle \tilde{\phi}_i, T_{k_X^+}\tilde{\phi}_j\right\rangle_{P_{\mathcal{X}}} = 0. \end{aligned} \tag{2.13}$$

Its difference from KISE is that it changes the constraint $\left\langle \tilde{\phi}_i, \tilde{\phi}_j\right\rangle_{P_{\mathcal{X}}} = 0$, which enforces the different dimensions to be orthogonal, to $\left\langle \tilde{\phi}_i, T_{k_X^+}\tilde{\phi}_j\right\rangle_{P_{\mathcal{X}}} = 0$. Interestingly, this small change allows the extraction of the exact eigenfunctions. It should be emphasized that the d eigenfunctions are obtained simultaneously, not sequentially.

Theorem 2.14 (Proof in Appendix A.7)**.** *Let the optimal Φ^* of Eqn.* (2.13) *be $[\phi_1^*, \cdots, \phi_d^*]$, then we can choose the eigenfunctions $\mu_1, \mu_2, \cdots$ of $T_{k_X^+}$ that have non-increasing eigenvalues and form an ONB of $L^2(P_{\mathcal{X}})$, such that $\tilde{\phi}_1^*, \cdots, \tilde{\phi}_d^*$ is a permutation of $\mu_1, \cdots, \mu_d$.*

With T-access, we can use multi-view learning to extract the exact eigenfunctions.

$$\underset{\Psi:\mathcal{A}\to\mathbb{R}^d}{\text{minimize}}\ \mathbb{E}\left[\left\|\tilde{\Psi}(A)-\tilde{\Psi}(A^+)\right\|_2^2\right] \quad \text{s.t.}\ \ \left\|\tilde{\psi}_i\right\|_{P_\mathcal{A}}^2=1,\ \ \forall i\neq j:\ \mathbb{E}[\tilde{\psi}_i(A)\tilde{\psi}_j(A^+)]=0,$$

where A, A^+ are positive samples drawn from $P^+(\cdot|X)$ of the same X. We can prove that $\tilde{\Psi}$ extracts the exact top-d eigenfunctions of $T_{k_A^+}$ in the same way as Theorem 2.14. And by Lemma 1.4, the average encoder $\tilde{\Phi}=T_{P^+}\tilde{\Psi}$ also extracts the exact top-d eigenfunctions of $T_{k_X^+}$ multiplied by some constants.

Post-hoc approach. Suppose we have a pretrained Φ that learns the contexture, can we obtain the exact eigenvalues and eigenfunctions of $T_{k_X^+}$? The answer is yes. Since Φ does not necessarily contain the information of the exact eigenvalues and eigenfunctions, we still need to use the context, via either kernel access or pair access.

First, suppose we have kernel access to the context. Since $\tilde{\Phi}$ spans the same space as $\mu_1,\cdots,\mu_d$, it suffices to learn a matrix $\boldsymbol{Q}\in\mathbb{R}^{d\times d}$ such that $\tilde{\Phi}\boldsymbol{Q}=[\alpha_1\mu_1,\cdots,\alpha_d\mu_d]$ for some $\alpha_1,\cdots,\alpha_d\neq 0$. Then, these α_i can be eliminated by normalizing $\tilde{\Phi}\boldsymbol{Q}$ to have unit variance in each dimension. The matrix $\boldsymbol{Q}$ and the eigenvalues can be estimated as follows:

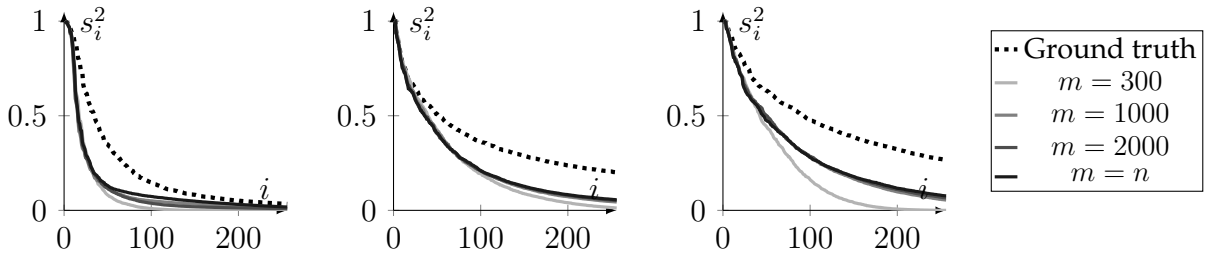
1. Estimate the covariance matrix $\boldsymbol{C}_\Phi\in\mathbb{R}^{d\times d}=\mathrm{Cov}_{P_\mathcal{X}}[\Phi]$ with Monte Carlo.
2. Estimate $\boldsymbol{B}_\Phi\in\mathbb{R}^{d\times d}$, where $\boldsymbol{B}_\Phi[i,j]=\left\langle\tilde{\phi}_i,T_{k_X^+}\tilde{\phi}_j\right\rangle_{P_\mathcal{X}}$, with Monte Carlo.
3. Solve the generalized eigenvalue problem $\boldsymbol{B}_\Phi\boldsymbol{v}=\lambda\boldsymbol{C}_\Phi\boldsymbol{v}$. Let the eigenvalues be $\lambda_1\geq\cdots\geq\lambda_d\geq 0$, and the orthonormal eigenvectors be $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_d$. Then, $\boldsymbol{Q}=[\boldsymbol{v}_1,\cdots,\boldsymbol{v}_d]$, and λ_i is an estimation of the i-th eigenvalue of $T_{k_X^+}$, which is s_i^2. Since d is not very large, this eigen-decomposition is efficient.

Let us elaborate on why this method works. For simplicity, assume that the top-d eigenvalues of $T_{k_X^+}$ are distinct; without this assumption, the result can still be proved with a more verbose proof. Let $U=[\mu_1,\cdots,\mu_d]$, and suppose $\tilde{\Phi}=U\boldsymbol{R}$ for some invertible $\boldsymbol{R}\in\mathbb{R}^{d\times d}$. Since $\mathrm{Cov}[U]=\boldsymbol{I}$ and $\left\langle U,T_{k_X^+}U\right\rangle_{P_\mathcal{X}}=\mathrm{diag}\{s_1^2,\cdots,s_d^2\}$, we have $\boldsymbol{C}_\Phi=\boldsymbol{R}^\top\boldsymbol{R}$ and $\boldsymbol{B}_\Phi=\boldsymbol{R}^\top\mathrm{diag}\{s_1^2,\cdots,s_d^2\}\boldsymbol{R}$. Thus, the generalized eigenvalues are equal to $s_1^2,\cdots,s_d^2$, and $\boldsymbol{R}\boldsymbol{v}_i=\alpha_i\boldsymbol{e}_i$ for some $\alpha_i\neq 0$, where $\boldsymbol{e}_i=[0,\cdots,0,1,0,\cdots,0]$. This implies that $\boldsymbol{R}\boldsymbol{Q}=\mathrm{diag}\{\alpha_1,\cdots,\alpha_d\}$. Hence, $\tilde{\Phi}\boldsymbol{Q}=[\alpha_1\mu_1,\cdots,\alpha_d\mu_d]$.

Second, if we have pair access instead of k-access, then the eigenvalues and $\boldsymbol{Q}$ can be obtained as follows: Let Φ and Ψ be trained via SVME. We can estimate $\boldsymbol{C}_\Phi=\mathrm{Cov}_{P_\mathcal{X}}[\Phi]$ and $\boldsymbol{B}_\Psi=\mathrm{Cov}_{P_\mathcal{A}}[\Psi]$ via Monte Carlo. Then similarly, we solve the generalized eigenvalue problem $\boldsymbol{B}_\Phi\boldsymbol{v}=\lambda\boldsymbol{C}_\Phi\boldsymbol{v}$ to obtain the eigenvalues and $\boldsymbol{Q}$.

Finally, if our goal is only to estimate the eigenvalues but not the eigenfunctions, then Φ need not to be trained with the entire dataset. In fact, [125] showed that for any fixed d, the sum $s_1^2+\cdots+s_d^2$ can be estimated with low error using $\Theta(d)$ *i.i.d.* samples. By union bound, all $s_1^2,\cdots,s_d^2$ can be estimated with low error using $\Theta(d\log d)$ *i.i.d.* samples.

Estimating eigenvalues with the post-hoc approach. Let us demonstrate the post-hoc method on 3 real datasets from OpenML [146]: `abalone`, `fifa`, and `kings_county`. We only extract the eigenvalues here, and later in Section 2.6 we will investigate the eigenfunctions. We use KNN with $K=60$ as context, where $\mathcal{A}=\mathcal{X}$, and $P^+(x'|x)=K^{-1}$ if x' is a K-nearest neighbor of x and 0 otherwise. For this context, we can exactly compute k_X^+, and thus we can obtain the exact eigenvalues (ground truth) using kernel



(a) abalone ($n = 4177$) (b) fifa ($n = 19178$) (c) kings_county ($n = 21613$)

Figure 2.3: Estimating the eigenvalues using the post-hoc approach with m samples.

Dataset	$m = 100$	$m = 300$	$m = 600$	$m = 1000$	$m = 2000$	Full dataset
abalone	0.157	0.124	0.088	0.104	0.110	0.088
fifa	0.218	0.151	0.137	0.134	0.133	0.131
kings_county	0.278	0.264	0.190	0.183	0.177	0.177

Table 2.1: Average estimation error of the top-256 eigenvalues.

PCA. Meanwhile, we pretrain Φ with one of the variational objectives using a random subset of m samples, and estimate the eigenvalues using the post-hoc approach. Then, we compare the estimation with the ground truth.

We use a 2-layer wide Tanh-activated neural network with embedding dimension $d = 512$ and hidden dimension 20,000 as Φ. We train the model through non-contrastive learning Eqn. (2.6), with the orthonormality constraint implemented by VICReg, and AdamW [86, 104] as the optimizer. We vary m and compare the estimated top-d_0 eigenvalues with the ground truth, where $d_0 = 256$. The estimated eigenvalues and the ground truth are plotted in Figure 2.3. From the plots, we observe that the eigenvalues estimated by our estimation method decay faster than the ground truth, even if the full dataset is used. We hypothesize that the main reason is that even though we use a very wide neural network, its function class is still a subset of $L^2(P_{\mathcal{X}})$. Consequently, the inductive bias of the model architecture has an impact on the encoder, and therefore the learned contexture can be viewed as a mixture of the inductive bias and the original KNN context. This mixture causes the eigenvalues to decay faster, which explains the observation in Figure 2.3. Another reason is related to optimization. Since the model is non-convex, gradient methods cannot find the minima of the objective.

The average estimation error of the top-256 eigenvalues is reported in Table 2.1. The error is defined as $\frac{1}{d_0}\sum_{i=1}^{d_0}|\hat{s}_i^2 - s_i^2|$, where $\hat{s}_i^2$ is the estimated eigenvalue. The table shows that when $m \in [600, 1000] \approx [0.5 d_0 \log d_0, 0.7 d_0 \log d_0]$, the performance is comparable to using the full dataset, which verifies the theoretical result of [125]. The estimation error is not zero even if the full dataset is used due to the aforementioned reasons. In summary, the post-hoc method can estimate the eigenvalues using a small subset of samples, but the estimated eigenvalues decay faster than the ground truth.

2.6 Implications on the Scaling Law

It has been widely observed that the performance of deep neural networks on many real tasks increases with the model size, which is known as the **scaling law** [84]. Furthermore, it has been observed that models of different architectures, such as ResNets [61]

and ViTs [36], learn highly aligned representations under the metrics in [89] when the models are sufficiently large, even if they are trained with different objectives.

To explain this phenomenon, [73] proposed the **platonic representation hypothesis**, which states that "neural networks, trained with different objectives on different data and modalities, are converging to a shared statistical model of reality in the representation spaces". The assumptions of this hypothesis are that the neural networks are large enough, and there is a sufficient amount of data.

The contexture theory provides a new perspective on the role of scaling. It implies that the so-called "reality" is in fact the top eigenfunctions of $T_{k_X^+}$. When using a neural network as Φ, the function class of Φ is a subset of $L^2(P_{\mathcal{X}})$; and when scaling up the model size, this subset gets closer to the entire space $L^2(P_{\mathcal{X}})$, and thus the learned representation becomes closer to the top-d eigenfunctions of $T_{k_X^+}$, which are independent of the model architecture (so this is not a special property of neural networks). This also explains why recently it has been observed that increasing the model size is producing a diminishing return. When the model is large enough so that the learned representation is highly aligned with the top-d eigenfunctions, then further increasing the model size will be less useful. [73] also observed that models trained in different modalities all align with the same shared representation. This suggests that commonly used contexts in different modalities have similar top eigenfunctions. Note that this cannot be true for all contexts. For a very weird context such as randomly shuffling the pixels of an image, its top eigenfunctions are surely not aligned with this shared representation.

In this section, we use an experiment to show that the representation learned by a neural network is indeed aligned with the top-d eigenfunctions, which provides empirical evidence to support the above arguments.

Experiment overview. The purpose of this experiment is to examine whether a large neural network can learn the contexture well, and whether scaling up the model size makes the learned representation more aligned to the top-d eigenfunctions. We compare between two encoders. The first encoder is obtained via kernel PCA on the dual kernel, so it consists of the exact top-d eigenfunctions. The second encoder is obtained via training a large neural network to optimize an objective that can learn the contexture. Then, we compute the representational alignment of these two encoders. The most classical metric is the canonical-correlation analysis (CCA) metric R^2_{CCA}, which is invariant under invertible linear transformations to the encoders. [89] proposed a variant called linear CKA, which is only invariant under orthogonal transformations. In our setting, since we only care about the span of $\phi_1, \cdots, \phi_d$, we would like the metric to be invariant under all invertible transformations, which is why we use CCA. In addition, we also use the mutual KNN metric with 10 neighbors proposed by [73], which measures the intersection over union (IoU) of nearest neighbors between the two representations. This metric is not invariant under invertible linear transformations, so we whiten the two representations such that their covariance matrices are both identities.

Setup. We use the `abalone` dataset from OpenML, and split the dataset into a pretrain set, a downstream train set and a downstream test set by 70%-15%-15%. Like what we used earlier, we choose K-nearest neighbors (KNN) with $K = 30$ to be the context. The embedding dimension is set to be $d = 128$. For the second encoder, we train a fully-connected neural network with Tanh activation and skip connections for a sufficient number of steps with full-batch AdamW, and vary the depth and width of the

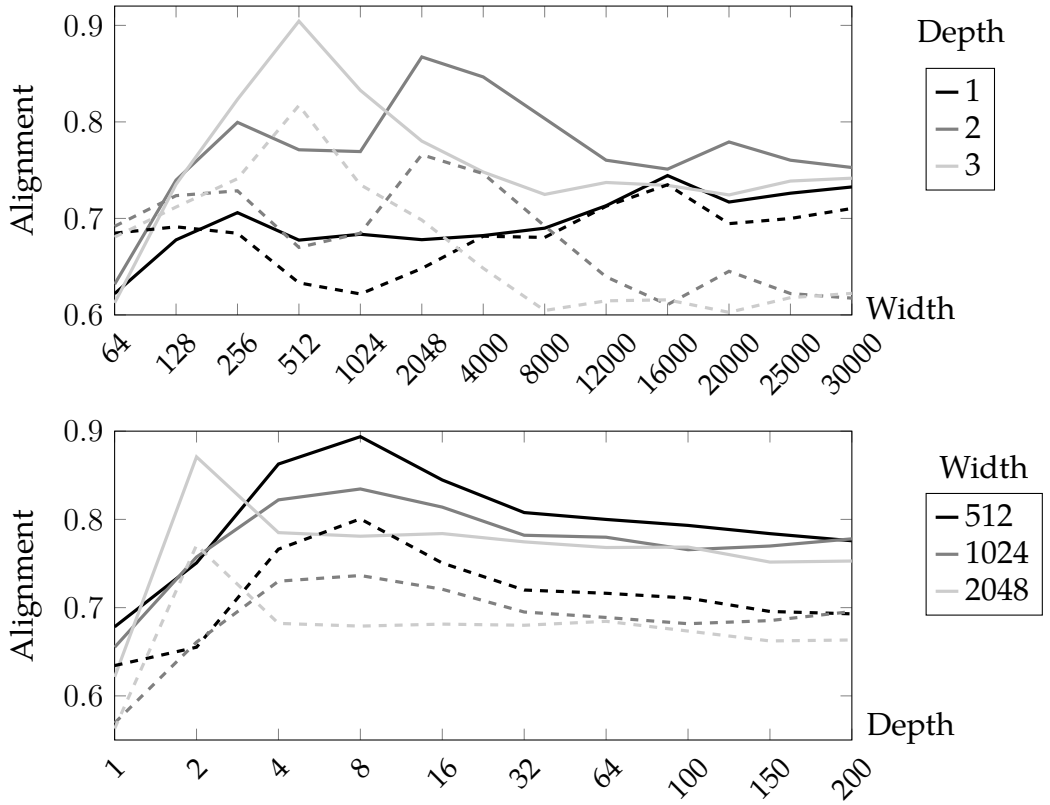


Figure 2.4: Alignment between the learned representation and the top-d eigenfunctions of $T_{k_X^+}$ on the `abalone` dataset. Solid curves: CCA. Dashed curves: mutual KNN. Depth here means the number of hidden layers.

network so that we can study their effect on the alignment. Here, "depth" refers to the number of hidden layers—for example, a 2-layer neural network has depth 1. For each width and depth, we run the experiments 15 times with different random initializations, and report the average alignment.

In our experiments, we observe the **dimension collapse** problem [80]—if we set the output dimension of the neural network to be d, then the rank of the learned representation will usually be less than d, meaning that it can only extract the top-d' eigenspace for some $d' < d$. [80] proved that this problem can be caused by the training dynamics of self-supervised learning, that is a large neural network trained with a gradient method cannot find the minima, but will find a low-rank solution instead.

To fix this issue, we set the output dimension of the neural network to be $d_1 = 512 > d$. After we obtain the d_1-dimensional encoder, similar to Section 2.5 we estimate the matrices $\boldsymbol{C}_\Phi$ and $\boldsymbol{B}_\Phi$, and solve the generalized eigenvalue problem $\boldsymbol{B}_\Phi \boldsymbol{v} = \lambda \boldsymbol{C}_\Phi \boldsymbol{v}$. Let $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d] \in \mathbb{R}^{d_1 \times d}$ be the top-d eigenvectors; then, we use $\tilde{\Phi}\boldsymbol{V}$ as the d-dimensional representation. In other words, we use the 128 principal components of the 512-dimensional embedding.

Results. Figure 2.4 plots the alignment between the two encoders while varying the depth and width of the neural network. we can see that when the depth and width are chosen correctly, the CCA can be as high as 0.9, and the mutual KNN can be over 0.8. Note that these alignment metric values are very high. For example, in [73], the mutual KNN metric value is usually below 0.2. Hence, the representation learned by the neural

network is highly aligned with the top-d eigenfunctions.

The top plot studies neural networks with increasing widths. We can see that when the neural network is not so wide, increasing the width will make the alignment higher. However, once the neural network is wide enough, further increasing the width might have a negative effect. For example, when the depth is 3, the alignment is the highest when the width is 512, and the alignment becomes lower when the network is wider than 512. Since increasing the width can only make the function class of Φ larger, this phenomenon is not due to the expressivity of the neural network. We hypothesize that this is because optimizing a larger model is harder. Consequently, with the same number of pretraining steps, a larger model will be farther away from the minima, and the alignment decreases.

The bottom plot studies neural networks with increasing depths, and the observation is similar. When the network is shallow, increasing the depth makes the alignment higher. However, once the network is deep enough, further increasing the depth might have a negative effect. We also observe from the bottom plot that a width-512 network has higher alignment than widths 1024 and 2048. In addition, the alignment cannot reach 1. This is because the model is non-convex, so the real optima (the precise top-d eigenspace) cannot be found by gradient methods.

In summary, we draw two conclusions from this experiment: (i) the representation learned by a large neural network is highly aligned with the top-d eigenfunctions; (ii) once the neural network is wide and deep enough, further increasing its size will not make the alignment higher, and might even have a negative effect. Hence, we put forward the following argument about the scaling law: Once the model is large enough such that Φ is already highly aligned with the top-d eigenfunctions, further increasing the model size inevitably yields diminishing returns.

When the model is already large enough, a better context is necessary for further improvement. The next part of this thesis studies how to obtain a better context. There are two questions that need to be addressed. First, what context is good and what context is "better"? In particular, if we want the representation to be transferable to a wide variety of downstream tasks, then how should we evaluate the representation without testing it on a specific task? Second, how to obtain better contexts? Creating new contexts from scratch is obviously difficult, so are there easier options? These questions will be addressed in the following two chapters.

Code and data availability. The code for the experiment in this section can be found at `https://1drv.ms/u/c/ea9fe908498c8b82/EWb_t4e-27VKsoSTWV6J_yQBUFsuWSAXbRHJK4GoUEynjw?e=r4ROdU`. The data can be downloaded from OpenML.

Chapter 3

Intrinsic Evaluation: The Optimality of Learning the Contexture

This chapter focuses on the evaluation of encoders and contexts. How to tell if an encoder is good or not? There are two methods as summarized in [15, Sec. 4]. The first method is extrinsic evaluation, where we choose a specific downstream task, fit a predictor on top of the encoder, and use the performance of that predictor as the evaluation of the encoder. In practice, this performance is ultimately what we care about. The second method is intrinsic evaluation, which is independent of any downstream task. Intrinsic evaluation is more useful in the analysis of pretraining for two reasons: (i) We would like the encoder to be transferable to a wide variety of downstream tasks, so only evaluating it on one task is usually insufficient; (ii) Foundation models are often used on tasks they are not designed for, so we cannot assume that we know all the tasks at pretrain time.

Obviously, no encoder is good for all tasks. For any encoder, one can adversarially construct a task for which the encoder is poor. However, such tasks are not what we care about. For example, in NLP, downstream tasks that we care about include sentiment analysis, sentence completion, translation, question answer, etc. Counting how many words in a paragraph contain the letter "e" is a valid task, and large language models are not really good at it (see Figure 3.1), but this task is not one that we would usually care about. Therefore, we need to define a class of relevant tasks, and the intrinsic evaluation of the encoder is conducted on this class.

How to define such a class of relevant tasks at pretrain time? The only information we have at pretrain time is the context, and the prior knowledge that the context should be useful for learning an encoder for any task of interest. Note that a downstream task need not to “look similar” to the context; for example, a BERT trained with masked token prediction can be used for sentiment analysis, though these two tasks look quite different. Thus, we need a quantitative description of the usefulness of a context for a task, for which we define the **compatibility** between a context and a task.

Once we define the class of compatible tasks, we can prove that the encoder that learns the contexture is optimal for this class of tasks, in the sense that it minimizes the worst-case approximation error. On the other hand, if the task is not compatible, then it is likely that the encoder will have a bad performance. This explains why LLMs are so powerful that they can perform well on various tasks, but they struggle with easy tasks such as answering factual questions, where they often produce seemingly correct but completely wrong answers, a phenomenon known as hallucination [69]. In short, answering factual questions is not compatible with the context of next token prediction.

The significance of this result is that it reduces the problem of pretraining to the

```
User: Count how many words in the following paragraph
contains the letter "e":
Overall, although there does not ... the proposed metric
is useful for choosing hyperparameters and comparing
contexts in practice.

Claude 3.7 Sonnet: I'll count the words containing the
letter "e" in the paragraph you provided. Let me go
through it word by word:
1. "Overall" - contains "e"
...
58. "and" - no "e"
59. "comparing" - contains "e"
60. "contexts" - contains "e"
61. "in" - no "e"
62. "practice" - contains "e"
Counting the words with the letter "e": 42 words out
of 62 total words contain the letter "e".
```

Figure 3.1: An example where the LLM Claude 3.7 Sonnet (as of April 14, 2025) makes a mistake on a task that is not quite relevant to NLP. Line 59 is wrong because "comparing" does not contain the letter "e". The final answer is also wrong.

problem of finding a good context. Once we have the context, we can use any variational objective in the last chapter, and the resulting foundation model is guaranteed to have the optimal performance if it can learn the contexture well. We will also discuss how to intrinsically evaluate an arbitrary encoder, which depends on two key concepts—the induced RKHS and the ratio trace. Thus, even if the foundation model does not learn the contexture, we are able to predict its performance on compatible tasks.

Finally, we study what contexts are good contexts. The key finding is that a good context should have a moderate association between X and A, so that the decay rate of the singular values is neither too fast nor too slow. Qualitatively, if the association is too weak, then there will be very few tasks that are compatible with the context, so the resulting encoder will not be transferable to a wide range of tasks. On the other hand, if the association is too strong, then the sample complexity of learning the contexture and the downstream predictor will be very high. Quantitatively, we propose a metric that measures the usefulness of a context. This metric only depends on the singular values, so it can be efficiently estimated using the post-hoc approach described in Section 2.5. Experiments show that the metric correlates well with the actual performance of the encoder on real datasets.

3.1 Compatibility, Optimality of Contexture

A downstream task is represented by a **target function**. Most downstream tasks, such as prediction, clustering, and segmentation, can be associated with a target function $f^* \in L^2(P_{\mathcal{X}})$. For example, multi-class classification can be associated with multiple one-vs-all labeling functions. After Φ is pretrained, and the training samples of the downstream task are revealed, there are a number of ways to use Φ, such as fitting a small neural network on top, using a kernel method, supervised fine-tuning (SFT), etc. This thesis

focuses on the simplest way called a **linear probe**, where one fits the downstream data with a linear predictor on top of Φ. Specifically, the downstream predictor is given by $\hat{f}(x) = \boldsymbol{W}\Phi(x) + \boldsymbol{b}$. Since $\boldsymbol{W}$ and $\boldsymbol{b}$ can be arbitrarily chosen, the mean and variance of f^* have no impact on the difficulty of learning f^*. Hence, our definition of compatibility should be independent of them.

Given a context P^+, we say that a task is compatible with it, if the information the context provides makes it easier to learn a predictor for the task. Formally, consider a training set for supervised learning $\{(a_i, y_i)\}_{i=1}^n$, where $y_i = f^*(x_i) + \xi_i \in \mathbb{R}$ for some small random noise ξ_i. In this dataset, the original inputs x_i are unknown, and only one $a_i \sim P^+(\cdot|x_i)$ is provided for each i. The most straightforward method of learning a predictor for this task is as follows: we first fit a predictor $\hat{g} : \mathcal{A} \to \mathbb{R}$ on this dataset, and then convert it to $\hat{f} : \mathcal{X} \to \mathbb{R}$ with $\hat{f} = T_{P^+}\hat{g}$, assuming that we have T-access to the context. f^* is said to be compatible with the context if this method works well, for which two conditions are necessary:

(i) There exists a $g^* \in L^2(P_{\mathcal{A}})$ such that $f^* = T_{P^+}g^*$.

(ii) The variance of g^* conditioned on x, denoted by $\mathrm{Var}[g^*(A)|x]$, is low on average.

Condition (i) says that the labels y_i can be approximated by a function in the range of T_{P^+}, up to the small noise ξ_i. Condition (ii) ensures that $g^*(a_i) \approx f^*(x_i) \approx y_i$; without this condition, the $\hat{g}$ fit on $\{(a_i, y_i)\}$ cannot approximate g^*.

The compatibility defined below is based on the above insights.

Definition 3.1. *The **compatibility** with P^+ of $f \in L^2(P_{\mathcal{X}})$ such that $\tilde{f} \neq \mathbf{0}$ is defined as*

$$\rho(f, P^+) = \max_{g \in L^2(P_{\mathcal{A}}), g \neq \mathbf{0}} \frac{\left\langle \tilde{f}, T_{P^+}g \right\rangle_{P_{\mathcal{X}}}}{\left\| \tilde{f} \right\|_{P_{\mathcal{X}}} \|g\|_{P_{\mathcal{A}}}} \in [0, 1]. \tag{3.1}$$

The compatibility is defined this way so that it is independent of the mean and variance of f. Here is a formula for the compatibility of any f^* with P^+. For simplicity, let f^* has mean zero. Let $f^* = \sum_{i \geq 1} u_i \mu_i$ and $g^* = \sum_{i \geq 1} v_i \nu_i$. Then, $\rho(f^*, P^+) = \max_{v_i} \frac{\sum_i s_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}} = \sqrt{\frac{\sum s_i^2 u_i^2}{\sum u_i^2}}$ by Cauchy-Schwarz inequality (the maximum is attained when $v_i \propto s_i u_i$). To estimate $\rho(f^*, P^+)$ for any f^*, the most straightforward way is to find the g^* ths maximizes the fraction on the right-hand side. This is much more efficient than training a d-dimensional encoder.

Under this definition, the class of $(1-\epsilon)$-compatible tasks is defined as

$$\mathcal{F}_\epsilon(P^+) = \left\{ f \in L^2(P_{\mathcal{X}}) : \rho(f, P^+) \geq 1 - \epsilon \right\}, \tag{3.2}$$

for any $\epsilon > 0$. When ϵ is small, this is a class of compatible downstream tasks, and it can be shown that this class satisfies the two conditions aforementioned:

Theorem 3.2 (Proof in Appendix B.1)**.** *For any $f^* \in \mathcal{F}_\epsilon(P^+)$, there exists a $g^* \in L^2(P_{\mathcal{A}})$ such that $f^*(x) = \mathbb{E}[g^*(A)|x]$, and g^* satisfies*

$$\mathop{\mathbb{E}}_{X \sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A, A' \sim P^+(\cdot|X)} \left[\left(g^*(A) - g^*(A') \right)^2 \right] \leq 4\epsilon \|g^*\|_{P_{\mathcal{A}}}^2. \tag{3.3}$$

Next, we show why an encoder that learns the contexture is optimal. First, we need to define "optimal", which involves the evaluation of an encoder. The ultimate evaluation

of Φ is the performance of the downstream predictor. Let $f(x) = \boldsymbol{W}\Phi(x) + \boldsymbol{b}$ be the downstream linear probe. The performance of f can be measured by the mean squared error (MSE):

$$\text{err}(f, f^*) = \|f - f^*\|_{P_\mathcal{X}}^2 = \mathbb{E}_{X\sim P_\mathcal{X}}[(f(X) - f^*(X))^2].$$

Let f_Φ be the projection of f^* onto the linear space spanned by $\phi_1, \cdots, \phi_d$. Then, when using a linear probe, the above error can be decomposed as

$$\text{err}(f, f^*) = \underbrace{\|f_\Phi - f^*\|_{P_\mathcal{X}}^2}_{\text{Approximation error}} + \underbrace{\|f - f_\Phi\|_{P_\mathcal{X}}^2}_{\text{Estimation error}}.$$

The embedding dimension d controls the trade-off between the two errors. If d is larger, then the span of Φ will become larger, so the approximation error will be lower; meanwhile, the downstream sample complexity will be higher, so the estimation error will increase. There are two ways in which Φ affects the final prediction error. First, the distance from f^* to the span of Φ decides the approximation error. Second, the smoothness of $\phi_1, \cdots, \phi_d$ affects the sample complexity of both pretraining and downstream. The second part will be studied in Chapter 5, and this chapter evaluates Φ by the approximation error. More specifically, Φ is evaluated on the class of compatible tasks $\mathcal{F}_\epsilon(P^+)$ by its worst-case approximation error.

Definition 3.3. *Let $\mathcal{F} \subset L^2(P_\mathcal{X})$ be a function class where $f \in \mathcal{F} \Rightarrow \alpha f \in \mathcal{F}$ for all $\alpha \in \mathbb{R}$. The **worst-case approximation error** of $\Phi : \mathcal{X} \to \mathbb{R}^d$ on $\mathcal{F}$ is defined as*

$$\text{err}(\Phi; \mathcal{F}) = \max_{f\in\mathcal{F}(P^+),\, \|f\|_{P_\mathcal{X}}=1} \text{err}(\Phi, f);$$

$$\textit{where} \quad \text{err}(\Phi, f) = \mathbb{E}_\Phi\left[\min_{\boldsymbol{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} \left\|\boldsymbol{w}^\top\Phi + b - f\right\|_{P_\mathcal{X}}^2\right].$$

Here, $\mathbb{E}_\Phi$ is taken over the randomness of Φ. When Φ is randomized, one first samples a deterministic Φ from the distribution, and then fits $\boldsymbol{w}$ and b accordingly.

The following result is one of the main results of the contexture theory. It says two things. First, if we know a priori the downstream task is compatible, then learning the contexture is the optimal thing to do, because it minimizes the worst-case approximation error. Second, what if the task is incompatible? We cannot argue that no encoder works for an incompatible task. In a hypothetical scenario, if an oracle tells us the target function f^* in advance, then we can set $\phi_1 = f^*$ to achieve perfect performance. What we can argue is that for any low compatibility level and any encoder Φ, there exists an f^* on that level such that Φ is poor for f^*.

Theorem 3.4 (Proof in Appendix B.2)**.** *Suppose $1 - \epsilon \le s_1$. For any d, among all $\Phi = [\phi_1, \cdots, \phi_d]$ where $\phi_i \in L^2(P_\mathcal{X})$, Φ minimizes $\text{err}(\Phi; \mathcal{F}_\epsilon(P^+))$ if and only if it learns the contexture of T_{P^+}. The error is given by*

$$\min_{\Phi:\mathcal{X}\to\mathbb{R}^d,\, \phi_i\in L^2(P_\mathcal{X})} \text{err}\big(\Phi; \mathcal{F}_\epsilon(P^+)\big) = \frac{s_1^2 - (1-\epsilon)^2}{s_1^2 - s_{d+1}^2}.$$

Conversely, for any d-dimensional encoder Φ and any $\epsilon > 0$, there exists $f \in L^2(P_\mathcal{X})$ such that $\rho(f, P^+) = 1 - \epsilon$, and $\text{err}(\Phi, f) \ge \frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}$.

3.2 Intrinsic Evaluation of an Arbitrary Encoder

Given a context that is compatible with the task, the encoder that learns the contexture is optimal. Now what about an arbitrary encoder Φ? Is it possible to bound its worst-case approximation error on the class of compatible tasks? To derive such a bound, two key objects are necessary: the induced RKHS and the ratio trace.

Denote the range of $T^*_{P^+}$ by $R(T^*_{P^+}) = \{T^*_{P^+} f \mid f \in L^2(P_{\mathcal{X}})\}$.

Definition 3.5. *The **induced RKHS** of P^+, denoted by $\mathcal{H}_{P^+}$, is the Hilbert space $R(T^*_{P^+})$ with the inner product given by $\langle T^*_{P^+} f_1, T^*_{P^+} f_2 \rangle_{\mathcal{H}_{P^+}} = \langle f_1, f_2 \rangle_{P_{\mathcal{X}}}$.*

An alternative formula is that for any $h_1, h_2 \in \mathcal{H}_{P^+}$ where $h_1 = \sum u_i \nu_i$ and $h_2 = \sum v_i \nu_i$, there is $\langle h_1, h_2 \rangle_{\mathcal{H}_{P^+}} = \sum \frac{u_i v_i}{s_i^2}$.

Proposition 3.6. *The induced RKHS $\mathcal{H}_{P^+}$ has the following properties:*

(i) k_A^+ is the reproducing kernel, such that $h(a) = \langle h, k_A^+(a, \cdot) \rangle_{\mathcal{H}_{P^+}}$ for all $h \in \mathcal{H}_{P^+}$.

(ii) $\mathcal{H}_{P^+}$ is isometric to $\operatorname{span}\{\mu_i : s_i > 0\}$, which is a subspace of $L^2(P_{\mathcal{X}})$.

*(iii) $f^* \in \mathcal{F}_\epsilon(P^+)$ is equivalent to $h^* = T^*_{P^+} f^*$ satisfying the following **isometry property**:*

$$(1-\epsilon)\left\|\tilde{h}^*\right\|_{\mathcal{H}_{P^+}} \le \left\|\tilde{h}^*\right\|_{P_{\mathcal{A}}} \le \left\|\tilde{h}^*\right\|_{\mathcal{H}_{P^+}}. \tag{3.4}$$

Proof For any $h \in \mathcal{H}_{P^+}$ where $h = T^*_{P^+} f$ and $f = \sum u_i \mu_i$, by Corollary 1.6 we have

$$\langle h, k_A^+(a, \cdot) \rangle_{\mathcal{H}_{P^+}} = \left\langle \sum s_i u_i \nu_i, \sum s_i^2 \nu_i(a) \nu_i \right\rangle_{\mathcal{H}_{P^+}} = \sum s_i u_i \nu_i(a) = h(a),$$

which proves (i). (ii) is obvious. Regarding (iii), recall that $f^* = \sum u_i \mu_i \in \mathcal{F}_\epsilon(P^+)$ is equivalent to $\sum_{i \ge 1} s_i^2 u_i^2 \ge (1-\epsilon)^2 \sum_{i \ge 1} u_i^2$, and this is $\left\|\tilde{h}^*\right\|_{P_{\mathcal{A}}} \ge (1-\epsilon)\left\|\tilde{h}^*\right\|_{\mathcal{H}_{P^+}}$. Meanwhile, it is obvious that $\left\|\tilde{h}^*\right\|_{P_{\mathcal{A}}} \le \left\|\tilde{h}^*\right\|_{\mathcal{H}_{P^+}}$ always holds. ■

Definition 3.7. *Define covariance matrices $\boldsymbol{C}_\Phi = \operatorname{Cov}_{P_{\mathcal{X}}}[\Phi]$, and $\boldsymbol{B}_\Phi = \operatorname{Cov}_{P_{\mathcal{A}}}[T^*_{P^+}\Phi]$. If $\boldsymbol{C}_\Phi$ is invertible, then the **ratio trace** of Φ w.r.t. P^+ is defined as $\operatorname{RT}(\Phi; P^+) = \operatorname{RT}(\phi_1, \cdots, \phi_d; P^+) := \operatorname{Tr}(\boldsymbol{C}_\Phi^{-1} \boldsymbol{B}_\Phi)$; otherwise, let $\Phi' = [\phi_{i_1}, \cdots, \phi_{i_t}]$ be the maximal linearly independent subset of $[\phi_1, \cdots, \phi_d]$, and define the ratio trace of Φ the same as the ratio trace of Φ'.*

The ratio trace of any Φ essentially measures how well Φ is aligned with the contexture of P^+. Multiplying Φ by any invertible matrix does not change its ratio trace. The matrices $\boldsymbol{C}_\Phi$ and $\boldsymbol{B}_\Phi$ here are the same as in Section 2.5. If Φ learns the contexture, then its ratio trace is $s_1^2 + \cdots + s_d^2$, which can be easily shown by setting $\phi_i = \mu_i$. In fact, this is the maximum ratio trace of any d-dimensional encoder.

Lemma 3.8. *Suppose $\phi_1, \cdots, \phi_d$ are orthonormal and all have zero mean. Then, we have*

$$\|T^*_{P^+}\phi_1\|^2_{P_{\mathcal{A}}} + \cdots + \|T^*_{P^+}\phi_d\|^2_{P_{\mathcal{A}}} \le s_1^2 + \cdots + s_d^2.$$

Proof Let $\phi_i = \sum_{j \ge 1} q_{ij} \mu_j$ for $i \in [d]$. Then, $\boldsymbol{Q} = (q_{ij})$ is a matrix with d orthonormal rows and infinitely many columns. It is easy to see that the left-hand side is equal to $\operatorname{Tr}(\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top)$, where $\boldsymbol{D} = \operatorname{diag}\{s_1^2, s_2^2, \cdots\}$. Let $\boldsymbol{q}_j$ be the j-th column of $\boldsymbol{Q}$. For all $j \in [d]$, there is $\sum_{i=1}^j \boldsymbol{q}_i^\top \boldsymbol{q}_i \le j$; and for any $j > d$, there is $\sum_{i=1}^j \boldsymbol{q}_i^\top \boldsymbol{q}_i \le d$. Thus, using Abel

transformation, we have

$$\mathrm{Tr}(\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top) = \mathrm{Tr}(\boldsymbol{D}\boldsymbol{Q}^\top\boldsymbol{Q}) = \sum_{j=1}^{\infty} s_j^2 \boldsymbol{q}_j^\top \boldsymbol{q}_j = \sum_{j=1}^{\infty}\left(\sum_{i=1}^{j} \boldsymbol{q}_i^\top \boldsymbol{q}_i\right)\left(s_j^2 - s_{j+1}^2\right) \le \sum_{j=1}^{d} s_j^2,$$

as desired. ■

The ratio trace induces a key quantity in the approximation error bound called the trace gap, which reflects the gap between Φ and the top-d singular functions. The larger the trace gap is, the larger the approximation error will be. A simple definition is $s_1^2 + \cdots + s_{d+1}^2 - \mathrm{RT}(\Phi; P^+)$, whose lower bound s_{d+1}^2 can be achieved by the top-d singular functions, the optimal encoder. However, there is an issue with this definition. For example, consider an encoder with $d = 1000$. It learns the top-10 singular functions, but the other 990 dimensions are complete noise that has zero contribution to $\mathrm{RT}(\Phi; P^+)$. The approximation error of this encoder should be no higher than that of the top-10 singular functions, because adding more dimensions will never make the approximation error higher. However, if d becomes larger and $\mathrm{RT}(\Phi; P^+)$ stays the same, then $s_1^2 + \cdots + s_{d+1}^2 - \mathrm{RT}(\Phi; P^+)$ will become larger, so this quantity does not correlate with the approximation error in this scenario. The following definition fixes this issue.

Definition 3.9. *For any linearly independent $f_1, \cdots, f_{d'} \in L^2(P_{\mathcal{X}})$, denote $F = [f_1, \cdots, f_{d'}]$, $\boldsymbol{C}_F = \mathrm{Cov}_{P_{\mathcal{X}}}[F]$, and $\boldsymbol{B}_F = \mathrm{Cov}_{P_{\mathcal{A}}}[F]$. The **trace gap** of Φ w.r.t. P^+ is defined as*

$$\mathrm{TG}(\Phi; P^+) := \inf_{d' \le d} \inf_{f_1, \cdots, f_{d'}} \left\{ s_1^2 + \cdots + s_{d'+1}^2 - \mathrm{Tr}(\boldsymbol{C}_F^{-1}\boldsymbol{B}_F) \right\}.$$

Obviously, this definition of trace gap is upper bounded by $s_1^2 + \cdots + s_{d+1}^2 - \mathrm{RT}(\Phi; P^+)$. It solves the issue in the previous example, because having completely noisy dimensions does not affect the trace gap. The following result bounds the approximation error.

Theorem 3.10. *Suppose $\mathrm{TG}(\Phi; P^+) < s_1^2$, and $\epsilon > 1 - s_1$. Then,*

$$\mathrm{err}(\Phi; \mathcal{F}_\epsilon(P^+)) \le \frac{s_1^2 - (1-\epsilon)^2 + s_1 \mathrm{TG}(\Phi; P^+)}{s_1^2 - \mathrm{TG}(\Phi; P^+)^2}.$$

Remark 3.11. *This bound is fairly tight. If Φ learns the contexture, then by Theorem 3.4 we have $\mathrm{err}(\Phi; \mathcal{F}_\epsilon(P^+)) = \frac{s_1^2 - (1-\epsilon)^2}{s_1^2 - s_{d+1}^2}$, and $\mathrm{TG}(\Phi; P^+) = s_{d+1}$. Compared to this exact formula, the above upper bound only has an extra $s_1 \mathrm{TG}(\Phi; P^+)$ term in the numerator.*

Proof Let $f_1, \cdots, f_{d'}$ be the functions that minimize $s_1^2 + \cdots + s_{d'+1}^2 - \mathrm{Tr}(\boldsymbol{C}_F^{-1}\boldsymbol{B}_F)$. Without loss of generality, assume that $f_1, \cdots, f_{d'}$ have zero mean and are orthonormal. Let $\mathcal{F} = \mathrm{span}\{f_1, \cdots, f_{d'}\}$, and $\mathcal{H} = \mathrm{span}\{T_{P^+}^* f_1, \cdots, T_{P^+}^* f_{d'}\}$. For any $f \in \mathcal{F}_\epsilon(P^+)$ with $\|f\|_{P_{\mathcal{X}}} = 1$, let $h = T_{P^+}^* f \in \mathcal{H}_{P^+}$, and let $f_{\mathcal{F}}$ be the projection of f onto $\mathcal{F}$. Since $\mathrm{err}(\Phi; \mathcal{F}_\epsilon(P^+))$ is upper bounded by $\|f - f_{\mathcal{F}}\|_{P_{\mathcal{X}}}^2$, it suffices to show that $\|f - f_{\mathcal{F}}\|_{P_{\mathcal{X}}}^2$ is upper bounded by the right-hand side.

Let $\alpha^2 = \|f_{\mathcal{F}}\|_{P_{\mathcal{X}}}^2$, and $\beta^2 = \|f - f_{\mathcal{F}}\|_{P_{\mathcal{X}}}^2$, where α and β are non-negative. Then, $\alpha^2 + \beta^2 = \|f\|_{P_{\mathcal{X}}}^2 = 1 = \|h\|_{\mathcal{H}_{P^+}}^2$. The isometry property says that $(1-\epsilon)^2(\alpha^2 + \beta^2) \le \|h\|_{P_{\mathcal{A}}}^2$. Let $f - f_{\mathcal{F}} = \beta f_0$ where $\|f_0\|_{P_{\mathcal{X}}} = 1$. Let $h_{\mathcal{F}} = T_{P^+}^* h_{\mathcal{F}}$ and $h_0 = T_{P^+}^* f_0$. Then, we have $\|h_{\mathcal{F}}\|_{P_{\mathcal{A}}}^2 \le s_1^2 \|f_{\mathcal{F}}\|_{P_{\mathcal{X}}}^2 = s_1^2 \alpha^2$. Meanwhile, since f_0 is orthogonal to $f_1, \cdots, f_{d'}$, by Lemma 3.8 we have $\left\|T_{P^+}^* f_0\right\|_{P_{\mathcal{A}}}^2 + \left\|T_{P^+}^* f_1\right\|_{P_{\mathcal{A}}}^2 + \cdots + \left\|T_{P^+}^* f_{d'}\right\|_{P_{\mathcal{A}}}^2 \le s_1^2 + \cdots + s_{d'+1}^2$, which

implies that $\left\|T^*_{P^+} f_0\right\|^2_{P_{\mathcal{A}}} \le s_1^2 + \cdots + s_{d'+1}^2 - \operatorname{Tr}(\boldsymbol{C}_F^{-1}\boldsymbol{B}_F^{-1})$. Let $\tau = \operatorname{TG}(\Phi; P^+)$. Then, we have

$$\|h\|^2_{P_{\mathcal{A}}} = \|h_{\mathcal{F}} + \beta h_0\|^2_{P_{\mathcal{A}}} \le \|h_{\mathcal{F}}\|^2_{P_{\mathcal{A}}} + \beta^2\|h_0\|^2_{P_{\mathcal{A}}} + 2\beta\|h_{\mathcal{F}}\|_{P_{\mathcal{A}}}\|h_0\|_{P_{\mathcal{A}}} \le s_1^2\alpha^2 + \tau^2\beta^2 + 2s_1\tau\alpha\beta.$$

Thus, we have $(1-\epsilon)^2(\alpha^2+\beta^2) \le s_1^2\alpha^2 + \tau^2\beta^2 + 2s_1\tau\alpha\beta$, which implies that $(s_1^2 - \tau^2)\beta^2 \le [s_1^2 - (1-\epsilon)^2](\alpha^2+\beta^2) + 2s_1\tau\alpha\beta \le [s_1^2 - (1-\epsilon)^2 + s_1\tau](\alpha^2+\beta^2)$, as desired. ■

Connection to Fisher discriminant analysis. Fisher discriminant analysis [7, 101, 106], or more generally linear discriminant analysis (LDA), is a classical method of learning linear classifiers in statistics. Here we show that Fisher discriminant analysis has a strong connection to the contexture theory. Suppose $\mathcal{X} \subseteq \mathbb{R}^{d_{\mathcal{X}}}$. Fisher discriminant analysis defines the following **between-class covariance matrix** $\boldsymbol{S}_B \in \mathbb{R}^{d_{\mathcal{X}} \times d_{\mathcal{X}}}$ and **within-class covariance matrix** $\boldsymbol{S}_W \in \mathbb{R}^{d_{\mathcal{X}} \times d_{\mathcal{X}}}$:

$$\boldsymbol{S}_B = \iint \Big\{(\mathbb{E}[X \mid A = a_1] - \mathbb{E}[X \mid A = a_2])(\mathbb{E}[X \mid A = a_1] - \mathbb{E}[X \mid A = a_2])^\top\Big\};$$
$$\boldsymbol{S}_W = \int \mathbb{E}_{P^+}\Big[(X - \mathbb{E}[X \mid A = a])(X - \mathbb{E}[X \mid A = a])^\top \;\Big|\; A = a\Big] dP_{\mathcal{A}}(a).$$

In the original formulation of Fisher discriminant analysis, A is the label of X. Here we extend it to a general context variable. Consider a linear encoder $\Phi(x) = \boldsymbol{W}x$, where $\boldsymbol{W} \in \mathbb{R}^{d \times d_{\mathcal{X}}}$. Then, one solves the following optimization problem to find $\boldsymbol{W}$:

$$\underset{\boldsymbol{W} \in \mathbb{R}^{d \times d_{\mathcal{X}}}}{\text{maximize}}\; J(\boldsymbol{W}) = \operatorname{Tr}\Big[\big(\boldsymbol{W}\boldsymbol{S}_B\boldsymbol{W}^\top\big)\big(\boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^\top\big)^{-1}\Big] \quad \text{s.t.} \quad \boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^\top \text{ is invertible.}$$

Here, $J(\boldsymbol{W})$ is called the **Fisher discriminant**. Define $\Psi(a) = \mathbb{E}_{P^+}[\boldsymbol{W}X | A = a]$. Then, we can see that

$$\boldsymbol{W}\boldsymbol{S}_B\boldsymbol{W}^\top = \iint (\Psi(a_1) - \Psi(a_2))(\Psi(a_1) - \Psi(a_2))^\top dP_{\mathcal{A}}(a_1) dP_{\mathcal{A}}(a_2);$$
$$\boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^\top = \int \mathbb{E}_{P^+}\Big[(\Phi(X) - \Psi(a))(\Phi(X) - \Psi(a))^\top \;\Big|\; A = a\Big] dP_{\mathcal{A}}(a).$$

Let $\boldsymbol{C}_\Phi = \mathbb{E}[\tilde{\Phi}(X)\tilde{\Phi}(X)^\top]$ and $\boldsymbol{B}_\Phi = \mathbb{E}[\tilde{\Psi}(A)\tilde{\Psi}(A)^\top]$. Then, we have

$$\boldsymbol{W}\boldsymbol{S}_B\boldsymbol{W}^\top = 2\big\{\mathbb{E}\big[\Psi(A)\Psi(A)^\top\big] - \bar{\Psi}\bar{\Psi}^\top\big\} = 2\mathbb{E}\Big[\tilde{\Psi}(A)\tilde{\Psi}(A)^\top\Big] = 2\boldsymbol{B}_\Phi;$$
$$\begin{aligned}\boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^\top &= \int \mathbb{E}_{P^+}\big[\Phi(X)\Phi(X)^\top - \Psi(a)\Psi(a)^\top \;\big|\; A = a\big] dP_{\mathcal{A}}(a) \\ &= \mathbb{E}\big[\Phi(X)\Phi(X)^\top\big] - \mathbb{E}\big[\Psi(A)\Psi(A)^\top\big] \\ &= \mathbb{E}\Big[\tilde{\Phi}(X)\tilde{\Phi}(X)^\top\Big] - \mathbb{E}\Big[\tilde{\Psi}(A)\tilde{\Psi}(A)^\top\Big] = \boldsymbol{C}_\Phi - \boldsymbol{B}_\Phi.\end{aligned}$$

Therefore, $J(\boldsymbol{W}) = 2\operatorname{Tr}[(\boldsymbol{C}_\Phi - \boldsymbol{B}_\Phi)^{-1}\boldsymbol{B}_\Phi]$, which is very similar to the ratio trace defined in Definition 3.7. Recall that an encoder that learns the contexture maximizes the ratio trace. A well-known result is that $J(\boldsymbol{W})$ is maximized when $\boldsymbol{W}$ consists of the top-d eigenvectors of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$. Hence, Fisher discriminant analysis is almost equivalent to contexture learning under the constraint that the encoder must be linear.

3.3 Evaluating Context Usefulness

The previous chapter argued that better contexts are essential to further improve foundation models, but how to create better contexts is a challenging open problem. This section studies an easier problem—how to evaluate a context before pretraining. We say that a context is **useful** for a downstream task, if it can lead to an encoder with good performance on this task. However, since we might not know the task at pretrain time, we can only predict if the context is useful *in general*. Solving this problem is a prerequisite, because if we cannot even decide whether a context is good or not, then there will be no way for us to create better contexts.

Evaluating a context is more difficult than evaluating an encoder. When evaluating an encoder, we can assume that the task is known to be compatible with the context. However, when evaluating a context, we cannot make this assumption. Instead, a context is better if it is compatible with more tasks, because such a context can lead to more transferable encoders, and our evaluation should take this into consideration.

Given a context and a dataset, the only things we can use are the singular values and singular functions of the context. However, estimating the singular functions is as hard as pretraining an encoder, whereas the singular values can be efficiently estimated using the post-hoc approach in Section 2.5 with a small subset of samples. Therefore, a metric will be more ideal if it only uses the singular values (the spectrum).

In this section, we show that it is possible to evaluate the usefulness of a context with only its singular values. This might seem counter-intuitive, because suppose the encoder learns the contexture, then only the singular functions will affect the performance of the encoder. When $\mu_1, \cdots, \mu_d$ are fixed, the space spanned by $\phi_1, \cdots, \phi_d$ is also fixed and is independent of $s_1, \cdots, s_d$. The reason why this is possible is that the singular values and the singular functions are intrinsically connected. Recall that $k_X^+(x, x') = \sum s_i^2 \mu_i(x)\mu_i(x')$, and that it must satisfy $k_X^+(x, x') \geq 0$ for all x, x', which is quite a strict constraint. Therefore, when $\mu_1, \mu_2, \cdots$ are fixed, one cannot choose $s_1^2, \cdots, s_d^2$ arbitrarily. Conversely, the singular values limit the possible choices of the singular functions.

In this section, we propose a metric for evaluating context usefulness, and the metric only depends on the singular values. Note that there does not exist a universal metric, because whether the context is useful on a task or not depends on their compatibility, which cannot be estimated without the knowledge of the task. However, our experiments show that the proposed metric generally works well on real datasets.

Qualitative analysis. In Section 1.3 we showed that the shape of the spectrum depends on the association strength between X and A, also called the association of the context. The singular values decay slower if the association is stronger. Here we show that the decay rate of the singular values has a great impact on the usefulness of the context.

The central argument is: **A useful context should have a moderate association**. To get an intuition, consider the two extreme cases in Section 1.3: (i) A is independent of X, then there is only one positive singular value; (ii) $A = X$, then all singular values are 1. Both contexts are clearly useless because they provide no additional information. In what follows, we qualitatively explain why a context is not very useful if its association is too strong or too weak.

Recall that any compatible task $f^* = \sum u_i \mu_i \in \mathcal{F}_\epsilon(P^+)$ as defined in Eqn. (3.2) needs to satisfy $\sum_{i \geq 1} s_i^2 u_i^2 \geq (1-\epsilon)^2 \sum_{i \geq 1} u_i^2$. This is easier to satisfy if s_i are large. Thus, if the association is too weak and the singular values decay too fast, then $\mathcal{F}_\epsilon(P^+)$ will be a very small set. Consequently, very few tasks will be compatible with the context, so an

encoder trained on the context will not be transferable to various tasks.

On the other hand, when the association is too strong, there are two consequences. First, more singular functions have large singular values, and to learn all of them, one needs to use a larger d, which makes the sample complexity of learning the downstream linear predictor higher. Second, the singular functions with large singular values become less smooth, so learning them requires more pretraining samples. In Chapter 5 we will define the context complexity that quantitatively measures the smoothness of the top singular functions, and we will show that the context complexity is higher when the context has a stronger association.

Quantitative metric. We now propose a metric that quantitatively measures the general usefulness of a context. The metric only depends on the singular values of the context. The metric assumes that the pretraining method aims to learn the contexture of the context, and it does not work for an arbitrary encoder.

Our metric is defined as

$$\tau_d = \frac{1}{1 - s_{d+1}^2} + \beta \frac{\sum_{i=1}^{d} s_i^2}{\sum_{i=1}^{d_0} s_i^2}, \tag{3.5}$$

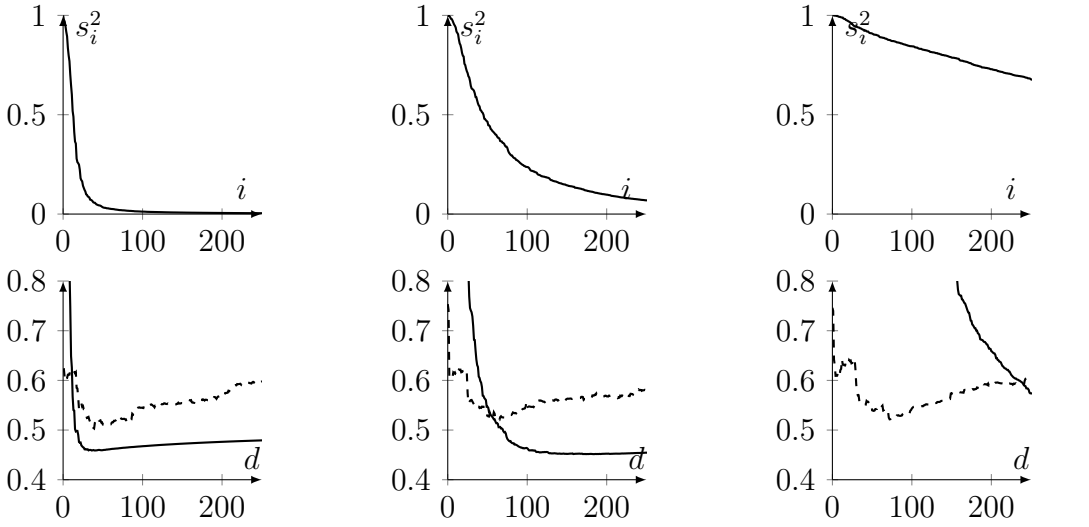
where $\beta > 0$ is a parameter, and d_0 is the maximum d we consider. Typically d_0 ranges from 512 to 8192. We choose $\beta = 1$ and $d_0 = 512$ in our experiments. τ_d is a proxy of the prediction error when the embedding dimension is d. Thus, the d that minimizes τ_d can be viewed as the optimal embedding dimension predicted by the metric, and τ evaluates the context when d is chosen optimally. Since this metric only depends on the singular values, it can be efficiently estimated using the post-hoc approach in Section 2.5, with which we can estimate the spectrum using a subset of $\Theta(d_0 \log d_0)$ samples.

This metric is derived in the following way. Let the target function be $f^* = f_0 + f_1$, where $\langle f_0, f_1 \rangle_{P_{\mathcal{X}}} = 0$, f_0 is not compatible with the context, and f_1 is compatible with the context. Then, the prediction error can be decomposed into three components:

(i) The approximation error of f_1
(ii) The approximation error of f_0
(iii) The estimation error

By Theorem 3.4, component (i) can be bounded by $\frac{s_1^2 - (1-\epsilon)^2}{s_1^2 - s_{d+1}^2}$, and we simplify this to the first term because s_1 is very close to 1 in most real cases, and the numerator is a constant. Component (ii) is smaller if the context has a stronger association; thus, it should be negatively correlated with $\sum_{i=1}^{d_0} s_i^2$. Component (iii) is larger if the context has a stronger association, or if d is large. Based on the result in Chapter 5, it is positively correlated with $\sum_{i=1}^{d} s_i^2$. The second term of the metric combines these two components, and it is designed to be bounded by 1.

Compared to previous metrics. Some previously proposed metrics are also based on the decay rate of the spectrum. [3] proposed a metric based on the eigenvalues of $\Phi\Phi^\top$ for a particular pretrained encoder Φ; that is, they use the λ that satisfies $\langle \Phi, \Phi \rangle_{P_{\mathcal{X}}} f = \lambda f$ for some $f \neq \mathbf{0}$. Here, $\langle \Phi, \Phi \rangle_{P_{\mathcal{X}}}$ is the covariance matrix, assuming that Φ is centered. In contrast, our metric is based on the general eigenvalues satisfying $\langle \Phi, T_{P^+} \Phi \rangle_{P_{\mathcal{X}}} f = s^2 \langle \Phi, \Phi \rangle_{P_{\mathcal{X}}} f$ for some $f \neq \mathbf{0}$, as discussed in Section 2.5. These two sets of eigenvalues are fundamentally different. The eigenvalues s_i^2 we use are invariant under invertible linear transformations on Φ, while λ_i are not. This suggests that our metric is more



(a) Weak association (b) Moderate association (c) Strong association

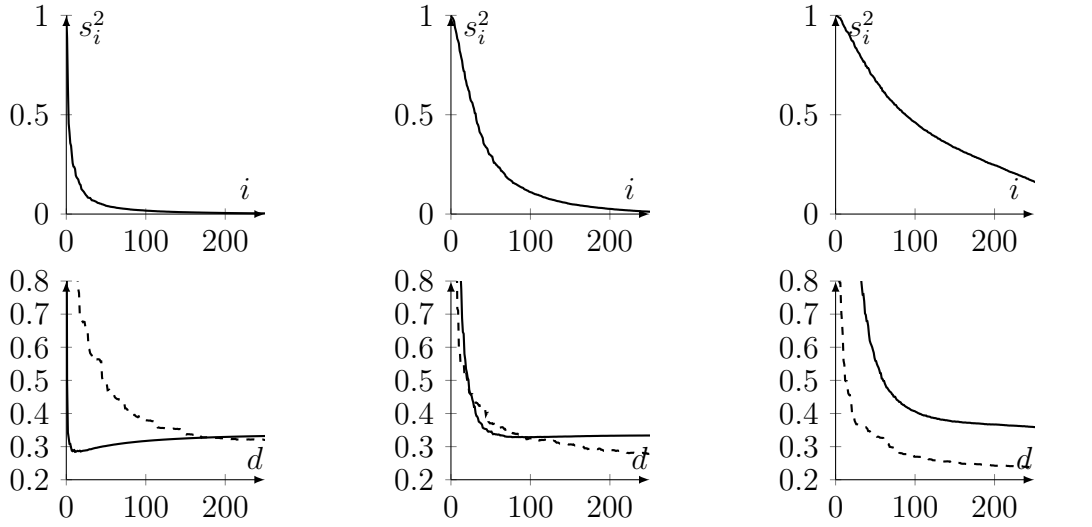
Figure 3.2: Metric illustration on `abalone`. **Top row:** context spectra. **Bottom row:** solid curves are τ_d divided by 6; dashed curves are the actual downstream prediction error. We divide τ_d by 6 to fit it in the same plot.

desirable since invertible linear transformations on Φ do not affect the performance of the downstream linear probe.

Now we empirically examine τ_d on the two datasets. First, we apply the metric to the `abalone` dataset and use KNN as the context, similar to Section 2.6. We adjust the association of the context by changing K. In particular, we choose $K = 150$ (weak), $K = 30$ (moderate) and $K = 5$ (strong). We obtain the exact eigenvalues and eigenfunctions of $T_{k_X^+}$ using kernel PCA. In Figure 3.2, we plot the spectra of the three contexts in the top row. Then, in the bottom row, we compare τ_d against the prediction error of the linear probe under different d. We can see that when the association is weak or moderate, τ_d first decreases and then increases, which tracks the actual error. However, when the association is too strong, τ_d monotonically decreases with d, and it cannot track the actual error.

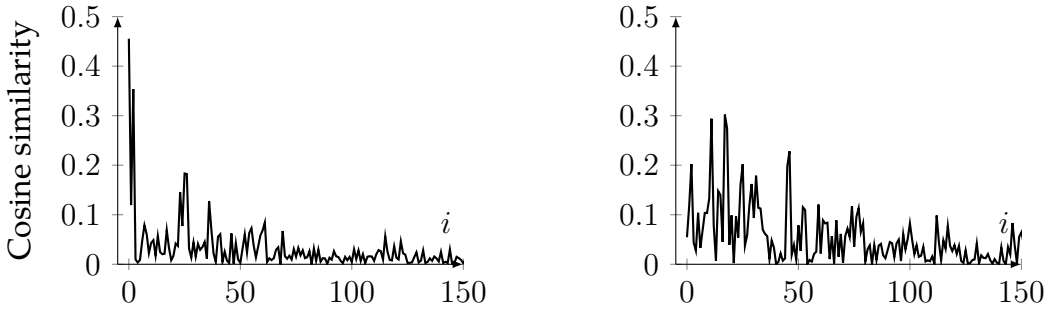
Second, we apply the metric to the `MNIST` dataset. The context is random cropping with crop ratio α. We adjust the association of the context by changing α. In particular, we choose $\alpha = 0.5$ (weak), $\alpha = 0.2$ (moderate) and $\alpha = 0.05$ (strong). Since kernel PCA is not scalable to datasets as large as MNIST, we instead train a neural network. Specifically, we train a LeNet [95] using the non-contrastive learning objective formulated earlier, and the AdamW optimizer. Then, we estimate the top eigenvalues using the post-hoc approach in Section 2.5. The downstream task is a binary classification task—whether the digit is greater than 4. After pretraining, a linear probe is fit on top of Φ using ridge regression. The result is plotted in Figure 3.3.

From Figure 3.3, we can see that when the association is not too strong, τ_d first decreases and then increases, similar to Figure 3.2. However, on `MNIST`, the downstream error monotonically decreases with d, unlike `abalone`. This disparity is due to the difference between the two downstream tasks. To demonstrate this, in Figure 3.4 we plot the cosine similarity between the target function f^* and the estimated i-th eigenfunction on the two datasets. We can see that the variance of f^* on `abalone` is mostly concentrated on the top-5 eigenfunctions, with the first cosine similarity being almost 0.5. In contrast,



(a) Weak association (b) Moderate association (c) Strong association

Figure 3.3: Metric illustration on MNIST, similar to Figure 3.2.



(a) abalone (b) MNIST

Figure 3.4: Comparison of the downstream task between abalone and MNIST.

the variance of f^* on MNIST is more scattered, and the cosine similarity is still close to 0.1 for the 150-th eigenfunction. Consequently, having a large d on abalone will have a little impact on the approximation error but will increase the estimation error significantly. On the other hand, having a larger d on MNIST will decrease the approximation error more than it increases the estimation error, which is why the total error monotonically decreases with d.

The takeaway from this experiment is that although in general a context with a moderate association is good, in reality it still depends on the actual downstream task. For example, on abalone the weakest context actually leads to the lowest error, because the variance of f^* is concentrated on the top-5 eigenfunctions. On the other hand, on MNIST the strongest context leads to the lowest error, because the variance of f^* is scattered among a lot of features, and a stronger association allows more features to be discovered. Hence, no evaluation metric would universally work for all contexts and downstream tasks, but a metric would still be useful if it correlates well with the actual error in most scenarios, and thus can provide insights into choosing the right context and the right hyperparameters, such as the mask or crop ratio.

Empirical verification of the proposed metric. Although our metric is derived from the decomposition of the prediction error, it is still heuristic and is not an upper bound of the actual error. Here we show that the metric correlates well with the actual error on many real datasets. Therefore, our metric is useful as it can help practitioners to select among various pretraining methods or choose the hyperparameters efficiently.

We use 28 real classification and regression datasets from OpenML that are widely used in machine learning research. Each dataset is randomly split into a pretrain set, a labeled downstream training set, and a downstream test set by 70%-15%-15%. For each d, we obtain the top-d eigenfunctions of $T_{k_X^+}$ via kernel PCA, and then fit a linear probe on top of it using ridge regression. Then, we select the best d^* that achieves the lowest test mean squared error, which is denoted by err_{d^*}. The correlation between τ and err_{d^*} is reported. The following four types of contexts are used in the experiment.

- RBF kernels: $k(x, a) = \exp(-\gamma\|x - a\|^2)$. Define $P^+(a|x) \propto k(x, a)$ for each x.
- KNN: $P^+(a|x) = K^{-1}$ if a is a KNN of x; otherwise, it is 0.
- RBF $\star$ Masking: First randomly mask 20% of the features, and then apply RBF kernels to the other features. The $T_{k_X^+}$ of this context can be estimated as follows: first randomly draw 50 masks, and then compute their average $T_{k_X^+}$.
- KNN $\star$ Masking: 20% random masking and then apply KNN.

For each of these contexts, $\mathcal{A} = \mathcal{X}$. For each type, 35 contexts are obtained by adjusting the γ for RBF kernels, and K for KNN. The association between X and A for these 35 contexts are different, and the experiment makes sure that the contexts in every type range from very weak to very strong association. "$\star$ Masking" here means the convolution with masking, as mentioned in Section 2.4. We do not use masking alone because the dual kernel of masking is hard to estimate.

Table 3.1 reports the correlation between τ and err_{d^*} over all the 140 contexts. The most common metric is the Pearson correlation, but it can only detect linear correlations, while the correlation between τ and err_{d^*} is not necessarily linear. Thus, we also report the distance correlation [135], which is another common metric that can detect nonlinear correlations but cannot tell if the correlation is positive or negative because this metric is always non-negative.

The median reported in the table shows that on more than half of the datasets, the Pearson correlation is over 0.5, which is generally considered a strong correlation. The distance correlation is even higher. As expected, the metric does not work on all datasets. For example, the Pearson correlation is very negative on `brazilian_houses` and `fifa`.

To understand when our metric might fail, we further visualize the results by plotting τ against err_{d^*} on five of the datasets in Figure 3.5. In this figure, plots (a), (b) and (c) are three success cases where a clear positive correlation can be observed, and plots (d) and (e) display two failure cases. Plot (d) shows a common failure case: if τ is very close to $2 = \beta + 1$, meaning that the metric believes that the association is extremely weak or extremely strong, then the metric will predict that the context is bad. However, a generally bad context can still be good on some tasks. For example, a very weak context still works well on a task that only uses the top-3 singular functions of the context. Therefore, it is advisable to abstain from using the metric when it is too close to $\beta + 1$.

Plot (e) shows a case where the metric is generally good for every single context type but has poor cross-type behavior. Specifically, it fails to predict that KNN is worse than RBF on this dataset. This suggests that our metric might not be able to compare different types of contexts. For example, if two contexts of completely different types have similar spectra, then our metric will indicate that they are similarly useful. This is because our

Dataset	Size (↑)	#Feature	Type	Pearson	Distribution
credit-approval	690	15	Cls	0.583	0.683
breast-w	699	9	Cls	0.072	0.255
diabetes	768	8	Cls	0.737	0.740
solar_flare	1066	10	Reg	0.019	0.262
Moneyball	1232	14	Reg	0.680	0.650
yeast	1269	8	Cls	0.221	0.256
cmc	1473	9	Cls	0.867	0.860
Wine	1599	11	Reg	-0.084	0.212
scene	2407	299	Cls	0.608	0.685
dna	3186	180	Cls	0.881	0.843
splice	3190	60	Cls	0.831	0.801
kr-vs-kp	3196	36	Cls	0.543	0.512
abalone	4177	8	Reg	0.028	0.470
spambase	4601	57	Cls	0.775	0.858
colleges	7603	44	Reg	0.155	0.387
mushroom	8124	22	Cls	0.185	0.340
kin8nm	8192	8	Reg	0.805	0.760
pumadyn32nh	8192	32	Reg	0.938	0.961
cpu_activity	8192	21	Reg	0.709	0.825
SpeedDating	8378	120	Cls	0.590	0.656
grid_stability	10000	12	Reg	0.925	0.911
sulfur	10081	6	Reg	-0.180	0.487
brazilian_houses	10692	9	Reg	-0.290	0.563
fifa	19178	28	Reg	-0.349	0.663
superconductivity	21263	81	Reg	0.141	0.367
kings_county	21613	21	Reg	0.842	0.882
health_insurance	22272	11	Reg	0.601	0.749
cps88wages	28155	6	Reg	0.250	0.479
			Mean	0.431	0.611
			Median	0.587	0.659

Table 3.1: Correlation between τ and the actual error err_{d^*} on all 4 types of contexts.

metric only depends on the spectrum. However, it could be possible that for a particular task, one context is good and the other is bad, and our metric cannot reflect this disparity.

Overall, although there does not exist a universal metric that works for all contexts and tasks, and our metric does have failure cases, the experiment results here provide empirical evidence that more often than not, the proposed metric correlates well with the actual prediction error of the downstream linear probe. Hence, the proposed metric is useful for choosing hyperparameters and comparing contexts in practice.

In summary, this chapter first defined the compatibility between a task and a context, and then discussed the intrinsic evaluation of encoders and contexts. The key takeaway is that when we know a priori that the downstream task is compatible with the context we are given, then the optimal thing to do is learning the contexture. The intrinsic evaluation of an arbitrary encoder depends on two key concepts—the induced RKHS and the ratio trace. For the intrinsic evaluation of contexts, the key takeaway is that a good context should have a moderate association. Moreover, we proposed a metric that only

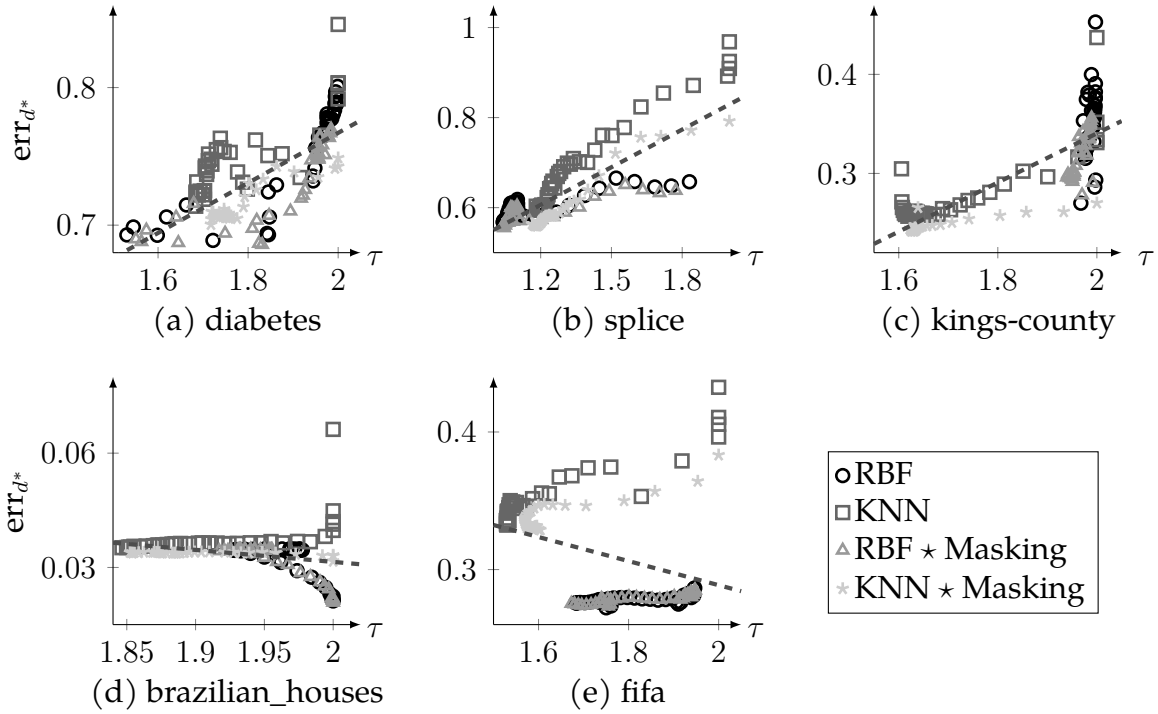


Figure 3.5: Scatter plots of τ versus err$_{d^*}$. Dashed line: Linear fit.

depends on the spectrum of the context. The metric correlates with the actual prediction error on many real datasets.

Code and data availability. The code for Figure 3.2 can be found at `https://1drv.ms/u/c/ea9fe908498c8b82/EV11dVAUVCdCnkw31CHrZC0BOEmsudZ7swpTKvzcfod5uA?e=axm6XG`. The code for Table 3.1 is at `https://1drv.ms/u/c/ea9fe908498c8b82/EcqvS70ynvdCsma6MoSDRcwBZPRw5uThHUXQcz9P7vNQZQ?e=7ULQey`. All datasets can be downloaded from OpenML.

Chapter 4

Mixing Multiple Contexts

We have shown that creating better contexts is imperative for further improving the performance of foundation models. However, creating new contexts from scratch is extremely challenging. Normally, it requires new domain knowledge obtained from scientific research, or collecting new data through experiments or human labeling such as in RLHF, both of which require considerable effort. As a result, the contexts used for pretraining rarely change. For example, in computer vision, although various objectives have been proposed such as contrastive and non-contrastive learning and masked autoencoders, the context is always based on image corruption, such as random cropping, masking and color distortion. In NLP, base language models have always been pretrained on the context of masking tokens like BERT [34]. There are many variants of BERT, such as determining whether a sentence completion is correct instead of actually completing a masked sentence like Electra [27], predicting the next token rather than tokens in the middle of a sentence like GPT [117], and predicting the next group of tokens like Medusa [21]. In all these variants, the context variable A is a masked version of X.

This chapter introduces a much easier way to obtain better contexts. We know that a good context should have a moderate association. Consider the following scenario: we have a number of contexts, but none of them is useful enough because the association is either too strong or too weak. Then, how can we get a better context? The idea is to mix these contexts together, so that we can obtain a context with a moderate association.

In fact, mixing multiple contexts is quite common in practice, though it has never been fully formalized as a unified framework like this chapter. One very widely used method is composing different data augmentation techniques together. For example, common data augmentations for images include translation, rotation, random cropping, color distortion, etc. Each augmentation defines one context, and sequentially applying them to the same image leads to the **convolution** of their contexts.

Another common practice when there are multiple contexts is optimizing a weighted sum of different objectives. Suppose there are r desiderata we want our model to satisfy, and desideratum j can be achieved by minimizing objective $\mathcal{R}_j$. A natural idea is to minimize $\sum w_j \mathcal{R}_j$ for some $w_1, \cdots, w_r > 0$, so that we minimize all $\mathcal{R}_j$ simultaneously. In machine learning, if $\mathcal{R}_1$ is the main objective, then other $\mathcal{R}_j$ are also called penalty terms. Common penalty terms include the L^2 or RKHS norm of the model weight, the local smoothness of the model *w.r.t.* a certain manifold, the distance to a reference model, etc. The weighted sum of the objectives learns the contexture of what we call the **convex combination** of the r contexts.

To get an intuition of convolution and convex combination, consider the example in Figure 4.1. In this example, $|\mathcal{X}| = 4$. We know that P^+ induces a joint distribution on

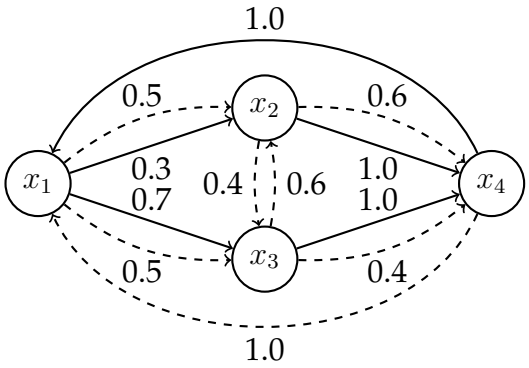


Figure 4.1: An example of two contexts (solid and dashed edges) where $|\mathcal{X}| = 4$.

$\mathcal{X} \times \mathcal{X} : P^+(x, x') = \int P^+(x|a)P^+(x'|a)dP_{\mathcal{A}}(a)$. We can then obtain $P^+(x'|x)$, which are labeled as edge weights in Figure 4.1. In other words, each context induces a random walk on the graph. There are two contexts in Figure 4.1. P_1^+ corresponds to the solid edges, and P_2^+ corresponds to the dashed edges.

Convolution, denoted by $\star$, is equivalent to a multi-step random walk. $P_2^+ \star P_1^+$ means that we first walk one step on the solid edges (P_1^+), and then walk one step on the dashed edges (P_2^+). For example, $(P_2^+ \star P_1^+)(x_4|x_1) = \sum_x P_2^+(x_4|x)P_1^+(x|x_1) = 0.6 \times 0.3 + 0.4 \times 0.7 = 0.46$. Similarly, $P_1^+ \star P_2^+ \star P_1^+$ is equivalent to a solid-dashed-solid random walk. In general, convolution does not have the commutative property.

Convex combination, denoted by $+$, is equivalent to a stochastic one-step random walk. $0.6P_1^+ + 0.4P_2^+$ means that we walk one step; with probability 0.6 the step is on the solid edges, and with probability 0.4 the step is on the dashed edges. For example, if we are standing at x_1, then under $0.6P_1^+ + 0.4P_2^+$, with probability $0.6 \times 0.3 + 0.4 \times 0.5 = 0.38$ we will walk to x_2, and with probability 0.62 we will walk to x_3. Convex combination can be combined with convolution. For example, $P_2^+ \star (0.6P_1^+ + 0.4P_2^+)$ means that we first take one step using $0.6P_1^+ + 0.4P_2^+$, and then take another step using P_2^+.

We can show that the set of contexts forms a near-ring with scalar multiplication (also called a module), where convolution is multiplication and convex combination is addition with scalar multiplication. "Near"-ring means that it only has the right distributive property but not the left one; that is, $(0.6P_1^+ + 0.4P_2^+) \star P_2^+ = 0.6P_1^+ \star P_2^+ + 0.4P_2^+ \star P_2^+$, but $P_2^+ \star (0.6P_1^+ + 0.4P_2^+)$ is not distributive.

There is a third operation for mixing multiple contexts called concatenation. Concatenation is completely different from the other two operations, because convolution and convex combination act on the input space $\mathcal{X}$, whereas concatenation acts on the output space of the encoder. Specifically, given r contexts, one trains an individual encoder Φ_j for each context, and then concatenate them as $\Phi(x) = [\Phi_1(x), \cdots, \Phi_r(x)]$. Concatenation is a classical and very popular method, and it is connected to a whole field in data science known as feature engineering. In feature engineering, people create different features based on different signals (formulated as contexts in this thesis), and then train a model on their concatenation.

Mixture of experts (MOE) [76] is a popular method that is a stronger version of concatenation. In MOE, we first train individual encoders $\Phi_1, \cdots, \Phi_r$ for the r contexts, and then at the downstream stage, we train a gating function $g : \mathcal{X} \to \mathbb{R}^r$ that assigns weights to the r encoders for each x. The overall encoder is $\Phi(x) = \sum_{j=1}^r g(x)_j \Phi_j(x)$, and finally a linear probe is fit on top of Φ. The common practice is to implement g as a small neural network, and its output space is usually restricted to Δ^{r-1}, the $(r-1)$-dimensional

unit simplex, so that $g(x)$ is a probability distribution over the r encoders. This can be easily done by adding a softmax layer to the end of the neural network. Concatenation is a special case of MOE, where g is restricted to be a constant function.

In what follows, we conduct a deep analysis on the three base operations for mixing multiple contexts: convolution, convex combination and concatenation. Importantly, we discuss in what situations each operation should be used. After that, we apply the three operations to real-world tabular datasets, and we find that they can improve the performance of state-of-the-art methods such as XGBoost [22].

4.1 Convolution

Suppose we have r contexts given by $P_1^+, \cdots, P_r^+$. Let $\mathcal{A}_j$ be the context space of P_j^+, and let Q_j^+ be the heuristic inverse of P_j^+ (Definition 1.11). As mentioned earlier, convolution is similar to composing multiple data augmentation techniques. Assuming that we have T-access to every context, we can transform X by $X \xrightarrow{P_1^+} A_1 \xrightarrow{Q_1^+} X_1 \xrightarrow{P_2^+} A_2 \xrightarrow{Q_2^+} X_2 \xrightarrow{P_3^+} \cdots \xrightarrow{Q_{r-1}^+} X_{r-1} \xrightarrow{P_r^+} A_r$. This A_r is defined as the context variable of the convolution $P_r^+ \star \cdots \star P_1^+$. The heuristic inverse of $P_r^+ \star \cdots \star P_1^+$ is defined as Q_r^+. Then, it is easy to see that convolution has the associative property.

In the general case, we might not have T-access to all contexts, but we can assume that each context has either k-access or T-access. This is a weak assumption since it has been shown previously that any access can be converted to k-access.

For every $j \in [r]$, define a kernel k_j as follows:

- If it has k-access, let k_j be the kernel we have access to.
- If it has T-access, let $k_j(x, x') = \int Q_j^+(x'|a) dP_j^+(a|x)/P_{\mathcal{X}}(x')$ if $j < r$. If $j = r$, then define $k_r = k_{Xr}^+$ as the exact dual kernel of context r.

Proposition 4.1. *The integral operator of the dual kernel of the convolution of the r contexts is equal to $T_{k_1} T_{k_2} \cdots T_{k_r} \cdots T_{k_2} T_{k_1}$.*

Remark 4.2. *When each T_{k_j} is a bounded compact self-adjoint operator, this operator is also bounded compact self-adjoint, so the Hilbert-Schmidt theorem still applies. Note that this operator is not equal to $T_{k_r} \cdots T_{k_1}$, which is not necessarily a self-adjoint operator.*

Proof We prove by induction on r. When $r = 1$ this is obvious. Suppose the result holds for $r - 1$. Let k_j^+ be the dual kernel between X_1 and A_j. Then, we have

$$k_r^+(x, x') = \iint k_1(x, z) k_{r-1}^+(z, z') k_1(z', x') dP_{\mathcal{X}}(z) dP_{\mathcal{X}}(z'),$$

so it is easy to see that $T_{k_r^+} = T_{k_1} T_{k_{r-1}^+} T_{k_1}$, which shows that the result holds for r. ■

We now discuss how to learn the contexture of a convolution. If we have T-access to every context, then we can simply transform $X \to A_r$, and use SVME on X and A_r. If we have T-access to $P_1^+, \cdots, P_{r-1}^+$ but k-access to P_r^+, then it is also very simple. We can transform $X \to X_{r-1}$, and use KISE on X and X_{r-1}.

The more difficult scenario is when we have k-access to P_j^+ for some $j < r$. Let $j_1 < j_2 < \cdots < j_l$ be all such j. We now present an algorithm that learns the contexture. First, initialize $\Phi^0 = \Phi$, and $X_0 = X$. Second, transform $X_0 \to X_{j_1 - 1}$ using the T-access to $P_1^+, \cdots, P_{j_1 - 1}^+$. Third, let $\Phi^1(X) = (T_{k_{j_1}} \Phi^0)(X_{j_1 - 1})$, which can be estimated with Monte

Algorithm 1 Extracting top-d eigenspace of a convolution of contexts

Initialize encoder $\Phi : \mathcal{X} \to \mathbb{R}^d$; If have T-access to context r, initialize $\Psi : \mathcal{A}_r \to \mathbb{R}^d$
for each training step **do**
  Sample a batch of samples $\{x_1, \cdots, x_m\}$
  Center Φ: $\Phi \leftarrow \Phi - \frac{1}{m}\sum_{i=1}^m \Phi(x_i)$
  for $i \in [m]$ **do** $x_i^0 \leftarrow x_i$; set $\boldsymbol{B} \leftarrow \boldsymbol{0} \in \mathbb{R}^{d\times m}$, $k \leftarrow$ **null**
  for $j = 1, \cdots, r-1$ **do**
    if have k-access to context j **then**
      if $k =$ **null then**
        $\boldsymbol{B} \leftarrow \left[\Phi(x_1^{j-1}), \cdots, \Phi(x_m^{j-1})\right] \in \mathbb{R}^{d\times m}$
      else
        Set Gram matrix $\boldsymbol{G} \in \mathbb{R}^{m\times m}$ as $\boldsymbol{G}[p,q] = k(x_p, x_q^{j-1})$; $\boldsymbol{B} \leftarrow \frac{1}{m}\boldsymbol{B}\boldsymbol{G}$
      $k \leftarrow k^+_{X^j}$; **for** $i \in [m]$ **do** $x_i^j \leftarrow x_i$ ▷ reset x_i^j to the original input
    else ▷ T-access to context j
      Sample $a_i^j \sim P_j^+(\cdot|x_i^{j-1}),\ x_i^j \sim Q_j^+(\cdot|a_i^j)$
  if $k =$ **null then** ▷ for every $j \le r-1$, context j has T-access
    $\boldsymbol{B} \leftarrow [\Phi(x_1^{r-1}), \cdots, \Phi(x_m^{r-1})] \in \mathbb{R}^{d\times m}$
  else
    Set Gram matrix $\boldsymbol{G} \in \mathbb{R}^{m\times m}$ as $\boldsymbol{G}[p,q] = k(x_p, x_q^{r-1})$; $\boldsymbol{B} \leftarrow \frac{1}{m}\boldsymbol{B}\boldsymbol{G}$
  if has k-access to r **then**
    Set Gram matrix $\boldsymbol{G}_r \in \mathbb{R}^{m\times m}$ as $\boldsymbol{G}_r[p,q] = k^+_{X^r}(x_p, x_q)$; $\boldsymbol{C} \leftarrow \frac{1}{m}\boldsymbol{B}\boldsymbol{G}_r \in \mathbb{R}^{d\times m}$
    Define $\mathcal{L} = \frac{1}{m}\left[\sum_{i=1}^m \|\Phi(x_i)\|_2^2 - \langle \boldsymbol{B}, \boldsymbol{C}\rangle\right]$ ▷ $\langle \boldsymbol{B}, \boldsymbol{C}\rangle = \mathrm{Tr}\left(\boldsymbol{B}\boldsymbol{C}^\top\right)$
  else ▷ T-access to context r
    Sample $a_i^r \sim P_r^+(\cdot|x_i)$; Center Ψ: $\Psi \leftarrow \Psi - \frac{1}{m}\sum_{i=1}^m \Psi(a_i)$
    $\boldsymbol{C} \leftarrow [\Psi(a_1^r), \cdots, \Psi(a_m^r)] \in \mathbb{R}^{d\times m}$
    Define $\mathcal{L} = \frac{1}{m}\left[\sum_{i=1}^m \|\Phi(x_i)\|_2^2 + \|\boldsymbol{C}\|_F^2 - 2\langle \boldsymbol{B}, \boldsymbol{C}\rangle\right]$ ▷ $\|\boldsymbol{C}\|_F^2 = \sum_{i,j}\boldsymbol{C}[i,j]^2$
  Update Φ to minimize loss $\mathcal{L}$, subject to $\mathrm{Cov}_{P_\mathcal{X}}[\Phi] = \boldsymbol{I}$

Carlo using the k-access to k_{j_1}. Fourth, let $X_{j_1} = X$, and transform $X_{j_1} \to X_{j_2-1}$ using the T-access to $P^+_{j_1+1}, \cdots, P^+_{j_2-1}$. Fifth, let $\Phi^2(X) = (T_{k_{j_2}}\Phi^1)(X_{j_2-1})$. Repeat these two steps until we get Φ^l, and transform $X_{j_l} \to X_{r-1}$. Finally, the learning objective is SVME if we have T-access to P_r^+, or KISE if we have k-access to P_r^+.

The detailed algorithm is listed in Algorithm 1. One can prove that for any $t \in [0, l]$ and any $j \in [j_t, j_{t+1} - 1]$, $\Phi^t(X_j)$ has the same distribution as $(T_{k_j}T_{k_{j-1}}\cdots T_{k_1}\Phi)(X)$. Hence, $\Phi^l(X_{r-1})$ has the same distribution as $(T_{k_{r-1}}T_{k_{j-1}}\cdots T_{k_1}\Phi)(X)$. By Theorem 2.11, this algorithm extracts the top-d eigenspace of the operator.

Standardizing kernels. If we only have k-access to some contexts, then it is important to standardize these kernels. We say that a *p.s.d.* kernel k_j is **standardized**, if $\mu_0 \equiv 1$ is an eigenfunction of T_{k_j} with eigenvalue 1, and all eigenvalues of T_{k_j} belong to $[0, 1]$. To see why standardizing $k_1, \cdots, k_r$ is necessary, consider a case where k_1 has much larger eigenvalues than the other kernels; then, when mixing these contexts, context 1 could dominate over the other contexts. We can standardize any *p.s.d.* kernel in three steps:

(i) Center the kernel: $k(x, x') \leftarrow \tilde{k}(x, x')$, which makes μ_0 an eigenfunction with eigenvalue 0. The centered kernel is defined as follows.

Definition 4.3. *For any p.s.d. kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, its **centered kernel** is given by*

$$\tilde{k}(x, x') = k(x, x') - \int k(z, x') dP_{\mathcal{X}}(z) - \int k(x, z') dP_{\mathcal{X}}(z') + \iint k(z, z') dP_{\mathcal{X}}(z) dP_{\mathcal{X}}(z'),$$

which is a p.s.d. kernel that satisfies $\int k(x, z) dP_{\mathcal{X}}(z) = \int k(z, x) dP_{\mathcal{X}}(z) = 0$ for any x.

(ii) Divide k by its largest eigenvalue, which makes all its eigenvalues at most 1.

(iii) $k(x, x') \leftarrow k(x, x') + 1$, which makes μ_0 an eigenfunction with eigenvalue 1.

The only "hyperparameter" we need to tune for a convolution is the order of the contexts, because convolution is not commutative. In practice today, the order usually does not matter too much. For example, it matters very little whether we first translate an image and then crop it, or we first crop it and then perform the translation. However, as we obtain more complex contexts especially through context scaling, this order could become very important.

Apart from supervised learning with feature maps discussed in Section 2.4, another example of convolution is supervised contrastive learning [85]. Given a sample X, supervised contrastive learning first randomly samples X_1 that has the same class as X, and then augments $X_1 \to A_2$ via cropping, flipping, etc. Supervised contrastive learning learns the convolution of the class context P_1^+ and the augmentation context P_2^+.

Finally, let us discuss when we should use convolution. In practice, convolution is used to create "harder" pretraining tasks. For example, in self-supervised learning, multiple weak data augmentations are composed together to create a stronger augmentation. One great example is SimCLR [23], whose success is largely due to its aggressive crop ratio and color distortion, both of which make the augmentation stronger. When an augmentation is stronger, the association between X and A becomes weaker. Hence, convolution should be used when all contexts have strong associations, because it always weakens the association.

4.2 Convex Combination

A convex combination of $P_1^+, \cdots, P_r^+$ is written as $w_1 P_1^+ + \cdots + w_r P_r^+$. Usually we require that $\boldsymbol{w} = [w_1, \cdots, w_r] \in \Delta^{r-1}$, so that it is a probability distribution over the r contexts. In this case, the convex combination can be understood as follows: given an input X, one first samples one P_j^+ from the probability distribution $\boldsymbol{w}$ over the r contexts, and then samples $A \sim P_j^+(\cdot|X)$. This A is the context variable of the convex combination. Theoretically speaking, $w_1, \cdots, w_r$ can be any real values, but we assume that $\boldsymbol{w} \in \Delta^{r-1}$ in this chapter unless stated otherwise.

Learning the contexture of a convex combination is simple: we only need to use a weighted sum of individual learning objectives. Assume that we have either pair access or k-access to every context. For each context j, we can learn its contexture using either SVME or KISE. Define its individual objective as

$$\mathcal{R}_j = \begin{cases} \mathbb{E}_{X \sim P_{\mathcal{X}}} \mathbb{E}_{A_j \sim P_j^+(\cdot|X)} \left[\|\Phi(X) - \Psi_j(A_j)\|_2^2 \right], & \text{pair access to } P_j^+; \\ \mathbb{E}_{X \sim P_{\mathcal{X}}} \left[\left\| \tilde{\Phi}(X) \right\|_2^2 - \left\langle \tilde{\Phi}(X), T_{k_j} \tilde{\Phi}(X) \right\rangle \right], & k\text{-access to } P_j^+. \end{cases}$$

Then, we can learn the contexture of their convex combination by minimizing $\sum_j w_j \mathcal{R}_j$ subject to $\mathrm{Cov}_{P_{\mathcal{X}}}[\Phi] = \boldsymbol{I}$. This approach might need more than two encoders, since it

requires one Ψ_j for each context j with pair access. An alternative approach is to convert every pair access to k-access first, but this leads to a huge overhead.

If we have pair access to context j, let $k_j = k^+_{X_j}$; if we have k-access to context j, let k_j be the kernel we have access to, and assume that it has been standardized. The following result shows that the dual kernel of the convex combination is $w_1k_1 + \cdots + w_rk_r$, the linearly combined kernel. If every k_j is standardized and $\boldsymbol{w} \in \Delta^{r-1}$, then the linearly combined kernel is also standardized. Linearly combining multiple kernels is a classical technique in multiple kernel learning [47].

Theorem 4.4 (Proof in Appendix C.1)**.** *Let Φ^* be a minimizer of the weighted sum of objectives $\sum_j w_j\mathcal{R}_j$ subject to $\mathrm{Cov}_{P_\mathcal{X}}[\Phi] = \boldsymbol{I}$. Then, $\tilde{\Phi}^*$ extracts the top-d eigenspace of $\sum_j w_jk_j$.*

Convolution and convex combination have the right distributive property.

Proposition 4.5. $(w_1P^+_1 + \cdots + w_rP^+_r) \star P^+_0 = w_1P^+_1 \star P^+_0 + \cdots + w_rP^+_r \star P^+_0$.

Proof Let k_L be the dual kernel of the left, and k_R be the dual kernel of the right. It suffices to show that $T_{k_L} = T_{k_R}$. Using what was proved earlier, we have

$$T_{k_L} = T_{k_0}(w_1T_{k_1} + \cdots + w_rT_{k_r})T_{k_0} = w_1T_{k_0}T_{k_1}T_{k_0} + \cdots + w_rT_{k_0}T_{k_r}T_{k_0} = T_{k_R},$$

as desired. ■

Remark 4.6. *The left distributive property does not hold, that is, $P^+_0 \star (w_1P^+_1 + \cdots + w_rP^+_r)$ and $w_1P^+_0 \star P^+_1 + \cdots + w_rP^+_0 \star P^+_r$ are not necessarily equal.*

The hyperparameters we need to tune for the convex combination are the weights $w_1, \cdots, w_r$, which should be chosen based on the associations of the contexts. Usually, if a context is very weak or strong, then we would give it a small weight to limit its impact. For example, in RLHF we have two contexts: the alignment context and the reference model context. The alignment context has a strong association—for a given prompt, there are multiple possible valid completions A, but alignment selects a small number of preferred completions from all valid ones. Consequently, alignment reduces the conditional entropy $H(A|X)$, so A has a stronger association with X. The more selective the critic, the stronger the association. On the other hand, the reference model context has a moderate association, provided that the reference model is well trained. Therefore, in practice, people usually give a much larger weight to the reference model context than to the alignment context, so that the model will not be too different from the reference model.

Convex combination is usually used to balance strong and weak associations. It can also be used when all contexts have very strong associations. In this case, we want to select the $w_1, \cdots, w_r$ that weaken the association as much as possible. We show that this can be achieved by playing a zero-sum game between a Φ-player who learns the encoders, and a $\boldsymbol{w}$-player who picks $\boldsymbol{w}$ to maximize the loss of the Φ-player. The game has the following minimax form.

$$\underset{\Phi:\mathcal{X}\to\mathbb{R}^d;\ \Psi_j:\mathcal{A}_j\to\mathbb{R}^d}{\text{minimize}} \quad \max_{\boldsymbol{w}\in\Delta^{r+1}} \mathcal{L}(\Phi, \Psi_1, \cdots, \Psi_r; \boldsymbol{w}) := \sum\nolimits_{j=1}^{r} w_j\mathcal{R}_j \quad \text{s.t. } \mathrm{Cov}_{P_\mathcal{X}}[\Phi] = \boldsymbol{I}. \quad (4.1)$$

Φ is allowed to be randomized, that is the Φ-player can use a mixed strategy. Obviously, the $\boldsymbol{w}$-player has an optimal pure strategy, so $\boldsymbol{w}$ need not to be randomized. Let us analyze the Nash equilibrium of this game. First, notice that the Φ-player only needs to pick Φ. Once Φ is picked, the optimal Ψ_j's can be determined as follows.

Algorithm 2 Convex combination: Solving the minimax game

Input: Step size $\eta > 0$

1: Initialize encoder $\Phi : \mathcal{X} \to \mathbb{R}^d$, and $\Psi_j : \mathcal{A}_j \to \mathbb{R}^d$ if needed; $\boldsymbol{w} \leftarrow [1/r, \cdots, 1/r]$

2: **for** training step $t = 1, 2, \cdots, T$ **do**

3: Fix $\boldsymbol{w}$, and find the optimal Φ, Ψ_j to Eqn. (4.1). Denote the optimal Φ at step t by Φ^t

4: Compute $\mathcal{R}_1, \cdots, \mathcal{R}_r$; Update $w_j \leftarrow w_j \cdot \exp(\eta \mathcal{R}_j)$, then normalize $w_j \leftarrow \frac{w_j}{\sum_{i=1}^r w_i}$

5: The Φ-player picks the uniform distribution over $\Phi^1, \Phi^2, \cdots, \Phi^T$ (a randomized Φ)

Proposition 4.7. *Suppose context j has pair access. Then, when Φ is fixed, the optimal Ψ_j that minimizes $\mathcal{R}_j$ is $\Psi_j^* = T^*_{P_j^+}\Phi$.*

Proof This is the same as the proof of Theorem 2.11. ■

When both players play optimally and the game reaches Nash equilibrium, the value of $\mathbb{E}_\Phi[\mathcal{L}(\Phi, \Psi_1, \cdots, \Psi_r; \boldsymbol{w})]$ is called the value of this game, denoted by $\mathcal{L}^*$. The expectation is taken over the randomness of Φ. The following result gives the optimal strategy of the $\boldsymbol{w}$-player, and the formula for $\mathcal{L}^*$.

Theorem 4.8. *Let $1 = \lambda_0(\boldsymbol{w}) \geq \lambda_1(\boldsymbol{w}) \geq \cdots$ be the eigenvalues of $T_{w_1 k_1 + \cdots + w_r k_r}$. Let $\boldsymbol{w}^*$ be the optimal strategy of the $\boldsymbol{w}$-player. Then, the game value of Eqn.* (4.1) *is*

$$\mathcal{L}^* = d - \sum\nolimits_{i=1}^d \lambda_i(\boldsymbol{w}^*), \quad \text{and } \boldsymbol{w}^* \text{ minimizes } \sum\nolimits_{i=1}^d \lambda_i(\boldsymbol{w}) \text{ over all } \boldsymbol{w} \in \Delta^{r+1}.$$

Proof For a fixed $\boldsymbol{w}$, when $\tilde{\Phi}$ learns the contexture of $\sum_j w_j k_j$, the loss is $\mathcal{L} = d - \sum_{i=1}^d \lambda_i(\boldsymbol{w})$. Thus, the optimal $\boldsymbol{w}^*$ that maximizes $\mathcal{L}$ must minimize $\sum_{i=1}^d \lambda_i(\boldsymbol{w})$. ■

To solve this game, we can use an algorithm similar to the Hedge algorithm in online learning [42]. The algorithm is listed in Algorithm 2. A standard result in online learning shows that this algorithm can find the value of this game.

Theorem 4.9 (Proof in Appendix C.2)**.** *Let $\mathcal{R}_j^t$ be the loss $\mathcal{R}_j$ in step t. Suppose there exists a constant $C > 0$ such that $\mathcal{R}_j^t \leq C$ holds for all t, j. Denote $\mathcal{L}(\boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^T \left(\sum_{j=1}^r w_j \mathcal{R}_j^t\right)$. If $\eta = \frac{\sqrt{\log r}}{C\sqrt{T}}$ where $T > \log r$, then*

$$\sup_{\boldsymbol{w} \in \Delta^{r+1}} \mathcal{L}(\boldsymbol{w}) \leq \max_{\boldsymbol{w} \in \Delta^{r+1}} \min_{\Phi, \Psi_1, \cdots, \Psi_r} \sum_{j=1}^r w_j \mathcal{R}_j + \frac{2C\sqrt{\log r}}{\sqrt{T}} = \mathcal{L}^* + \frac{2C\sqrt{\log r}}{\sqrt{T}},$$

which implies that $\mathcal{L}(\boldsymbol{w}) \to \mathcal{L}^*$ as $T \to \infty$.

Convex combination cannot be used when all contexts have weak associations, because the decay rate of its eigenfunctions is upper bounded by that of the strongest context. That is, convex combination cannot make the association stronger.

4.3 Concatenation

Concatenation is easy to implement: For each context j, one trains an encoder Φ_j, and then concatenates them into a single encoder by $\Phi(x) = [\Phi_1(x), \cdots, \Phi_r(x)]$. Concatenation is used when all contexts have weak associations. We cannot learn rich features from

Base operation	Concatenation	Convolution	Convex combination
When to use?	All weak associations	All strong associations	Mixed weak/strong

Table 4.1: When to use each base operation of mixing contexts.

a context with a weak association, but if we have several such contexts and concatenate them, then the features will be richer. Since concatenation could lead to redundant features that increase the sample complexity, it should not be used for contexts with very strong associations.

To further elaborate on concatenation, let us suppose $\mathcal{X}$ is a finite set, and $|\mathcal{X}| = N$. Then, the dual kernel of each context j is an $N \times N$ matrix denoted by $\boldsymbol{K}_j$. The dual kernel of the concatenation can be understood as $\begin{pmatrix} \boldsymbol{K}_1 & & & 0 \\ & \boldsymbol{K}_2 & & \\ & & \ddots & \\ 0 & & & \boldsymbol{K}_r \end{pmatrix}$, whose eigenvalues are the union of the eigenvalues of every individual $\boldsymbol{K}_j$. Therefore, the eigenvalues of the concatenation decay more slowly than those of any individual context.

The hyperparameters of concatenation are the dimensions of $\Phi_1, \cdots, \Phi_r$, denoted by $d_1, \cdots, d_r$. One good way to select them is the following: first, estimate the singular values $s_1^{(j)}, s_2^{(j)}, \cdots$ of each context j using the post-hoc approach in Section 2.5; then, select $d_1, \cdots, d_r$ such that $s_{d_1+1}^{(1)}, s_{d_2+1}^{(2)}, \cdots, s_{d_r+1}^{(r)}$ are all close and small.

Table 4.1 summarizes when to use each of the three base operations. We can see that these three operations cover all possible scenarios.

4.4 Application to Tabular Data

The methods of mixing contexts provide us with an opportunity to create better contexts almost effortlessly. Here we test these methods on real-world tabular datasets. We focus on tabular data for two reasons:

(i) It is important: Tabular data is the most common type of data in industry.

(ii) It is challenging: So far deep learning has not been as successful on tabular data as it is on other modalities. In particular, XGBoost [22] has long been the state of the art on tabular data.

There are several complications of tabular data that make it more difficult for deep learning than image and text:

(i) Heterogeneous features, such as categorical and numerical columns.

(ii) Features have different meanings, unlike *e.g.* images where all features are pixels.

(iii) Much lower signal-to-noise ratio than other modalities, and missing values (NaNs).

Setup. We use 118 datasets and run a grid search on tens of thousands of hyperparameter combinations for each method and dataset. Using a large number of datasets and hyperparameters ensures that the bias of dataset and hyperparameter selection is reduced as much as possible. We focus on prediction tasks, and use 98 classification datasets and 20 regression datasets (labeled `reg-20`). The 98 classification datasets further consist of 56 smaller ones (labeled `cls-56`) with fewer than 1500 samples, and 42 larger ones (labeled `cls-42`). All datasets are real-world datasets from OpenML [146],

	Linear	Rand-Forest	CatBoost	XGBoost	MLP	ResNet	FT-Transformer	TabPFN
`cls-56` Perf	$79.03_{0.28}$	$82.01_{0.31}$	$82.79_{0.29}$	$82.03_{0.30}$	$83.59_{0.29}$	$\mathbf{83.63_{0.29}}$	$82.84_{0.30}$	$82.96_{0.30}$
`cls-56` Rank	5.38	4.93	4.11	5.00	3.14	**2.95**	3.84	3.75
Prior work*	61	76	85	74	57	77	75	84
`cls-42` Perf	$85.05_{0.09}$	$89.03_{0.08}$	$89.71_{0.09}$	$\mathbf{90.43_{0.08}}$	$90.04_{0.10}$	$90.03_{0.09}$	$90.42_{0.09}$	$87.98_{0.08}$
`cls-42` Rank	6.26	4.36	4.07	2.83	3.17	3.05	**2.81**	5.10
`reg-20` Perf	$56.60_{1.20}$	$78.76_{0.43}$	$\mathbf{80.35_{0.37}}$	$80.26_{0.41}$	$78.22_{0.41}$	$78.37_{0.44}$	$57.41_{0.36}$	N/A
`reg-20` Rank	6.40	4.10	3.20	**2.55**	3.30	3.65	3.95	N/A

Table 4.2: Baseline average performances (accuracy or R^2-score) (%) and rankings. Standard deviations reported in the subscripts. *For `cls-56`, we compare with the numbers reported in the prior work by [105, Table 2], whose caption says "57 data sets" but one data set is actually duplicated.

and they cover a wide range of domains. We use 10 train-val-test splits (which we call 10 folds) for each dataset and report the standard deviation of performance. For evaluation, our metrics are the accuracy for classification, and the R^2-score for regression. We consider two performances close if their difference is less than one standard deviation.

Baselines. We start by evaluating the baseline methods and two recent methods that are widely compared to in the literature—FT-Transformer [49] and TabPFN [65]. Table 4.2 reports the average performances and rankings of eight baseline methods. For the rankings, we rank the methods from 1 to 8 on each dataset (ties get the same ranking) and take the average for every method. From the table we can see that:

- Compared to the prior work [105], most methods (except CatBoost and TabPFN) get much higher performance in our experiments. In their paper, ResNet was reported to be much better than MLP, which is not very reasonable. On the other hand, in our experiments, MLP and ResNet have almost the same performance. This shows that careful hyperparameter tuning gives us more accurate baselines.
- To our surprise, MLP and ResNet perform the best on the 56 small classification data sets, though conventional wisdom suggests that deep learning is bad on small data sets. However, XGBoost has a much higher performance than MLP on the 42 larger classification data sets and the regression data sets. Based on this observation, we use MLP and XGBoost as our main baselines.

Methods. We consider the following four types of contexts:

(i) Y-Linear kernel (Y-Lin): We use the centered linear kernel on Y defined after Theorem 2.6, that is $k(y, y') = \langle \tilde{y}, \tilde{y}' \rangle$. More specifically, we use an STK of the kernel, which transforms all eigenvalues above threshold $c = 0.1$ to 1 and the rest to 0.

(ii) XGBoost (XGB): An XGBoost model consists of d_t trees. We construct a teacher model $\Phi_t : \mathcal{X} \to \mathbb{R}^{d_t}$, where each dimension is the output of one tree.

(iii) SCARF [4] (SF): Randomly masks some columns and replaces them with random values sampled from $\mathrm{Unif}[c_{\min}, c_{\max}]$, where $c_{\min}$ and $c_{\max}$ are the smallest and largest values of this column in the training set.

(iv) Cutmix (CM): Randomly masks some columns and replace them with values from the same column but other random rows.

We use $\oplus$ to denote concatenation. For example, "$\oplus$ XGB" refers to concatenation with the XGBoost teacher model. We use "$+$" to denote convex combination, where the

	XGBoost	MLP	Y-Lin	SCARF	Cutmix	SCARF + Y-Lin	Cutmix + Y-Lin
`cls-56` Perf	$82.03_{0.30}$	$83.59_{0.29}$	$83.58_{0.31}$	$82.08_{0.30}$	$81.93_{0.30}$	$83.49_{0.32}$	$\mathbf{83.64_{0.32}}$
`cls-56` Rank	4.89	**2.96**	3.25	4.00	4.18	3.36	3.05
`cls-42` Perf	$90.43_{0.08}$	$90.04_{0.10}$	$\mathbf{90.60_{0.07}}$	$88.15_{0.09}$	$88.06_{0.09}$	$90.55_{0.08}$	$90.56_{0.08}$
`cls-42` Rank	3.19	3.12	**2.45**	4.81	4.88	**2.45**	2.50

Table 4.3: Results on the 98 classification datasets.

	XGBoost	MLP	Y-Lin	SF + Y-Lin	CM + Y-Lin	(SF + Y-Lin) $\oplus$ XGB	(CM + Y-Lin) $\oplus$ XGB
`reg-20` Perf	$80.26_{0.41}$	$78.22_{0.41}$	$78.80_{0.61}$	$79.17_{0.45}$	$78.99_{0.53}$	$\mathbf{81.07_{0.47}}$	$81.04_{0.52}$
`reg-20` Rank	3.55	3.95	3.90	3.60	4.10	**2.50**	2.65

Table 4.4: Results on the 20 regression datasets. We omit SCARF and Cutmix because their performances are low.

weights $\boldsymbol{w}$ are the minimax weights in Eqn. (4.1).

Results. Table 4.3 reports the performance of seven methods on the classification datasets. We can see that only using SCARF or Cutmix leads to pretty bad performance, largely because their association is too strong, as shown in Section 3.3. However, when they are mixed with Y-Lin, the mixture has a weaker association, and thus the performance becomes much better. The performance of Y-Lin is very close to MLP on the 56 small datasets, but much higher than MLP on the 42 large datasets. This suggests that representation learning usually works better with larger datasets.

Table 4.4 reports the performance of seven methods on the regression datasets. We can see that Y-Lin is better than MLP but worse than XGBoost. Mixing Y-Lin with SCARF or Cutmix slightly improves the performance, and concatenating with XGBoost further significantly improves the performance. The mixture of SCARF, Y-Lin and XGBoost achieves a much higher average performance than XGBoost.

The above results show that if we mix the right set of contexts, then we can achieve a higher performance than a single context. Moreover, on all three benchmarks our methods significantly improve over XGBoost, and on all but `cls-56` our methods significantly improve over MLP. These experiments showcase the practical value of the contexture theory. Note that these experiments are only a start, and we expect there to be a large room of improvement if we can find better contexts for tabular data.

In summary, when we have multiple contexts with either strong or weak associations, we can mix them to obtain a better context with a moderate association. This chapter introduced three base operations: convolution, convex combination and concatenation. While mixing multiple contexts is a useful method, to achieve a revolutionary breakthrough, we still need to create new contexts that are completely different from existing ones. In other words, we are not suggesting that context scaling can be achieved by solely mixing existing contexts.

Chapter 5

Statistical Learning Bounds for Representation Learning

This chapter studies representation learning in the finite sample regime, that is when there are only finite pretraining and downstream samples, how well one can learn the encoder and the downstream predictor. Recall that the prediction error can be decomposed as the sum of the approximation error and the estimation error. The key takeaways of this chapter are summarized as follows.

(i) When the embedding dimension d increases, the approximation error decreases, but the estimation error increases.

(ii) The approximation error consists of two parts: (a) the distance from the target function to the function class; (b) how well the function class can be approximated.

(iii) Contexture learning (extracting the top-d eigenspace) can be viewed as a spectrally transformed kernel (STK). It transforms all eigenvalues other than the top-d to be zero. It loses some information, but achieves the fastest eigenvalue decay.

(iv) Other STKs such as the inverse Laplacian are popular in semi-supervised learning, because they are more efficient than extracting the top-d eigenspace.

This chapter first defines the context complexity, and then proves the generalization bounds for contexture learning. After that, we generalize contexture learning to spectrally transformed kernel regression (STKR), and prove generalization bounds for STKR.

5.1 Context Complexity

So far, our intrinsic evaluation of a context has only relied on the singular values of the context. However, the singular functions of the context, particularly their smoothness, also have a great impact on the generalization performance. Consider two contexts with similar spectra, but the first one has smoother singular functions than the second one. Then, given the same number of pretrain samples, it is easier to approximate the top-d singular functions of the first context. The context complexity is defined to mathematically characterize such smoothness.

Definition 5.1. *The **context complexity** of P^+ is defined as $\kappa := \left\|k_X^+\right\|_\infty^{1/2}$, such that*

$$k_X^+(x,x) = \sum s_i^2 \mu_i(x)^2 = \int \frac{P^+(x|a)P^+(a|x)}{P_{\mathcal{X}}(x)} da = D_{\chi^2}\big(P^+(\cdot|x) \parallel P_{\mathcal{A}}\big) + 1 \le \kappa^2$$

holds for $P_{\mathcal{X}}$-almost all x, where $D_{\chi^2}(P \parallel Q) = \int (\frac{dP}{dQ} - 1)^2 dQ$ is the χ^2-divergence.

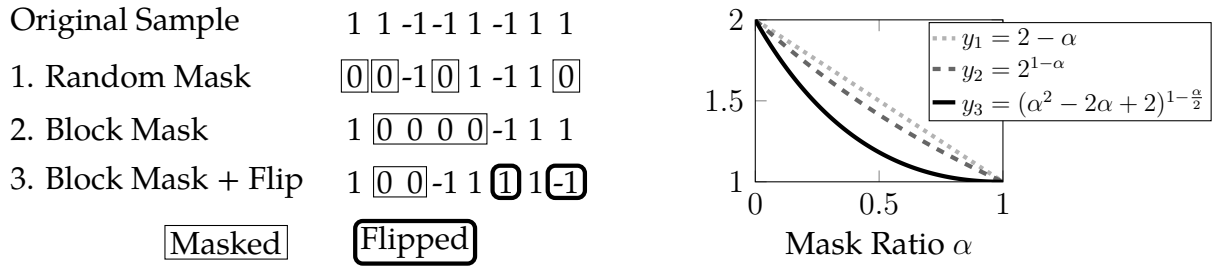


Figure 5.1: **Left:** Three mask-type data augmentations on the hypercube data model. **Right:** Their theoretical $\kappa^{2/d_{\mathcal{X}}}$ with different mask ratio α.

This κ was initially introduced as the *augmentation complexity* by [167] in the context of self-supervised learning. If κ is finite, then $\kappa^2 \geq \int k_X^+(x,x)dP_{\mathcal{X}}(x) = \sum s_i^2$, which means that $T_{k_X^+}$ is a trace-class operator. If k is the centered kernel of k_X^+, then $k(x,x) = \sum_{i\geq 1} s_i^2 \mu_i(x)^2 \leq \kappa^2 - 1$. Usually $\kappa \gg 1$, so we use $k(x,x) \leq \kappa^2$ for simplicity.

Now let us see some examples of masking, and estimate their context complexity. Intuitively, the context of a data augmentation is more complex if the augmentation is stronger, that is the association between X and A is weaker. For masking, the mask ratio clearly controls the context complexity. In addition, the complexity also depends on the type of masking. For example, consider a checkerboard-style masking, where for any two adjacent pixels, exactly one of them is masked. This masking has a mask ratio of 50%, but clearly it is much weaker than the one that puts all 50% masking on the center of the image where the object is located. The context complexity provides a quantitative way to measure the strength of a data augmentations of different types.

Hypercube data model. Consider the hypercube data model introduced by [122]: $\mathcal{X} = \{-1,1\}^{d_{\mathcal{X}}}$, and $P_{\mathcal{X}}$ is the uniform distribution over $\mathcal{X}$. Consider three random masking methods similar to those studied in [20]: (i) Independent random masking; (ii) Cutout-like block masking [35]; (iii) BERT-like masking. See Figure 5.1 (left) for an illustration. Denote the mask ratio by α. Let us compute the κ for these three masking methods, which are denoted by $\kappa_r, \kappa_c, \kappa_b$, respectively.

Example 5.2. *Consider a* random masking *augmentation, i.e. for any $x \in \mathcal{X}$, each coordinate $x^{(i)}$ is randomly and independently masked to be* 0 (*i.e.* 0 *denotes the* `[MASK]` *token*) *with probability $\alpha \in (0,1)$. Then, its context complexity is given by $\kappa_r^2 = (2-\alpha)^{d_{\mathcal{X}}}$.*

Example 5.3. *Consider* random block masking, *i.e. masking $x^{(i)}, x^{(i+1)}, \cdots, x^{(i+r-1)}$ for $r = \lceil \alpha d_{\mathcal{X}} \rceil$ and a uniformly random $i \in [d_{\mathcal{X}} - r]$, for any $x \in \mathcal{X}$. Then, $\kappa_c^2 \leq [2^{(1-\alpha)}]^{d_{\mathcal{X}}}$.*

Example 5.4. *Consider* random block masking with flipping, *where for any $x \in \mathcal{X}$, first mask $x^{(i)}, \cdots, x^{(i+r-1)}$ to be* 0 *for $r = \lceil \alpha d_{\mathcal{X}} \rceil$ and a uniformly random $i \in [d_{\mathcal{X}} - r]$, then randomly flip the sign of each remaining coordinate independently with probability $\frac{\alpha}{2}$. Then, its context complexity is bounded by $\kappa_b^2 \leq \left[(\alpha^2 - 2\alpha + 2)^{(1-\alpha/2)}\right]^{d_{\mathcal{X}}}$.*

See Appendix D.1 for the derivation of the above κ. Figure 5.1 (right) plots the $\kappa^{2/d_{\mathcal{X}}}$ for all three examples. We can see that κ becomes lower as the mask ratio α increases. Moreover, when $\alpha \in (0,1)$ is fixed, Cutout-like masking has a lower κ than independent random masking, and BERT-like masking has a lower κ than Cutout-like masking. Cutout has a weaker association than random masking, and BERT has an even weaker

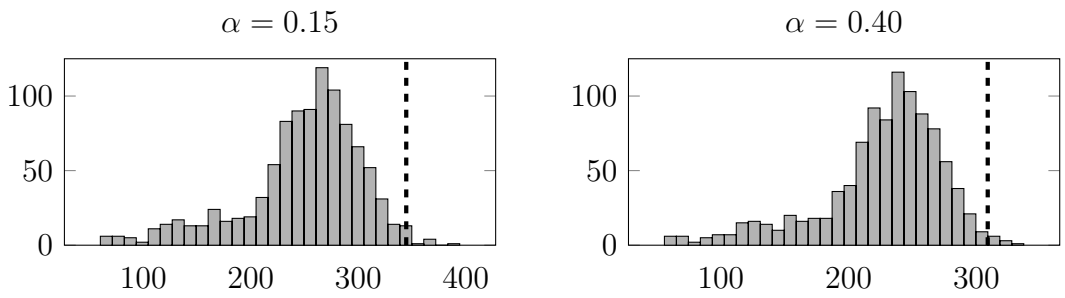


Figure 5.2: Histograms of $\log k_X^+(x,x)^2$ for random masking on `wikipedia-simple` with mask ratio α. The dashed vertical line in each plot indicates the 99^{th} percentile.

association. Thus, these examples show that κ is lower when the association is weaker.

Another observation is that all three κ have an exponential dependency on $d_{\mathcal{X}}$. This is a manifestation of the typical curse of dimensionality in high-dimensional statistics. One way to make κ polynomial in $d_{\mathcal{X}}$ is to use a context with very weak association, such as a very strong data augmentation. For example, if $\mathcal{A}$ is a finite set with a small size, then κ will be polynomial. However, contexts with such weak association usually lead to substantially worse performance in practice. The bounds to be proved in this chapter depend on κ polynomially, meaning that they are not really useful in the high-dimensional scenario. In practice, however, representation learning can still achieve good performances when the data dimension is high. How to address this discrepancy is posed as an open problem.

Real language models. Now let us estimate the κ of some real language models. We use the NLP dataset `wikipedia-simple`, and consider the context of masking tokens, where x is a complete text whereas a is a masked version of x. Recall that κ^2 is an upper bound of $k_X^+(x,x) = \int \frac{P^+(x|a)P^+(a|x)}{P_{\mathcal{X}}(x)} da$. For a fixed x, this integration can be estimated with Monte Carlo (by sampling a set of $a \sim P^+(\cdot|x)$), and then κ^2 can be estimated by its maximum over $x \in \mathcal{X}$. For $x = [x^{(1)}, \cdots, x^{(l)}]$ where $x^{(i)}$ is the i^{th} token, we have

$$\log P^+(x|a) = \log P^+\left(x^{(1)}\middle|a\right) + \log P^+\left(x^{(2)}\middle|a, x^{(1)}\right) + \cdots + \log P^+\left(x^{(l)}\middle|a, x^{(1)}, \cdots, x^{(l-1)}\right).$$

We can leverage a bi-directional masked language model such as a BERT, and then compute $P^+(x^{(i)}|a, x^{(<i)})$ auto-regressively: For each $i \in [l]$, use the BERT to output $P^+(x^{(i)}|a, x^{(<i)})$, and then replace $a^{(i)}$ with $x^{(i)}$ for $i+1$. As such, we can estimate $P^+(x|a)$, and $P_{\mathcal{X}}(x)$ can be estimated by $P^+(x|a_0)$ where a_0 is a fully masked text.

A natural idea is to estimate $k_X^+(x,x)$ for a random subset of samples, and output their maximum as an estimate of κ^2. However, this approach has two issues. First, $\sup_x k_X^+(x,x)$ is statistically impossible to estimate from a subset of data without any extra assumptions on the distribution of $k_X^+(x,x)$. Second, almost all real datasets contain outliers, which are very different from most samples. These outliers have very large $k_X^+(x,x)$, but given that these outliers have little impact on the actual pretraining, we do not want to take these large $k_X^+(x,x)$ into account. To fix these two issues, we can use the 99^{th} percentile of $k_X^+(x,x)$. First, the percentile can be estimated with a finite confidence interval via sampling regardless of the distribution of $k_X^+(x,x)$ [54, Section 5.2]. Second, we get rid of the outliers if they are fewer than 1 percent. Figure 5.2 plots the histograms of $\log k_X^+(x,x)^2$ for random masking on `wikipedia-simple`. The dashed line in each plot indicates the 99^{th} percentile. We can see from the plots that the 99^{th} percentile is a good choice, as it picks out the outliers where $k_X^+(x,x)$ is too large.

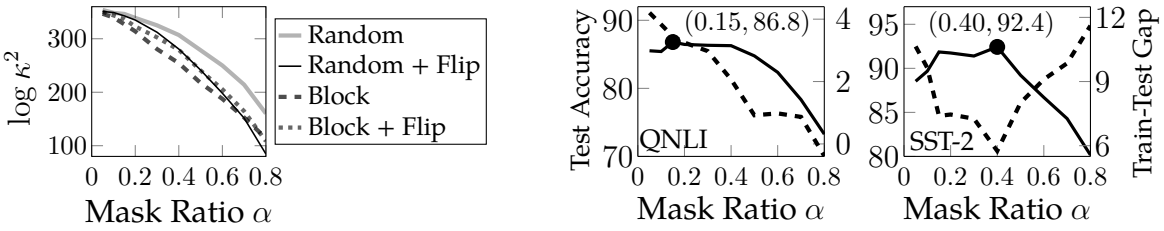


Figure 5.3: **Left:** Estimated $\log \kappa^2$ (99^{th} percentile) on `wikipedia-simple`, which is the average of five runs with different random seeds. **Right:** Downstream performance on QNLI and SST-2. The solid line is the test accuracy and the dashed line is the train-test gap. The highest test accuracy is labeled on each plot.

Figure 5.3 (left) plots the 99^{th} percentile of $\log \kappa^2$ of four contexts: random masking, random masking with flipping, block masking, and block masking with flipping. Masking randomly masks α of the tokens. Masking with flipping masks $\alpha/2$ of the tokens and replaces another $\alpha/2$ of the tokens with random tokens. Note that this replace rate is higher than the common 80-10-10 strategy in NLP, because we want to magnify the effect of flipping. From the plot, we can see that the complexity drops as α increases as expected. One observation is that the "Random + Flip" curve intersects with "Block" and "Block + Flip", suggesting that block masking has a stronger effect when α is small, whereas flipping has a stronger effect when α is large.

Figure 5.3 (right) plots the real downstream performance of BERT with different mask ratios on QNLI [149] and SST-2 [130]. The models are `roberta-large` trained with the fast pretraining recipe in [158]. The context is random masking without the 80-10-10 strategy. At downstream, the encoder is fine-tuned along with the linear head following common practice. From the plot, we can see that the highest test accuracy (solid line) is achieved at $\alpha = 0.15$ on QNLI and at $\alpha = 0.40$ on SST-2. This is because the association between X and A is moderate when α is neither too big nor too small. The dashed line is the gap between the train accuracy and test accuracy. On QNLI, the gap monotonically decreases with α; on SST-2, it is U-shaped, with the lowest at $\alpha = 0.40$. We will come back to explain this observation after we prove the generalization bounds.

5.2 Generalization Bounds for Contexture Learning

Suppose there are m pretrain samples $x_1, \cdots, x_m$ and n downstream samples $\tilde{x}_1, \cdots, \tilde{x}_n$, jointly *i.i.d.* sampled from $P_{\mathcal{X}}$. Usually $m \gg n$. This section proves an error bound when an encoder is learned with $x_1, \cdots, x_m$, and then a predictor is fit with $\tilde{x}_1, \cdots, \tilde{x}_n$. Assume that the context has kernel access, and assume that the kernel k that we have access to is the centered kernel of k_X^+. In practice there could be a difference between k_X^+ and k, but since this difference depends on what k we have, this is not something we can analyze in our study of generalization. Therefore, for simplicity we ignore this difference. The scenario where the context has pair or transformation access will be discussed later.

Let s_i^2, μ_i be the actual eigenvalues and eigenfunctions of T_k. The **empirical top-d eigenfunctions** of T_k can be estimated with kernel PCA as introduced at the beginning of Chapter 2, which is reiterated as follows.

(i) Compute the Gram matrix $\boldsymbol{G} \in \mathbb{R}^{m \times m} : \boldsymbol{G}[i,j] = k(x_i, x_j)$.

(ii) Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$ be the eigenvalues of $\boldsymbol{G}$, with orthonormal eigenvectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m$. Note that λ_1 has the same order as m, so it can be much greater than 1.

(iii) Assume that $\lambda_d > 0$. Let $\phi_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^m \boldsymbol{v}_i[j] k(x, x_j)$, for $i \in [d]$.

Although ideally we would like a bound for any target function in $\mathcal{F}_\epsilon(P^+)$, we cannot prove a universal bound for the entire $\mathcal{F}_\epsilon(P^+)$. To see why, consider the following example. Suppose f^* is a $(1-\epsilon/2)$-compatible function that is easy to learn, and $\|f^*\| = 1$. Suppose there is a μ_D with an extremely large D, such that the functiion is very noisy and it has large values on all the samples we have. Let $f' = f^* + \frac{\epsilon}{4}\mu_D$. It is easy to show that $f' \in \mathcal{F}_\epsilon(P^+)$, but learning f' with the samples we have is impossible. In order to make learning possible, we have to get rid of the extremely noisy components.

To this end, we define a variant of the induced RKHS in Section 3.2.

Definition 5.5. *Define the **induced RKHS** of k as $\mathcal{H}_k = \{f = \sum s_i u_i \mu_i \mid \sum u_i^2 < \infty\}$. For $f_1 = \sum u_i \mu_i, f_2 = \sum v_i \mu_i \in \mathcal{H}_k$, define their inner product as $\langle f_1, f_2 \rangle_{\mathcal{H}_k} = \sum \frac{u_i v_i}{s_i^2}$.*

Note that $\mathcal{H}_k$ is a subset of $L^2(P_\mathcal{X})$, rather than $L^2(P_\mathcal{A})$. Obviously $\mathcal{H}_k$ is the RKHS of k, that is for any $f \in \mathcal{H}_k$, we have $\langle f, k(\cdot, x)\rangle_{\mathcal{H}_k} = f(x)$. In particular, we have $\langle k(\cdot, x), k(\cdot, x')\rangle_{\mathcal{H}_k} = k(x, x')$. This implies that for all $i, j \in [d]$, we have

$$\langle \phi_i, \phi_j \rangle_{\mathcal{H}_k} = \frac{1}{\sqrt{\lambda_i \lambda_j}} \sum_{q,r=1}^m \boldsymbol{v}_i[q] \boldsymbol{v}_j[r] \langle k(\cdot, x_q), k(\cdot, x_r)\rangle_{\mathcal{H}_k} = \frac{1}{\sqrt{\lambda_i \lambda_j}} \boldsymbol{v}_i^\top \boldsymbol{G} \boldsymbol{v}_j = \mathbb{I}[i = j].$$

That is, $\phi_1, \cdots, \phi_d$ are orthonormal in the Hilbert space $\mathcal{H}_k$. Note that when there are finite samples, it is impossible to make $\phi_1, \cdots, \phi_d$ orthonormal in $L^2(P_\mathcal{X})$ since we have no access to $P_\mathcal{X}$. Given $\mathcal{H}_k$, define the following set of compatible functions.

Definition 5.6. *The set of functions $(\mathbf{1-\epsilon})$-**compatible with $\mathcal{H}_k$** is defined as*

$$\mathcal{F}_{\mathcal{H}_k}(P^+) = \left\{ f \in \mathcal{H}_k \;\middle|\; \left\|\tilde{f}\right\|_{\mathcal{H}_k} \geq \left\|\tilde{f}\right\|_{P_\mathcal{X}} \geq (1-\epsilon)\left\|\tilde{f}\right\|_{\mathcal{H}_k} \right\}.$$

Proposition 5.7. $\mathcal{F}_{\mathcal{H}_k}(P^+) \subseteq \mathcal{F}_\epsilon(P^+)$.

Proof Let $f = \sum_i u_i \mu_i \in \mathcal{F}_{\mathcal{H}_k}(P^+)$. Then, $\sum_i u_i^2 \geq (1-\epsilon)\sum_i \frac{u_i^2}{s_i^2}$. By Cauchy-Schwarz inequality, we have $(\sum_i s_i^2 u_i^2)\left(\sum_i \frac{u_i^2}{s_i^2}\right) \geq (\sum_i u_i^2)^2$. Thus, $\sum_i s_i^2 u_i^2 \geq \sum_i u_i^2$. ■

The following lemma can be proved in the same way as Lemma 3.8, which is left as an exercise to the reader.

Lemma 5.8. *For any $f_1, \cdots, f_d \in \mathcal{H}_k$ such that $\langle f_i, f_j \rangle_{\mathcal{H}_k} = \mathbb{I}[i = j]$, we have*

$$\|f_1\|_{P_\mathcal{X}}^2 + \cdots + \|f_d\|_{P_\mathcal{X}}^2 \leq s_1^2 + \cdots + s_d^2.$$

Approximation error bound. Our goal is to prove a universal bound for $\mathcal{F}_{\mathcal{H}_k}(P^+)$. First, let us bound the approximation error. Previously, the approximation error was defined as the distance from f^* to the span of Φ in space $L^2(P_\mathcal{X})$. However, since we cannot compute the distance in space $L^2(P_\mathcal{X})$ with only finite samples, here we define it with the distance in space $\mathcal{H}_k$. Specifically, let f_Φ be the projection of f^* onto the span of Φ in space $\mathcal{H}_k$, that is $\langle f^* - f_\Phi, f_\Phi \rangle_{\mathcal{H}_k} = 0$. Let $f^* - f_\Phi = \beta f_0$, where $\beta \geq 0$ and $\|f_0\|_{\mathcal{H}_k} = 1$. The above lemma implies that $\|\phi_1\|_{P_\mathcal{X}}^2 + \cdots + \|\phi_d\|_{P_\mathcal{X}}^2 + \|f_0\|_{P_\mathcal{X}}^2 \leq s_1^2 + \cdots + s_{d+1}^2$.

Hence, to bound $\|f_0^2\|$, it suffices to prove a lower bound for $\|\phi_1\|_{P_\mathcal{X}}^2 + \cdots + \|\phi_d\|_{P_\mathcal{X}}^2$. Using the definition of eigenvectors, it is not hard to show that $\frac{1}{m}\sum_{j=1}^m \phi_i(x_j)^2 = \frac{\lambda_i}{m}$ for

all $i \in [d]$, which is the empirical L^2 norm of ϕ_i. Therefore, we need to bound two things: (i) the gap between $\frac{\lambda_1+\cdots+\lambda_d}{m}$ and $s_1^2+\cdots+s_d^2$; (ii) the gap between the empirical L^2 norm and the actual L^2 norm.

Using [13, Theorem 3.2], for any $\delta \in (0,1)$, gap (i) can be bounded by

$$\frac{\lambda_1+\cdots+\lambda_d}{m} \geq s_1^2+\cdots+s_d^2 - \frac{\kappa^2}{\sqrt{m}}\sqrt{\frac{1}{2}\log\frac{6}{\delta}} \qquad \text{with probability at least } 1-\frac{\delta}{2}. \quad (5.1)$$

Gap (ii) can be bounded using classical generalization bounds with the Rademacher complexity.

Definition 5.9. *Let $\mathcal{F}$ be a function class. Let $S = \{x_1, \cdots, x_m\}$ be i.i.d. drawn from $P_{\mathcal{X}}$. The **empirical Rademacher complexity** of $\mathcal{F}$ on S is defined as*

$$\hat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^m \sigma_i f(x_i)\right],$$

*where $\sigma_1, \cdots, \sigma_m$ are Rademacher variables, which are i.i.d. uniform random variables taking values in $\{-1, +1\}$. The **Rademacher complexity** of $\mathcal{F}$ is defined as*

$$\mathfrak{R}_m(\mathcal{F}) = \mathbb{E}_S[\hat{\mathfrak{R}}_S(\mathcal{F})].$$

Define the following function class:

$$\mathcal{F}_d := \left\{F = f_1^2+\cdots+f_d^2 \;\middle|\; f_i \in \mathcal{H}_k;\ \forall i,j\in[d],\ \langle f_i, f_j\rangle_{\mathcal{H}_k} = \mathbb{I}[i=j]\right\}.$$

Since $\phi_1^2+\cdots+\phi_d^2 \in \mathcal{F}_d$, it suffices to bound $\left|\frac{1}{m}\sum_{i=1}^m F(x_i) - \mathbb{E}[F(X)]\right|$ for all $F \in \mathcal{F}_d$, which requires $\mathfrak{R}_m(\mathcal{F}_d)$ and a bound for $|F(x)|$. Let $\phi_i = \sum_j u_{ij}s_j\mu_j$, and $\boldsymbol{U} = (u_{ij}) = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d]$, which is a matrix with d columns and infinitely many rows. Then, $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_d$. Let $\boldsymbol{M}(x) = [s_1\mu_1(x), s_2\mu_2(x), \cdots]$. Then, by Definition 5.1, we have $\|\boldsymbol{M}(x)\|_2^2 = \sum s_i^2\mu_i(x)^2 \leq \kappa^2$ for $P_{\mathcal{X}}$-almost all x. Thus, for $P_{\mathcal{X}}$-almost all x, we have

$$|F(x)| = \left|\boldsymbol{M}(x)^\top\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{M}(x)\right| \leq \|\boldsymbol{M}(x)\|_2^2\left\|\boldsymbol{U}\boldsymbol{U}^\top\right\|_2 \leq \kappa^2 \quad \text{for all } F \in \mathcal{F}_d.$$

Regarding $\mathfrak{R}_m(\mathcal{F}_d)$, we have the following result.

Lemma 5.10 (Proof in Appendix D.2)**.** $\mathfrak{R}_m(\mathcal{F}_d) \leq \frac{\sqrt{d}}{\sqrt{m}}\kappa^2$.

Hence, by [148, Theorem 4.10], with probability at least $1-\frac{\delta}{2}$ we have

$$\left|\frac{1}{m}\sum_{i=1}^m F(x_i) - \mathbb{E}_{P_{\mathcal{X}}}[F(X)]\right| \leq \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d}+\sqrt{2\log\frac{2}{\delta}}\right) \qquad \text{for all } F \in \mathcal{F}_d,$$

which implies that

$$\|\phi_1\|_{P_{\mathcal{X}}}^2+\cdots+\|\phi_d\|_{P_{\mathcal{X}}}^2 \geq \frac{\lambda_1+\cdots+\lambda_d}{m} - \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d}+\sqrt{2\log\frac{2}{\delta}}\right). \quad (5.2)$$

Combining Eqns. (5.1) and (5.2), with probability at least $1-\delta$ we have

$$\|\phi_1\|_{P_{\mathcal{X}}}^2+\cdots+\|\phi_d\|_{P_{\mathcal{X}}}^2 \geq s_1^2+\cdots+s_d^2 - \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d}+3\sqrt{\log\frac{6}{\delta}}\right).$$

Thus, by Lemma 5.8, we get

$$\|f_0\|_{P_{\mathcal{X}}}^2 \le s_{d+1}^2 + \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d} + 3\sqrt{\log\frac{6}{\delta}}\right).$$

Let $\alpha = \|f_\Phi\|_{\mathcal{H}_k}^2 \ge s_1^{-2}\|f_\Phi\|_{P_{\mathcal{X}}}^2$. Then, we have

$$\alpha^2+\beta^2 = \|f^*\|_{\mathcal{H}_k}^2 \le \frac{\|f^*\|_{P_{\mathcal{X}}}^2}{(1-\epsilon)^2} \le \frac{\left(\|f_\Phi\|_{P_{\mathcal{X}}} + \beta\|f_0\|_{P_{\mathcal{X}}}\right)^2}{(1-\epsilon)^2} \le \frac{s_1^2\alpha^2 + \beta^2\|f_0\|_{P_{\mathcal{X}}}^2 + (\alpha^2+\beta^2)s_1\|f_0\|_{P_{\mathcal{X}}}}{(1-\epsilon)^2}.$$

With some simple algebra, we get

$$(s_1^2 - \|f_0\|_{P_{\mathcal{X}}}^2)\beta^2 \le \left[s_1^2 - (1-\epsilon)^2 + s_1\|f_0\|_{P_{\mathcal{X}}}\right](\alpha^2+\beta^2) \le \left[s_1^2 - (1-\epsilon)^2 + s_1\|f_0\|_{P_{\mathcal{X}}}\right]\frac{\|f^*\|_{P_{\mathcal{X}}}^2}{(1-\epsilon)^2}.$$

Note that $\|f^* - f_\Phi\|_{\mathcal{H}_k} = \beta$, and $\|f^* - f_\Phi\|_{P_{\mathcal{X}}} = \beta\|f_0\|_{P_{\mathcal{X}}}$. Hence, we have proved the following bound for the approximation error.

Theorem 5.11. *Let f_Φ be the projection of any $f^* \in \mathcal{F}_{\mathcal{H}_k}(P^+)$ onto the span of Φ, such that $\langle f^* - f_\Phi, f_\Phi\rangle_{\mathcal{H}_k} = 0$. Suppose $(1-\epsilon)^2 < s_1^2$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ we have*

$$\|f^* - f_\Phi\|_{\mathcal{H}_k}^2 \le \frac{s_1^2 - (1-\epsilon)^2 + s_1\sqrt{s_{d+1}^2 + \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d} + 3\sqrt{\log\frac{6}{\delta}}\right)}}{s_1^2 - s_{d+1}^2 - \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d} + 3\sqrt{\log\frac{6}{\delta}}\right)} \cdot \frac{\|f^*\|_{P_{\mathcal{X}}}^2}{(1-\epsilon)^2};$$

$$\|f^* - f_\Phi\|_{P_{\mathcal{X}}}^2 \le \frac{s_1^2 - (1-\epsilon)^2 + s_1\sqrt{s_{d+1}^2 + \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d} + 3\sqrt{\log\frac{6}{\delta}}\right)}}{s_1^2 - s_{d+1}^2 - \frac{\kappa^2}{\sqrt{m}}\left(2\sqrt{d} + 3\sqrt{\log\frac{6}{\delta}}\right)} \cdot \frac{s_1^2\|f^*\|_{P_{\mathcal{X}}}^2}{(1-\epsilon)^2},$$

assuming that the denominator is positive.

Comparing to Theorem 3.4, we can see that this bound is fairly tight when m is sufficiently large. The only differences are that the numerator has an extra term that is close to $s_1 s_{d+1}$ (s_{d+1} should be pretty small), and $\|f^*\|_{P_{\mathcal{X}}}^2$ is multiplied by $\frac{s_1^2}{(1-\epsilon)^2}$.

Estimation error bound. Next, we bound the estimation error of the predictor fit on top of Φ using the n labeled downstream samples. The bound consists of two parts:

(i) Assume that the labels are generated from $f_\Phi(\tilde{x}_i)$. Let $\hat{f}$ be the learned predictor. Then, the gap $\left\|\hat{f} - f_\Phi\right\|_{P_{\mathcal{X}}}$ can be bounded using standard generalization bounds.

(ii) However, the real labels are generated from $f^*(\tilde{x}_i)$ instead of $f_\Phi(\tilde{x}_i)$. Thus, we need to bound the gap between $f^*(\tilde{x}_i)$ and $f_\Phi(\tilde{x}_i)$.

Let the downstream training set be $\{(\tilde{x}_i, y_i)\}_{i=1}^n$ *i.i.d.* drawn from a distribution $P_{\mathcal{X},\mathcal{Y}}$ with marginal distribution $P_{\mathcal{X}}$, and we assume the following **moment condition** [41]:

$$\mathbb{E}_{P_{\mathcal{X},\mathcal{Y}}}[|Y - f^*(X)|^r] \le \frac{1}{2}r!\sigma^2 L^{r-2} \quad \text{for all } r \ge 2 \text{ and } P_{\mathcal{X}}\text{-almost all } x, \tag{5.3}$$

for some $\sigma, L > 0$. For example, if $Y \sim \mathcal{N}(f^*(X), \sigma^2)$, then this condition holds with $L = \sigma$. Denote $\boldsymbol{y} = [y_1, \cdots, y_n]$, $\boldsymbol{y}^* = [f^*(\tilde{x}_1), \cdots, f^*(\tilde{x}_n)]$, and $\boldsymbol{y}_\Phi^* = [f_\Phi(\tilde{x}_1), \cdots, f_\Phi(\tilde{x}_n)]$. Then, $\boldsymbol{y}_\Phi = \boldsymbol{y} - \boldsymbol{y}^* + \boldsymbol{y}_\Phi^*$ consists of the original labels shifted by $f_\Phi - f^*$. Part (i) bounds the estimation error assuming that the labels are $\boldsymbol{y}_\Phi$. Then, part (ii) bounds the gap between $\boldsymbol{y}_\Phi$ and $\boldsymbol{y}$.

Part (i) Estimation error bound for kernel ridge regression. Let the downstream predictor be $f(x) = \boldsymbol{w}^\top \Phi(x)$. Here we consider an unbiased linear predictor for simplicity. The proof when the predictor is biased is the same but a bit more verbose. Consider fitting $\boldsymbol{w}$ with ridge regression given by

$$\tilde{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left\{ \frac{1}{n} \big(\boldsymbol{y}_\Phi[i] - \boldsymbol{w}^\top \Phi(\tilde{x}_i) \big)^2 + \beta_n \|\boldsymbol{w}\|_2^2 \right\},$$

where the regularization parameter β_n can change with n. When n is larger, typically we would like β_n to be smaller, because we have more samples so can use a smaller regularization to achieve the same level of generalization. Let $\tilde{f}(x) = \tilde{\boldsymbol{w}}^\top \Phi(x)$. Note that $\tilde{f} \in \mathcal{H}_k$, and $\left\| \tilde{f} \right\|_{\mathcal{H}_k}^2 = \sum \tilde{w}_i^2 \|\phi_i\|_{\mathcal{H}_k}^2 = \|\tilde{\boldsymbol{w}}\|_2^2$. Thus, the above regression is equivalent to a kernel ridge regression on $\mathcal{H}_k$, where the regularization term is $\beta_n \left\| \tilde{f} \right\|_{\mathcal{H}_k}^2$.

Denote $\boldsymbol{\Phi} = [\Phi(\tilde{x}_1), \cdots, \Phi(\tilde{x}_n)] \in \mathbb{R}^{d \times n}$. A standard result in statistics shows that

$$\tilde{\boldsymbol{w}} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d \right)^{-1} \boldsymbol{\Phi} \boldsymbol{y}_\Phi. \tag{5.4}$$

$\left\| \tilde{f} - f_\Phi \right\|_{P_\mathcal{X}}^2$ can be bounded using the results in [41]. To use their results, apart from the moment condition Eqn. (5.3), three additional conditions are required:

- **Eigenvalue decay (EVD):** Let $\zeta_1 \geq \zeta_2 \geq \cdots$ be the eigenvalues of the RKHS. Then, $\zeta_i \leq c_1 i^{-\frac{1}{p}}$ for some constant $c_1 > 0$ and $p \in (0, 1]$.
- **Embedding condition (EMB):** There exists a constant $c_2 > 0$ such that for any $f \in \mathcal{H}_k$, $\|f\|_\infty \leq c_2 \|f\|_{\mathcal{H}_k}$. Here $\|f\|_\infty$ is the smallest B such that $|f(x)| \leq B$ for $P_\mathcal{X}$-almost all x.
- **Source condition (SRC):** There exists a constant $c_3 > 0$ such that $\|f\|_{\mathcal{H}_k} \leq c_3$.

Space $\mathcal{H}_k$ obviously satisfies (EVD) because its rank is d, so the condition holds for any $p \in (0, 1]$. For $f = f_\Phi$, it satisfies (EMB) with $c_2 = \kappa$, because for any $x_0 \in \mathcal{X}$, we have $f(x_0) = \langle f, k(x_0, \cdot) \rangle_{\mathcal{H}_k} \leq \|f\|_{\mathcal{H}_k} \|k(x_0, \cdot)\|_{\mathcal{H}_k}$, and $\|k(x_0, \cdot)\|_{\mathcal{H}_k}^2 = \|\sum s_i^2 \mu_i(x_0) \mu_i(\cdot)\|_{\mathcal{H}_k}^2 = \sum s_i^2 \mu_i(x_0)^2 \leq \kappa^2$ for $P_\mathcal{X}$-almost all x_0. Moreover, assuming that f^* is fixed, $f = f_\Phi$ also satisfies (SRC) with $c_3 = \|f^*\|_{P_\mathcal{X}} / (1 - \epsilon)$ because $\|f_\Phi\|_{\mathcal{H}_k} \leq \|f^*\|_{\mathcal{H}_k}$.

With all these conditions, invoking Theorem 3.1 in [41], we get the following.

Theorem 5.12. *For any $p \in (0, 1]$ and sufficiently large $n \geq 1$ and $\eta > 0$, suppose we choose $\beta_n = \Theta\left(n^{-\frac{1}{1+p}} \right)$. Then, there exists a constant $A > 0$ independent of n and η, such that with probability at least $1 - 4e^\eta$, we have $\left\| \tilde{f} - f_\Phi \right\|_{P_\mathcal{X}} \leq s_1 \left\| \tilde{f} - f_\Phi \right\|_{\mathcal{H}_k}$, and*

$$\left\| \tilde{f} - f_\Phi \right\|_{\mathcal{H}_k}^2 \leq 2\sqrt{\beta_n} \frac{\|f^*\|_{P_\mathcal{X}}^2}{(1-\epsilon)^2} + A\eta^2 \left[\frac{\kappa^2 \left(\sigma^2 + \frac{\|f^*\|_{P_\mathcal{X}}^2}{(1-\epsilon)^2} \right)}{n \beta_n^{\frac{1}{2}+p}} + \frac{\kappa^2 \max\left\{ L^2, 4\kappa^2 \frac{\|f^*\|_{P_\mathcal{X}}{}^2}{(1-\epsilon)^2} \right\}}{n^2 \sqrt{\beta_n}} \right].$$

We can see that as $n \to \infty$, there is $\left\| \tilde{f} - f_\Phi \right\|_{P_\mathcal{X}} \to 0$. when p is very small, we can think of this bound as $O(\frac{\kappa^2}{\sqrt{n}})$. In the ideal case, all singular values except the top-d are 0; then, κ^2 can be viewed as an approximation of $\sum_{i=1}^d s_i^2$. This is the bound that we used to derive the metric in Section 3.3.

Part (ii) Shift in the labels. The actual downstream predictor is given by

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)^{-1}\boldsymbol{\Phi}\boldsymbol{y}, \tag{5.5}$$

similar to Eqn. (5.4). Let $\hat{f}(x) = \hat{\boldsymbol{w}}^\top \Phi(x)$. Then, we have

$$\left\|\hat{f} - \tilde{f}\right\|_{\mathcal{H}_k}^2 = \left\|\left[\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)^{-1}\boldsymbol{\Phi}(\boldsymbol{y} - \boldsymbol{y}_\Phi)\right]^\top \Phi\right\|_{\mathcal{H}_k}^2 = \left\|\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)^{-1}\boldsymbol{\Phi}(\boldsymbol{y} - \boldsymbol{y}_\Phi)\right\|_2^2.$$

Let $\boldsymbol{Q} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d$. It suffices to bound $\|\boldsymbol{Q}^{-1}\boldsymbol{\Phi}\|_2^2$ and $\|\boldsymbol{y} - \boldsymbol{y}_\Phi\|_2^2$. To bound these, we need the following lemma, which can be proved using the Rademacher complexity.

Lemma 5.13 (Proof in Appendix D.3)**.** *Let* $\mathcal{F} = \left\{f_1 f_2 \;\middle|\; f_1, f_2 \in \mathcal{H}_k; \|f_1\|_{\mathcal{H}_k}, \|f_2\|_{\mathcal{H}_k} \le 1\right\}$. *Then,* $\mathfrak{R}_n(\mathcal{F}) \le \frac{\kappa^2}{\sqrt{n}}$. *If* $m \ge n$*, then for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\left\|\boldsymbol{\Phi}^\top \boldsymbol{u}\right\|_2^2 \ge \frac{n}{m}\lambda_d - \kappa^2\sqrt{n}\left(4 + 2\sqrt{2\log\frac{2}{\delta}}\right) \quad \textit{for any unit vector } \boldsymbol{u} \in \mathbb{R}^d.$$

$\|\boldsymbol{Q}^{-1}\boldsymbol{\Phi}\|_2^2$ is equal to the largest eigenvalue of $\boldsymbol{\Phi}^\top \boldsymbol{Q}^{-2}\boldsymbol{\Phi}$, which is also the largest eigenvalue of $\boldsymbol{Q}^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ by Sylvester's theorem. Let $\xi_1 \ge \cdots \ge \xi_d \ge 0$ and $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_d$ be the eigenvalues and orthonormal eigenvectors of $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \in \mathbb{R}^{d\times d}$. If $\xi_i = 0$, then $\boldsymbol{\alpha}_i$ is also an eigenvector of $\boldsymbol{Q}^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ with eigenvalue 0. If $\xi > 0$, then we have

$$\boldsymbol{Q}\boldsymbol{\alpha}_i = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)\boldsymbol{\alpha}_i = (\xi_i + n\beta_n)\boldsymbol{\alpha}_i,$$

which implies that $\boldsymbol{Q}^2\boldsymbol{\alpha}_i = (\xi_i + n\beta_n)^2\boldsymbol{\alpha}_i = \frac{(\xi_i + n\beta_n)^2}{\xi_i}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{\alpha}_i$. Thus, $\boldsymbol{\alpha}_i$ is an eigenvector of $\boldsymbol{Q}^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ with eigenvalue $\frac{\xi_i}{(\xi_i + n\beta_n)^2}$. We can therefore conclude that $\left(\frac{\xi_i}{(\xi_i + n\beta_n)^2}\right)$ are all the eigenvalues of $\boldsymbol{Q}^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$. Meanwhile, by Lemma 5.13, we have

$$\xi_d = \boldsymbol{\alpha}_d^\top \boldsymbol{\Phi}\boldsymbol{\Phi}^\top \boldsymbol{\alpha}_d = \left\|\boldsymbol{\Phi}^\top \boldsymbol{\alpha}_d\right\|_2^2 \ge \frac{n}{m}\lambda_d - \kappa^2\sqrt{n}\left(4 + 2\sqrt{2\log\frac{2}{\delta}}\right).$$

When $n \ge \frac{4\kappa^4 m^2}{\lambda_d^2}\left(4 + 2\sqrt{2\log\frac{2}{\delta}}\right)^2$, we have $\xi_d \ge \frac{n\lambda_d}{2m}$, which implies that $\|\boldsymbol{Q}^{-1}\boldsymbol{\Phi}\|_2^2 \le \frac{2m}{n\lambda_d}$. Remember that λ_1 has the same order as m, so we should view λ_d as also having roughly the same order as m.

Regarding $\|\boldsymbol{y} - \boldsymbol{y}_\Phi\|_2^2$, it is equal to $\sum_{i=1}^n F(\tilde{x}_i)$ where $F(x) = (f^*(x) - f_\Phi(x))^2$, and we have already bounded $\|f^* - f_\Phi\|_{\mathcal{H}_k}^2$ in Theorem 5.11. We can bound F using Lemma 5.13.

Corollary 5.14 (Proof in Appendix D.4)**.** *With probability at least* $1-\delta$*, we have*

$$\frac{\|\boldsymbol{y} - \boldsymbol{y}_\Phi\|_2^2}{n} \le \|f^* - f_\Phi\|_{P_\mathcal{X}}^2 + \|f^* - f_\Phi\|_{\mathcal{H}_k}^2 \frac{\kappa^2}{\sqrt{n}}\left(2 + \sqrt{2\log\frac{2}{\delta}}\right).$$

Next, note that $G(x) = \frac{(\hat{f}(x) - \tilde{f}(x))^2}{\|\hat{f} - \tilde{f}\|_{\mathcal{H}_k}^2} \in \mathcal{F}$ defined in Lemma 5.13, and

$$\sum_{i=1}^n \left(\hat{f}(\tilde{x}_i) - \tilde{f}(\tilde{x}_i)\right)^2 = \left\|\boldsymbol{\Phi}^\top\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)^{-1}\boldsymbol{\Phi}(\boldsymbol{y} - \boldsymbol{y}_\Phi)\right\|_2^2 \le \|\boldsymbol{y} - \boldsymbol{y}_\Phi\|_2^2$$

because $\left\|\boldsymbol{\Phi}^\top\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + n\beta_n \boldsymbol{I}_d\right)^{-1}\boldsymbol{\Phi}\right\|_2 \leq 1$. Thus, using Corollary 5.14, we get

$$\begin{aligned}
\left\|\hat{f} - \tilde{f}\right\|_{P_\mathcal{X}}^2 &\leq \frac{1}{n}\sum_{i=1}^n \left(\hat{f}(\tilde{x}_i) - \tilde{f}(\tilde{x}_i)\right)^2 + \left\|\hat{f} - \tilde{f}\right\|_{\mathcal{H}_k}^2 \frac{\kappa^2}{\sqrt{n}}\left(2 + \sqrt{2\log\frac{2}{\delta}}\right) \\
&\leq \frac{\|\boldsymbol{y} - \boldsymbol{y}_\Phi\|_2^2}{n}\left[1 + \frac{2\kappa^2}{\lambda_d\sqrt{n}}\left(2 + \sqrt{2\log\frac{2}{\delta}}\right)\right] \\
&\leq \frac{3}{2}\left(\|f^* - f_\Phi\|_{P_\mathcal{X}}^2 + \frac{\lambda_d}{4}\|f^* - f_\Phi\|_{\mathcal{H}_k}^2\right),
\end{aligned}$$

where the last step uses $n \geq \frac{4\kappa^4}{\lambda_d^2}\left(4 + 2\sqrt{2\log\frac{2}{\delta}}\right)^2$. Combining the above inequality with Theorem 5.12, we obtain a bound for $\left\|\hat{f} - f_\Phi\right\|_{P_\mathcal{X}}^2$.

Dependency on the context complexity. Our prediction error bound decreases with κ, meaning that the weaker the association of the context, the better the generalization will be. Let us revisit the experimental results in Figure 5.3 (right). On QNLI, the train-test gap decreases with the mask ratio α, which matches our bound. However, on SST-2, the train-test gap first decreases and then increases. The reason is that our bound assumes that $f^* \in \mathcal{F}_{\mathcal{H}_k}(P^+)$, but as the association becomes weaker, $\mathcal{F}_{\mathcal{H}_k}(P^+)$ becomes smaller and smaller. When the association becomes too weak, f^* might no longer belong to $\mathcal{F}_{\mathcal{H}_k}(P^+)$, and our bound will no longer hold. In this case, further weakening the association might make the train-test gap larger.

Pair/transformation access. The above analysis assumed that we have access to the centered dual kernel of the context. What if we have pair or transformation access instead? Recall that the dual kernel is $k_X^+(x, x') = \int \frac{P^+(a|x)P^+(a|x')}{P_\mathcal{A}(a)} da$. If we have transformation access, then we can assume that we have full access to $P^+(a|x)$ for any given x. The main problem appears in the denominator, because we can only estimate $P_\mathcal{A}(a)$ by $\frac{1}{m}\sum_{i=1}^m P^+(a|x_i)$. Although their difference can be shown to be small if $P^+(a|x)$ is assumed to be smooth in x, if $P_\mathcal{A}(a)$ itself is very small, then a tiny estimation error of $P_\mathcal{A}(a)$ will have a huge impact on k_X^+ because it is in the denominator. For this reason, we also need to assume that $P_\mathcal{A}(a)$ is bounded away from zero for all a, in order to get a reasonable generalization bound. Such a bound was proved in [167].

The case with pair access to the context is even more challenging, because in this case we have to estimate $P^+(a|x)$, so we need to consider the sample complexity of a as well. Deriving a bound for this case is posed as an open problem.

5.3 Spectrally Transformed Kernel Regression

The previous section showed that fitting the linear probe with ridge regression is equivalent to kernel ridge regression on a subspace of the induced RKHS, spanned by $\phi_1, \cdots, \phi_d$. In fact, this is a special case of spectrally transformed kernel regression (STKR) [168].

Definition 5.15. *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a p.s.d. kernel such that $k(x, x') = \sum_{i=1}^\infty \lambda_i \mu_i(x)\mu_i(x')$, where $\lambda_1 \geq \lambda_2 \geq \cdots$ and $\mu_1, \mu_2, \cdots$ are the eigenvalues and eigenfunctions of T_k. Then, a **spectrally transformed kernel (STK)** of k is defined as $k_s(x, x') = \sum_{i=1}^\infty s(\lambda_i)\mu_i(x)\mu_i(x')$, for some transformation function $s : [0, +\infty) \to [0, +\infty)$ such that $s(0) = 0$.*

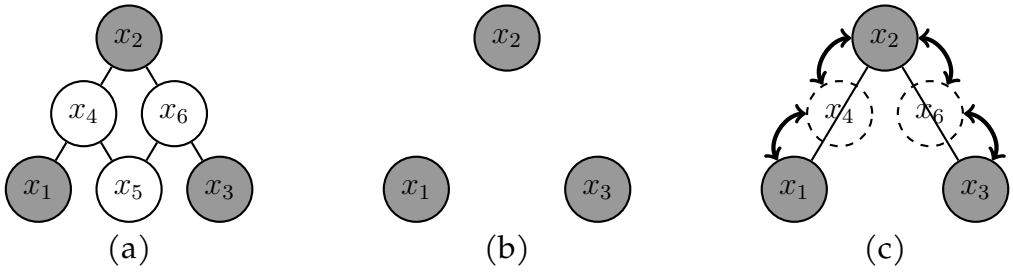


Figure 5.4: (a) A graph example where the kernel is the adjacency matrix. Shaded nodes are labeled and white nodes are unlabeled. (b) In KRR, the unlabeled nodes are useless and can be removed, so the graph becomes three isolated nodes. (c) With a two-step random walk k^2, x_1 and x_2 are connected, and x_2 and x_3 are connected.

An STK has the same eigenfunctions as the original kernel. In general one would like s to be a monotonically non-decreasing function, so that the spectral transformation does not change the order of the eigenfunctions,

Example 5.16. *If $\phi_1, \cdots, \phi_d$ are obtained by kernel PCA described in the previous section, and the downstream linear probe is fit using ridge regression, then this process is equivalent to doing kernel ridge regression on an STK k_s, where $s(\lambda_i) = \lambda_i \mathbb{I}[i \leq d]$ is called the truncation function.*

STKR is extremely useful in a semi-supervised learning setting where there are much more unlabeled samples than labeled samples. One might ask what is the point of using an STK or extracting the top-d eigenspace, instead of directly performing kernel ridge regression with k. The answer is that in many cases, regression with k would fail, but with k_s it would not. This can be demonstrated with a concrete example. Consider the graph in Figure 5.4 (a), where the shaded nodes are labeled but the white nodes are not, and the context is that connected nodes are similar. Then, k can be the adjacency matrix of this graph. Let $\mathcal{H}_k$ be the RKHS of k. Recall that given a labeled dataset $\{(\tilde{x}_i, y_i)\}_{i=1}^n$, kernel ridge regression (KRR) with k is given by

$$\hat{f} \in \operatorname*{arg\,min}_{f \in \mathcal{H}_k} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(f(\tilde{x}_i) - y_i \right)^2 + \beta_n \|f\|_{\mathcal{H}_k}^2 \right\},$$

for which there is a classical **Representer Theorem** [124, Theorem 4.2]:

Theorem 5.17. *All minimizers of KRR admit the form $\hat{f}^*(x) = \sum_{j=1}^n \alpha_i^* k(x, \tilde{x}_j)$, where*

$$\boldsymbol{\alpha}^* \in \operatorname*{arg\,inf}_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{j=1}^{n} \alpha_j k(\tilde{x}_i, \tilde{x}_j) - y_i \right]^2 + \beta_n \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\tilde{x}_i, \tilde{x}_j) \right\}.$$

What this theorem implies is that KRR only uses the labeled samples, and the large number of unlabeled samples are not used at all. This means that in Figure 5.4 (a), the white nodes are not used by KRR. When all the white nodes are removed, the graph becomes (b)—a graph with three isolated nodes, which is a useless context.

So what goes wrong here? The key is that the graph only says that x_1 and x_4 are similar, and x_4 and x_2 are similar. However, it does not say that x_1 and x_2 are similar, that is it cannot imply the transitivity of similarity on its own. Therefore, in the eyes of this graph, x_1, x_2, x_3 are three completely independent nodes, which is why the graph is useless in KRR. However, STKR assumes that similarity is transitive, and this allows us

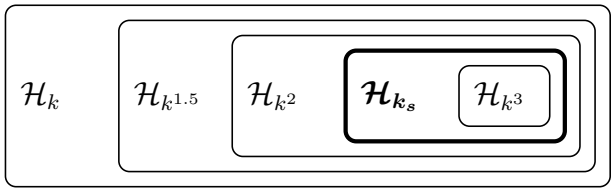


Figure 5.5: Illustration of the multiscale smoothness induced by diffusion, producing the chain $\mathcal{H}_k \supseteq \mathcal{H}_{k^{1.5}} \supseteq \mathcal{H}_{k^2} \supseteq \mathcal{H}_{k^3} \supseteq \cdots$. The RKHS of the spectrally transformed kernel k_s is $\mathcal{H}_{k_s}$ marked in bold, which is in this chain but not necessarily equal to any $\mathcal{H}_{k^p}$.

to start at any point x and do a random walk up to some number of steps, and the node we land on is still similar to x. For example, if we allow random walks up to two steps, then the graph becomes Figure 5.4 (c), which no longer consists of three isolated nodes.

To formalize this idea of random walk, define the following power spaces.

Definition 5.18. *A **power space** is an RKHS associated with k^p for any $p \geq 1$, where*

$$k^p(x, x') = \sum_{i=1}^{\infty} \lambda_i^p \mu_i(x)\mu_i(x');$$
$$\mathcal{H}_{k^p} = \left\{ f = \sum u_i \mu_i \;\middle|\; \sum \frac{u_i^2}{\lambda_i^p} < \infty \right\}, \; \left\langle \sum u_i\mu_i, \sum v_i\mu_i \right\rangle_{\mathcal{H}_{k^p}} = \sum \frac{u_i v_i}{\lambda_i^p}.$$

The proof of the following proposition is left as an exercise.

Proposition 5.19. *For any $p \geq 1$, there is $k^{p+1}(x, x') = \int k^p(x, z)k(z, x')dP_{\mathcal{X}}(z)$.*

This formula shows that when p is an integer, k^p can be viewed as a p-step random walk. When p is a real number, k^p is essentially a continuous random walk, which is called a **diffusion process**. This definition requires $p \geq 1$, because when $p < 1$, $\mathcal{H}_{k^p}$ is not necessarily an RKHS, due to the following classical result [124, p. 36].

Proposition 5.20. *Let $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$. For any $x \in \mathcal{X}$, define an evaluation functional $L_x : \mathcal{H} \to \mathbb{R}$ as $(L_x f) = f(x)$. Then, $\mathcal{H}$ is an RKHS if and only if for all $x \in \mathcal{X}$, L_x is a continuous operator, that is there exists a constant $M_x > 0$ such that $|f(x)| \leq M_x \|f\|_{\mathcal{H}}$ for all $f \in \mathcal{H}$. (Note: $\sup_x M_x = \infty$ is allowed.)*

In the last section, it has been shown that $M_x = \kappa$ when $p = 1$. Obviously, for any $p > 1$ and $f \in \mathcal{H}_{k^p}$, we have $\|f\|^2_{\mathcal{H}_{k^p}} \geq \lambda_1^{1-p}\|f\|^2_{\mathcal{H}_k}$. Thus, $\mathcal{H}_{k^p}$ is still an RKHS when $p > 1$. However, when $p < 1$, it is easy to construct an example where $\mathcal{H}_{k^p}$ is not an RKHS.

k^p is an example of an STK, and $\{k^p\}_{p\geq 1}$ forms a chain of function classes: $L^2(P_{\mathcal{X}}) \supset \mathcal{H}_{k^1} \supset \mathcal{H}_{k^{1.5}} \supset \mathcal{H}_{k^2} \supset \cdots$, as illustrated in Figure 5.5. We say that any function $f \in \mathcal{H}_{k_s}$ is **smooth** *w.r.t.* the kernel k_s, and the diffusion process induces **multiscale smoothness**. The kernel metric of k^p is $d_{k^p}(x, x') = \|k^p(x, \cdot) - k^p(x', \cdot)\|_{\mathcal{H}_{k^p}} = \sum \lambda_i^p(\mu_i(x) - \mu_i(x'))^2$, which is equivalent to the diffusion distance defined in [29].

The key result of this section is the generality of STK. Suppose the target function is smooth *w.r.t.* a certain measure of smoothness. Then, under some mild conditions, the class of all smooth functions under this measure must be the RKHS of some STK. This is a quite insightful result because it essentially says that if we know how to deal with $\mathcal{H}_{k_s}$, then we can deal with almost any kind of downstream task.

Specifically speaking, each k^p defines a measure of smoothness. We are interested in a certain measure of smoothness called the **target smoothness**, which satisfies the following condition: For any two functions f_1 and f_2, if all k^p say that f_1 is smoother than

f_2, then the target smoothness must also say that f_1 is smoother than f_2. This is called the condition of **preserving relative smoothness**. This is a weak condition, because usually f_1 is smoother at some scale p while f_2 is smoother at some other scale q. However, if f_1 is smoother at all scales, then it must also be smoother under the target smoothness. In addition, there are two more assumptions: (i) All smooth functions under the target smoothness form a Hilbert space $\mathcal{H}_t$; (ii) $\mathcal{H}_t \subseteq \mathcal{H}_k$.

To prove the result, we need to formally define the term "smoothness".

Definition 5.21. *For any Hilbert space $\mathcal{H} \subset L^2(P_\mathcal{X})$, the **smoothness** of $f \in \mathcal{H}$ w.r.t. $\mathcal{H}$ is defined as $r_\mathcal{H}(f) = \frac{\|\tilde{f}\|_{P_\mathcal{X}}^2}{\|\tilde{f}\|_{\mathcal{H}_{k_s}}^2}$.*

Smoothness is an alias of compatibility, because $r_\mathcal{H}(f) \geq (1-\epsilon)^2$ is equivalent to f being $(1-\epsilon)$-compatible with $\mathcal{H}$ as per Definition 5.6. The key result is as follows.

Theorem 5.22 (Proof in Appendix D.5)**.** *Suppose $\mathcal{H}_t \subseteq \mathcal{H}_k$ preserves relative smoothness: for any $f_1, f_2 \in L^2(P_\mathcal{X})$, if $r_{\mathcal{H}_{k^p}}(f_1) \geq r_{\mathcal{H}_{k^p}}(f_2)$ for all $p \geq 1$, then $r_{\mathcal{H}_t}(f_1) \geq r_{\mathcal{H}_t}(f_2)$. Then, $\mathcal{H}_t$ is an RKHS, whose reproducing kernel is an STK for a transformation function s such that:*

(i) s is monotonically non-decreasing;
(ii) $s(\lambda) \leq M\lambda$ for some constant $M > 0$;
(iii) s is continuous on $[0, +\infty)$;
(iv) s is C^∞ on $(0, +\infty)$.

STKR is the generalization of contexture learning from representation learning to semi-supervised learning. In semi-supervised learning, STKR is usually much more efficient than extracting the top-d eigenspace. The following is a popular example.

Example 5.23 (Inverse Laplacian)**.** *For $\eta \in (0, \lambda_1^{-1})$, define k_s such that ${k_s}^{-1}(x, x') = k^{-1}(x, x') - \eta k^0(x, x')$. k^{-1} and k^0 are STKs with $s(\lambda) = \lambda^{-1}$ and $s(\lambda) = \lambda^0$. Then, the reciprocal of s is given by $s^{-1}(\lambda) = \lambda^{-1} - \eta > 0$ for $\lambda \in (0, \lambda_1]$, which means that $s(\lambda) = \frac{\lambda}{1-\eta\lambda} = \sum_{p=1}^\infty \eta^{p-1}\lambda^p$, and $\|f\|_{\mathcal{H}_{k_s}}^2 = \|f\|_{\mathcal{H}_k}^2 - \eta\|f\|_{P_\mathcal{X}}^2$.*

The inverse Laplacian used to be very popular for semi-supervised learning, because it can be implemented very efficiently via a method called label propagation [170, 171].

5.4 Implementation and Generalization Analysis of STKR

Now we develop the algorithms for STKR for a wide variety of STKs, including the inverse Laplacian. After that, we derive generalization bounds for the algorithms. In particular, we consider polynomial STKs with $s(\lambda) = \sum_{p=1}^\infty \pi_p \lambda^p$, where $\pi_p \geq 0$ for all p.

Since $k_s = \sum_{p=1}^\infty \pi_p k^p$, it suffices to show how to estimate k^p. For example, when $p = 2$, we have $k^2(x, x') = \int k(x, z)k(x', z)dP_\mathcal{X}(z) \approx \frac{1}{m+n}\sum_{i=1}^{m+n} k(x, x_i)k(x', x_i)$, that is we estimate k^2 by Monte Carlo. Here, $x_{m+i} = \tilde{x}_i$. Recall that KRR does not use the m unlabeled samples at all, but here we can make use of the unlabeled samples when estimating k^p. Similarly, we can estimate k^p for all positive integer p as follows:

1. Compute Gram matrix $\boldsymbol{G}_k \in \mathbb{R}^{(m+n)\times(m+n)}$, where $\boldsymbol{G}_k[i, j] = k(x_i, x_j)$.
2. Let $\hat{k}^1 = k$. Define $\boldsymbol{v}_k(x) \in \mathbb{R}^{m+n}$ as $\boldsymbol{v}_k(x)[i] = k(x, x_i)$.
3. Compute $\hat{k}^p(x, x') = \frac{\boldsymbol{v}_k(x)^\top \boldsymbol{G}_k^{p-2} \boldsymbol{v}_k(x')}{(m+n)^{p-1}}$ iteratively for $p = 2, 3, \cdots$.

Then, an estimate of k_s is given by $\hat{k}_s = \sum_{p=1}^\infty \pi_p \hat{k}^p$. One difference is that previously we only use the m unlabeled samples to estimate the top-d eigenfunctions, but here we

Algorithm 3 STKR-Prop for simple s

Input: $\boldsymbol{G}_k$, $\boldsymbol{G}_{k,n}$, $\boldsymbol{F}$, s, β_n, $\boldsymbol{y}$, γ, ϵ

1: Initialize: $\hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{0} \in \mathbb{R}^n$
2: **while** True **do**
   # *Compute* $\boldsymbol{u} = (\boldsymbol{G}_{\hat{k}_s,n} + n\beta_n \boldsymbol{I}_n)\hat{\boldsymbol{\alpha}}$
3: $\quad \tilde{\boldsymbol{\alpha}} \leftarrow \frac{1}{m+n}\boldsymbol{F}\hat{\boldsymbol{\alpha}}$, $\boldsymbol{v} \leftarrow \boldsymbol{0} \in \mathbb{R}^{m+n}$
4: $\quad$ **for** $p = q, \cdots, 2$ **do** $\boldsymbol{v} \leftarrow \frac{\boldsymbol{G}_k \boldsymbol{v}}{m+n} + \pi_p \tilde{\boldsymbol{\alpha}}$
5: $\quad \boldsymbol{u} \leftarrow \boldsymbol{F}^\top \boldsymbol{v} + \pi_1 \boldsymbol{G}_{k,n}\hat{\boldsymbol{\alpha}} + n\beta_n \hat{\boldsymbol{\alpha}}$
6: $\quad$ **if** $\|\boldsymbol{u} - \boldsymbol{y}\|_2 < \epsilon \|\boldsymbol{y}\|_2$ **then return** $\hat{\boldsymbol{\alpha}}$
7: $\quad \hat{\boldsymbol{\alpha}} \leftarrow \hat{\boldsymbol{\alpha}} - \gamma(\boldsymbol{u} - \boldsymbol{y})$

Algorithm 4 STKR-Prop for simple s^{-1}

Input: $\boldsymbol{G}_k$, $s^{-1}(\lambda)$, β_n, $\boldsymbol{y}$, γ, ϵ

1: Initialize: $\boldsymbol{\theta} \leftarrow \boldsymbol{0} \in \mathbb{R}^{m+n}$, $\tilde{\boldsymbol{y}} \leftarrow [\boldsymbol{y}, \boldsymbol{0}_m]^\top$
2: **while** True **do**
   # *Compute* $\boldsymbol{u} = \boldsymbol{M}\boldsymbol{\theta}$
3: $\quad \boldsymbol{v} \leftarrow \boldsymbol{0} \in \mathbb{R}^{m+n}$
4: $\quad$ **for** $p = q-1, \cdots, 0$ **do** $\boldsymbol{v} \leftarrow \frac{\boldsymbol{G}_k \boldsymbol{v}}{m+n} + \xi_p \boldsymbol{\theta}$
5: $\quad \boldsymbol{u} \leftarrow \left[\left(\frac{\boldsymbol{G}_k^r}{(m+n)^{r-1}}\boldsymbol{\theta}\right)[1:n], \boldsymbol{0}_m\right]^\top + n\beta_n \boldsymbol{v}$
6: $\quad \boldsymbol{a} \leftarrow \boldsymbol{u} - \tilde{\boldsymbol{y}}$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \boldsymbol{a}$
7: $\quad$ **if** $\|\boldsymbol{a}\|_2 < \epsilon \|\boldsymbol{y}\|_2$ **then return** $\boldsymbol{\theta}$

use all $(m+n)$ samples to estimate $\hat{k}_s$. This difference is small in practice provided that $m \gg n$. Later we will see why we can use all $(m+n)$ samples in this situation.

We use $\boldsymbol{G}_{k,n} \in \mathbb{R}^{n \times n}$ to denote the Gram matrix on the n labeled samples, and define $\boldsymbol{v}_{k,n}(x) \in \mathbb{R}^n$ as $\boldsymbol{v}_{k,n}(x)[i] = k(x, \tilde{x}_i)$. Similarly, define $\boldsymbol{G}_{k_s,n}, \boldsymbol{v}_{k_s,n}, \boldsymbol{G}_{\hat{k}_s,n}$ and $\boldsymbol{v}_{\hat{k}_s,n}$. Let $\tilde{f}$ and $\hat{f}$ be the predictor obtained from KRR with k_s and $\hat{k}_s$, respectively. The following closed-form formulas can be derived from the Representer Theorem:

$$\begin{cases} \tilde{f}(x) = \boldsymbol{v}_{k_s,n}(x)^\top \tilde{\boldsymbol{\alpha}}, & \tilde{\boldsymbol{\alpha}} = (\boldsymbol{G}_{k_s,n} + n\beta_n \boldsymbol{I}_n)^{-1} \boldsymbol{y}; \\ \hat{f}(x) = \boldsymbol{v}_{\hat{k}_s,n}(x)^\top \hat{\boldsymbol{\alpha}}, & \hat{\boldsymbol{\alpha}} = \left(\boldsymbol{G}_{\hat{k}_s,n} + n\beta_n \boldsymbol{I}_n\right)^{-1} \boldsymbol{y}. \end{cases} \tag{5.6}$$

Here, $\boldsymbol{y} = [y_1, \cdots, y_n]$. To obtain $\hat{f}$, it suffices to solve $\boldsymbol{A}\hat{\boldsymbol{\alpha}} = \boldsymbol{y}$ for $\boldsymbol{A} = \boldsymbol{G}_{\hat{k}_s,n} + n\beta_n \boldsymbol{I}_n$. Let us consider two scenarios: (i) s is simple: For some q, $\pi_p = 0$ for all $p > q$; (ii) s^{-1} is simple: $s^{-1}(\lambda) = \sum_{p=0}^{q-1} \xi_p \lambda^{p-r}$, such as the inverse Laplacian. Here "simple" means that the polynomial contains only a few terms.

For scenario (i), directly computing $\boldsymbol{A}$ is slow because it involves lots of matrix-matrix multiplications. A faster alternative is iterative methods, such as Richardson iteration [119], which solves a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by iteratively computing $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \gamma(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(t)})$ for some $\gamma > 0$. Richardson iteration is guaranteed to converge to the solution when γ is chosen correctly. While computing $\boldsymbol{A}$ is slow, computing $\boldsymbol{A}\boldsymbol{x}^{(t)}$ is very efficient because it only involves matrix-vector multiplication. This method is called STKR propagation (STKR-Prop), because it is a generalization of label propagation (Label-Prop) for the inverse Laplacian. Define $\boldsymbol{F} \in \mathbb{R}^{(m+n) \times n}$ as $\boldsymbol{F}[i,j] = k(x_i, \tilde{x}_j)$. The algorithm is listed in Algorithm 3.

The next question is, given $\hat{\boldsymbol{\alpha}}$, how to efficiently compute $\hat{f}(x)$ for a test input x? We do not want to compute $\boldsymbol{v}_{\hat{k}_s,n}(x)$, which involves another set of matrix-vector multiplications. An efficient way is that we can store the $\boldsymbol{v}$ computed in line 4 of Algorithm 3 in the memory. Then, $\hat{f}(x) = \sum_{i=1}^{m+n} k(x_i, x)\boldsymbol{v}[i] + \pi_1 \sum_{j=1}^{n} k(\tilde{x}_j, x)\hat{\boldsymbol{\alpha}}[j]$ for any $x \in \mathcal{X}$, which only needs $O(m+n)$ time to compute.

For scenario (ii) where s could be complex but $s^{-1}(\lambda) = \sum_{p=0}^{q-1} \xi_p \lambda^{p-r}$ is simple, we can no longer estimate $\boldsymbol{G}_{\hat{k}_s,n}\boldsymbol{\alpha}$. However, we can do the following transformation: Let $\boldsymbol{Q} = \sum_{p=0}^{q-1} \xi_p \left(\frac{\boldsymbol{G}_k}{m+n}\right)^p$. Then, we have $\boldsymbol{G}_{\hat{k}_s}\boldsymbol{Q} = (m+n)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r$, where $\boldsymbol{G}_{\hat{k}_s} \in \mathbb{R}^{(m+n)\times(m+n)}$ is the Gram matrix of $\hat{k}_s$ on all $(m+n)$ samples. Therefore, we can efficiently compute $\boldsymbol{G}_{\hat{k}_s}\boldsymbol{Q}\boldsymbol{x}$ for any vector $\boldsymbol{x}$, and this motivates us to find a $\boldsymbol{\theta} \in \mathbb{R}^{m+n}$ such that $\boldsymbol{Q}\boldsymbol{\theta} = [\hat{\boldsymbol{\alpha}}, \boldsymbol{0}_m]^\top$. To solve for $\boldsymbol{\theta}$, we need $(m+n)$ linear equations. The last m elements of $\boldsymbol{Q}\boldsymbol{\theta}$ are all zero,

which provides us with m linear equations. Since $\boldsymbol{A}\hat{\boldsymbol{\alpha}} = \boldsymbol{y}$, the first n elements of $\boldsymbol{AQ\theta}$ must be $\boldsymbol{y}$, which provides us with another n linear equations. Overall, we can find $\boldsymbol{\theta}$ by solving $\boldsymbol{M}\theta = \tilde{\boldsymbol{y}}$, where

$$\boldsymbol{M} = (m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}, \quad \tilde{\boldsymbol{y}} = [\boldsymbol{y}, \boldsymbol{0}_m]^\top.$$

Here, $\tilde{\boldsymbol{I}}_n = \text{diag}\{1, \cdots, 1, 0, \cdots, 0\}$, with n ones and m zeros. Once again, we can solve for $\boldsymbol{\theta}$ using Richardson iteration. The algorithm is listed in Algorithm 4. After running this algorithm, we can store $\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r-1}\boldsymbol{\theta}$ in the memory. Then, for any test input x, we can compute $\hat{f}(x)$ with $\hat{f}(x) = \boldsymbol{v}_k(x)^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r-1}\boldsymbol{\theta}$ in $O(m+n)$ time.

Next, let us study the time complexity of these two algorithms. Assume that computing $k(x, x')$ for any $x, x' \in \mathcal{X}$ takes $O(1)$ time. To start with, let us review a classical result about Richardson iteration.

Lemma 5.24. *Consider solving $\boldsymbol{Ax} = \boldsymbol{b}$ with Richardson iteration, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is positive definite, and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$. Let $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and smallest eigenvalue of $\boldsymbol{A}$, and let $\tau = \frac{\lambda_{\max}}{\lambda_{\min}}$ be the condition number of $\boldsymbol{A}$. Then, by choosing $\gamma = \frac{2}{\lambda_{\max}+\lambda_{\min}}$ and setting the stop criterion as $\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_2 < \epsilon\|\boldsymbol{b}\|_2$, the iteration stops in $O\left(\tau\log\frac{1}{\epsilon}\right)$ steps.*

Proof Let $\boldsymbol{x}^*$ be the solution such that $\boldsymbol{Ax}^* = \boldsymbol{b}$. By $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \gamma(\boldsymbol{b} - \boldsymbol{Ax}^{(t)})$, we have $\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^* = (\boldsymbol{I}_n - \gamma\boldsymbol{A})(\boldsymbol{x}^{(t)} - \boldsymbol{x}^*)$. If we start with $\boldsymbol{x}^{(0)} = \boldsymbol{0}$, then this implies that $\boldsymbol{x}^* - \boldsymbol{x}^{(t)} = (\boldsymbol{I}_n - \gamma\boldsymbol{A})^t\boldsymbol{x}^*$. When $\gamma = \frac{2}{\lambda_{\max}+\lambda_{\min}}$, we have

$$\left\|\boldsymbol{x}^* - \boldsymbol{x}^{(t)}\right\|_2 \le \|\boldsymbol{I}_n - \gamma\boldsymbol{A}\|_2^t\|\boldsymbol{x}^*\|_2 = \left(1 - \frac{2\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\right)^t\|\boldsymbol{x}^*\|_2 \le \exp\left(-\frac{2\lambda_{\min}t}{\lambda_{\max}+\lambda_{\min}}\right)\|\boldsymbol{x}^*\|_2.$$

This implies that $\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_2 \le 2\exp\left(-\frac{2\lambda_{\min}t}{\lambda_{\max}+\lambda_{\min}}\right)\|\boldsymbol{x}^*\|_2 \le 2\exp(-\frac{t}{\tau})\|\boldsymbol{x}^*\|_2$. Thus, the iteration stops when $t = O\left(\tau\log\frac{1}{\epsilon}\right)$. Moreover, when the iteration stops, we have $\left\|\boldsymbol{Ax}^{(t)} - b\right\|_2 = \gamma^{-1}\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_2 < \gamma^{-1}\epsilon\|\boldsymbol{b}\|_2$. ■

Let λ_1 be the largest eigenvalue of T_k. When m and n are sufficiently large, the largest eigenvalue of $\boldsymbol{G}_{\hat{k}_s,n}$ is close to $ns(\lambda_1)$. Meanwhile, the smallest eigenvalue of $\boldsymbol{A} = \boldsymbol{G}_{\hat{k}_s,n} + n\beta_n\boldsymbol{I}_n$ is at least $n\beta_n$. Therefore, τ can be upper bounded by $O(\beta_n^{-1}s(\lambda_1))$. Moreover, each iteration in Algorithm 3 has a time complexity of $O((m+n)^2q)$. Thus, the overall time complexity of Algorithm 3 is $O((m+n)^2q\beta_n^{-1}s(\lambda_1)\log\frac{1}{\epsilon})$.

For Algorithm 4, the analysis is similar but much more complex. We have the following result regarding its time complexity.

Theorem 5.25 (Proof in Appendix D.6)**.** *Let $\rho(\lambda) = \frac{\lambda^r}{s(\lambda)} = \sum_{p=0}^{q-1}\xi_p\lambda_p$, where $\rho(0) = \xi_0 > 0$. Then, $\rho(\lambda)$ is a continuous function on $[0, +\infty)$. Denote its maximum and minimum on $[0, \lambda_1]$ by $\rho_{\max}$ and $\rho_{\min}$. Then, with $\gamma = (n\lambda_1^r)^{-1}$, Algorithm 4 has a total time complexity of $O\left((m+n)^2\frac{\max\{q,r\}\lambda_1^r\beta_n^{-1}}{\rho_{\min}}\log\left[\max\left\{\frac{1}{\epsilon}, \frac{\lambda_1^r\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n^2\rho_{\min}^2\|\hat{\boldsymbol{\alpha}}_*\|_2}\right\}\right]\right)$, where $\hat{\boldsymbol{\alpha}}_*$ is the exact solution.*

We have seen how to implement STKR when s is a polynomial. Now let us derive the generalization bound for the general STKR. Define the following function class:

$$\mathcal{F}_\epsilon(\mathcal{H}_{k_s}) = \left\{f \in \mathcal{H}_{k_s} \;\middle|\; \|f\|_{\mathcal{H}_{k_s}}^2 \le \epsilon\|f\|_{P\mathcal{X}}^2\right\}.$$

This class is similar to $\mathcal{F}_{\mathcal{H}_k}(P^+)$, and it contains functions with smoothness at least ϵ^{-1} *w.r.t.* $\mathcal{H}_{k_s}$. Our result consists of an approximation error bound and an estimation error bound. Recall the definition of $\tilde{f}$ and $\hat{f}$ in Eqn. (5.6). The approximation error is the gap between $\tilde{f}$ and f^*, and the estimation error is the gap between $\hat{f}$ and $\tilde{f}$. The approximation error can be bounded by the result in [41].

Theorem 5.26 (Proof in Appendix D.7)**.** *Let $\lambda_1 \geq \lambda_2 \geq \cdots$ be the eigenvalues of T_k. Let M be given by Theorem 5.22. Suppose the moment condition Eqn.* (5.3) *holds, and the eigenvalues decay by order p^{-1} for some $p \in (0, 1]$, that is $s(\lambda_i) = O(i^{-\frac{1}{p}})$ for all i. Choose $\beta_n = \Theta(n^{-\frac{1}{1+p}})$. Then, there exists a constant $c_0 > 0$ independent of $n \geq 1$ and $\tau \geq \kappa^{-1}M^{-\frac{1}{2}}$ (recall that $\kappa^2 = \|k\|_\infty$), such that*

$$\left\|\tilde{f} - f^*\right\|_{P_\mathcal{X}}^2 \leq c_0\tau^2\kappa^2 M\left[\left(\epsilon\|f^*\|_{P_\mathcal{X}}^2 + \sigma^2\right)n^{-\frac{1}{1+p}} + \max\left\{L^2, \kappa^2 M\epsilon\|f^*\|_{P_\mathcal{X}}^2\right\}n^{-\frac{1+2p}{1+p}}\right]$$

holds for all $f^* \in \mathcal{F}_\epsilon(\mathcal{H}_{k_s})$ with probability at least $1 - 4e^{-\tau}$, given that n is sufficiently large.

Remark 5.27. [41] *showed that the learning rate $O(n^{-\frac{1}{1+p}})$ is minimax optimal, which means that one can construct an example where the learning rate is at most $O(n^{-\frac{1}{1+p}})$. When $\kappa^2 = \|k\|_\infty < \infty$, one can always choose $p = 1$ because $i \cdot s(\lambda_i) \leq \sum_{j=1}^i s(\lambda_j) \leq M\sum\lambda_j \leq M\kappa^2$.*

Regarding the estimation error, we prove the following result.

Theorem 5.28 (Proof in Appendix D.8)**.** *Let $\hat{\lambda}_1$ be the largest eigenvalue of $\frac{\boldsymbol{G}_k}{m+n}$, and denote $\lambda_{\max} = \max\left\{\lambda_1, \hat{\lambda}_1\right\}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ we have*

$$\left\|\hat{f} - \tilde{f}\right\|_{P_\mathcal{X}}^2 \leq 8s(\lambda_{\max})\,\nabla_\lambda\left(\frac{s(\lambda)}{\lambda}\right)\Big|_{\lambda=\lambda_{\max}} \frac{\beta_n^{-2}\kappa^4}{\sqrt{m+n}}\left(2 + \sqrt{2\log\frac{1}{\delta}}\right)\frac{\|\boldsymbol{y}\|_2^2}{n}.$$

This result requires $\hat{\lambda}_1$, for which there is the following result.

Lemma 5.29 ([125], Theorem 2)**.** *For any $\delta \in (0, 1)$, with probability at least $1 - \delta$ we have*

$$\hat{\lambda}_1 \leq \lambda_1 + \frac{\kappa^2}{\sqrt{m+n}}\left[2\sqrt{2} + \sqrt{19\log\frac{2(m+n+1)}{\delta}}\right].$$

The key to prove Theorem 5.28 is using the complexity of $\mathcal{H}_{k_s}$ to prove a uniform deviation bound for $\left|\hat{k}_s(x, x_i) - k_s(x, x_i)\right|$ for all x and i. Thus, we can see why we only used the m unlabeled samples to extract the top-d eigenspace, but can use all $(m + n)$ samples in STKR. The reason is that for uniform deviation bounds, the function class must be independent of the training samples. For the top-d eigenfunctions, the downstream function class is the span of $\phi_1, \cdots, \phi_d$, and the proof of Theorem 5.12 was based on this function class. This function class depends on the m unlabeled samples. If the n labeled samples were also used, then the uniform deviation bound would not hold. On the other hand, in Theorem 5.28, the function class if $\mathcal{H}_{k_s}$, which is independent of all $(m + n)$ samples. Hence, we can use all the $(m + n)$ samples in STKR.

5.5 Empirical Study of Contexture Learning and STKR

This section conducts some experiments on STKR with the following goals:

	Classes	Nodes	Edges	Train	Validation
Cora	7	2,708	10,556	5.17	18.46
CiteSeer	6	3,327	9,104	3.61	15.03
PubMed	3	19,717	88,648	0.3	2.54
Amazon - Computers	10	13,752	491,722	1.45	1.45
Amazon - Photos	8	7,650	238,162	2.09	2.09
Coauthor - CS	15	18,333	163,788	1.64	1.64
Coauthor - Physics	5	34,493	495,924	0.29	0.29
DBLP	4	17,716	105,734	0.45	0.45
CoraFull	70	19,793	126,842	7.07	7.07

Table 5.1: Number of classes, nodes, edges, and fractions (%) of train/validation sets.

(i) Verify that STKR-Prop (Algorithms 3 and 4) works with general polynomial s, such as the inverse Laplacian, and compare them to label propagation (Label-Prop).

(ii) Explore possible reasons why the inverse Laplacian works so well in practice, by examining the effect of p on the performance when using STKR with $s(\lambda) = \lambda^p$.

(iii) Compare extracting the top-d eigenspace with STKR and Label-Prop.

The experiments here focus on graph node classification tasks. The datasets used are listed in Table 5.1, and they all come from the *PyTorch Geometric* library [40]. Each dataset is split into four sets: train, validation (val), test and other. Among them, train and val contain labeled samples, while test and other contain unlabeled samples. Both the transductive and the inductive settings are tested.

- In the **transductive setting**, the samples in all four sets are available at train time. The learner hides the labels of val samples for validation. Thus, n is the size of the train set, while m is the size of all other three sets combined.
- In the **inductive setting**, samples in the test set are invisible at train time. The learner hides the entire val set (samples and labels) for validation. Thus, n is the size of the train set, while m is the size of the other set.

Label propagation (Label-Prop) only works for the transductive setting, and is implemented as follows: Let $\boldsymbol{W}$ be the adjacency matrix of the graph, such that $\boldsymbol{W}[i,j] = 1$ if x_i and x_j are connected, and 0 otherwise. Let $\boldsymbol{D}$ be a diagonal matrix such that $\boldsymbol{D}[i,i] = \sum_j \boldsymbol{W}[i,j]$. Let $\boldsymbol{S} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$. Following [170], Label-Prop solves

$$(\boldsymbol{I}_{m+n} - \eta\boldsymbol{S})\hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}}, \quad \text{where } \boldsymbol{y} = [\boldsymbol{y}, \boldsymbol{0}_m].$$

Then, $\hat{\boldsymbol{y}}$ contains the predicted labels for all $(m+n)$ samples. On the other hand, STKR works for both transductive and inductive settings. Its base kernel k is defined as

$$k(x,x') = (m+n)\frac{W(x,x')}{\sqrt{D(x)D(x')}},$$

where $W(x_i,x_j) = \boldsymbol{W}[i,j]$. For the transductive setting, $D(x_i) = \boldsymbol{D}[i,i]$; for the inductive setting, $D(x_i) = \sum_{j \notin \text{ test nodes}} W(x_i,x_j)$, that is the sum is taken over only visible nodes.

We use 1% of the samples as test samples. Each experiment is run with 10 random seeds for splitting the dataset. The results are reported in Table 5.2, from which we make the following observations:

- STKR works pretty well with general polynomial $s(\lambda)$ in the inductive setting. In the transductive setting, the performance of STKR-Prop with the inverse Laplacian

	CS	CiteSeer	Computers	Cora	CoraFull	DBLP	Photo	Physics	PubMed
LP (t)	$79.07_{2.19}$	$\mathbf{52.73_{7.72}}$	$77.30_{3.05}$	$\underline{73.33}_{6.00}$	$\mathbf{54.47_{3.24}}$	$\mathbf{66.44_{3.78}}$	$83.95_{5.78}$	$84.33_{4.86}$	$\mathbf{72.28_{5.55}}$
Lap (t)	$78.96_{2.53}$	$\underline{52.12}_{7.67}$	$77.81_{3.94}$	$\mathbf{77.04_{5.74}}$	$\underline{53.81}_{2.34}$	$\underline{65.42}_{5.02}$	$84.08_{6.52}$	$84.22_{4.86}$	$71.93_{4.86}$
Poly (t)	$\mathbf{79.13_{2.29}}$	$48.79_{8.51}$	$76.72_{4.12}$	$71.48_{5.80}$	$53.25_{3.54}$	$64.52_{4.20}$	$79.21_{7.20}$	$\underline{84.45}_{4.89}$	$\underline{72.18}_{4.66}$
Topd (t)	$78.80_{3.22}$	$46.06_{1.08}$	$\mathbf{80.80_{3.06}}$	$69.26_{7.82}$	$50.36_{2.85}$	$64.86_{4.60}$	$84.61_{6.30}$	$83.20_{2.25}$	$65.38_{5.66}$
Lap (i)	$78.42_{2.81}$	$46.06_{6.97}$	$77.15_{2.64}$	$67.78_{7.62}$	$53.30_{3.24}$	$65.20_{4.92}$	$\underline{84.87}_{5.66}$	$83.11_{5.09}$	$70.36_{4.80}$
Poly (i)	$79.02_{2.42}$	$44.55_{9.15}$	$71.97_{4.13}$	$65.19_{9.11}$	$51.98_{3.88}$	$64.52_{4.05}$	$78.42_{7.80}$	$\mathbf{84.68_{4.83}}$	$70.76_{4.28}$
Topd (i)	$\underline{79.13}_{3.35}$	$41.52_{6.71}$	$\underline{80.80}_{3.28}$	$63.70_{6.00}$	$47.41_{3.39}$	$63.16_{3.41}$	$\mathbf{85.53_{5.68}}$	$82.44_{3.88}$	$64.31_{4.95}$
KRR (i)	$13.11_{2.29}$	$13.64_{5.93}$	$26.35_{4.34}$	$28.52_{8.56}$	$19.80_{2.22}$	$44.80_{3.86}$	$33.95_{7.07}$	$19.74_{1.46}$	$20.76_{2.06}$

Table 5.2: The test accuracy (%) of Label-Prop (LP), STKR-Prop with inverse Laplacian (Lap), with polynomial $s(\lambda) = \lambda^8$ (Poly), with kernel PCA (Topd), and with $s(\lambda) = \lambda$ (KRR). (t) and (i) indicate transductive and inductive. Standard deviations are given across ten random seeds. Best/second-best results are in bold/underlined.

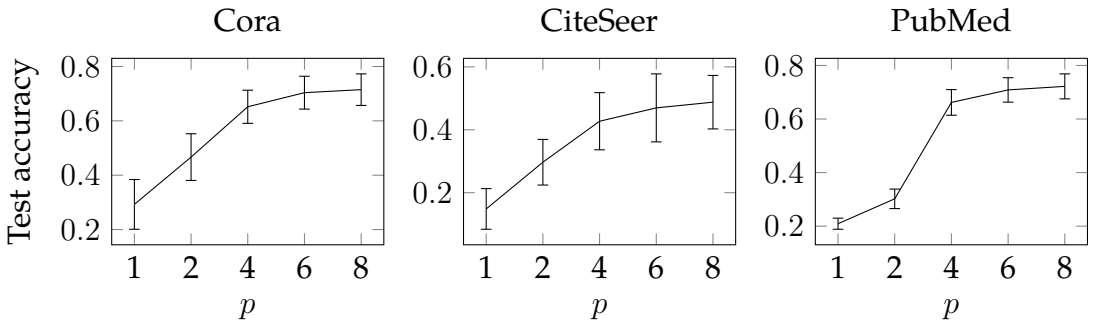


Figure 5.6: Performance of STKR-Prop with $s(\lambda) = \lambda^p$ and 8 iterations. The best β_n with the highest test accuracy is selected. Each experiment is run with 10 random seeds.

is similar to Label-Prop, and STKR-Prop with polynomial s is slightly worse. The performance under the inductive setting is slightly worse than that under the transductive setting, which is reasonable since there is less information at train time for the inductive setting. It is also worth noting that the running time of STKR-Prop is similar to that of Label-Prop with the same number of iterations.

- STKR with $s(\lambda) = \lambda^p$ for $p > 1$ is much better than KRR (where $p = 1$). Moreover, we test STKR with $s(\lambda) = \lambda^p$ for more choices of $p \in \{1, 2, 4, 6, 8\}$, and report the results on three data sets in Figure 5.6. It is clear that a larger p leads to higher performance. This suggests one possible reason why inverse Laplacian works so well empirically: It contains k^p for $p = 1, 2, \cdots$, so it can use multi-step similarity information up to infinitely many steps.
- Extracting the top-d eigenspace with kernel PCA can achieve pretty high performance. Specifically, on 3 of the 9 datasets we use, such as `Computers`, STKR with top-d truncation is better than Label-Prop and STKR with inverse Laplacian. This shows that STKR with inverse Laplacian and STKR with top-d truncation plus kernel PCA are two parallel methods—neither is superior to the other one.

In conclusion, this chapter first derived generalization bounds for contexture learning, that is extracting the top-d eigenspace with finite samples. The key quantity is the context complexity, which reflects the smoothness of the eigenfunctions. Next, this chapter introduced a more general formulation called STKR, and presented its implementation as well as generalization bounds. The key takeaway from this chapter is how the context affects the sample complexity of representation learning.

Chapter 6

Generalization Under Distribution Shift

The contexture theory characterizes the mechanism of representation learning, thereby advancing the science of foundation models. However, one main assumption of the contexture theory is that the data distribution $P_{\mathcal{X}}$ is fixed. This is a very strong assumption that is hardly true in practice. In reality, the distributions of the pretrain data and the downstream data are always different. Whether a model trained on one distribution can still achieve good performance on another distribution is called the problem of **out-of-distribution (OOD) generalization**. The main takeaway from this chapter is that OOD generalization is extremely hard, both in theory and in practice.

There is a rich body of work on OOD generalization in machine learning, statistics, applied probability and optimization [12, 68, 115, 128]. In deep learning, there are two types of research on OOD generalization. The first type studies how to transfer a model trained on one domain to another domain, also known as transfer learning [111] or domain adaptation [151]. The second type studies how to train robust models against distribution shift, that is preserving the model's performance on the new distribution. Such research is valuable in safety-critical applications or domains where the data is constantly changing, such as finance. Both types of research are relevant to foundation models. For example, how to apply an LLM trained on Wikipedia to a dataset of Python codes is the first type of research. How to make sure that an LLM always generates proper responses to prompts it has never seen is the second type of research. The first type is more related to the fine-tuning stage rather than the representation learning stage of foundation model training. Therefore, this chapter focuses on the second type.

In representation learning, a foundation model is pretrained on one distribution P, and then applied to another distribution Q. This chapter studies an easier problem, which is the standard problem in the literature: A predictor is trained on P and then evaluated on Q. We assume that the distribution shift only contains **covariate shift**, where the ground truth target function is always fixed, and only the distribution of X changes. In practice, there are two types of covariate shift [88].

- In **domain generalization**, the support of Q might contain samples that are not in the support of P. This is the typical scenario in domain adaptation.
- In **subpopulation shift**, the support of Q is a subset of the support of P. Alternatively we can write $Q \ll P$, that is Q is absolutely continuous to P. Mathematically this means that for any set A, $P(A) = 0$ implies $Q(A) = 0$.

In the context of foundation models, it suffices to study subpopulation shift. There are two main reasons. First, the pretraining set is very large, so it is very unlikely to have test samples that do not appear in the pretraining set. Second, even in domain adaptation, the common practice is to fine-tune the foundation model on the new distribution

before using it. It is rarely the case that a model is used on a completely new distribution without any fine-tuning. For these reasons, this chapter focuses on subpopulation shift exclusively.

6.1 Reweighting and DRO

In subpopulation shift, P and Q are different distributions on the same set of samples. Thus, one can view Q as assigning different weights to the samples than P. For example, given a training set $\{(x_i, y_i)\}_{i=1}^n$, P is usually defined as the uniform distribution over the n samples, while Q gives different weights to these samples. Since the model is evaluated on Q, we care more about those samples where Q place larger weights than P, because these samples have greater impact on the evaluation. We call them **upweighted samples**.

Empirical risk minimization (ERM) is the standard training algorithm in deep learning. It assumes that the training samples are *i.i.d.* sampled from the data distribution, and minimizes the average model risk over the training samples. Let $\ell(\hat{y}, y)$ be the loss function. Then, ERM minimizes the following empirical risk:

$$\hat{\mathcal{R}}_{\text{ERM}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i).$$

However, if P and Q give different weights to the samples, then ERM could achieve low performance on Q if its risk on the upweighted samples on higher than average [14, 67, 137]. One common example of this is **class imbalance**, which was briefly discussed in Section 2.1. In a classification task, if some classes are significantly smaller than the other classes, then a model trained via ERM will typically have high risk on these small classes. However, the model is required to have good performance on every class, meaning that Q places larger weights on these small classes than P. As a result, the ERM model is poor in the class imbalance situation.

Given the above discussion, the most straightforward way to tackle the subpopulation shift is reweighting, also known as **importance weighting** [128]. The idea is to assign a different weight P' to the samples in the training loss, such that $P' = Q$. The importance weighting (IW) empirical risk is

$$\hat{\mathcal{R}}_{\text{IW}}(f) = \frac{1}{n}\sum_{i=1}^{n} \frac{Q(x_i)}{P(x_i)} \ell(f(x_i), y_i).$$

Note that $P(x)$ and $Q(x)$ are the weights of x, and they are not necessarily the density functions. For example, in class imbalance, $P(x)$ is proportional to the size of the class of x. If the goal is to have good performance on every class, then $Q(x)$ is the same for all x. Consequently, $\mathcal{R}_{\text{IW}}$ divides the loss on each sample by its class size, thereby assigning larger weights to samples in smaller classes. Note that this is different from Eqn. (2.2), which divides the sample loss by the square root of its class size.

Importance weighting assumes that we know Q at train time. What if Q is unknown? This is the more common case in training foundation models. The pretraining dataset is a very large and comprehensive dataset, while the downstream task usually focuses on only one domain, that is a small part of the pretraining data. However, since we may not know the downstream task at pretrain time, we do not know which data it will focus on.

Distributionally robust optimization (DRO) is the most popular approach when Q is unknown. It aims to minimize the model risk on the worst Q whose distance to P

is bounded by some $\rho > 0$. The distance from one distribution to another is also known as a divergence function, and is denoted by $D(Q \parallel P)$. Note that $D(Q \parallel P)$ might not be symmetric, that is we could have $D(Q \parallel P) \neq D(P \parallel Q)$. The DRO risk is defined as

$$\mathcal{R}_{D,\rho}(f; P) = \sup_{Q} \left\{ \mathbb{E}_{(X,Y)\sim Q}[\ell(f(X), Y)] \;\middle|\; D(Q \parallel P) \leq \rho \right\}. \tag{6.1}$$

The constraint $D(Q \parallel P) \leq \rho$ comes from our prior knowledge about Q. The following are two examples of DRO.

Example 6.1. ***Conditional value at risk* (*CVaR*)** *[38, Example 3] aims to maximize the model's performance on the worst α fraction of the data, for some fixed $\alpha \in (0, 1)$. For example, CVaR is widely used in finance, where the model needs to perform well when the market is at its lowest point. In this scenario, we have $Q(x) \leq \alpha^{-1} P(x)$ for all x, that is the sample weight under P is at least α times the weight under Q. Thus, $D(Q \parallel P) = \sup \frac{Q(x)}{P(x)}$, and $\rho = \alpha^{-1}$.*

Example 6.2. *In **group DRO** (**GDRO**) [121], the data is divided into a number of groups, and the model is required to perform well on the worst group. This is a common scenario in the field of machine learning fairness. For example, a credit approval model is required to be fair across all races. Then, the groups are defined by the races. In this case, $P(x)$ and $Q(x)$ are the weights on the group of x, $D(Q \parallel P) = \sup \frac{Q(x)}{P(x)}$, and ρ is the number of groups.*

Both reweighting and DRO are quite heuristic, but do they actually work in practice? The results are mixed. For example, balancing the classes can usually give the model higher performance on small classes, but at the cost of the average performance. Meanwhile, [121] showed that reweighting and DRO methods can overfit very easily, and they usually require much stronger regularization than ERM, or early stopping. Furthermore, [51] conducted a large-scale empirical study, and showed the surprising negative result that reweighting are not better than ERM on most real datasets. The authors argued that reweighting were reported better in prior work only because the ERM baseline had not been sufficiently tuned.

The rest of this chapter will show two reasons why reweighting and DRO might fail. The first reason is related to the training dynamics of these methods. The second reason is related to their sensitivity to the outliers in the dataset. Possible solutions to each failure mode will be discussed. These analyses are based on my work [164, 165, 166].

6.2 Generalized Reweighting (GRW) Versus ERM

Let the input space be $\mathcal{X} \subseteq \mathbb{R}^{d_{\mathcal{X}}}$, and we assume that all $x \in \mathcal{X}$ satisfies $\|x\|_2 \leq 1$. Consider learning a target function $f^* : \mathcal{X} \to \mathbb{R}$ using a training set $\{(x_i, y_i)\}_{i=1}^n$. Denote $\boldsymbol{X} = (x_1, \cdots, x_n) \in \mathbb{R}^{d_{\mathcal{X}} \times n}$, and $\boldsymbol{Y} = (y_1, \cdots, y_n) \in \mathbb{R}^n$. For any function $g : \mathcal{X} \mapsto \mathbb{R}^m$, we overload notation and denote $g(\boldsymbol{X}) = (g(x_1), \cdots, g(x_n)) \in \mathbb{R}^{m \times n}$.

The difference between reweighting and DRO is that Q is fixed in reweighting are fixed, while it is not in DRO. Here we present a general formulation called **generalized reweighting (GRW)**. At training step t, GRW minimizes the following weighted empirical risk:

$$\hat{\mathcal{R}}_{\boldsymbol{q}^{(t)}}(f) = \sum_{i=1}^{n} q_i^{(t)} \ell(f(x_i), y_i), \tag{6.2}$$

where $\boldsymbol{q}^{(t)} = (q_1^{(t)}, \cdots, q_n^{(t)})$ is the sample weight vector, such that $q_1^{(t)} + \cdots + q_n^{(t)} = 1$. If $\boldsymbol{q}^{(t)}$ does not change with t, we call it static GRW; if $\boldsymbol{q}^{(t)}$ can change with t, we call it dynamic GRW. Note that ERM is a special case of static GRW.

Importance weighting is obviously an example of static GRW. Group DRO is an example of dynamic GRW, and here is how it is usually implemented. Let there be K groups. Denote the empirical risk over group k by $\hat{\mathcal{R}}_k(f)$, and the model at time t by $f^{(t)}$. For all $k \in [K]$, group DRO iteratively sets $q_i^{(t)} = g_k^{(t)}/n_k$ for all (x_i, y_i) in group k, where $g_k^{(t)}$ is the group weight that is updated by

$$g_k^{(t)} \propto g_k^{(t-1)} \exp\left[\nu \hat{\mathcal{R}}_k\big(f^{(t-1)}\big)\right]$$

for some $\nu > 0$. The group weights are normalized so that $q_1^{(t)} + \cdots + q_n^{(t)} = 1$. [121, Proposition 2] showed that for convex settings, the Group DRO risk of iterates converges to the global minimum with the rate $O(t^{-1/2})$ if ν is sufficiently small.

The key result of this section is that GRW cannot improve over ERM, because the models they produce are too similar. It relies on two key assumptions. First, the model is over-parameterized, meaning that the number of parameters is (much) greater than the number of samples. This is the usual case in deep learning. Second, the optimizer is gradient descent with a sufficiently small learning rate. Our results also hold for other gradient methods such as momentum SGD and Adam, as long as the training loss converges to zero.

Let us first gain some insights from linear models, using an analysis similar to [52]. Then, we will study neural networks.

Insights from linear models. Consider a regression task, where the loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ is the squared loss. Consider using a linear model denoted by $f(x) = \langle \theta, x \rangle$ for $\theta \in \mathbb{R}^{d_{\mathcal{X}}}$. We assume that the model is over-parameterized, that is $d_{\mathcal{X}} > n$. The weight update rule of GRW under gradient descent (GD) is the following:

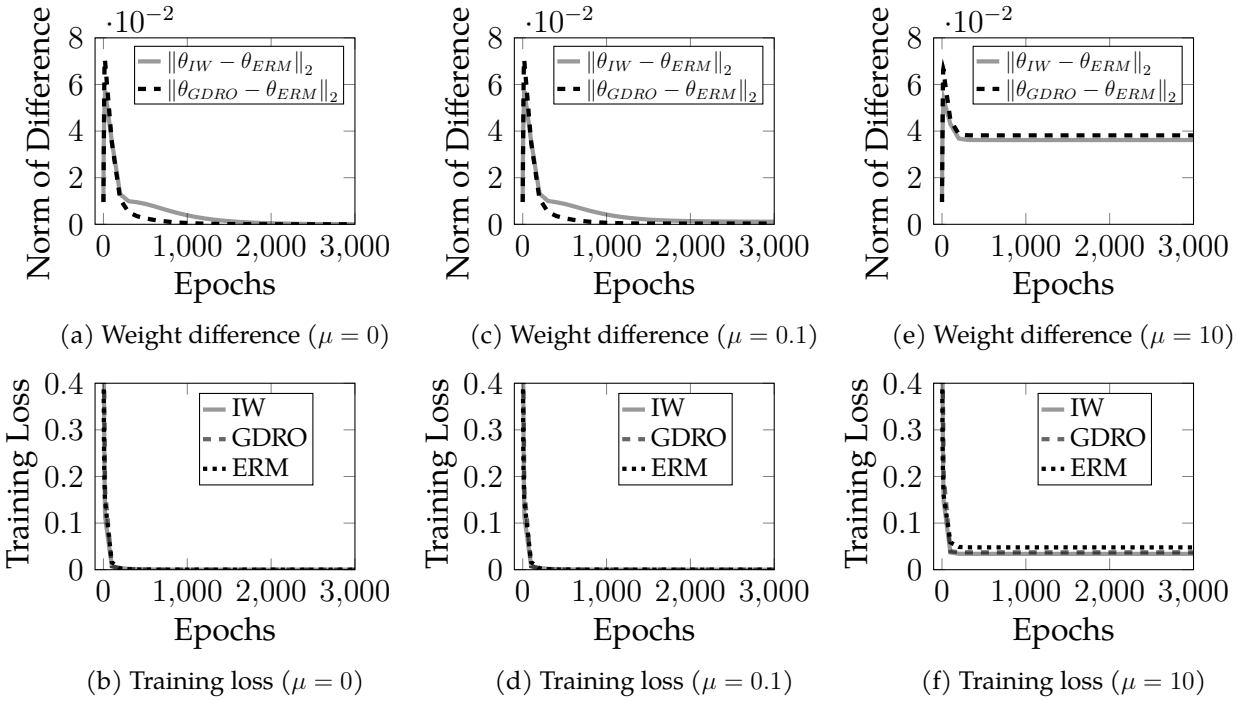
$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^{n} q_i^{(t)} \nabla_\theta \ell(f^{(t)}(x_i), y_i) = \theta^{(t)} - \eta \sum_{i=1}^{n} q_i^{(t)} (f^{(t)}(x_i) - y_i) x_i, \tag{6.3}$$

where $\eta > 0$ is the learning rate. We now show that, under some assumptions, as $t \to \infty$, $\theta^{(t)}$ must converge to a common θ^* for all GRW and ERM methods. The proof consists of two steps. First, we prove that the training loss will converge to zero. In this case, as long as $x_1, \cdots, x_n$ are linearly independent, θ^* must be an interpolator, which means that $\langle \theta^*, x_i \rangle = y_i$ for all $i \in [n]$. Second, we prove that there is a unique interpolator θ^*. The interpolator only depends on the starting point $\theta^{(0)}$ and the training samples, but it does not depend on the sample weights $\boldsymbol{q}^{(t)}$. These results require the following assumption about the sample weights.

Assumption 6.3. *There are constants $q_1, \cdots, q_n$ such that for all $i \in [n]$, there is $q_i^{(t)} \to q_i$ as $t \to \infty$. Moreover, $\min_i q_i = q^* > 0$.*

This assumption says two things. First, the sample weights become stable after a sufficient amount of training. It guarantees that the training loss will not change too much between two consecutive training steps. Second, all samples have positive weights, which means that no sample is deleted from the training set. This is important because the unique θ^* depends on the training samples. If samples were deleted, θ^* would not be the same. The following theorem shows that the training loss will converge to zero.

Theorem 6.4 (Proof in Appendix E.1)**.** *If $x_1, \cdots, x_n$ are linearly independent, then there exists a constant $\eta_0 > 0$ such that: For any GRW algorithm satisfying Assumption 6.3, under the update rule Eqn.* (6.3) *with $\eta \le \eta_0$, the empirical ERM risk $\hat{\mathcal{R}}_{\text{ERM}}(f^{(t)}) \to 0$ as $t \to \infty$.*



(a) Weight difference ($\mu = 0$) (c) Weight difference ($\mu = 0.1$) (e) Weight difference ($\mu = 10$)

(b) Training loss ($\mu = 0$) (d) Training loss ($\mu = 0.1$) (f) Training loss ($\mu = 10$)

Figure 6.1: Experiment results of ERM, importance weighting (IW) and group DRO (GDRO) with the squared loss and L^2 regularization on six MNIST images with a linear model. μ is the regularization coefficient. All norms are L^2 norms.

When $x_1, \cdots, x_n$ are linearly independent, Eqn. (6.3) implies that $\theta^{(t+1)} - \theta^{(t)}$ is always a linear combination of $x_1, \cdots, x_n$. As a result, for all t, $\theta^{(t)} - \theta^{(0)} \in \text{span}\{x_1, \cdots, x_n\}$. This is an n-dimensional subspace of $\mathbb{R}^d$. By Cramer's rule, there is exactly one $\tilde{\theta}$ in this subspace such that $\langle \tilde{\theta} + \theta^{(0)}, x_i \rangle = y_i$ for all $i \in [n]$. Therefore, $\theta^* = \tilde{\theta} + \theta^{(0)}$ is unique and independent of the sample weights. It only relies on $\theta^{(0)}$ and $x_1, \cdots, x_n$.

Finally, note that ERM is a special case of GRW. Thus, GRW produces the exact same model as ERM, so it cannot improve over ERM. One way to solve this problem is adding regularization. The regularization has two effects: (i) moving $\theta^{(t+1)} - \theta^{(t)}$ out of the span of $x_1, \cdots, x_n$; (ii) preventing the model from achieving zero loss, that is interpolating the training samples. Another solution is to add some new samples to (via data augmentation for example), or delete som samples from the dataset.

Let us demonstrate this result with a simple experiment. The experiment is conducted on a training set of six MNIST images, five of which are digit 0 and one is digit 1. The two different digits define two groups. We use a 784-dimensional linear model and run ERM, importance weighting and group DRO (with $\nu = 1$). The results are reported in Figure 6.1, in which (a) and (b) are the results when no regularization is applied. From (a) we can see that the three models will converge to the same θ; from (b) we can see that the training loss of all three models will converge to zero. Then, we apply L^2 regularization and run the experiment again. From (c) and (d) we can see that when the regularization is small, the training loss will still converge to zero, and the three models will still converge to the same θ. From (e) and (f) we can see that the three models converge to different points only when the regularization is so large that the training loss does not converge to zero.

To conclude, we gain the following insight from the above analysis: Without a very

large regularization, early stopping or altering the training set, GRW and ERM will produce very similar models, so GRW cannot be better than ERM.

Wide neural networks, regression tasks. With this insight, we now study neural networks. In particular, we focus on *sufficiently wide fully-connected neural networks* within the neural tangent kernel (NTK) regime [77]. The neural network is defined as

$$\boldsymbol{h}^{l+1} = \frac{\boldsymbol{W}^l}{\sqrt{d_l}}\boldsymbol{x}^l + \beta \boldsymbol{b}^l \ ; \qquad \boldsymbol{x}^0 = x, \ \boldsymbol{x}^{l+1} = \sigma(\boldsymbol{h}^{l+1}). \quad (l = 0, \cdots, L)$$

Here σ is a non-linear activation function, $\boldsymbol{W}^l \in \mathbb{R}^{d_{l+1} \times d_l}$ and $\boldsymbol{W}^L \in \mathbb{R}^{1 \times d_L}$, and $d_0 = d_{\mathcal{X}}$. The parameter vector θ consists of $\boldsymbol{W}^0, \cdots, \boldsymbol{W}^L$ and $\boldsymbol{b}^0, \cdots, \boldsymbol{b}^L$ (θ is the concatenation of all flattened weights and biases). The final output is $f(x) = \boldsymbol{h}^{L+1}$. A wide neural network has large $d_1, \cdots, d_L$. Moreover, the neural network is initialized as

$$\begin{cases} \boldsymbol{W}_{i,j}^{l(0)} \sim \mathcal{N}(0,1) \\ \boldsymbol{b}_j^{l(0)} \sim \mathcal{N}(0,1) \end{cases} \quad (l = 0, \cdots, L-1) \qquad \text{and} \qquad \begin{cases} \boldsymbol{W}_{i,j}^{L(0)} = 0 \\ \boldsymbol{b}_j^{L(0)} \sim \mathcal{N}(0,1) \end{cases}$$

Finally, we assume that σ is differentiable everywhere; and both σ and its first-order derivative $\dot{\sigma}$ are Lipschitz, meaning that there exists a constant $L > 0$ such that $|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|_2$ for all x_1, x_2. In the rest of this chapter, we will use *wide NN* to refer to a neural network that satisfies all the above conditions.

The *neural tangent kernel* (NTK) is defined as $\Theta^{(0)}(x, x') = \nabla_\theta f^{(0)}(x)^\top \nabla_\theta f^{(0)}(x')$. Our result is based on the following NTK theorem proved in [77].

Lemma 6.5. *If σ is Lipschitz and $d_l \to \infty$ for $l = 1, \cdots, L$ sequentially, then $\Theta^{(0)}(x, x')$ converges in probability to a non-degenerate deterministic limiting kernel $\Theta(x, x')$. Here "non-degenerate" means that $\Theta(x, x')$ depends on x and x' and is not a constant.*

The *kernel Gram matrix* $\Theta = \Theta(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{n \times n}$ is a positive semi-definite symmetric matrix. Denote its largest and smallest eigenvalues by $\lambda^{\max}$ and $\lambda^{\min}$. Note that Θ is non-degenerate, so we can assume that $\lambda^{\min} > 0$ (which is almost surely true when $d_L \gg n$).

Theorem 6.6 (Proof in Appendix E.2)**.** *Let $f^{(t)}$ be a wide NN trained by any GRW method satisfying Assumption 6.3 with the squared loss. Let $f_{\text{ERM}}^{(t)}$ be the same model trained by ERM from the same initial point. Suppose $d_1 = \cdots = d_L = \tilde{d}$, $\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)$ are linearly independent, and $\lambda^{\min} > 0$. Then, with a sufficiently small η, for any $\delta > 0$, there exists $\tilde{D}(\lambda) > 0$ such that when $\tilde{D}(\lambda) \leq \tilde{d} \to \infty$, with probability at least $(1 - \delta)$ over random initialization, there is*

$$\limsup_{t \to \infty} \left| f^{(t)}(x) - f_{\text{ERM}}^{(t)}(x) \right| = O(\tilde{d}^{-1/4}) \to 0 \qquad \textit{for all } x \in \mathbb{R}^d \textit{ such that } \|x\|_2 \leq 1.$$

This theorem says that on any test point x in the unit ball, the GRW model and the ERM model produce almost the same output. Thus, the two models must have similar OOD generalization performance. Note that for simplicity, we only prove for $d_1 = \cdots = d_L = \tilde{d} \to \infty$, but the result can be very easily extended to the case where $d_l/d_1 \to \alpha_l$ for $l = 2, \cdots, L$ for some constants $\alpha_2, \cdots, \alpha_L$, and $d_1 \to \infty$.

The key of proving this theorem is to consider the following *linearized neural network*:

$$f_{\text{lin}}^{(t)}(x) = f^{(0)}(x) + \left\langle \theta^{(t)} - \theta^{(0)}, \nabla_\theta f^{(0)}(x) \right\rangle, \tag{6.4}$$

which is a linear model *w.r.t.* $\nabla_\theta f^{(0)}(x)$. If $\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)$ are linearly independent (which is almost surely true when the model is overparameterized so that θ has a very high dimension), then our previous insight tells us that the linearized network will converge to the unique interpolator. It then suffices to show that the wide NN can be approximated by its linearized version uniformly throughout training.

Now let us study the effect of L^2 regularization, with which the GRW learning objective becomes

$$\hat{\mathcal{R}}^{\mu}_{\boldsymbol{q}^{(t)}}(f) = \sum_{i=1}^{n} q_i^{(t)} \ell(f(x_i), y_i) + \frac{\mu}{2} \left\| \theta - \theta^{(0)} \right\|_2^2. \tag{6.5}$$

Adding regularization does make a difference to the model regardless of how big μ is. However, to make it possible for GRW to improve over ERM, the regularization must be large enough to *significantly lower the training performance*. Otherwise, the final model would still be too close to the unregularized ERM model.

Theorem 6.7 (Proof in Appendix E.3)**.** *Let $f^{(t)}_{\text{reg}}$ be a regularized wide NN trained by any GRW algorithm satisfying Assumption 6.3 with the squared loss. Suppose there exists $M_0 > 0$ such that $\left\| \nabla_\theta f^{(0)}(x) \right\|_2 \le M_0$ for all $\|x\|_2 \le 1$. Suppose $\lambda^{\min} > 0$, $\mu > 0$, $d_1 = \cdots = d_L = \tilde{d}$, $\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)$ are linearly independent, and the learning rate is sufficiently small. If the empirical training risk of $f^{(t)}_{\text{reg}}$ satisfies $\limsup_{t\to\infty} \hat{\mathcal{R}}(f^{(t)}_{\text{reg}}) < \epsilon$ for some $\epsilon > 0$, then as $\tilde{d} \to \infty$, with probability close to 1 over random initialization there is*

$$\limsup_{t\to\infty} \left| f^{(t)}_{\text{reg}}(x) - f^{(t)}_{\text{ERM}}(x) \right| = O(\tilde{d}^{-1/4} + \sqrt{\epsilon}) \to O(\sqrt{\epsilon}) \quad \text{for all } x \in \mathbb{R}^d \text{ such that } \|x\|_2 \le 1.$$

Linear models, classification tasks. So far we have been discussing regression tasks. We now move on to binary classification, where the label space is $\mathcal{Y} = \{+1, -1\}$, and the loss is the logistic loss $\ell(\hat{y}, y) = \log(1 + \exp(-\hat{y}y))$. The big difference here is that *the logistic loss does not have finite minimizers*. The logistic loss converging to zero means that the model weight "explodes" to infinity instead of converging to a finite point.

Again consider the linear model $f(x) = \langle \theta, x \rangle$. Prior work has shown a couple of negative results under this setting. For example, [19] empirically observed that importance weighting does not improve over ERM for linear models, and [162] proved that for importance weighting, as $t \to \infty$, $\|\theta^{(t)}\|_2 \to \infty$ and $\theta^{(t)}/\|\theta^{(t)}\|_2$ converges to a unit vector that does not depend on the sample weights, so it does not improve over ERM.

We extend these results to GRW. First, we show that when the training error goes to zero, $\theta^{(t)}$ will converge to the *max-margin classifier* defined as

$$\hat{\theta}_{\text{MM}} = \arg\max_{\theta : \|\theta\|_2 = 1} \left\{ \min_{i=1,\cdots,n} y_i \cdot \langle \theta, x_i \rangle \right\}.$$

Theorem 6.8 (Proof in Appendix E.4)**.** *Suppose $x_1, \cdots, x_n$ are linearly independent. Suppose we use GRW such that for all $i \in [n]$, $\liminf_{t\to\infty} q_i^{(t)} > 0$. As $t \to \infty$, if the empirical risk $\hat{\mathcal{R}}(f^{(t)})$ converges to zero and $\theta^{(t)}/\left\|\theta^{(t)}\right\|_2 \to \boldsymbol{u}$ for some unit vector $\boldsymbol{u}$, then $\boldsymbol{u} = \hat{\theta}_{\text{MM}}$.*

This result is an extension of [133]. It implies that all GRW methods including ERM, if converge, must converge to the same point $\hat{\theta}_{\text{MM}}$ that does not depend on the sample weights $q_i^{(t)}$. Next, we show that any GRW satisfying Assumption 6.3 does converge.

Definition 6.9. *A first-order differentiable function f on $\mathcal{D}$ is called **L-smooth** for $L > 0$ if*

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|_2^2 \qquad \text{for all } x, y \in \mathcal{D}.$$

An equivalent definition is that f is L-smooth if it satisfies

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x - y\|_2 \qquad \textit{for all } x, y \in \mathcal{D}.$$

Theorem 6.10 (Proof in Appendix E.5)**.** *Suppose the loss function $\ell(\hat{y}, y)$ is convex, L-smooth in $\hat{y}$ and strictly monotonically decreasing to zero as $y\hat{y} \to +\infty$. Consider the linear model $f(x) = \langle \theta, x \rangle$. Suppose $x_1, \cdots, x_n$ are linearly independent. For any GRW that satisfies Assumption 6.3 with q_i, denote $F(\theta) = \sum_{i=1}^n q_i \ell(\langle \theta, x_i \rangle, y_i)$. Then, for a sufficiently small learning rate η:*

(i) $F(\theta^{(t)}) \to 0$ *as* $t \to \infty$.

(ii) $\left\|\theta^{(t)}\right\|_2 \to \infty$ *as* $t \to \infty$.

(iii) Let $\theta_R = \arg\min_\theta \{F(\theta) : \|\theta\|_2 \le R\}$. *$\theta_R$ is unique for any R such that* $\min_{\|\theta\|_2 \le R} F(\theta) < \min_i q_i \ell(0, y_i)$. *And if* $\lim_{R\to\infty} \frac{\theta_R}{R}$ *exists, then* $\lim_{t\to\infty} \frac{\theta^{(t)}}{\left\|\theta^{(t)}\right\|_2}$ *also exists and they are equal.*

This result is an extension of Theorem 1 in [79]. We show that the logistic loss satisfies the conditions of the above theorem and $\lim_{R\to\infty} \frac{\theta_R}{R} = \hat{\theta}_{\text{MM}}$ in Appendix E.6. Thus, Theorem 6.8 and Theorem 6.10 imply that for a linear model, all GRW methods satisfying Assumption 6.3 (including ERM) will converge to the same point.

Wide neural networks, classification tasks. We now study wide NNs with regularization. But before that, we have to point out that it is impossible to extend Theorem 6.10 to a wide NN without regularization. This is because for a neural network, if $\left\|\theta^{(t)}\right\|_2$ goes to infinity, then $\|\nabla_\theta f\|_2$ will also go to infinity, unlike a linear model where this gradient is a constant. Consequently, the gap between the neural networks and its linearized counterpart will "explode" under gradient descent, so we cannot prove that the wide NN can always be approximated by its linearized version similar to the previous section. However, with regularization, an approximation theorem can be proved.

Consider minimizing the regularized risk Eqn. (6.5) with a wide NN, and ℓ is the logistic loss. Define the max-margin linearized NN as

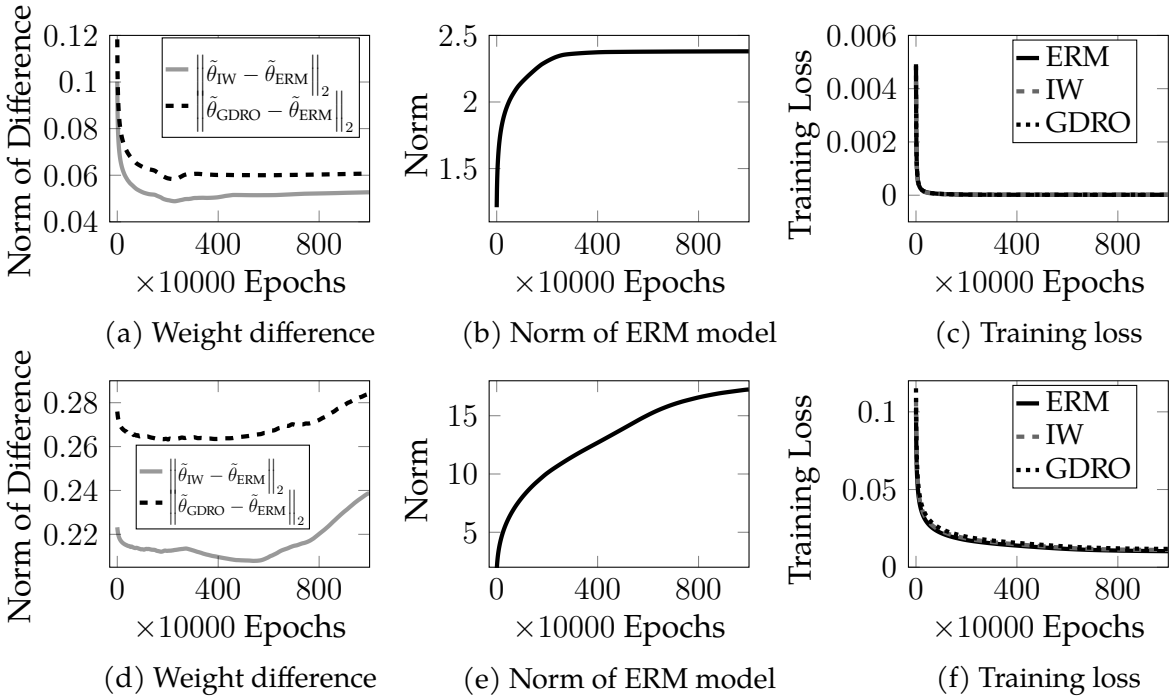
$$f_{\text{MM}}(x) = \left\langle \hat{\theta}_{\text{MM}}, \nabla_\theta f^{(0)}(x) \right\rangle \quad \text{where} \quad \hat{\theta}_{\text{MM}} = \arg\max_{\|\theta\|_2 = 1} \left\{ \min_{i=1,\cdots,n} y_i \cdot \left\langle \theta, \nabla_\theta f^{(0)}(x_i) \right\rangle \right\}$$

Note that f_{MM} does not depend on $q_i^{(t)}$. We can again show that regularization only works when it is large enough to sufficiently downgrade the training performance.

Theorem 6.11 (Proof in Appendix E.7)**.** *Suppose $\left\|\nabla_\theta f^{(0)}(x)\right\|_2$ is bounded. Under the same conditions of Theorem 6.7, when ℓ is the logistic loss, for any $\delta > 0$ there exists a constant $C(\delta) > 0$ such that with probability at least $1 - \delta$, the following holds as $\tilde{d} \to \infty$:*
For any $\epsilon \in (0, \frac{1}{4})$, if the training error has $\limsup_{t\to\infty} \hat{\mathcal{R}}^{\mu}_{\boldsymbol{q}^{(t)}}(f^{(t)}_{\text{reg}}) < \epsilon$, then for any x such that $|f_{\text{MM}}(x)| > C(\delta) \cdot (-\log 2\epsilon)^{-1/2}$, $f^{(t)}_{\text{reg}}(x)$ has the same sign as $f_{\text{MM}}(x)$ for a sufficiently large t.

Apart from adding a large regularization, is there any other way to prevent GRW from obtaining almost the same model as ERM in classification? The main reason why GRW always converges to the max-margin classifier regardless of the weights $q_i^{(t)}$ is that the logistic loss is exponentially tailed. Thus, one way to ensure that the sample weights have an impact is to use a polynomially tailed loss. For example, [150] defined the following polynomially tailed loss for linear classifiers:

$$\ell_{\alpha,\beta}(\hat{y}, y) = \begin{cases} \ell_{\text{left}}(\hat{y}, y), & \text{if } \hat{y}y < \beta; \\ \dfrac{1}{[\hat{y}y - (\beta - 1)]^\alpha}, & \text{if } \hat{y}y \ge \beta, \end{cases}$$



(a) Weight difference (b) Norm of ERM model (c) Training loss

(d) Weight difference (e) Norm of ERM model (f) Training loss

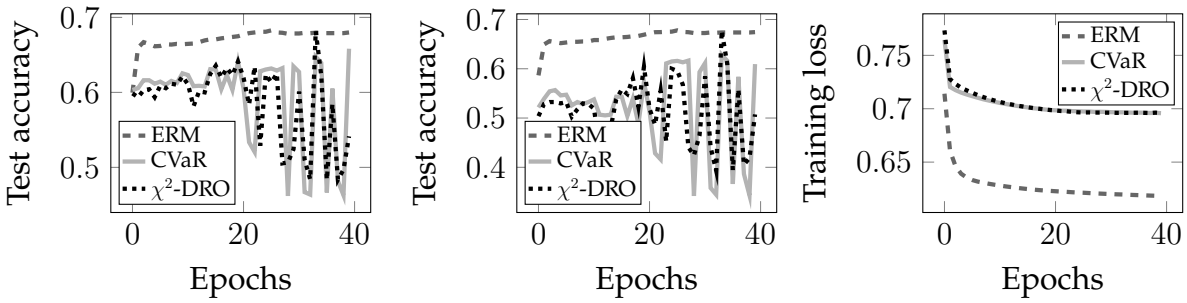
Figure 6.2: Experiment results of ERM, importance weighting (IW) and group DRO (GDRO) with the logistic loss (**top row**) and the polynomially tailed loss (**bottom row**) on a linear model. All norms are L^2 norms. $\tilde{\theta} = \theta/\|\theta\|_2$.

where ℓ_{left} is any function such that the overall loss function $\ell_{\alpha,\beta}$ is convex, differentiable and strictly decreasing. Here we empirically compare between the logistic loss and this polynomially tailed loss on the six MNIST images we used earlier. The results are plotted in Figure 6.2, and we can observe the following:

- For either loss function, the training loss of each method converges to 0.
- In theory the norm of the ERM model will explode to infinity, but in reality it will not because once the training loss becomes extremely small, it will turn into zero in the floating number representation, and thus the training halts.
- We can see a fundamental difference between the logistic loss and the polynomially tailed loss. For the logistic loss, the norm of the gap between importance weighting (or Group DRO) and ERM will converge to around 0.06 when the training stops, while for the polynomially tailed loss, the norm will be larger than 0.22 and will keep growing, which shows that for the polynomially tailed loss the normalized model weights do not converge to the same point.

6.3 Sensitivity to Outliers

Another issue with DRO is its sensitivity to outliers, which are samples significantly different from most of the sample in the dataset. Let us use CVaR as an example to see why DRO is particularly sensitive to outliers. CVaR places all the weights on the worst α fraction of the training samples. "Worst" here means that the model gets the highest loss on these samples. However, almost all real datasets contain outliers, and by their very definition, the model tends to have high loss on the outliers. This means that CVaR is very likely to place large weights on many outliers, which will make training very unstable, and the final performance pretty bad.



(a) Average accuracy (b) Worst-group accuracy (c) Training loss

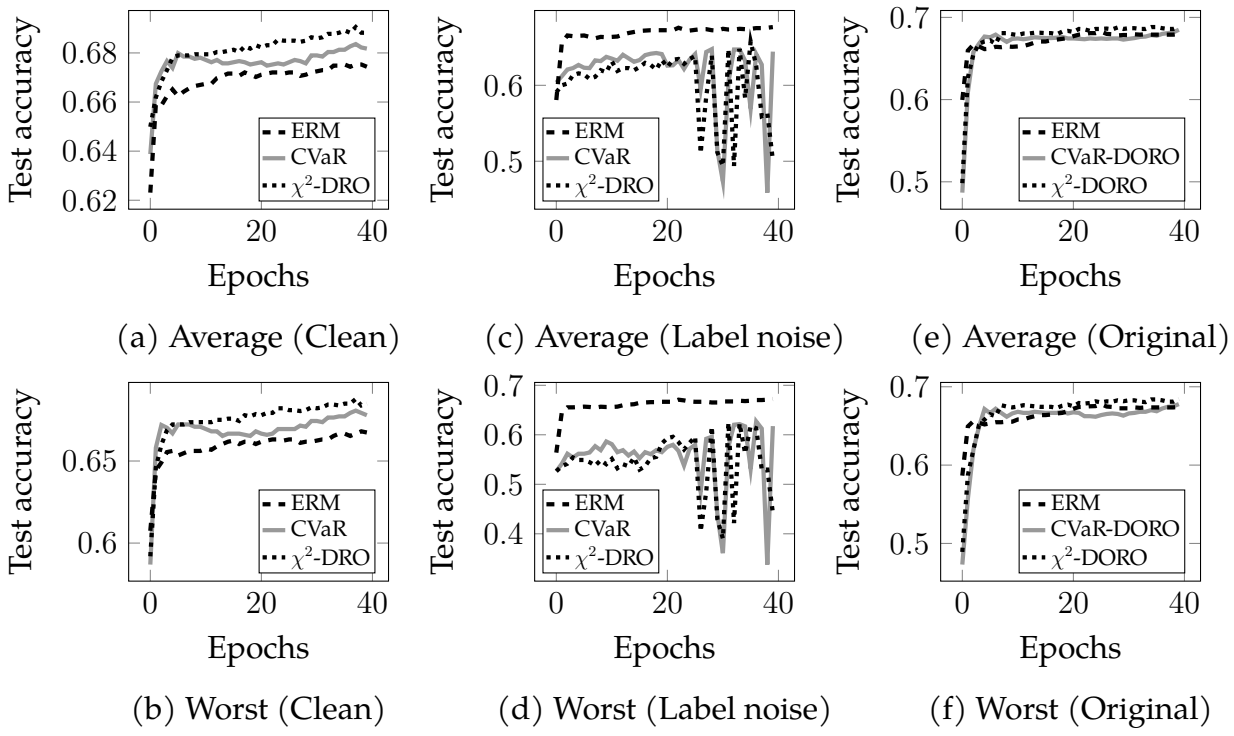
Figure 6.3: Results of ERM and two DRO methods on the original COMPAS data set.

Let us use an experiment on a real dataset to show that this is a real problem in practice. We use the COMPAS dataset [94], a recidivism prediction data set with 5049 training instances (after pre-processing and train-test splitting). We construct four groups on this data set with two sensitive features: race and sex. These two features define four overlapping groups (demographic groups): *White*, *Non-white*, *Male* and *Female*. We use a two-layer feed-forward neural network with ReLU activation, and train it with three methods: ERM, CVaR, and χ^2-DRO, which is a DRO method to be introduced later. The results are plotted in Figure 6.3. We report the average test accuracy, the the minimum test accuracy on any group (the worst-group accuracy). From figures (a) and (b), we can see that for both average and worst-group test accuracies, the two DRO methods are worse than ERM. Moreover, the two DRO curves are jumping up and down, showcasing the huge volatility of DRO training, in stark contrast to the stable curve of ERM. In addition, the training loss is plotted in figure (c), and we can see that the loss curve is pretty stable, meaning that this high volatility is not caused by optimization.

Next, we examine if this instability and poor performance is caused by the outliers in the dataset. For this purpose, we "clean" the dataset by removing from it 1000 "potential outliers", which are the samples on which the ERM model has a high loss. Note that these are not 100% outliers, but removing them has a huge impact on the performance of the two DRO methods. We "clean" the dataset in five rounds. In each round, we train a model from scratch on the samples with ERM, and then remove 200 training samples that incur the highest loss on this model. After five rounds, 1000 instances are removed in total, and we get a "clean" data set with 4049 samples. Figure 6.4 (a), (b) show the results on the "clean" dataset, from which we can see that the two DRO methods become very stable and better than ERM. It should be emphasized that up to this point, we have only removed samples from the dataset and added nothing into it. The outliers naturally exist in the original data set.

One might argue that these high-loss samples are not necessarily outliers. Thus, to further substantiate our claim, we conduct a third experiment where we add some outliers to the "clean" dataset. We use a common source of noise: incorrect labels. Specifically, we randomly flip 20% of the labels of the "clean" COMPAS dataset, and run the three methods again. The results are plotted in Figure 6.4 (c), (d). We can see that while the label noise only slightly influences ERM, it significantly downgrades the performance of the two DRO methods, and makes their training highly unstable again.

The experiment results so far should have convinced the reader that outliers have a much greater impact on DRO than ERM, and they can make DRO poor and highly unstable. In the following section, we will introduce a solution called DORO. Figure 6.4



(a) Average (Clean) (c) Average (Label noise) (e) Average (Original)

(b) Worst (Clean) (d) Worst (Label noise) (f) Worst (Original)

Figure 6.4: Average and worst-group test accuracies on: (a), (b) the "clean" dataset where potential outliers are removed; (c), (d) the noisy set where label noise is added to the "clean" set; (e), (f) the original COMPAS dataset, but with DORO algorithms.

(e), (f) plot the performance of DORO on the original COMPAS dataset (that is, before cleaning). We can see that DORO is much better and more stable than DRO.

6.4 Distributionally and Outlier Robust Optimization

The idea of distributionally and outlier robust optimization, or DORO, can be summarized by Figure 6.5. We know that CVaR places large weights on the worst samples, which may contain lots of outliers. Instead, CVaR-DORO ignores the worst of the worst samples, and places weights on the second-worst fraction of the samples, in order to avoid potential outliers. We can extend this idea to a family of DRO method called the Cressie-Read family.

Recall that a DRO method is defined by the constraint $D(Q \parallel P) \leq \rho$, where the divergence function $D(Q \parallel P)$ measures the difference between Q and P. There are two large families of divergence functions, namely integral probability metrics (IPMs) and f-divergences. An IPM is defined as $D(Q \parallel P) = \sup_{f \in \mathcal{F}} [\mathbb{E}_{X \sim Q} f(X) - \mathbb{E}_{Y \sim P} f(Y)]$ for some function class $\mathcal{F}$. It is symmetric, so it is also written as $D(Q, P)$. Examples include the total variation (TV) distance defined as $D(Q, P) = \frac{1}{2} \int |Q(x) - P(x)| dx$, maximum mean discrepancy (MMD) defined as $D(Q, P) = \|\mathbb{E}_{X \sim Q} \pi(X) - \mathbb{E}_{Y \sim P} \pi(Y)\|_{\mathcal{H}}$ for some feature map π and some RKHS $\mathcal{H}$, and the Wasserstein distance defined as $D(Q, P) = \inf_{\gamma \in \Gamma(Q,P)} \int \rho(x, y) d\gamma(x, y)$ for some metric function $\rho(\cdot, \cdot)$.

An f-divergence is defined as $D_f(Q \parallel P) = \int f\left(\frac{dQ}{dP}\right) dP$ for some function f. An f-divergence is not necessarily symmetric. For example, when $f(t) = -\log t$, then the f-divergence becomes the popular KL-divergence $D_f(Q \parallel P) = D_{KL}(P \parallel Q) =$

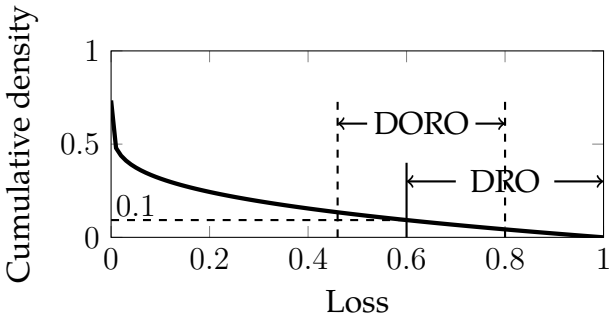


Figure 6.5: Comparison between DRO and DORO for CVaR.

$\int \log\left(\frac{P(x)}{Q(x)}\right) dP(x)$ (note that P and Q are reversed). The TV distance is the only non-trivial divergence function that is both an IPM and an f-divergence.

The Cressie-Read family of Rényi divergence [30] is a family of f-divergences. For any $\beta > 1$, the divergence is defined as

$$D_\beta(Q \parallel P) = \int f_\beta\left(\frac{dQ}{dP}\right) dP, \qquad \text{where } f_\beta(t) = \frac{1}{\beta(\beta-1)}\left(t^\beta - \beta t + \beta - 1\right).$$

The reason why we are interested in this family is that the DRO risk Eqn. (6.1) *w.r.t.* D_β has the following dual characterization (see [38, Lemma 1]):

$$\mathcal{R}_{D_\beta,\rho}(f;P) = \inf_{\eta\in\mathbb{R}} \left\{ c_\beta(\rho)\mathbb{E}_{(X,Y)\sim P}\left[(\ell(f(X),Y)-\eta)_+^{\beta_*}\right]^{\frac{1}{\beta_*}} + \eta \right\}, \tag{6.6}$$

where $(x)_+ = \max\{x, 0\}$, $\beta_* = \frac{\beta}{\beta-1}$, and $c_\beta(\rho) = (1+\beta(\beta-1)\rho)^{1/\beta}$.

We first show the relationship between the DRO loss and the worst-group risk. Suppose there are K groups $\mathcal{D}_1, \cdots, \mathcal{D}_K$. Let $P_k(x,y) = P((x,y)|(x,y) \in \mathcal{D}_k)$. Then, the worst-group risk is given by

$$\mathcal{R}_{\max}(f;P) = \max_{k=1,\cdots,K} \mathcal{R}(f;P_k) = \max_{k=1,\cdots,K} \mathbb{E}_{(X,Y)\sim P}[\ell(f(X),Y)|(X,Y)\in\mathcal{D}_k].$$

Consider the scenario where we do not know $\mathcal{D}_1, \cdots, \mathcal{D}_K$ at train time, but only know that the smallest group is at least $\alpha \in (0,1)$ of the population size. Then, we can use the DRO risk given by the Cressie-Read family to obtain a surrogate of the worst-group risk, as shown by the following result.

Proposition 6.12. *Let $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k) \le \exp(-1) \approx 36.8\%$ be the minima group size, and define $\rho = f_\beta(\frac{1}{\alpha})$, then we have $\mathcal{R}_{\max}(f;P) \le \mathcal{R}_{D_\beta,\rho}(f;P)$.*

Proof Note that $f'_\beta(t) = \frac{1}{\beta-1}\left(t^{\beta-1}-1\right)$. Thus, $f'_\beta(t)$ is decreasing when $t \in [0,1]$ and increasing when $t \in \left[1, \frac{1}{\alpha}\right]$. This implies that $f_\beta(t) \le \max\left\{f_\beta(0), f_\beta\left(\frac{1}{\alpha}\right)\right\}$. We can further verify that $f_\beta\left(\frac{1}{\alpha}\right) - f_\beta(0) = \frac{1}{\beta(\beta-1)}\left(\frac{1}{\alpha^\beta} - \frac{\beta}{\alpha}\right)$, which is non-negative when $\alpha \le \beta^{-\frac{1}{\beta-1}}$. Since $\beta > 1$, we have $\beta^{-\frac{1}{\beta-1}} \ge \exp(-1)$. Thus, we have essentially proved that when $\alpha \le \exp(-1)$, there is

$$\forall t \in \left[0, \frac{1}{\alpha}\right],\ f_\beta(t) \le f_\beta\left(\frac{1}{\alpha}\right).$$

For any k, there is $\frac{dP_k}{dP} \le \frac{1}{\alpha}$. Thus, we have

$$D_\beta(P_k \parallel P) = \int f_\beta\left(\frac{dP_k}{dP}\right) dP \le \int f_\beta\left(\frac{1}{\alpha}\right) dP = f_\beta\left(\frac{1}{\alpha}\right),$$

Method	β	β_*	ρ	$c_\beta(\rho)$	$D_\beta(Q \parallel P)$	Risk notation
CVaR	∞	1	$-\log\alpha$	α^{-1}	$\sup \log \frac{dQ}{dP}$	$\text{CVaR}_\alpha(f;P)$
χ^2-DRO	2	2	$\frac{1}{2}\left(\frac{1}{\alpha}-1\right)^2$	$\sqrt{1+\left(\frac{1}{\alpha}-1\right)^2}$	$\frac{1}{2}\int\left(\frac{dQ}{dP}-1\right)^2 dP$	$\mathcal{R}_{D_{\chi^2,\rho}}(f;P)$

Table 6.1: The two DRO risks studied in our analysis.

which combined with the definition of $\mathcal{R}_{D_\beta,\rho}$ completes the proof. ■

While the Cressie-Read family only defines the f-divergence for finite $\beta \in (1,+\infty)$, it can be shown that the dual characterization Eqn. (6.6) is valid for $\beta=\infty$ as well, for which D_β becomes CVaR. Our analysis below focuses on two cases: (i) $\beta=\infty$, which corresponds to CVaR; (ii) $\beta=2$, which corresponds to the χ^2-DRO risk used in [59]. Table 6.1 summarizes the relevant information about these two DRO risks. We will denote the CVaR risk by $\text{CVaR}_\alpha(f;P)$, and the χ^2-DRO risk by $\mathcal{R}_{D_{\chi^2,\rho}}(f;P)$.

From Eqn. (6.6), we derive the following dual form of CVaR:

$$\text{CVaR}_\alpha(f;P) = \inf_{\eta\in\mathbb{R}} \left\{\alpha^{-1}\mathbb{E}_{(X,Y)\sim P}\left[(\ell(f(X),Y)-\eta)_+\right]+\eta\right\}. \tag{6.7}$$

It is easy to see that the optimal η of Eqn. (6.7) is the α-quantile of the loss defined as

$$q_{f;P}(\alpha) = \inf_q \left\{P_{(X,Y)\sim P}[\ell(f(X),Y)>q] \le \alpha\right\}. \tag{6.8}$$

Proposition 6.12 implies that both CVaR and χ^2-risks are upper bounds of $\mathcal{R}_{\max}$.

Corollary 6.13. *Let $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k) \le \exp(-1)$ be the minimal group size, and $\rho = \frac{1}{2}\left(\frac{1}{\alpha}-1\right)^2$. Then, we have $\mathcal{R}_{\max}(f;P) \le \text{CVaR}_\alpha(f;P) \le \mathcal{R}_{D_{\chi^2,\rho}}(f;P)$.*

Proof Denote $p_k = P(\mathcal{D}_k)$ for all k. Then, $P(x,y) = p_k P(x,y|\mathcal{D}_k) + (1-p_k)P(x,y|\overline{\mathcal{D}_k})$. Let $Q=P_k$ and $Q'(x,y) = \frac{p_k-\alpha}{1-\alpha}P(x,y|\mathcal{D}_k) + \frac{1-p_k}{1-\alpha}P(x,y|\overline{\mathcal{D}_k})$. Then, $P=\alpha Q+(1-\alpha)Q'$, which by the definition of CVaR implies that $\mathbb{E}_{P_k}[\ell(f(X),Y)] \le \text{CVaR}_\alpha(f;P)$. Thus, $\mathcal{R}_{\max}(f;P) \le \text{CVaR}_\alpha(f;P)$. On the other hand, for any Q such that there exists Q' that satisfies $P=\alpha Q+(1-\alpha)Q'$, there is $\frac{dQ}{dP}(x,y) \le \frac{1}{\alpha}$ *a.e.*. Thus, $D_{\chi^2}(Q \parallel P) \le \frac{1}{2}\left(\frac{1}{\alpha}-1\right)^2 = \rho$. This implies that $\text{CVaR}_\alpha(f;P) \le \mathcal{R}_{D_{\chi^2,\rho}}(f;P)$. ■

Method. We model the outliers in the dataset with **Huber's ϵ-contamination model** [71]. Let P be the clean distribution without outliers, and P_{train} be the observed contaminated training distribution. Then, this model assumes that

$$P_{\text{train}} = (1-\epsilon)P + \epsilon\tilde{P}, \quad \text{where } \tilde{P} \text{ is an arbitrary distribution, and } 0<\epsilon<\frac{1}{2}.$$

As we saw in Figure 6.5, DORO ignores the worst samples where the loss is the highest. With this insight, we define the DORO risk as follows.

Definition 6.14. *The **expected ϵ-DORO risk** is defined as*

$$\mathcal{R}_{D,\rho,\epsilon}(f;P_{\text{train}}) = \inf_{P'}\left\{\mathcal{R}_{D,\rho(f;P')} \;\middle|\; \exists\, \tilde{P}' \text{ s.t. } P_{\text{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'\right\}. \tag{6.9}$$

The DORO risk has the following relationship with the total variation.

Lemma 6.15. *Let* $\mathrm{TV}(P,Q) = \frac{1}{2}\int_{\mathcal{X}\times\mathcal{Y}} |P(z)-Q(z)|dz$ *be the total variation, and* P_{train} *be given by the Huber's model. Then the DORO risk can be lower bounded by*

$$\mathcal{R}_{D,\rho,\epsilon}(\theta; P_{\mathrm{train}}) \geq \inf_{P''} \left\{ \mathcal{R}_{D,\rho}(\theta; P'') : \mathrm{TV}(P, P'') \leq \frac{\epsilon}{1-\epsilon} \right\}.$$

Proof For any P' such that $P_{\mathrm{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'$ for some $\tilde{P}'$. Let $U(z) = \min\{P(z), P'(z)\}$ for all $z \in \mathcal{X}\times\mathcal{Y}$. Then, we have

$$(1-\epsilon)U(z) + \epsilon\tilde{P}(z) + \epsilon\tilde{P}'(z) \geq P_{\mathrm{train}}(z) \quad \text{for any } z \in \mathcal{X}\times\mathcal{Y},$$

as both $\tilde{P}(z)$ and $\tilde{P}'(z)$ are non-negative. Integrating both sides produces $\int_{\mathcal{X}\times\mathcal{Y}} U(z)dz \geq \frac{1-2\epsilon}{1-\epsilon}$, which implies that $\mathrm{TV}(P,P') \leq \frac{\epsilon}{1-\epsilon}$. Thus, we have

$$\mathcal{R}_{D,\rho}(\theta; P') \geq \inf_{P''} \left\{ \mathcal{R}_{D,\rho}(\theta, P'') : \mathrm{TV}(P, P'') \leq \frac{\epsilon}{1-\epsilon} \right\},$$

which combined with the definition of the DORO risk proves the result. ■

With the Cressie-Read family, the DORO risk has the following dual formula:

Proposition 6.16 (Proof in Appendix E.8)**.** *Let* ℓ *be a continuous non-negative loss function, and suppose* P_{train} *is a continuous distribution. Then, we have*

$$\mathcal{R}_{D,\rho,\epsilon}(f; P_{\mathrm{train}}) = \inf_{\eta} \left\{ c_\beta(\rho) \mathop{\mathbb{E}}_{(X,Y)\sim P_{\mathrm{train}}} \left[(\ell-\eta)_+^{\beta_*} \,\middle|\, \Pr_{(X',Y')\sim P_{\mathrm{train}}} \{\ell > \ell'\} \geq \epsilon \right]^{\frac{1}{\beta_*}} + \eta \right\},$$

where $\ell = \ell(f(X), Y)$*, and* $\ell' = \ell(f(X'), Y')$*.*

With this dual formula, the DORO risk can be minimized using Algorithm 5. For each batch of samples, this algorithm first sorts the samples by their training losses, and then finds the optimal η^* in the above dual form. For example, we can use Brent's method [17] to find η^*. Then, this η^* is fixed and θ is updated to minimize the dual form. This algorithm is inspired by the ITLM algorithm [127], in which an alternative approach to making DRO more robust to outliers was proposed—removing the potential outliers from the dataset via data preprocessing. In comparison, DORO does not throw away any data. In addition, preprocessing methods cannot cope with online data where new samples are received in a stream, but DORO is still feasible.

Theoretical guarantee. We now show that the DORO risk is a surrogate of the worst-group risk, meaning that the DORO risk is an upper bound. This result parallels Corollary 6.13 in the uncontaminated setting, and guarantees that minimizing the DORO risk over P_{train} efficiently minimizes $\mathcal{R}_{\max}$ over P.

Theorem 6.17 (Proof in Appendix E.9)**.** *Let* $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k)$*, and* $\rho = \frac{1}{2}\left(\frac{1}{\alpha}-1\right)^2$*. Suppose* P_{train} *satisfies the Huber's* ϵ*-contamination model. Suppose* ℓ *is a non-negative loss function with a uniformly bounded second moment:* $\mathbb{E}_{(X,Y)\sim P}[\ell(f_\theta(X), Y)^2] \leq \sigma^2$ *for all* θ*. Then, we have*

$$\begin{aligned} \mathcal{R}_{\max}(f_\theta; P) &\leq \max\left\{ 3\mathrm{CVaR}_{\alpha,\epsilon}(f_\theta; P_{\mathrm{train}}), 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}} \right\} \\ &\leq \max\left\{ 3D_{\chi^2,\rho,\epsilon}(f_\theta; P_{\mathrm{train}}), 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}} \right\}. \end{aligned} \tag{6.10}$$

Algorithm 5 Minimizing the DORO risk with D_β Divergence

Input: Batch size n, outlier fraction ϵ, minimal group size α, initial model weight θ
for each iteration **do**
Sample a batch $(x_1, y_1), \cdots, (x_n, y_n) \sim P_{\text{train}}$
Compute the sample losses: $\ell_i = \ell(f_\theta(x_i), y_i)$ for $i = 1, \cdots, n$
Sort the sample losses: $\ell_{i_1} \geq \cdots \geq \ell_{i_n}$
Define $F(\theta, \eta) = c_\beta(\rho) \cdot [\frac{1}{n - \lfloor \epsilon n \rfloor} \sum_{j=\lfloor \epsilon n \rfloor + 1}^{n} (\ell(f_\theta(x_{i_j}), y_{i_j}) - \eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \eta$
Find $\eta^* = \arg\min_\eta F(\theta, \eta)$
Update θ by one step to minimize $\ell(\theta) = F(\theta, \eta^*)$ with some gradient method

Dataset	Method	Average Accuracy	Worst-group Accuracy
COMPAS	ERM	69.31 ± 0.19	68.83 ± 0.18
	CVaR	68.52 ± 0.31	68.22 ± 0.30
	CVaR-DORO	69.38 ± 0.10	69.11 ± 0.05
	χ^2-DRO	67.93 ± 0.40	67.32 ± 0.60
	χ^2-DORO	69.62 ± 0.16	69.22 ± 0.11
CelebA	ERM	95.01 ± 0.38	53.94 ± 2.02
	CVaR	82.83 ± 1.33	66.44 ± 2.34
	CVaR-DORO	92.91 ± 0.48	72.17 ± 3.14
	χ^2-DRO	83.85 ± 1.42	67.76 ± 3.22
	χ^2-DORO	82.18 ± 1.17	68.33 ± 1.79
CivilComments-Wilds	ERM	92.04 ± 0.24	64.62 ± 2.48
	CVaR	89.11 ± 0.76	63.90 ± 4.42
	CVaR-DORO	90.45 ± 0.70	68.00 ± 2.10
	χ^2-DRO	90.08 ± 0.92	65.55 ± 1.51
	χ^2-DORO	90.11 ± 1.09	67.19 ± 2.51

Table 6.2: Average/worst-group test accuracies of the selected models. (%)

Empirical evaluation. We test DORO on three datasets: COMPAS, CelebA [103] and CivilComments-Wilds [16, 88]. CelebA is a facial recognition dataset, where the target is whether the person has blond hair. CivilComments-Wilds is a toxicity identification NLP dataset, where the target is whether an online post contains toxic contents. All targets are binary. For COMPAS, we randomly sample 70% of the samples to be the training data. The other two datasetse have official train-val-test splits. On COMPAS we define 4 groups, and on each of the other datasets we define 16 groups.

We use a two-layer ReLU-activated feed-forward neural network on the COMPAS dataset, a ResNet18 [61] on CelebA, and a BERT-base-uncased [34] on CivilComments-Wilds. Each algorithm is run 300 epochs on COMPAS, 30 epochs on CelebA and 5 epochs on CivilComments-Wilds. The best model is selected based on the worst-group accuracy on the validation set. Note that in reality, the worst-group accuracy is not available because we might now know the group membership of each sample. Therefore, this model selection strategy is an oracle one. We will discuss more on this point later.

Table 6.2 reports the 95% confidence intervals of the mean test accuracies on each dataset. For DRO and DORO, we do a grid search to pick the best α and ϵ that achieve the best worst-group accuracy. Each experiment is repeated 10 times on COMPAS and

Method	Average	Worst-group
ERM	0.73 ± 0.06	8.59 ± 0.90
CVaR	11.53 ± 1.72	21.47 ± 0.71
CVaR-DORO	4.03 ± 1.57	16.84 ± 0.91
χ^2-DRO	8.88 ± 2.98	19.06 ± 1.18
χ^2-DORO	1.60 ± 0.34	13.01 ± 1.40

Table 6.3: Standard deviations of test accuracies during training on CelebA.

CelebA, and 5 times on CivilComments-Wilds with different random seeds. From the table, we can conclude that DORO consistently outperforms DRO in terms of the average and worst-group test accuracies.

Next, we show that DORO enhances training stability. We compute the standard deviations of the average and worst-group test accuracies across epochs during training on CelebA, and the results are reported in Table 6.3. We can see that DORO lowers the standard deviations, which means that its training is more stable.

Difficulty of model selection. In our experiments, we select the best models based on their worst-group validation accuracies, which is an oracle strategy. However, model selection without access to the worst-group accuracy is extremely hard. We tried the following three strategies of model selection in our experiments:

- Max average accuracy
- Min CVaR risk
- Min CVaR-DORO risk

All of these strategies are significantly worse than the oracle strategy. The main reason is that though we have shown that CVaR and CVaR-DORO risks are surrogates of the worst-group risk, they do not necessarily have a monotonic relationship. Therefore, we pose model selection in this scenario as an open problem.

To summarize this chapter, we first introduced the most popular solutions to subpopulation shift—reweighting and DRO. Then, we demonstrated two issues they have. First, they might not be able to improve over ERM because the models they lead to are too similar. Thus, a large regularization, early stopping or a different loss function (such as the polynomially tailed loss) is neccessary for them to work. Second, DRO is very sensitive to outliers because it places large weights on them. To solve this problem, we proposed the DORO risk which avoids the potential outliers.

The key takeaway from this chapter is that generalization under distribution shift, or **distributionally robust generalization (DRG)**, is extremely hard, much harder than DRO. Over the past decade a lot of methods for DRO have been proposed, but whether these models can lead to better DRG is questionable. In fact, as mentioned earlier, [51] showed that many of these methods do not have better DRG. The same applies to representation learning. For example, upweighting the data in a certain domain during pretraining does not necessarily lead to a higher downstream performance on that domain. Thus, we need to be very cautious of our heuristics.

Chapter 7

Conclusion

Here is a summary of the key results from the contexture theory:

1. Representations are learned from the association between input X and a context variable A, and we call this association the contexture.
2. Learning the contexture, or extracting the top-d eigenspace, can preserve the most information of T_{P^+}, the expectation operator.
3. The top-d eigenspace can be extracted by training a large encoder to optimize certain variational objectives. This is how deep representation learning works.
4. Learning the contexture is optimal if the task is compatible with the context.
5. Making models larger inevitably produces diminishing returns, and creating better contexts is necessary for further advancements of pretraining.
6. A good context should have a moderate association between X and A.
7. Mixing multiple existing contexts with convolution, convex combination and concatenation can lead to better contexts.
8. The context complexity affects the sample complexity of representation learning.
9. Learning the contexture can be viewed as STKR with the truncation function.
10. Analyzing the distribution shift from pretraining to downstream is very hard.

Let us conclude this thesis by discussing its limitations, and posing some open problems for future work.

Effect of optimization and model architecture. Throughout this thesis, we did not analyze the effect of optimization and model architecture on representation learning, though in Section 2.6 we showed that scaling up the model size brings the learned representation closer to the top-d singular functions of T_{P^+}. In reality, the *implicit bias* of optimization and the *inductive bias* of model architecture are both very important to the encoder. For optimization, [28] showed that if the model is trained with popular gradient methods such as gradient descent or Adam [86], then after a sufficient amount of training time, the model weights will oscillate around what they called the *edge of stability*, instead of converging to any optimal weights. Thus, the first open problem is whether the representation is always close to the top-d singular functions when it is oscillating. And even better, can we characterize this oscillating representation as a dynamical system? The inductive bias of the model architecture contains our prior knowledge about the task, and thus should be considered as a part of the context. For example, in Section 2.4 we proved that when the model is an encoder composed with a feature map, then the representation trained via supervised learning will be the contexture of the convolution of two contexts—the feature map and the label. Thus, the second open

problem is how to express the inductive bias of any arbitrary model architecture as a context, and how the context affects the learned representation.

Context scaling. The contexture theory makes an important prediction that the next major breakthrough in pretraining will be the result of context scaling, where a much better and more complex context is obtained from the real world rather than human heuristics. For example, base large language models are pretrained with masked token prediction. To further improve these models, RLHF [110] created new contexts by collecting human preferences of different completions of the same prompt. Such large-scale data collection serves as a major source of new contexts, and deep learning has been proved to be quite capable of finding the patterns within these contexts. For example, AlphaFold [82] can learn from the context between protein sequences and their structures through data collected from lab experiments. The major downside is that these experiments are usually quite expensive, and could sometimes take years to conduct.

Another source of new contexts that has been explored quite a lot recently is multi modalities [98]. In Section 2.2 we showed that multi-modal models such as CLIP can learn from the context between different modalities. Apart from image and text, there are many modalities to be explored, such as videos, tables, graphs, etc. The problem is how to merge multiple modalities through the contexture theory in order to learn a "world model".

Towards system 2 thinking (reasoning). It was mentioned in the introduction that the contexture learning only covers system 1 thinking, that is the fast, automatic and associative thinking. Currently it is not applicable to system 2 thinking, such as logical reasoning. Since system 2 thinking is slow and effortful for humans, we hypothesize that pretraining alone is not sufficient for learning system 2 thinking—post-training is necessary. Indeed, this has been a hot topic recently in LLM research [53, 78, 155].

The high-level idea is to train a model that does not produce outputs so quickly. In pretraining, Φ is nothing more than a function, and computing $\Phi(x)$ on a computer usually requires only a fraction of a second. For system 2 thinking, we would like the computation of $\Phi(x)$ to take a much longer time, which is called *test-time scaling*. It usually involves a chain of thinking steps, such that $a_1 = \Phi(x), a_2 = \Phi(x; a_1), a_3 = \Phi(x; a_1, a_2)$ and so on, and the final output is a_L. Here, $a_1, a_2, \cdots$ are the intermediate results, also known as a scratch pad. Recent theoretical studies [39, 97] also showed that such a sequential procedure is necessary for LLMs to carry out certain logical reasoning.

The open problem is: Will test-time scaling always improve the performance, or will it achieve diminishing returns after some point? For example, when an LLM thinks for 3 minutes, its performance is usually significantly better than if it only thinks for 1 minute. But is an LLM that thinks for 3 weeks significantly better than an LLM thinking for 1 week? What about 3 years versus 1 year? If an LLM thinks for 3 years, will it be able to solve problems that no human beings can solve, such as proving $P \neq NP$? To answer these questions, we need a scientific understanding of the mechanism of system 2 thinking, similar to what we have done for system 1 thinking in this thesis.

Appendix A

Proofs for Chapter 2

A.1 Proof of Theorem 2.2

Let us first prove the following lemma.

Lemma A.1. *$T_{P^+}\Lambda T_{P^+}^*$ is the integral kernel operator of the following kernel*

$$k(x,x') = \iint k_\Lambda(a,a')P^+(a|x)P^+(a'|x')dada'.$$

Proof

By definition, we have $(T_{P^+}^* h)(a') = \int h(x')P^+(x'|a')dx'$, which implies that

$$\begin{aligned}(\Lambda T_{P^+}^* h)(a) &= \int (T_{P^+}^* h)(a')k_\Lambda(a,a')P_{\mathcal{A}}(a')da' \\ &= \iint h(x')P^+(x'|a')k_\Lambda(a,a')P_{\mathcal{A}}(a')dx'da' \\ &= \iint h(x')P^+(a'|x')k_\Lambda(a,a')P_{\mathcal{X}}(x')dx'da'.\end{aligned}$$

This further implies that

$$\begin{aligned}(T_{P^+}\Lambda T_{P^+}^* h)(x) &= \int (\Lambda T_{P^+}^* h)(a)P^+(a|x)da \\ &= \iiint h(x')k_\Lambda(a,a')P^+(a|x)P^+(a'|x')P_{\mathcal{X}}(x')dada'dx' \\ &= \int h(x')k(x,x')P_{\mathcal{X}}(x')dx',\end{aligned}$$

as desired. ■

Now, we prove Theorem 2.2.

Proof Denote $\mathcal{R}(\Phi, \boldsymbol{W}) = \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\|A - \boldsymbol{W}\Phi(X)\|_2^2\right]$. Assuming, without loss of generality, that $\mathbb{E}_{X\sim P_{\mathcal{X}}}[\Phi_i\Phi_j] = \delta_{ij}$; otherwise one can perform Gram-Schmidt process on Φ_i and change $\boldsymbol{W}$ respectively. Thus, it amounts to minimize

$$\begin{aligned}\mathcal{R}(\Phi, \boldsymbol{W}) &= \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}}\ \mathop{\mathbb{E}}_{A\sim P^+(\cdot|X)}\left[\|A - \boldsymbol{W}\Phi(X)\|_2^2\right] \\ &= \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}}\|\boldsymbol{W}\Phi(X)\|_2^2 - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\langle A, \boldsymbol{W}\Phi(X)\rangle + \mathop{\mathbb{E}}_{A\sim P_{\mathcal{A}}}\|A\|_2^2 \\ &= \|\boldsymbol{W}\|_F^2 - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\langle A, \boldsymbol{W}\Phi(X)\rangle + \mathop{\mathbb{E}}_{A\sim P_{\mathcal{A}}}\|A\|_2^2.\end{aligned}$$

Denote $\boldsymbol{W} = (w_{ij})_{1\le i\le d_A, 1\le j\le d}$. We have

$$\frac{\partial \mathcal{R}}{\partial w_{ij}} = 2w_{ij} - 2 \mathop{\mathbb{E}}_{(X,A)\sim P^+} [A_i \Phi_j(X)],$$

which implies that for a fixed Φ, the optimal $\boldsymbol{W}$ that minimizes this loss should satisfy

$$w_{ij} = \mathop{\mathbb{E}}_{(X,A)\sim P^+} [A_i \Phi_j(X)].$$

Combining the minimizer of $\boldsymbol{W}$ with $\mathcal{R}$ and notice that $\mathbb{E}_{A\sim P_{\mathcal{A}}} \|A\|_2^2$ is a constant, it suffices to **maximize**

$$\begin{aligned} F(\Phi) &= \sum_{i,j} \left[\mathop{\mathbb{E}}_{(X,A)\sim P^+} A_i \Phi_j(X) \right]^2 \\ &= \int \sum_j \Phi_j(x_1)\Phi_j(x_2)\langle a_1, a_2\rangle P_{\mathcal{X}}(x_1) P^+(a_1|x_1) P_{\mathcal{X}}(x_2) P^+(a_2|x_2) dx_1 da_1 dx_2 da_2 \\ &= \iint \sum_j \Phi_j(x_1)\Phi_j(x_2)\hat{k}(x_1,x_2) P_{\mathcal{X}}(x_1) P_{\mathcal{X}}(x_2) dx_1 dx_2, \end{aligned}$$

where

$$\begin{aligned} \hat{k}(x_1,x_2) &= \iint \langle a_1, a_2\rangle P^+(a_1|x_1) P^+(a_2|x_2) da_1 da_2 \\ &= \iint \mathbb{I}[a_1 = a_2] P^+(a_1|x_1) P^+(a_2|x_2) da_1 da_2. \end{aligned} \tag{A.1}$$

Thus Φ^* is a minimizer of $\mathcal{R}(\Phi)$ if Φ^* extracts the top-d eigenfunctions of $\hat{k}(x_1, x_2)$. Combining with Lemma A.1 yields that $k_\Lambda(a, a') = \mathbb{I}[a = a']$. Furthermore, we have $(\Lambda g)(a) = \int g(a') k_\Lambda(a, a') dP_{\mathcal{A}}(a') = g(a) P_{\mathcal{A}}(a)$, as desired.

If all classes have the same size, we have $P_{\mathcal{A}}(a) \equiv c \in (0, 1)$ where c is a constant. Thus $(\Lambda g)(a) = g(a) P_{\mathcal{A}}(a) = c g(a)$, which implies that $T_{P^+} \Lambda T_{P^+}^* = c T_{P^+} T_{P^+}^*$. This concludes that $T_{P^+} \Lambda T_{P^+}^*$ and $T_{P^+} T_{P^+}^*$ share the same top-d eigenfunctions. ■

A.2 Proof of Theorem 2.4

Proof Denote $\mathcal{R}(\Phi, \boldsymbol{W}) = \mathbb{E}_{(X,A)\sim P^+} \left[\frac{1}{\sqrt{P_{\mathcal{A}}(A)}} \|A - \boldsymbol{W}\Phi(X)\|_2^2 \right]$. Assuming without loss of generality that $\mathbb{E}_{(X,A)\sim P^+} \left[\frac{1}{\sqrt{P_{\mathcal{A}}(A)}} \Phi_i \Phi_j \right] = \mathbb{I}[i = j]$; otherwise we can perform Gram-Schmidt process on Φ_i and change the value of $\boldsymbol{W}$ respectively. Thus, it amounts to

minimize

$$\begin{aligned}\mathcal{R}(\Phi, \boldsymbol{W}) &= \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\frac{1}{\sqrt{P_{\mathcal{A}}(A)}}\|A - \boldsymbol{W}\Phi(X)\|_2^2\right] \\ &= \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\frac{1}{\sqrt{P_{\mathcal{A}}(A)}}\|\boldsymbol{W}\Phi(X)\|_2^2\right] \\ &\quad - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\left\langle\frac{A}{\sqrt{P_{\mathcal{A}}(A)}}, \boldsymbol{W}\Phi(X)\right\rangle + \mathop{\mathbb{E}}_{A\sim P_{\mathcal{A}}}\left[\frac{\|A\|_2^2}{\sqrt{P_{\mathcal{A}}(A)}}\right] \\ &= \|\boldsymbol{W}\|_F^2 - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\left\langle\frac{A}{\sqrt{P_{\mathcal{A}}(A)}}, \boldsymbol{W}\Phi(X)\right\rangle + \mathop{\mathbb{E}}_{A\sim P_{\mathcal{A}}}\left[\frac{\|A\|_2^2}{\sqrt{P_{\mathcal{A}}(A)}}\right].\end{aligned}$$

Denote $\boldsymbol{W} = (w_{ij})_{1\le i\le d_A, 1\le j\le d}$. We have

$$\frac{\partial\mathcal{R}}{\partial w_{ij}} = 2w_{ij} - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\frac{A_i}{\sqrt{P_{\mathcal{A}}(A)}}\Phi_j(X)\right],$$

which implies that for a fixed Φ, the minimizer of $\boldsymbol{W}$ satisfies

$$w_{ij} = \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\frac{A_i}{\sqrt{P_{\mathcal{A}}(A)}}\Phi_j(X)\right].$$

Combining the minimizer of $\boldsymbol{W}$ with $\mathcal{R}$, it suffices to maximize

$$\mathcal{R}' = \sum_{i,j}\left[\mathop{\mathbb{E}}_{(X,A)\sim P^+}\frac{A_i}{\sqrt{P_{\mathcal{A}}(A)}}\Phi_j(X)\right]^2 = \iint\sum_j \Phi_j(x_1)\Phi_j(x_2)\hat{k}(x_1,x_2)P_{\mathcal{X}}(x_1)P_{\mathcal{X}}(x_2)dx_1dx_2,$$

where

$$\begin{aligned}\hat{k}(x_1,x_2) &= \iint\frac{\langle a_1,a_2\rangle}{\sqrt{P_{\mathcal{A}}(a_1)P_{\mathcal{A}}(a_2)}}P^+(a_1|x_1)P^+(a_2|x_2)da_1da_2 \\ &= \iint\frac{\mathbb{I}[a_1=a_2]}{\sqrt{P_{\mathcal{A}}(a_1)P_{\mathcal{A}}(a_2)}}P^+(a_1|x_1)P^+(a_2|x_2)da_1da_2 \\ &= \int\frac{P^+(a|x_1)P^+(a|x_2)}{P_{\mathcal{A}}(a)}dy.\end{aligned}$$

Thus Φ^* is a minimizer of $\mathcal{R}(\Phi)$ if Φ^* extracts the top-d eigenfunctions of $\hat{k}(x_1,x_2)$. Combining with Definitions 1.3 and 1.7 yields the desired results. ■

A.3 Proof of Theorem 2.6

Proof If the linear predictor is unbiased, then the proof is the same as Appendix A.1. Consider the case where the linear predictor is biased. Then the pretraining objective can be rewritten as

$$\mathcal{R}(\Phi, \boldsymbol{W}, \boldsymbol{b}) = \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\left\|\tilde{A} - \boldsymbol{W}\tilde{\Phi}(X)\right\|_2^2\right] + \left\|\hat{\boldsymbol{b}}\right\|_2^2,$$

where $\hat{\boldsymbol{b}} = \boldsymbol{W}\mathbb{E}_{X\sim P_{\mathcal{X}}}[\Phi(X)] - \mathbb{E}_{A\sim P_{\mathcal{A}}}[A] + \boldsymbol{b}$. Thus, if Γ and $\boldsymbol{W}$ are fixed, then the optimal $\boldsymbol{b}^* = \mathbb{E}_{A\sim P_{\mathcal{A}}}[A] - \boldsymbol{W}\mathbb{E}_{X\sim P_{\mathcal{X}}}[\Phi(X)]$.

Assuming, without loss of generality, that $\mathbb{E}_{X\sim P_{\mathcal{X}}}[\tilde{\Phi}_i\tilde{\Phi}_j] = \delta_{ij}$; otherwise we can perform Gram-Schmidt process on $\tilde{\phi}_i$ and change $\boldsymbol{W}$ accordingly. Then, it amounts to minimize

$$\mathcal{R}(\Phi, \boldsymbol{W}, \boldsymbol{b}^*) = \|\boldsymbol{W}\|_F^2 - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\left\langle \tilde{A}, \boldsymbol{W}\tilde{\Phi}(X)\right\rangle + \mathop{\mathbb{E}}_{A\sim P_{\mathcal{A}}}\left\|\tilde{A}\right\|_2^2,$$

for which we have

$$\frac{\partial \mathcal{R}}{\partial w_{ij}} = 2w_{ij} - 2\mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\tilde{A}_i\tilde{\Phi}_j(X)\right].$$

Thus, the optimal $\boldsymbol{W}$ is given by $w_{ij}^* = \mathbb{E}_{(X,A)\sim P^+}\left[\tilde{A}_i\tilde{\Phi}_j(X)\right]$. Note that $\mathbb{E}_{A\sim P_{\mathcal{A}}}\left\|\tilde{A}\right\|_2^2$ is a constant. Thus, minimizing $\mathcal{R}(\Phi, \boldsymbol{W}^*, \boldsymbol{b}^*)$ is equivalent to maximizing

$$\begin{aligned}
\mathcal{J} &= \sum_{i,j}\left[\mathop{\mathbb{E}}_{(X,A)\sim P^+}\tilde{A}_i\tilde{\Phi}_j(X)\right]^2 \\
&= \int\sum_j \tilde{\Phi}_j(x_1)\tilde{\Phi}_j(x_2)\langle\tilde{a}_1,\tilde{a}_2\rangle P_{\mathcal{X}}(x_1)P^+(a_1|x_1)P_{\mathcal{X}}(x_2)P^+(a_2|x_2)dx_1da_1dx_2da_2 \\
&= \iint\sum_j \tilde{\Phi}_j(x_1)\tilde{\Phi}_j(x_2)\hat{k}(x_1,x_2)P_{\mathcal{X}}(x_1)P_{\mathcal{X}}(x_2)dx_1dx_2,
\end{aligned}$$

where

$$\hat{k}(x_1,x_2) = \iint\langle\tilde{a}_1,\tilde{a}_2\rangle P^+(a_1|x_1)P^+(a_2|x_2)da_1da_2.$$

Then, we can complete the proof in the same way as Appendix A.1. ■

A.4 Proof of Theorem 2.8

Proof Without loss of generality, suppose $\bar{\Phi} = \boldsymbol{0}$. We have

$$(T_{P^+}f)(u) = \sum_v f(v)\frac{w(u,v)}{D(u)}; \quad \langle T_{P^+}f, g\rangle_{P_{\mathcal{X}}} = \sum_{u,v} f(u)g(v)\frac{w(u,v)}{D_{\text{sum}}} = \langle f, T_{P^+}g\rangle_{P_{\mathcal{X}}},$$

which implies that T_{P^+} is self-adjoint. Therefore, the eigenfunctions of T_{P^+} are the same as those of $T_{P^+}^*T_{P^+}$, with square root eigenvalues.

For the objective of Eqn. (2.4), we have

$$\begin{aligned}
\frac{1}{2}\mathbb{E}_{(u,v)\sim P^+}\left[\|\Phi(u)-\Phi(v)\|_2^2\right] &= \mathop{\mathbb{E}}_{(u,v)\sim P_w}\left[\|\Phi(u)\|_2^2 - \langle\Phi(u),\Phi(v)\rangle\right] \\
&= \sum_{i=1}^d\left(\|\phi_i\|_{P_{\mathcal{X}}}^2 - \langle\phi_i, T_{P^+}\phi_i\rangle_{P_{\mathcal{X}}}\right) \\
&= d - \sum_{i=1}^d\langle\phi_i, T_{P^+}\phi_i\rangle_{P_{\mathcal{X}}}.
\end{aligned}$$

Note that (u,v) and (v,u) can be drawn from P^+ with equal probability. We conclude that Φ extracts the top-d eigenfunctions of T_{P^+}, which are the same as the top-d eigenfunctions of $T_{P^+}^*T_{P^+}$. This implies that $\tilde{\Phi}$ learns the contexture of T_{P^+}. ■

A.5 Proof of Theorem 2.9

Proof

(i) The spectral contrastive loss is

$$\mathcal{R}(\Psi) = \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A,A^+\sim P^+(\cdot|X)}\left[-\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^+)\right\rangle + \frac{1}{2}\mathop{\mathbb{E}}_{A^-\sim P_{\mathcal{A}}}\left[\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^-)\right\rangle^2\right]\right].$$

Suppose $\psi_i = \sum_{j\geq 0} c_{ij}\nu_j$ where ν_j is the ONB of $L^2(P_{\mathcal{A}})$ in Lemma 1.4. Since ν_j is the ONB of $L^2(P_{\mathcal{A}})$ and $\nu_0 \equiv 1$, we can get for $j \geq 1$, $\mathbb{E}_{P_{\mathcal{A}}}[\nu_j(a)] = \delta_{0,j} = 0$. Thus we can get $\tilde{\psi}_i = \psi_i - \mathbb{E}[\psi_i] = \sum_{j\geq 1} c_{ij}\nu_j$.
Denote matrix $\boldsymbol{C} = (c_{ij})_{1\leq i\leq d, j\geq 1}$, matrix $\boldsymbol{B} = (b_{ij}) := \boldsymbol{C}^\top\boldsymbol{C}$, and matrix $\boldsymbol{D} = \mathrm{diag}(s_1^2, s_2^2, \cdots)$ where s_i is the singular value of T_{P^+}. We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A,A^+\sim P^+(\cdot|X)}\left[\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^+)\right\rangle\right] \\
&= \iiint \left\langle \tilde{\Psi}(a), \tilde{\Psi}(a^+)\right\rangle P^+(a|x)P^+(a^+|x)P_{\mathcal{X}}(x)dxdada^+ \\
&= \int \left\langle \int \tilde{\Psi}(a)P^+(a|x)dy, \int \tilde{\Psi}(a^+)P^+(a^+|x)da^+\right\rangle p(x)dx \\
&= \int \left\langle T_{P^+}\tilde{\Psi}(x), T_{P^+}\tilde{\Psi}(x)\right\rangle p(x)dx = \|T_{P^+}\tilde{\Psi}\|_{P_{\mathcal{X}}}^2 \\
&= \sum_i s_i^2 b_{ii};
\end{aligned}$$

and

$$\begin{aligned}
\mathop{\mathbb{E}}_{A,A^-\sim P_{\mathcal{A}}}\left[\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^-)\right\rangle^2\right] &= \iint \left[\sum_{i=1}^d \tilde{\psi}_i(a)\tilde{\psi}_i(a^-)\right]^2 dP_{\mathcal{A}}(a)dP_{\mathcal{A}}(a^-) \\
&= \sum_{1\leq i,j\leq d}\left[\int \tilde{\psi}_i(a)\tilde{\psi}_j(a)dP_{\mathcal{A}}(a)\right]^2 \\
&= \sum_{i,j} b_{ij}^2.
\end{aligned}$$

Thus, we have

$$\mathcal{R}(\Psi) = -\sum_i s_i^2 b_{ii} + \frac{1}{2}\sum_{i,j} b_{ij}^2 = \|\boldsymbol{B} - \boldsymbol{D}\|_F^2 - \|\boldsymbol{D}\|_F^2.$$

So if suffices to minimize $\|\boldsymbol{B} - \boldsymbol{D}\|_F^2$ where $\mathrm{rank}(\boldsymbol{B}) \leq d$. By Eckart-Young-Mirsky Theorem, we know the minimizer of $\boldsymbol{B}$ is $\boldsymbol{B}^* = \mathrm{diag}(s_1^2, \cdots, s_d^2)$. Thus the minimizer of $\boldsymbol{C}$ should be $\boldsymbol{C}^* = \boldsymbol{U}\mathrm{diag}(s_1, \cdots, s_d)$ where $\boldsymbol{U} \in \mathbb{R}^{d\times d}$ is an orthonormal matrix. This indicates the minimizer $\tilde{\Psi}^*$ extracts the top-d singular functions of T_{P^+}, and hence $\tilde{\Phi}^*$ learns the contexture of P^+.

(ii) Non-contrastive learning is done by minimizing

$$\mathcal{R}(\Psi) = \mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}}} \mathop{\mathbb{E}}_{A,A^+\sim P^+(\cdot|X)}\left[\left\|\Psi(A) - \Psi(A^+)\right\|_2^2\right],$$

subject to $\mathrm{Cov}_{P_{\mathcal{A}}}[\Psi] = \boldsymbol{I}$. This amounts to minimizing $F(\Psi) = -\mathbb{E}_{A,A^+}[\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^+)\right\rangle]$, because $\mathcal{R}(\Psi) - 2F(\Psi) = 2$ is a constant under the constraint $\mathrm{Cov}_{P_{\mathcal{A}}}[\Psi] = \boldsymbol{I}$.

Suppose $\psi_i = \sum_{j\geq 0} c_{ij}\nu_j$ where ν_j is the ONB of $L^2(P_{\mathcal{A}})$ in Lemma 1.5. Since $\mathbb{E}_{P_{\mathcal{A}}}[\nu_j(a)] = \delta_{0,j}$, we can get $\tilde{\psi}_i = \psi_i - \mathbb{E}[\psi_i] = \sum_{j\geq 1} c_{ij}\nu_j$.
Using the same calculation as in (i), we have

$$F(\Psi) = -\mathop{\mathbb{E}}_{X\sim P_{\mathcal{X}} A,A^+\sim P^+(\cdot|X)}\left[\left\langle \tilde{\Psi}(A), \tilde{\Psi}(A^+)\right\rangle\right] = -\|T_{P^+}\tilde{\Psi}\|^2_{P_{\mathcal{X}}} = -\sum_i s_i^2 b_{ii}.$$

By $\mathbb{E}_{P_{\mathcal{A}}}\left[\tilde{\psi}_i\tilde{\psi}_j\right] = \delta_{ij}$, we have

$$\sum_i b_{ii} = \sum_{i,j} c_{ij}^2 = d.$$

Since ν_i is an ONB of $L^2(P_{\mathcal{A}})$, $\tilde{\psi}_1, \cdots, \tilde{\psi}_d$ are orthogonal, we have

$$b_{ii} = \sum_{j=1}^d c_{ji}^2 = \sum_{j=1}^d \left\langle \tilde{\psi}_j, \nu_i\right\rangle^2_{P_{\mathcal{A}}} \leq \|\nu_i\|^2_{P_{\mathcal{A}}} = 1. \tag{A.2}$$

Thus, we conclude that

$$\mathcal{L}_{\mathrm{N}}(\Psi) + \sum_{i=1}^d s_i^2 = \sum_{i=1}^d s_i^2(1-b_{ii}) - \sum_{i>d} s_i^2 b_{ii} \geq \sum_{i=1}^d s_d^2(1-b_{ii}) - \sum_{i>d} s_d^2 b_{ii} = 0,$$

which implies that $\mathcal{L}_{\mathrm{N}}(\Psi) \geq -\sum_{i=1}^d s_i^2$. To attain equality, we will have $b_{ii} = 1$ for $i = 1, \cdots, d$, and $b_{ii} = 0$ for $i \geq d+1$. By Eqn. (A.2), we can know Ψ^* extracts the span of $\nu_1, \cdots, \nu_d$, indicating that $\tilde{\Psi}^*$ extracts the top-d singular functions of T_{P^+} and $\tilde{\Phi}^*$ learns the contexture of P^+.

■

A.6 Proof of Theorem 2.11

Proof We first show that when Φ is fixed in SVME, the optimal Ψ that minimizes the objective is $\Psi = T^*_{P^+}\Phi$. The SVME objective is

$$\mathcal{L} = \mathop{\mathbb{E}}_{(X,A)\sim P^+}\left[\|\Phi(X)\|_2^2 + \Psi(A)^\top(\Psi(A) - 2\Phi(X))\right] = \sum_{i=1}^d \left\{\|\phi_i\|^2_{P_{\mathcal{X}}} + \langle\psi_i, \psi_i - 2T^*_{P^+}\phi_i\rangle_{P_{\mathcal{A}}}\right\},$$

which implies that

$$\frac{\partial\mathcal{L}}{\partial\psi_i} = 2\psi_i - 2T^*_{P^+}\phi_i.$$

Setting it to zero yields $\psi_i = T^*_{P^+}\phi_i$. With this, the SVME objective becomes

$$\mathcal{L}(\Phi) = \sum_{i=1}^d \left\{\|\phi_i\|^2_{P_{\mathcal{X}}} - \|T^*_{P^+}\phi_i\|^2_{P_{\mathcal{A}}}\right\} = \sum_{i=1}^d \langle\phi_i, \phi_i - T_{P^+}T^*_{P^+}\phi_i\rangle_{P_{\mathcal{X}}}$$

because $T^*_{P^+}$ is the adjoint of T_{P^+}. This $\mathcal{L}$ is exactly the objective of KISE. Under the orthonormality constraint, minimizing $\mathcal{L}(\Phi)$ is equivalent to maximizing $\sum_{i=1}^d \left\langle \phi_i, T_{k_X^+}\phi_i\right\rangle_{P_{\mathcal{X}}}$.
Then, we can use the proof of Theorem 2.9 (ii) to show that when Φ^* is the optimal solution, $\tilde{\Phi}^*$ learns the contexture of P^+. ■

A.7 Proof of Theorem 2.14

Proof Without loss of generality assume that $T_{k_X^+}$ has at least $d+1$ positive eigenvalues (including λ_0). First we prove a simple result in linear algebra: For any positive definite matrix $\boldsymbol{A}$, if vectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k$ are pairwise orthogonal and satisfy $\boldsymbol{u}_i^\top \boldsymbol{A} \boldsymbol{u}_i = 1$ for all i, then $\|\boldsymbol{u}_1\|_2^2 + \cdots + \|\boldsymbol{u}_k\|_2^2 \le \lambda_1^{-1} + \cdots + \lambda_k^{-1}$, where $0 < \lambda_1 \le \lambda_2 \le \cdots$ are the eigenvalues of $\boldsymbol{A}$. To prove this, let $\boldsymbol{U}$ be the matrix whose columns are $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k$. Then, $\boldsymbol{U}^\top \boldsymbol{U}$ is a diagonal matrix and all elements on the diagonal of $\boldsymbol{U}^\top \boldsymbol{A} \boldsymbol{U}$ are 1. Denote the sorted diagonal elements of $\boldsymbol{U}^\top \boldsymbol{U}$ by $d_1 \ge d_2 \ge \cdots \ge d_k \ge 0$. Let $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{U}^\top$. A freshman linear algebra exercise states that the eigenvalues of $\boldsymbol{Q}^2$ are $d_1^2, \cdots, d_k^2$, and the rest are all zeros. Now consider $\boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{U}^\top \boldsymbol{A} \boldsymbol{U}$. It is easy to see that the diagonal elements of this matrix are $d_1, d_2, \cdots, d_k$. This implies that $\operatorname{Tr}(\boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{U}^\top \boldsymbol{A} \boldsymbol{U}) = \operatorname{Tr}(\boldsymbol{U}^\top \boldsymbol{U}) = d_1 + \cdots + d_k$. Thus, $\operatorname{Tr}(\boldsymbol{Q}^2 \boldsymbol{A}) = \operatorname{Tr}(\boldsymbol{Q}) = d_1 + \cdots + d_k$, because $\operatorname{Tr}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{Tr}(\boldsymbol{B}\boldsymbol{A})$. By von Neumann's trace inequality, there is $\operatorname{Tr}(\boldsymbol{Q}^2 \boldsymbol{A}) \ge \lambda_1 d_1^2 + \cdots + \lambda_k d_k^2$. So by Cauchy-Schwarz inequality, $(d_1 + \cdots + d_k)^2 \le (\lambda_1^{-1} + \cdots + \lambda_k^{-1}) \operatorname{Tr}(\boldsymbol{Q}^2 \boldsymbol{A}) = (\lambda_1^{-1} + \cdots + \lambda_k^{-1})(d_1 + \cdots + d_k)$, which implies that $\|\boldsymbol{u}_1\|_2^2 + \cdots + \|\boldsymbol{u}_k\|_2^2 = d_1 + \cdots + d_k \le \lambda_1^{-1} + \cdots + \lambda_k^{-1}$. The equality is only attained when the Cauchy-Schwarz inequality attains equality, that is $d_i = \lambda_i^{-1}$ for all i.

For simplicity, we assume that the eigenvalues of $T_{k_X^+}$ satisfy $\lambda_1 > \lambda_2 > \cdots > \lambda_d > 0$. If an eigenvalue has more multiplicity, the proof will be the same but much more verbose. Then, $\mu_1, \cdots, \mu_d$ are fixed. Let $\tilde{\phi}_i^* = \sum_k u_{ik} \mu_k$. Then, Eqn. (2.13) is equivalent to the following optimization problem:

$$
\begin{aligned}
\underset{u_{ik}}{\text{maximize}} \quad & \sum_{i=1}^{d} \sum_{k} \lambda_k u_{ik}^2 \\
\text{s.t.} \quad & \forall i \in [d] : \sum_k u_{ik}^2 = 1; \\
& \forall 1 \le i < j \le d : \sum_k \lambda_k u_{ik} u_{jk} = 0.
\end{aligned}
$$

Obviously, for any $\lambda_k = 0$, the optimal u_{ik} should be zero. So without loss of generality, we assume that all λ_k in the above problem are positive. Define matrix $\boldsymbol{U}$ whose (i,k)-th element is $\sqrt{\lambda_k} u_{ik}$. Denote the rows of $\boldsymbol{U}$ by $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d$. Let $\boldsymbol{D} = \operatorname{diag}\{\lambda_1^{-1}, \lambda_2^{-1}, \cdots\}$. Then, the constraints of the above problem become $\boldsymbol{u}_i^\top \boldsymbol{D} \boldsymbol{u}_i = 1$ for all i, and $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d$ are pairwise orthogonal. Thus, by the result we have just proved, the objective of this problem satisfies $\|\boldsymbol{u}_1\|_2^2 + \cdots + \|\boldsymbol{u}_d\|_2^2 \le \lambda_1 + \cdots + \lambda_d$. The equality is only attained when $\|\boldsymbol{u}_i\|_2^2 = \lambda_i$ for all $i \in [d]$, assuming that $\|\boldsymbol{u}_1\|_2 \ge \cdots \ge \|\boldsymbol{u}_d\|_2$. This is only possible when $u_{11}^2 = u_{22}^2 = \cdots = u_{dd}^2 = 1$, which proves the result. ■

A.8 Proof of Theorem 2.12

Proof It suffices to show that $T_\Omega^* T_{P^+} \Lambda T_{P^+}^* T_\Omega$ is the integral kernel operator of

$$
k(\omega, \omega') = \iint k_\Lambda(y, y') P_{Y|\omega}(y|\omega) P_{Y|\omega}(y'|\omega') dy dy'.
$$

After that, we can follow the proof in Appendix A.1 to prove the result.

Since x determines ω, y at the same time, we have $y \perp \omega | x$. This implies that

$$
P(\omega|x) = P(\omega|x, y), \quad P(y|x) = P(y|x, \omega).
$$

By definition, we have

$$(T_\Omega h)(x') = \int h(\omega')P(\omega'|x')d\omega',$$

which implies that

$$\begin{aligned}(T_{P+}^* T_\Omega h)(y') &= \int (T_\Omega h)(x')P_{X|Y}(x'|y')dx' = \iint h(\omega')P(\omega'|x')P_{X|Y}(x'|y')dx'd\omega' \\ &= \iint h(\omega')P(\omega'|x',y')P_{X|Y}(x'|y')dx'd\omega' = \int h(\omega')P_{\omega|Y}(\omega'|y')d\omega'.\end{aligned}$$

Thus, we have

$$\begin{aligned}(\Lambda T_{P+}^* T_\Omega h)(y) &= \int (T_{P+}^* T_\Omega h)(y')k_\Lambda(y,y')P_Y(y')dy' \\ &= \iint h(\omega')P_{\omega|Y}(\omega'|y')k_\Lambda(y,y')P_Y(y')d\omega'dy'.\end{aligned}$$

This implies that

$$\begin{aligned}(T_\Omega^* T_{P+}\Lambda T_{P+}^* T_\Omega h)(\omega) &= \int (T_{P+}\Lambda T_{P+}^* T_\Omega h)(x)P(x|\omega)dx \\ &= \iint (\Lambda T_{P+}^* T_\Omega h)(y)P(y|x)P_{X|\omega}(x|\omega)dydx \\ &= \iint (\Lambda T_{P+}^* T_\Omega h)(y)P(y|x,\omega)P_{X|\omega}(x|\omega)dydx \\ &= \int (\Lambda T_{P+}^* T_\Omega h)(y)P_{Y|\omega}(y|\omega)dy \\ &= \iiint h(\omega')P_{\omega|Y}(\omega'|y')k_\Lambda(y,y')P_Y(y')P_{Y|\omega}(y|\omega)d\omega'dy'dy \\ &= \iiint h(\omega')k_\Lambda(y,y')P_{Y|\omega}(y|w)P_{Y|\omega}(y'|w')P_\omega(\omega')dydy'd\omega',\end{aligned}$$

as desired. ■

Appendix B

Proofs for Chapter 3

B.1 Proof of Theorem 3.2

Proof Let $f^* = \sum u_i \mu_i$, and $g^* = \sum s_i u_i \nu_i$. For this f^*, the maximum in Eqn. (3.1) is attained by g^*, so it suffices to show that g^* satisfies Eqn. (3.3). By Bayes rule, we have

$$P^+(A'|A=a) = \int P^+(A'|X=x)P^+(x|A'=a)dx,$$

which implies that $P^+(A'=a'|A=a) = k_A^+(a,a')P_{\mathcal{A}}(a')$. Therefore, we have

$$\mathbb{E}_{X\sim P_{\mathcal{X}}}\mathbb{E}_{A,A'\sim P^+(\cdot|X)}[g^*(A)g^*(A')] = \mathbb{E}_{A\sim P_{\mathcal{A}}}\mathbb{E}_{A'\sim P(\cdot|A)}[g^*(A)g^*(A')]$$
$$= \mathbb{E}_A\left[g^*(A)\int g^*(a')P(a'|A)da'\right] = \mathbb{E}_A\left[g^*(A)\int g^*(a')k_A^+(a,a')P_{\mathcal{A}}(a')da'\right] = \left\langle g^*, T_{k_A^+}g^*\right\rangle_{P_{\mathcal{A}}}.$$

Since $T_{k_A^+}g^* = T^*_{P^+}T_{P^+}g^* = \sum s_i^3 u_i \nu_i$, Eqn. (3.3) is equivalent to $\sum(s_i^2 - s_i^4)u_i^2 \le 2\epsilon\sum s_i^2 u_i^2$. Meanwhile, we have $\sum s_i^2 u_i^2 \ge (1-\epsilon)^2\sum u_i^2 \ge (1-2\epsilon)\sum u_i^2$. By Cauchy-Schwarz inequality, we have $(\sum s_i^4 u_i^2)(\sum u_i^2) \ge (\sum s_i^2 u_i^2)^2 \ge (1-2\epsilon)(\sum u_i^2)(\sum s_i^2 u_i^2)$, which proves Eqn. (3.3). ■

B.2 Proof of Theorem 3.4

Proof Since span(Φ) is at most rank-d, thus there exists $f_1 \in \text{span}\{\mu_1, \cdots, \mu_{d+1}\}$ with $\|f_1\|_{P_{\mathcal{X}}} = 1$ that is orthogonal to span(Φ). Thus there exists $f_1, f_2 \in \text{span}\{\mu_1, \cdots, \mu_{d+1}\}$ with $\|f_1\|_{P_{\mathcal{X}}} = \|f_2\|_{P_{\mathcal{X}}} = 1$, f_1 is orthogonal to span(Φ) and $f_2 \in$ span(Φ) (thus $f_1 \perp f_2$), and $\mu_1 \in \text{span}\{f_1, f_2\}$. Suppose $\mu_1 = \alpha_1 f_1 + \alpha_2 f_2$ (without loss of generosity, assuming $\alpha_1, \alpha_2 \in [0,1]$) and denote $f_0 = \alpha_2 f_1 - \alpha_1 f_2$. Then $\|f_0\|_{P_{\mathcal{X}}} = 1$ and $\langle \mu_1, f_0\rangle_{P_{\mathcal{X}}} = 0$. Since $f_1, f_2 \in \text{span}\{\mu_1, \cdots, \mu_{d+1}\}$, we have $f_0 \in \text{span}\{\mu_2, \cdots, \mu_{d+1}\}$ and thus $\mathbb{E}[f_0] = 0$.

Consider $f = \beta_1\mu_1 + \beta_2 f_0 \in \mathcal{F}_\epsilon(P^+)$ where $\beta_1^2 + \beta_2^2 = 1$, $\beta_1, \beta_2 \in [0,1]$. Denote $f = \sum_{i\ge 1} u_i\mu_i$ Then, we have $\sum_i u_i^2 = 1$ and

$$\beta_2^2 \le \frac{s_1^2 - (1-\epsilon)^2}{s_1^2 - s_{d+1}^2} \implies \sum_{i\ge 1} s_i^2 u_i^2 \ge s_1^2\beta_1^2 + s_{d+1}^2\beta_2^2 = s_1^2 - (s_1^2 - s_{d+1}^2)\beta_2^2 \ge (1-\epsilon)^2\sum_i u_i^2.$$

Since $f = (\alpha_1\beta_1 + \alpha_2\beta_2)f_1 + (\alpha_2\beta_1 - \alpha_1\beta_2)f_2$, the approximation error of f is $(\alpha_1\beta_1 + \alpha_2\beta_2)^2$. Define a function $F(\alpha_1) = \alpha_1\beta_1 + \alpha_2\beta_2 = \alpha_1\beta_1 + \sqrt{1-\alpha_1^2}\beta_2$ ($\alpha_1 \in [0,1]$). We

can show that dis function first increases and then decreases with α_1. Thus, $F(\alpha_1)^2 \geq \min\{F(0)^2, F(1)^2\} = \min\{\beta_1^2, \beta_2^2\}$. If $\beta_2^2 = \frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2} \leq \frac{1}{2}$, then the approximation error is always at least $\frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}$. To attain this lower bound, we must have $\sum_{i\geq 1} s_i^2 u_i^2 = s_1^2\beta_1^2 + s_{d+1}^2\beta_2^2$. This implies that $f_1 = \mu_{d+1}$, indicating that $\text{span}(\phi_1, \cdots, \phi_d) = \text{span}(\mu_1, \cdots, \mu_d)$. Thus, Φ must learn the contexture of T_{P^+}.

On the other hand, if Φ learns the contexture, then the approximation of Φ on f will be $A := \sum_{i\geq d+1} u_i^2$. Then, we have

$$(1-\epsilon)^2 \leq \sum_{i\geq 1} s_i^2 u_i^2 \leq s_1^2 \sum_{i=1}^{d} u_i^2 + s_{d+1}^2 \sum_{i\geq d+1} u_i^2 = s_1^2 - (s_1^2 - s_{d+1}^2)A,$$

and this implies that

$$A = \min_{\boldsymbol{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} \left\|\boldsymbol{w}^\top \Phi + b - f\right\|_{P_{\mathcal{X}}}^2 \leq \frac{s_1^2 - (1-\epsilon)^2}{s_1^2 - s_{d+1}^2}.$$

When $u_1^2 = 1 - \frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}, u_{d+1}^2 = \frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}$, the equality holds. Thus, the lower bound $\frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}$ is attained if and only if Φ learns the contexture.

The converse part is obvious because for any Φ we can simply choose $f = \beta_1\mu_1 + \beta_2 f_0$ as defined above. Then, we have $f \in \mathcal{F}_\epsilon(P^+)$, and the approximation error of Φ on this f is at least $\frac{s_1^2-(1-\epsilon)^2}{s_1^2-s_{d+1}^2}$. ■

Appendix C

Proofs for Chapter 4

C.1 Proof of Theorem 4.4

Proof By the proof of Theorem 2.11, $\mathcal{R}_j$ can be written as

$$\mathcal{R}_j = d - \mathbb{E}_{A_j} \left\| \left(T^*_{P^+} \tilde{\Phi} \right)(A_j) \right\|_2^2 = d - \sum_{i=1}^{d} \left\langle \tilde{\phi}_i, T_{k_j} \tilde{\phi}_i \right\rangle_{P_{\mathcal{X}}}.$$

Therefore, the weighted sum $\mathcal{L} = \sum_j w_j \mathcal{R}_j$ is equivalent to

$$\mathcal{L} = \sum_{j=1}^{r} w_j \left(d - \sum_{i=1}^{d} \left\langle \tilde{\phi}_i, T_{k_j} \tilde{\phi}_i \right\rangle_{P_{\mathcal{X}}} \right) = d - \sum_{i=1}^{d} \left\langle \tilde{\phi}_i, \sum_{j=1}^{r} \left(w_j T_{k_j} \right) \tilde{\phi}_i \right\rangle_{P_{\mathcal{X}}},$$

where $\sum_j w_j T_{k_j}$ is equal to the integral operator of the linearly combined kernel $\sum_j w_j k_j$. Thus, minimizing this $\mathcal{L}$ subject to the orthonormality constraint will make $\tilde{\Phi}$ learn the contexture, as shown in Theorem 2.11. ■

C.2 Proof of Theorem 4.9

For ease of reading and better use of notations, we restate our algorithm in Algorithm 6 and use the notation defined there in our proof. It is easy to verify that they are equivalent. We first prove the following lemma.

Lemma C.1. *Suppose $\mathcal{R}_k^t \leq C$ holds for all t, k, for some constant C. If $\eta C < 1$, then for any $\boldsymbol{w} \in \Delta^r$,*

$$\sum_{t=1}^{T} \left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t \right) - \sum_{t=1}^{t} l^t \leq C^2 T \eta + \frac{1}{\eta} \log r. \tag{C.1}$$

Moreover, suppose $T > \log r$ and $\eta = \frac{\sqrt{\log r}}{C\sqrt{T}}$, Eqn. (C.1) *becomes*

$$\sum_{t=1}^{T} \left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t \right) - \sum_{t=1}^{T} l^t \leq 2C\sqrt{T \log r}.$$

Algorithm 6 Solving the game Eqn. (4.1) (Rewritten)

Input: Embedding dimension d, priors $P_1^+, \cdots, P_r^+$, step size η

1: Initialize: $\mathbf{L}^0 \leftarrow [0, \cdots, 0]$, $\Phi : \mathcal{X} \to \mathbb{R}^d$, $\Psi_k : \mathcal{A}_k \to \mathbb{R}^d$ for $k \in [r]$

2: **for** $t = 1, \cdots, T$ **do**

3: $W^t = \sum_k \exp(\eta \mathbf{L}_i^{t-1})$; $\mathrm{w}_k^t = \frac{\exp(\eta \mathbf{L}_k^{t-1})}{W^t}$

4: $\Phi^t, \Psi_1^t, \cdots, \Psi_r^t \leftarrow \arg\min \sum_k \mathrm{w}_k^t \mathcal{R}_k$, and get corresponding loss $\mathcal{R}_k^t$, $l^t = \sum_k \mathrm{w}_k^t \mathcal{R}_k^t$

5: Update the loss vector $\mathbf{L}^t \leftarrow \mathbf{L}^{t-1} + \mathcal{R}^t$

Proof Consider the following potential function:

$$\Omega(t) = \frac{1}{\eta} \log(W^t) = \frac{1}{\eta} \log \left(\sum_{k=1}^{r} \exp(\eta \mathbf{L}_k^t) \right).$$

Since $\mathrm{e}^x \leq 1 + x + x^2$ when $x \leq 1$, thus we have

$$\begin{aligned}
\Omega(t) - \Omega(t-1) &= \frac{1}{\eta} \log \frac{W^t}{W^{t-1}} = \frac{1}{\eta} \log \left(\sum_{k=1}^{r} \mathrm{w}_k^t \exp(\eta \mathcal{R}_k^t) \right) \\
&\leq \frac{1}{\eta} \log \left(\sum_{k=1}^{r} \mathrm{w}_k^t [1 + \eta \mathcal{R}_k^t + (\eta \mathcal{R}_k^t)^2] \right) \\
&= \frac{1}{\eta} \log \left(1 + \eta \sum_{k=1}^{r} \mathrm{w}_k^t [\mathcal{R}_k^t + \eta (\mathcal{R}_k^t)^2] \right) \\
&\leq \sum_{k=1}^{r} \mathrm{w}_k^t [\mathcal{R}_k^t + \eta (\mathcal{R}_k^t)^2] \\
&= \sum_{k=1}^{r} \mathrm{w}_k^t \mathcal{R}_k^t + \eta \sum_{k=1}^{r} \mathrm{w}_k^t (\mathcal{R}_k^t)^2 \\
&\leq \sum_{k=1}^{r} \mathrm{w}_k^t \mathcal{R}_k^t + C^2 \eta = l^t + C^2 \eta.
\end{aligned}$$

Summing over $t = 1, \cdots, T$ and we can get

$$\Omega(T) - \Omega(0) \leq \sum_{t=1}^{t} l^t + C^2 T \eta. \tag{C.2}$$

On the other hand, we know that for any $k \in [r]$, there is

$$\Omega(T) \geq \mathbf{L}_k^T = \sum_{t=1}^{T} \mathcal{R}_k^t.$$

Thus we can get for any $\boldsymbol{w} \in \Delta^r$,

$$\Omega(T) \geq \sum_{k=1}^{r} w_k \mathbf{L}_k^T = \sum_{t=1}^{T} \left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t \right). \tag{C.3}$$

Since $\Omega(0) = \frac{1}{\eta} \log r$, combining (C.2) and (C.3) yields

$$\sum_{t=1}^{T} \left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t \right) - \sum_{t=1}^{t} l^t \leq C^2 T \eta + \frac{1}{\eta} \log r$$

for any $w \in \Delta^r$.

Moreover, take $\eta = \frac{\sqrt{\log r}}{C\sqrt{T}} < \frac{1}{C}$ and Eqn. (C.1) becomes

$$\sum_{t=1}^{T}\left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t\right) - \sum_{t=1}^{T} l^t \leq 2C\sqrt{T\log r},$$

as desired. ■

Now, we finish the proof of Theorem 4.9.

Proof For any $\hat{\Phi}, \hat{\Psi}_1, \cdots, \hat{\Psi}_r$, by the optimality of $\Phi^t, \Psi_1^t, \cdots, \Psi_r^t$, we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} l_t &= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{r} \mathrm{w}_k^t \mathcal{R}_k^t \leq \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{r} \mathrm{w}_k^t \mathbb{E}_{X\sim P_{\mathcal{X}}}\mathbb{E}_{Y\sim P_k^+(\cdot|X)}\|\hat{\Phi}(X) - \hat{\Psi}_k(Y)\|_2^2 \\
&= \sum_{k=1}^{r} \frac{\sum_{t=1}^{T} \mathrm{w}_k^t}{T}\mathbb{E}_{X\sim P_{\mathcal{X}}}\mathbb{E}_{Y\sim P_k^+(\cdot|X)}\|\hat{\Phi}(X) - \hat{\Psi}_k(Y)\|_2^2 \\
&\leq \max_{w\in\Delta^r}\sum_{k=1}^{r} w_k \mathbb{E}_{X\sim P_{\mathcal{X}}}\mathbb{E}_{Y\sim P_k^+(\cdot|X)}\|\hat{\Phi}(X) - \hat{\Psi}_k(Y)\|_2^2.
\end{aligned}$$

This implies that

$$\frac{1}{T}\sum_{t=1}^{T} l_t \leq \min_{\Phi,\Psi}\max_{w\in\Delta^r}\sum_{k=1}^{r} w_k \mathcal{R}_k.$$

On the other hand, for any $w \in \Delta^r$,

$$\mathcal{L}(w) = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{k=1}^{r} w_k \mathcal{R}_k^t\right).$$

Applying Lemma C.1, we can get

$$\mathcal{L}(w) \leq \frac{1}{T}\sum_{t=1}^{T} l_t + \frac{2C\sqrt{\log r}}{\sqrt{T}} \leq \min_{\Phi,\Psi}\max_{w\in\Delta^r}\sum_{k=1}^{r} w_k \mathcal{R}_k + \frac{2C\sqrt{\log r}}{\sqrt{T}} = \mathcal{L}^* + \frac{2C\sqrt{\log r}}{\sqrt{T}}$$

for any $w \in \Delta^r$, completing our proof. ■

Appendix D

Proofs for Chapter 5

D.1 Context Complexity of Masking

Example D.1. *Consider a* random masking *augmentation, i.e. for any $x \in \mathcal{X}$, each coordinate $x^{(i)}$ is randomly and independently masked to be 0 (i.e. 0 denotes the `[MASK]` token) with probability $\alpha \in (0,1)$. Then, its context complexity is given by $\kappa_r^2 = (2-\alpha)^{d_\mathcal{X}}$.*

Proof We know that $\kappa^2 \geq \int \frac{P^+(a|x)^2}{P_\mathcal{A}(a)} da$, whose right-hand side is a constant for all x by symmetry. Given an a, suppose a has r coordinates masked and $(d_\mathcal{X} - r)$ coordinates unmasked. Then, there are 2^r possible x that can be masked to become a. For each of these x, $P^+(a|x) = \alpha^r(1-\alpha)^{d_\mathcal{X}-r}$. So $p(a) = \int P^+(a|x)P_\mathcal{X}(x)dx = 2^{r-d_\mathcal{X}}\alpha^r(1-\alpha)^{d_\mathcal{X}-r}$. Thus, we have

$$\begin{aligned}
\kappa^2 &= \int \frac{P^+(a|x)^2}{P_\mathcal{A}(a)} da = \sum_{r=0}^{d_\mathcal{X}} \binom{d_\mathcal{X}}{r} \frac{\alpha^{2r}(1-\alpha)^{2d_\mathcal{X}-2r}}{2^{r-d_\mathcal{X}}\alpha^r(1-\alpha)^{d_\mathcal{X}-r}} \\
&= \sum_{r=0}^{d_\mathcal{X}} \binom{d_\mathcal{X}}{r} \alpha^r(2-2\alpha)^{d_\mathcal{X}-r} \\
&= (\alpha + 2 - 2\alpha)^{d_\mathcal{X}} = (2-\alpha)^{d_\mathcal{X}},
\end{aligned}$$

which completes the proof. ■

Example D.2. *Consider* random block masking, *i.e. masking $x^{(i)}, x^{(i+1)}, \cdots, x^{(i+r-1)}$ for $r = \lceil \alpha d_\mathcal{X} \rceil$ and a uniformly random $i \in [d_\mathcal{X} - r]$, for any $x \in \mathcal{X}$. Then, $\kappa_c^2 \leq [2^{(1-\alpha)}]^{d_\mathcal{X}}$.*

Proof For any a, we have $P_\mathcal{A}(a) = \frac{1}{d_\mathcal{X}-r+1}\frac{1}{2^{d_\mathcal{X}-r}}$, and $P^+(a|x) = \frac{1}{d_\mathcal{X}-r+1}$ if a is a masked version of x. So there always is $\frac{P^+(a|x)}{P_\mathcal{A}(a)} = 2^{d_\mathcal{X}-r} \leq 2^{(1-\alpha)d_\mathcal{X}}$. Thus, we have $\kappa^2 \leq 2^{(1-\alpha)d_\mathcal{X}}$. ■

Example D.3. *Consider* random block masking with flipping, *where for any $x \in \mathcal{X}$, first mask $x^{(i)}, \cdots, x^{(i+r-1)}$ to be 0 for $r = \lceil \alpha d_\mathcal{X} \rceil$ and a uniformly random $i \in [d_\mathcal{X} - r]$, then randomly flip the sign of each remaining coordinate independently with probability $\frac{\alpha}{2}$. Then, its context complexity is bounded by $\kappa_b^2 \leq \left[(\alpha^2 - 2\alpha + 2)^{(1-\alpha/2)}\right]^{d_\mathcal{X}}$.*

Proof For any a, we have $P_\mathcal{A}(a) = \frac{1}{d_\mathcal{X}-r+1}\frac{1}{2^{d_\mathcal{X}-r}}$. Suppose a is a masked version of x, and among the unmasked $(d_\mathcal{X} - r)$ coordinates, a and x have k disagreeing coordinates. For a given k, there are $(d_\mathcal{X} - r + 1)\binom{d_\mathcal{X}-r}{k}$ possible a, and we have $P^+(a|x) = \frac{1}{d_\mathcal{X}-r+1}(\frac{\alpha}{2})^k(1-$

$\frac{\alpha}{2})^{d_{\mathcal{X}}-r-k}$. Thus, we have

$$\begin{aligned}
\int \frac{P^+(a|x)^2}{P_{\mathcal{A}}(a)} da &= \sum_{k=0}^{d_{\mathcal{X}}-r} (d_{\mathcal{X}}-r+1)\binom{d_{\mathcal{X}}-r}{k} \frac{\frac{1}{(d_{\mathcal{X}}-r+1)^2}(\frac{\alpha}{2})^{2k}(1-\frac{\alpha}{2})^{2d_{\mathcal{X}}-2r-2k}}{\frac{1}{d_{\mathcal{X}}-r+1}\frac{1}{2^{d_{\mathcal{X}}-r}}} \\
&= \sum_{k=0}^{d_{\mathcal{X}}-r} \binom{d_{\mathcal{X}}-r}{k} 2^{d_{\mathcal{X}}-r} \left(\frac{\alpha^2}{4}\right)^k \left(1-\alpha+\frac{\alpha^2}{4}\right)^{d_{\mathcal{X}}-r-k} \\
&= 2^{d_{\mathcal{X}}-r}\left(\frac{\alpha^2}{4}+1-\alpha+\frac{\alpha^2}{4}\right)^{d_{\mathcal{X}}-r} \\
&\le \left(\alpha^2-2\alpha+2\right)^{d_{\mathcal{X}}-r} \le \left(\alpha^2-2\alpha+2\right)^{(1-\alpha/2)d_{\mathcal{X}}},
\end{aligned}$$

which proves the bound. ■

D.2 Proof of Lemma 5.10

Proof Let $f_i = \sum_j u_{ij} s_j \mu_j$, and $\boldsymbol{U} = (u_{ij}) = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d]$. $\boldsymbol{U}$ is a matrix with d columns and infinitely many rows. Then, since $\langle f_i, f_j \rangle_{\mathcal{H}_k} = \mathbb{I}[i=j]$, we have $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_d$. Let $\boldsymbol{M}(x) = [s_1\mu_1(x), s_2\mu_2(x), \cdots]$, and for a set of samples $S = \{x_1, \cdots, x_m\}$ denote $\boldsymbol{M}_j = \boldsymbol{M}(x_j)$. Then, we have

$$\begin{aligned}
\hat{\mathfrak{R}}_S(\mathcal{F}_d) &= \mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\sup_{F\in\mathcal{F}_d} \frac{1}{m}\sum_{j=1}^m \sigma_i F(x_j)\right] \\
&\le \mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\sup_{\boldsymbol{U}:\boldsymbol{U}^\top\boldsymbol{U}=\boldsymbol{I}_d} \left|\frac{1}{m}\sum_{j=1}^m\sum_{i=1}^d \sigma_j \boldsymbol{u}_i^\top \boldsymbol{M}_j \boldsymbol{M}_j^\top \boldsymbol{u}_i\right|\right] \\
&= \mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\sup_{\boldsymbol{U}:\boldsymbol{U}^\top\boldsymbol{U}=\boldsymbol{I}_d} \left|\mathrm{Tr}\left\{\boldsymbol{U}^\top\left(\frac{1}{m}\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right)\boldsymbol{U}\right\}\right|\right] \\
&= \mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\sup_{\boldsymbol{U}:\boldsymbol{U}^\top\boldsymbol{U}=\boldsymbol{I}_d} \left|\mathrm{Tr}\left\{\left(\frac{1}{m}\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right)\boldsymbol{U}\boldsymbol{U}^\top\right\}\right|\right],
\end{aligned}$$

where $\sigma_1, \cdots, \sigma_m$ are Rademacher variables, which are *i.i.d.* uniform random variables taking values in $\{-1, +1\}$. Let $\beta_1 \ge \beta_2 \ge \cdots$ be the singular values of $\frac{1}{m}\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top$. For any x, $\boldsymbol{M}(x)^\top \boldsymbol{M}(x) = \sum s_i^2 \mu_i^2(x) \le \kappa^2$, which implies that $\boldsymbol{M}_j^\top \boldsymbol{M}_j \le \kappa^2$.

For any $\boldsymbol{U}$, the singular values of $\boldsymbol{U}\boldsymbol{U}^\top$ are d ones and lots of zeros. Moreover, $\left\|\frac{1}{m}\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right\|_F^2 = \sum_{i=1}^\infty \beta_i^2$. So by von Neumann's trace inequality, we have

$$\sup_{\boldsymbol{U}:\boldsymbol{U}^\top\boldsymbol{U}=\boldsymbol{I}_d} \left|\mathrm{Tr}\left\{\left(\frac{1}{m}\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right)\boldsymbol{U}\boldsymbol{U}^\top\right\}\right| \le \sum_{i=1}^d \beta_i \le \sqrt{d\sum_{i=1}^d \beta_i^2} \le \frac{\sqrt{d}}{m}\left\|\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right\|_F.$$

Thus, for any S, we have

$$\begin{aligned}
\hat{\mathfrak{R}}_S(\mathcal{F}_d) &\le \frac{\sqrt{d}}{m}\mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\left\|\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right\|_F\right] \\
&= \frac{\sqrt{d}}{m}\mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\mathrm{Tr}\left\{\left(\sum_{j=1}^m \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right)^\top\left(\sum_{l=1}^m \sigma_l \boldsymbol{M}_l\boldsymbol{M}_l^\top\right)\right\}^{1/2}\right] \\
&\le \frac{\sqrt{d}}{m}\sqrt{\mathbb{E}_{\sigma_1,\cdots,\sigma_m}\left[\mathrm{Tr}\left(\sum_{j,l=1}^m \sigma_j\sigma_l \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_l\boldsymbol{M}_l^\top\right)\right]} \qquad \text{(Jensen)} \\
&= \frac{\sqrt{d}}{m}\sqrt{\mathrm{Tr}\left(\sum_{j,l=1}^m \mathbb{E}[\sigma_j\sigma_l] \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_l\boldsymbol{M}_l^\top\right)} \\
&= \frac{\sqrt{d}}{m}\sqrt{\mathrm{Tr}\left(\sum_{j=1}^m \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_j\boldsymbol{M}_j^\top\right)} = \frac{\sqrt{d}}{m}\sqrt{\sum_{j=1}^m \left(\boldsymbol{M}_j^\top\boldsymbol{M}_j\right)^2} \le \frac{\sqrt{d}}{\sqrt{m}}\kappa^2.
\end{aligned}$$

Since $\mathfrak{R}_m(\mathcal{F}_d) = \mathbb{E}_S[\hat{\mathfrak{R}}_S(\mathcal{F}_d)]$, we have $\mathfrak{R}_m(\mathcal{F}_d) \le \frac{\sqrt{d}}{\sqrt{m}}\kappa^2$. ■

D.3 Proof of Lemma 5.13

Proof Let $f_1 = \sum u_i s_i \mu_i$ and $f_2 = \sum v_i s_i \mu_i$. Let $\boldsymbol{u} = [u_1, u_2, \cdots]$ and $\boldsymbol{v} = [v_1, v_2, \cdots]$. Then, $\|\boldsymbol{u}\|_2 \le 1$ and $\|\boldsymbol{v}\|_2 \le 1$. For any $S = \{x_1, \cdots, x_n\}$, let $\boldsymbol{M}(x) = [s_1\mu_1(x), s_2\mu_2(x)]$ and $\boldsymbol{M}_j = \boldsymbol{M}(x_j)$. Then, we have

$$\begin{aligned}
\hat{\mathfrak{R}}_S(\mathcal{F}) &\le \mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\sup_{\|\boldsymbol{u}\|_2\le 1, \|\boldsymbol{v}\|_2\le 1}\left|\frac{1}{n}\sum_{j=1}^n \sigma_j \boldsymbol{u}^\top \boldsymbol{M}_j\boldsymbol{M}_j^\top \boldsymbol{v}\right|\right] \\
&\le \frac{1}{n}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\left\|\sum_{j=1}^n \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right\|_2\right] \le \frac{1}{n}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\left\|\sum_{j=1}^n \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right\|_F\right] \\
&= \frac{1}{n}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\mathrm{Tr}\left\{\left(\sum_{j=1}^n \sigma_j \boldsymbol{M}_j\boldsymbol{M}_j^\top\right)^\top\left(\sum_{l=1}^n \sigma_l \boldsymbol{M}_l\boldsymbol{M}_l^\top\right)\right\}^{1/2}\right] \\
&\le \frac{1}{n}\sqrt{\mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\mathrm{Tr}\left\{\sum_{j,l=1}^n \sigma_j\sigma_l \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_l\boldsymbol{M}_l^\top\right\}\right]} \qquad \text{(Jensen)} \\
&= \frac{1}{n}\sqrt{\mathrm{Tr}\left\{\sum_{j,l=1}^n \mathbb{E}[\sigma_j\sigma_l] \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_l\boldsymbol{M}_l^\top\right\}} = \frac{1}{n}\sqrt{\mathrm{Tr}\left\{\sum_{j=1}^n \boldsymbol{M}_j\boldsymbol{M}_j^\top\boldsymbol{M}_j\boldsymbol{M}_j^\top\right\}} \le \frac{1}{n}\sqrt{n\kappa^4}.
\end{aligned}$$

Here the first line is not equality because of the absolute value. Since this holds for any S, we have $\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_s[\hat{\mathfrak{R}}_S(\mathcal{F})] \le \frac{\kappa^2}{\sqrt{n}}$.

By [148, Theorem 4.10], for any $\delta \in (0,1)$, with probability at least $1-\delta$, both of the following hold simultaneously for any $f \in \mathcal{F}$:

$$\begin{cases} \left| \frac{1}{n}\sum_{i=1}^{n} f(\tilde{x}_i) - \mathbb{E}_{X\sim P_{\mathcal{X}}}[f(X)] \right| \le \frac{\kappa^2}{\sqrt{n}}\left(2+\sqrt{2\log\frac{2}{\delta}}\right); \\ \left| \frac{1}{n}\sum_{i=1}^{m} f(x_i) - \mathbb{E}_{X\sim P_{\mathcal{X}}}[f(X)] \right| \le \frac{\kappa^2}{\sqrt{m}}\left(2+\sqrt{2\log\frac{2}{\delta}}\right). \end{cases}$$

For any unit vector $\boldsymbol{u} \in \mathbb{R}^d$, let $f_{\boldsymbol{u}}(x) = \boldsymbol{u}^\top \Phi(x)$. Then, $\|f_{\boldsymbol{u}}\|_{\mathcal{H}_k} = 1$, so $f_{\boldsymbol{u}}^2 \in \mathcal{F}$. Moreover, we have

$$\sum_{i=1}^{m} f_{\boldsymbol{u}}(x_i)^2 = \|\boldsymbol{G}[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d]\boldsymbol{u}\|_2^2 = \sum_{i=1}^{d} \lambda_i u_i^2 \ge \lambda_d.$$

Thus, using the above inequalities, assuming that $m \ge n$, we obtain

$$\frac{1}{n}\sum_{j=1}^{n} f(\tilde{x}_j)^2 \ge \frac{\lambda_d}{m} - \frac{\kappa^2}{\sqrt{n}}\left(4+2\sqrt{2\log\frac{2}{\delta}}\right),$$

which implies the result since $\|\boldsymbol{\Phi}\boldsymbol{u}\|_2^2 = \sum_{j=1}^{n} f(\tilde{x}_j)^2$. ■

D.4 Proof of Corollary 5.14

Proof Denote $F = f^* - f_\Phi$. Then, $\boldsymbol{y} - \boldsymbol{y}_\Phi = [F(\tilde{x}_1), \cdots, F(\tilde{x}_n)]$, and $\frac{F^2}{\|F\|_{\mathcal{H}_k}^2} \in \mathcal{F}$, where $\mathcal{F}$ was defined in Lemma 5.13. Therefore, by Lemma 5.13, we have

$$\frac{1}{n}\sum_{j=1}^{n} F(\tilde{x}_j) \le \mathbb{E}_{X\sim P_{\mathcal{X}}}[F(X)^2] + \|F\|_{\mathcal{H}_k}^2 \frac{\kappa^2}{\sqrt{n}}\left(2+\sqrt{2\log\frac{2}{\delta}}\right),$$

as desired. ■

D.5 Proof of Theorem 5.22

Proof Let $\mu_1, \mu_2, \cdots$ be the eigenfunctions of T_k. First, let us show that $\mathcal{H}_t$ must be an RKHS. since μ_1 is the common top-1 eigenfunction of T_{k^p} for all $p \ge 1$, we have $r_{k^p}(\mu_1) \ge r_{k^p}(f)$ for all $f \in \mathcal{H}_k$. By the condition of preserving relative smoothness, this implies that for all $f \in \mathcal{H}_t \subset \mathcal{H}_k$, we have $r_t(\mu_1) \ge r_t(f)$. Let $C_0 = r_t(\mu_1)$. Then, for any $f \in \mathcal{H}_t$, we have $\|f\|_{P_{\mathcal{X}}} \le \sqrt{C_0}\|f\|_{\mathcal{H}_t}$. In other words, $\|\cdot\|_{\mathcal{H}_t}$ is a stronger norm than $\|\cdot\|_{P_{\mathcal{X}}}$ on $\mathcal{H}_t$. Thus, for any sequence $(h_i) \in \mathcal{H}_t$ such that $\|h_i - h\|_{\mathcal{H}_k} \to 0$: first, we have $h \in \mathcal{H}_k$ because $\mathcal{H}_k$ is a Hilbert space; second, we have $\|h_i - h\|_{P_{\mathcal{X}}} \to 0$. Similarly, if $\|h_i - h'\|_{\mathcal{H}_k} \to 0$, then $\|h_i - h'\|_{P_{\mathcal{X}}} \to 0$.

Consider the inclusion map $I : \mathcal{H}_t \to \mathcal{H}_k$, where $Ih = h$. For any sequence $(h_i) \in \mathcal{H}_t$ such that $\|h_i - h\|_{\mathcal{H}_k} \to 0$ and $\|h_i - h'\|_{\mathcal{H}_k} \to 0$, h_i converges to both h and h' under $\|\cdot\|_{P_{\mathcal{X}}}$, so we must have $h' = h = Ih$. This means that the graph of I is closed, so the closed

graph theorem [18, Chapter 2] guarantees that I must be a bounded operator, meaning that there exists a constant C such that $\|f\|_{\mathcal{H}_k} \leq C\|f\|_{\mathcal{H}_t}$ for all $f \in \mathcal{H}_t$.

Let $\delta_x : f \mapsto f(x)$ be the evaluation functional at point x. Since $\mathcal{H}_k$ is an RKHS, there exists a constant $M_x > 0$ such that $|f(x)| \leq M_x\|f\|_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$. Thus, for any $f \in \mathcal{H}_t \subset \mathcal{H}_k$, we have $|f(x)| \leq M_x\|f\|_{\mathcal{H}_k} \leq M_xC\|f\|_{\mathcal{H}_t}$. Thus, by Proposition 5.20, $\mathcal{H}_t$ is also an RKHS. Let k_s be the reproducing kernel of $\mathcal{H}_t$. From now on, we will use $\mathcal{H}_{k_s}$ to denote $\mathcal{H}_t$.

Second, we prove by induction that $\mu_1, \cdots, \mu_d$ are the top-d eigenfunctions of T_{k_s}, and $\langle \mu_i, \mu_j \rangle_{\mathcal{H}_{k_s}} = 0$ for any $i \neq j$. We have already shown that μ_1 maximizes $r_{\mathcal{H}_{k_s}}(f)$ over all $f \in \mathcal{H}_{k_s}$. Thus, μ_1 must be the top-1 eigenfunction of $\mathcal{H}_{k_s}$. Suppose $d \geq 2$, and $\mu_1, \cdots, \mu_{d-1}$ are the top-$(d-1)$ eigenfunctions and are orthogonal to each other in $\mathcal{H}_{k_s}$. Let $\mathcal{H}_0 = \left\{ h \;\middle|\; \forall i \in [d-1] : \langle h, \mu_i \rangle_{P_\mathcal{X}} = 0 \right\}$. Obviously, $\mathcal{H}_0 \cap \mathcal{H}_{k^p}$ is a closed subspace of $\mathcal{H}_{k^p}$ for any $p \geq 1$. Moreover, for any $f \in \mathcal{H}_0 \cap \mathcal{H}_{k_s}$ and any $i \in [d-1]$, we have $\langle f, \mu_i \rangle_{\mathcal{H}_{k_s}} = s_i^{-1} \langle f, \mu_i \rangle_{P_\mathcal{X}} = 0$, where s_i is the eigenvalue of T_{k_s} corresponding to μ_i. Thus, $\mathcal{H}_{k_s} \cap \mathcal{H}_0$ is a closed subspace of $\mathcal{H}_{k_s}$. By the condition of preserving relative smoothness, μ_d maximizes $r_{\mathcal{H}_{k_s}}(f)$ over $f \in \mathcal{H}_{k_s} \cap \mathcal{H}_0$. Thus, μ_d is the d-th eigenfunction of T_{k_s}, and is orthogonal to $\mu_1, \cdots, \mu_{d-1}$ in $\mathcal{H}_{k_s}$.

Third, we prove by contradiction that $s_i \leq M\lambda_i$ for all i. If this is false, then obviously one can find $t_1 < t_2 < \cdots$ such that $s_{t_i} \geq 1 \cdot \lambda_{t_i}$ for all i. Consider $f = \sum_{i=1}^\infty \sqrt{i^{-1} \cdot \lambda_{t_i}} \mu_{t_i}$. Then, $\|f\|_{\mathcal{H}_k}^2 = \sum_i i^{-1} = +\infty$. Since $\mathcal{H}_{k_s} \subset \mathcal{H}_k$, this implies that $\|f\|_{\mathcal{H}_{k_s}}^2 = +\infty = \sum_i \frac{\lambda_{t_i}}{i \cdot s_{t_i}} \leq \sum_i \frac{1}{i^2} < +\infty$, which is a contradiction.

Fourth, we find a function $s(\lambda)$ that satisfies the conditions in the theorem to interpolate (λ_i, s_i) for all i. We first point out that we can without loss of generality assume that $\lambda_i < 2\lambda_{i+1}$ for all i: If there is an i that does not satisfy this condition, we simply insert some new λ's between λ_i and λ_{i+1}, whose corresponding s's are the linear interpolations between s_i and s_{i+1}, so that $s_i \leq M\lambda_i$ still holds. With this assumption, it suffices to construct a series of bump functions $\{f_i\}_{i=1}^\infty$, where $f_i \equiv 0$ if $\lambda_i = \lambda_{i+1}$; otherwise, $f_i(\lambda) = s_i - s_{i+1}$ for $\lambda \geq \lambda_i$ and $f_i(\lambda) = 0$ for $\lambda \leq \lambda_{i+1}$. Such bump functions are C^∞ and monotonically non-decreasing. Then, define $s(\lambda) = \sum_i f_i(\lambda)$ for $\lambda > 0$, and $s(0) = 0$. This sum of bump functions converges everywhere on $(0, +\infty)$, since it is a finite sum locally everywhere. Clearly this s is monotonic, interpolates all the points, continuous on $[0, +\infty)$ and C^∞ on $(0, +\infty)$. And for all λ that is not λ_i, for instance $\lambda \in (\lambda_{i+1}, \lambda_i)$, there is $s(\lambda) \leq s(\lambda_i) \leq M\lambda_i \leq 2M\lambda_{i+1} \leq 2M\lambda$. Thus, $s(\lambda) = O(\lambda)$ for $\lambda \in [0, +\infty)$. ■

Remark D.4. *In general, we cannot guarantee that $s(\lambda)$ is differentiable at $\lambda = 0$. Here is a counterexample: $\lambda_i = 3^{-i}$, and $s_i = 3^{-i}$ if i is odd and $2 \cdot 3^{-i}$ if i is even. Were $s(\lambda)$ to be differentiable at $\lambda = 0$, its derivative would be 1 and also would be 2, a contradiction.*

D.6 Proof of Theorem 5.25

Proof Let $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_{m+n}$ be the eigenvalues of $\frac{\boldsymbol{G}_k}{m+n}$. It is easy to show that $\boldsymbol{Q}$ has the same eigenvectors as $\frac{\boldsymbol{G}_k}{m+n}$, with eigenvalues $g(\hat{\lambda}_1), \cdots, g(\hat{\lambda}_2)$. By Lemma 5.29 and Borel-Cantelli lemma, as $n \to \infty$, $\hat{\lambda}_1 \xrightarrow{a.s.} \lambda_1$. For simplicity, let us assume that λ is slightly larger than λ_1, so almost surely there is $\hat{\lambda}_1 \leq \lambda$. Then, all eigenvalues of $\boldsymbol{Q}$ are in $[\rho_{\min}, \rho_{\max}]$.

The first part of this proof is to bound $\|\boldsymbol{u}_t\|_2$, where $\boldsymbol{u}_t := (m+n)\tilde{\boldsymbol{I}}_n \left(\frac{\boldsymbol{G}_k}{m+n}\right)^r (\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)$.

Let $\boldsymbol{\theta}_t$ be the $\boldsymbol{\theta}$ at iteration t, and $\boldsymbol{\theta}_*$ be the optimal solution. Since $\boldsymbol{\theta}_0 = \mathbf{0}$, we have

$$\begin{aligned}
\boldsymbol{\theta}_* - \boldsymbol{\theta}_t =& \left[\left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}\right]\right)\boldsymbol{\theta}_* + \gamma\tilde{\boldsymbol{y}}\right] \\
&- \left[\left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}\right]\right)\boldsymbol{\theta}_{t-1} + \gamma\tilde{\boldsymbol{y}}\right] \\
=& \left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}\right]\right)(\boldsymbol{\theta}_* - \boldsymbol{\theta}_{t-1}) \\
=& \left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}\right]\right)^t\boldsymbol{\theta}_*.
\end{aligned} \tag{D.1}$$

Note that

$$\begin{aligned}
&\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r + n\beta_n\boldsymbol{Q}\right]\right) \\
=& \left(\boldsymbol{I}_{m+n} - \gamma\left[(m+n)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2} + n\beta_n\boldsymbol{Q}\right]\right)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}.
\end{aligned}$$

Thus, by propagating $\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}$ from left to right, we get

$$\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t) = (\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^t\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{\theta}_*,$$

where $\boldsymbol{R} := (m+n)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2} + n\beta_n\boldsymbol{Q}$ is a *p.s.d.* matrix. Denote the smallest and largest eigenvalues of $\boldsymbol{R}$ by $\tilde{\lambda}_{\min}$ and $\tilde{\lambda}_{\max}$. Then, $\tilde{\lambda}_{\min} \geq n\beta_n\rho_{\min}$. In terms of $\tilde{\lambda}_{\max}$, we have

$$(m+n)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2} = \left(\frac{\boldsymbol{G}_k}{m+n}\right)^{\frac{r-1}{2}}\left(\boldsymbol{G}_k^{\frac{1}{2}}\tilde{\boldsymbol{I}}_n\boldsymbol{G}_k^{\frac{1}{2}}\right)\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{\frac{r-1}{2}}.$$

By Sylvester's theorem, all non-zero eigenvalues of $\boldsymbol{G}_k^{\frac{1}{2}}\tilde{\boldsymbol{I}}_n\boldsymbol{G}_k^{\frac{1}{2}}$ are the eigenvalues of $\tilde{\boldsymbol{I}}_n\boldsymbol{G}_k\tilde{\boldsymbol{I}}_n$, *i.e.* the non-zero eigenvalues of $\boldsymbol{G}_{K,n}$. By Lemma 5.29, $\frac{1}{n}\|\boldsymbol{G}_{K,n}\|_2 \xrightarrow{a.s.} \lambda_1$, so suppose $\|\boldsymbol{G}_{K,n}\|_2 \leq n\lambda$. Then, $\tilde{\lambda}_{\max} \leq n\lambda^r + n\beta_n\rho_{\max}$.

Since $\boldsymbol{M}\boldsymbol{\theta}_* = \tilde{\boldsymbol{y}}$, and $\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{M} = \boldsymbol{R}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}$, we have $\boldsymbol{R}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{\theta}_* = \left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{y}}$. Note that $\boldsymbol{R}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R}) = (\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})\boldsymbol{R}$. Thus, we have

$$\begin{aligned}
\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t) &= (\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^t\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{\theta}_* \\
&= \boldsymbol{R}^{-1}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^t\boldsymbol{R}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{\theta}_* \\
&= \boldsymbol{R}^{-1}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^t\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{y}}.
\end{aligned}$$

Now we bound $\|\boldsymbol{u}_t\|_2$. First, note that for any matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$ where $\boldsymbol{B}$ is *p.s.d.*, there is $\boldsymbol{u}^\top\boldsymbol{A}^\top\boldsymbol{B}\boldsymbol{A}\boldsymbol{u} \leq \|\boldsymbol{B}\|_2\|\boldsymbol{A}\boldsymbol{u}\|_2^2 \leq \|\boldsymbol{B}\|_2\|\boldsymbol{A}^\top\boldsymbol{A}\|_2\|\boldsymbol{u}\|_2^2$ for any $\boldsymbol{u} \in \mathbb{R}^d$, so $\|\boldsymbol{A}^\top\boldsymbol{B}\boldsymbol{A}\|_2 \leq$

$\|\boldsymbol{B}\|_2\|\boldsymbol{A}^\top\boldsymbol{A}\|_2$. Second, note that the last m elements of $\tilde{\boldsymbol{y}}$ are zeros, which means that $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{I}}_n\tilde{\boldsymbol{y}}$. Thus, we have

$$\begin{aligned}
\|\boldsymbol{u}_t\|_2 &= \left\|(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\right\|_2 \\
&= \left\|(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\boldsymbol{R}^{-1}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^t\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{y}}\right\|_2 \\
&= \left\|(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^{t/2}\boldsymbol{R}^{-1}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^{t/2}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{r/2}\tilde{\boldsymbol{I}}_n\tilde{\boldsymbol{y}}\right\|_2 \\
&\le \left\|(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^{t/2}\boldsymbol{R}^{-1}(\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R})^{t/2}\right\|_2\left\|(m+n)\tilde{\boldsymbol{I}}_n\left(\frac{\boldsymbol{G}_k}{m+n}\right)^r\tilde{\boldsymbol{I}}_n\right\|_2\|\tilde{\boldsymbol{y}}\|_2 \\
&\le \frac{1}{\tilde{\lambda}_{\min}}\|\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R}\|_2^t(n\lambda_1^r)\|\boldsymbol{y}\|_2,
\end{aligned}$$

where the last step is because we have already proved $\left\|(m+n)\tilde{\boldsymbol{I}}_n\big(\frac{\boldsymbol{G}_k}{m+n}\big)^r\tilde{\boldsymbol{I}}_n\right\|_2 \le n\lambda_1^r$.

Now, for $\gamma = \frac{1}{n\lambda^r}$, when n is sufficiently large it is less than $\frac{2}{\tilde{\lambda}_{\max}+\tilde{\lambda}_{\min}}$, because $\beta_n = o(1)$. Thus, $\|\boldsymbol{I}_{m+n} - \gamma\boldsymbol{R}\|_2 \le 1 - \frac{\tilde{\lambda}_{\min}}{n\lambda^r} \le 1 - \frac{\beta_n\rho_{\min}}{\lambda^r}$. Thus, we have

$$\|\boldsymbol{u}_t\|_2 \le \left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)^t\frac{\lambda^r}{\beta_n\rho_{\min}}\|\boldsymbol{y}\|_2.$$

The second part of this proof is to bound $\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2$. Let us return to Eqn. (D.1), which says that

$$\begin{aligned}
\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_{t+1})\|_2 &= \|(\boldsymbol{I}_{m+n} - \gamma n\beta_n\boldsymbol{Q})\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t) - \gamma\boldsymbol{Q}\boldsymbol{u}_t\|_2 \\
&\le \left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 + \frac{\rho_{\max}}{n\lambda^r}\|\boldsymbol{u}_t\|_2.
\end{aligned}$$

Here again, we assume that n is large enough so that $\lambda^r > \beta_n\rho_{\min}$. This implies that

$$\begin{aligned}
&\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_{t+1})\|_2 - t\left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)^t\frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n\rho_{\min}} \\
&\le \left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)\left[\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 - (t-1)\left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)^{t-1}\frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n\rho_{\min}}\right] \\
&\le \cdots \le \left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)^t\left[\left(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\right)\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2 + \frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n\rho_{\min}}\right].
\end{aligned}$$

Thus, there is $\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 \le \big(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\big)^t\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2 + t\big(1 - \frac{\beta_n\rho_{\min}}{\lambda^r}\big)^{t-1}\frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n\rho_{\min}}$. Using $1 - x \le e^{-x}$, we have

$$\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 \le \exp\left(-\frac{\beta_n\rho_{\min}t}{\lambda^r}\right)\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2 + t\exp\left(-\frac{\beta_n\rho_{\min}(t-1)}{\lambda^r}\right)\frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n\rho_{\min}}.$$

When $t = t_0 := \frac{4\lambda^r}{\beta_n\rho_{\min}}\log\frac{2\lambda^r\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n^2\rho_{\min}^2\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2}$, by $\log(2x) \le x$ for $x > 0$, we have

$$\exp\left(\frac{\beta_n\rho_{\min}}{\lambda^r}\frac{t}{2}\right) \ge \left(\frac{2\lambda^r\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n^2\rho_{\min}^2\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2}\right)^2 \ge \frac{4\lambda^r\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n^2\rho_{\min}^2\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2}\log\left(\frac{2\lambda^r\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n^2\rho_{\min}^2\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2}\right).$$

Let $F(t) := \exp\left(\frac{\beta_n \rho_{\min}}{2\lambda^r} t\right) - \frac{\rho_{\max}\|\boldsymbol{y}\|_2}{n\beta_n \rho_{\min}\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2} t$. Then we have $F(t_0) \geq 0$. And it is easy to show that for all $t \geq \frac{t_0}{2}$, there is $F'(t) \geq 0$. This means that when $t \geq t_0$, there is $F(t) \geq 0$, so we have

$$\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 \leq \exp\left(-\frac{\beta_n \rho_{\min} t}{\lambda^r}\right)\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2 + \exp\left(-\frac{\beta_n \rho_{\min}}{\lambda^r}\left(\frac{t}{2} - 1\right)\right)\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2.$$

Hence, when $t \geq \max\left\{\frac{2\lambda^r}{\beta_n \rho_{\min}} \log\frac{2}{\epsilon} + 2, t_0\right\}$, we have $\|\boldsymbol{Q}(\boldsymbol{\theta}_* - \boldsymbol{\theta}_t)\|_2 \leq \epsilon\|\boldsymbol{Q}\boldsymbol{\theta}_*\|_2$, which implies that the relative estimation error of $\hat{\boldsymbol{\alpha}}$ is less than ϵ. ■

D.7 Proof of Theorem 5.26

Proof Let us look at the three conditions used in Theorem 5.12.

- **Eigenvalue decay (EVD):** This is a condition of the theorem.
- **Embedding condition (EMB):** For any $f = \sum u_i \mu_i \in \mathcal{H}_k$, for $P_{\mathcal{X}}$-almost all x we have $f(x)^2 = (\sum u_i \mu_i(x))^2 \leq \left(\sum \frac{u_i^2}{\lambda_i}\right)(\sum \lambda_i \mu_i(x)^2) \leq \|f\|_{\mathcal{H}_k}^2 \kappa^2 \leq \kappa^2 M \|f\|_{\mathcal{H}_{k_s}}^2$. Thus, EMB holds with $c_2 = \kappa\sqrt{M}$.
- **Source condition (SRC):** This holds with $c_3 = \sqrt{\epsilon}\|f^*\|_{P_{\mathcal{X}}}$.

Thus, the theorem can be proved using the result in [41]. ■

D.8 Proof of Theorem 5.28

Proposition D.5. *For any p.s.d. matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$, we have* $\operatorname{Tr}(\boldsymbol{A}\boldsymbol{B}) \leq \|\boldsymbol{A}\|_2 \operatorname{Tr}(\boldsymbol{B})$.

Proof An elementary proof can be found at `https://math.stackexchange.com/questions/2241879/reference-for-trace-norm-inequality`. ■

Lemma D.6. *For any $\delta \in (0,1)$, with probability at least $1-\delta$ the following holds for all $p \geq 1$:*

$$\left|\hat{k}^p(x, x_j) - k^p(x, x_j)\right| \leq (p-1)\lambda_{\max}^{p-2} \frac{\kappa^4}{\sqrt{m+n}}\left(2 + \sqrt{2\log\frac{1}{\delta}}\right) \quad \textit{for all } j \in [m+n], x \in \mathcal{X},$$

which implies that

$$\left|\hat{k}_s(x, x_j) - k_s(x, x_j)\right| \leq \nabla_\lambda\left(\frac{s(\lambda)}{\lambda}\right)\Big|_{\lambda=\lambda_{\max}} \frac{\kappa^4}{\sqrt{m+n}}\left(2 + \sqrt{2\log\frac{1}{\delta}}\right).$$

Proof For any $x' \in \mathcal{X}$ and any $p \geq 1$, we have

$$\|k^p(\cdot, x')\|_{\mathcal{H}_k}^2 = \left\|\sum_i \lambda_i^p \mu_i(x')\mu_i(\cdot)\right\|_{\mathcal{H}_k}^2 = \sum_i \frac{\lambda_i^{2p}\mu_i(x')^2}{\lambda_i} \leq \lambda_1^{2p-2}\kappa^2.$$

Let $F_p(x) = \boldsymbol{u}^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p \boldsymbol{v}_k(x)$, where $\boldsymbol{u} \in \mathbb{R}^{m+n}$ satisfies $\|\boldsymbol{u}\|_1 \leq 1$. Since $\langle k(x_i,\cdot), k(x_j,\cdot)\rangle_{\mathcal{H}_k} = k(x_i, x_j)$, we have $\langle \boldsymbol{v}_k, \boldsymbol{v}_k\rangle_{\mathcal{H}_k} = \boldsymbol{G}_k$. Thus,

$$\|F_p\|_{\mathcal{H}_k}^2 = \left\langle \boldsymbol{u}^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p \boldsymbol{v}_k, \boldsymbol{u}^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p \boldsymbol{v}_k\right\rangle_{\mathcal{H}_k} = \boldsymbol{u}^\top \frac{\boldsymbol{G}_k^{2p+1}}{(m+n)^{2p}}\boldsymbol{u}.$$

Since $\boldsymbol{G}_k$ is *p.s.d.*, we can define $\boldsymbol{G}_k^{1/2}$. Using Proposition D.5, we have

$$\begin{aligned}\|F_p\|_{\mathcal{H}_k}^2 &= \boldsymbol{u}^\top \frac{\boldsymbol{G}_k^{2p+1}}{(m+n)^{2p}}\boldsymbol{u} = \operatorname{Tr}\left\{\boldsymbol{u}^\top \boldsymbol{G}_k^{1/2}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{2p}\boldsymbol{G}_k^{1/2}\boldsymbol{u}\right\} \\ &= \operatorname{Tr}\left\{\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{2p}\boldsymbol{G}_k^{1/2}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}_k^{1/2}\right\} \le \hat{\lambda}_1^{2p}\operatorname{Tr}\left(\boldsymbol{G}_k^{1/2}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}_k\right).\end{aligned}$$

Moreover, $\operatorname{Tr}\left(\boldsymbol{G}_k^{1/2}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}_k\right) = \boldsymbol{u}^\top\boldsymbol{G}_k\boldsymbol{u} = \sum_{i,j=1}^{m+n} u_i u_j k(x_i,x_j) \le \sum_{i,j=1}^{m+n}|u_i u_j k(x_i,x_j)| \le \kappa^2\|\boldsymbol{u}\|_1^2 \le \kappa^2$. Thus, we have $\|F_p\|_{\mathcal{H}_k} \le \hat{\lambda}_1^p\kappa$ for all $p \ge 1$.

For any $p \ge 1$, define $\boldsymbol{v}_{k^p}(x) \in \mathbb{R}^{m+n}$ as $\boldsymbol{v}_{k^p}(x)[i] = k^p(x,x_i)$ for $i \in [m+n]$. Then,

$$\begin{aligned}&\left|k^p(x,x_j) - \hat{k}^p(x,x_j)\right| \\ =& \left|k^p(x,x_j) - \frac{1}{(m+n)^{p-1}}\boldsymbol{v}_k(x)^\top\boldsymbol{G}_k^{p-2}\boldsymbol{v}_k(x_j)\right| \\ \le& \left|k^p(x,x_j) - \frac{1}{m+n}\boldsymbol{v}_{k^{p-1}}(x)^\top\boldsymbol{v}_k(x_j)\right| \\ &+ \sum_{q=1}^{p-2}\frac{1}{(m+n)^q}\left|\boldsymbol{v}_{k^{p-q}}(x)^\top\boldsymbol{G}_k^{q-1}\boldsymbol{v}_k(x_j) - \boldsymbol{v}_{k^{p-q-1}}(x)^\top\frac{\boldsymbol{G}_k^q}{m+n}\boldsymbol{v}_k(x_j)\right|.\end{aligned}$$

Since $f(z) = k^{p-1}(x,z)k(x_j,z) \in \mathcal{F}$, where $\mathcal{F}$ was defined in Lemma 5.13, the first term can be bounded as

$$\begin{aligned}&\left|k^p(x,x_j) - \frac{1}{m+n}\boldsymbol{v}_{k^{p-1}}(x)^\top\boldsymbol{v}_k(x_j)\right| \\ =& \left|\int k^{p-1}(x,z)k(x_j,z)dP_{\mathcal{X}}(z) - \frac{1}{m+n}\sum_{i=1}^{m+n}k^{p-1}(x,x_i)k(x_j,x_i)\right| \\ \le& \lambda_1^{p-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right),\end{aligned}$$

where the last step uses $\|k^{p-1}(x,\cdot)\|_{\mathcal{H}_k} \le \lambda_1^{p-2}\kappa$, and $\|k(x_j,\cdot)\|_{\mathcal{H}_k} \le \kappa$.

For the second term, since $\boldsymbol{v}_k(x_j) = \boldsymbol{G}_k\boldsymbol{e}_j$ where $\boldsymbol{e}_j = [0,\cdots,0,1,0,\cdots,0]$, we have

$$\begin{aligned}&\sum_{q=1}^{p-2}\frac{1}{(m+n)^q}\left|\boldsymbol{v}_{k^{p-q}}(x)^\top\boldsymbol{G}_k^{q-1}\boldsymbol{v}_k(x_j) - \boldsymbol{v}_{k^{p-q-1}}(x)^\top\frac{\boldsymbol{G}_k^q}{m+n}\boldsymbol{v}_k(x_j)\right| \\ =& \left|\int k^{p-q-1}(x,z)\left[\boldsymbol{e}_j^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^q\boldsymbol{v}_k(z)\right]dP_{\mathcal{X}}(z) - \frac{1}{m+n}\sum_{j=1}^{m+n}k^{p-q-1}(x,x_j)\left[\boldsymbol{e}_j^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^q\boldsymbol{v}_k(x_j)\right]\right| \\ \le& \lambda_1^{p-q-2}\hat{\lambda}_1^q\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right),\end{aligned}$$

where the last step uses $\|k^{p-q-1}(x,\cdot)\|_{\mathcal{H}_k} \le \lambda_1^{p-q-2}\kappa$, and $\left\|\boldsymbol{e}_j^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^q\boldsymbol{v}_k\right\|_{\mathcal{H}_k} \le \hat{\lambda}_1^q\kappa$ since $\|\boldsymbol{e}_j\|_1 = 1$. Finally, note that $\nabla_\lambda\left(\frac{s(\lambda)}{\lambda}\right) = \sum_{p=1}^\infty \pi_p(p-1)\lambda^{p-2}$. Combining all of the above yields the result. ■

Corollary D.7. *Under the settings of Lemma D.6, we have*

$$\begin{aligned}
&\left|k_{s^2}(x_i,x_j)-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_{\mathcal{X}}}\right|+\left|\langle\hat{k}_s(x_i,\cdot),\hat{k}_s(x_j,\cdot)\rangle_{P_{\mathcal{X}}}-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_{\mathcal{X}}}\right|\\
&\leq 2s(\lambda_{\max})\,\nabla_\lambda\left(\frac{s(\lambda)}{\lambda}\right)\Big|_{\lambda=\lambda_{\max}}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right)
\end{aligned}$$

holds for all $i,j\in[m+n]$*, where* $\lambda_{\max}=\max\left\{\lambda_1,\hat{\lambda}_1\right\}$.

Proof Let $F_{p,q}(x)=\boldsymbol{u}^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p\boldsymbol{v}_{k^q}(x)$ for any $\|\boldsymbol{u}\|_1\leq 1$ and any $p\geq 0,q\geq 1$. By Proposition D.5, we have

$$\begin{aligned}
\|F_{p,q}\|_{\mathcal{H}_k}^2&=\boldsymbol{u}^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p\boldsymbol{G}_{k^{2q-1}}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^p\boldsymbol{u}\\
&=\mathrm{Tr}\left(\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-1/2}\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-1/2}\boldsymbol{G}_k^{1/2}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}_k^{1/2}\right)\\
&\leq\hat{\lambda}_1^{2p-1}\left\|\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\right\|_2\mathrm{Tr}\left(\boldsymbol{G}_k^{1/2}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}_k^{1/2}\right)\\
&=\hat{\lambda}_1^{2p-1}\left\|\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\right\|_2\boldsymbol{u}^\top\boldsymbol{G}_k\boldsymbol{u}\leq\hat{\lambda}_1^{2p-1}\left\|\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\right\|_2\kappa^2.
\end{aligned}$$

For any unit vector $\boldsymbol{w}\in\mathbb{R}^{m+n}$, we have

$$\hat{\lambda}_1\geq\boldsymbol{w}^\top\frac{\boldsymbol{G}_k}{m+n}\boldsymbol{w}=\frac{1}{m+n}\sum_{i,j=1}^{m+n}w_iw_jK(x_i,x_j)=\frac{1}{m+n}\sum_t\lambda_t\boldsymbol{w}^\top\boldsymbol{M}_t\boldsymbol{w},$$

where $\boldsymbol{M}_t\in\mathbb{R}^{(m+n)\times(m+n)}$ is defined as $\boldsymbol{M}_t[i,j]=\mu_t(x_i)\mu_t(x_j)$. Thus, we have

$$\boldsymbol{w}^\top\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\boldsymbol{w}=\frac{1}{m+n}\sum_t\lambda_t^{2q-1}\boldsymbol{w}^\top\boldsymbol{M}_t\boldsymbol{w}\leq\lambda_1^{2q-2}\frac{1}{m+n}\sum_t\lambda_t\boldsymbol{w}^\top\boldsymbol{M}_t\boldsymbol{w}\leq\lambda_1^{2q-2}\hat{\lambda}_1,$$

which implies that $\left\|\frac{\boldsymbol{G}_{k^{2q-1}}}{m+n}\right\|_2\leq\lambda_1^{2q-2}\hat{\lambda}_1$. Thus, $\|F_{p,q}\|_{\mathcal{H}_k}^2\leq\lambda_1^{2q-2}\hat{\lambda}_1^{2p}\kappa^2$.

Note that $\langle\boldsymbol{v}_k,\boldsymbol{v}_k\rangle_{P_{\mathcal{X}}}=\boldsymbol{G}_{k^2}$. So for any $p,q\geq 1$ and any $i,j\in[m+n]$, there is:

$$\begin{aligned}
&\left|k^{p+q}(x_i,x_j)-\left\langle\hat{k}^p(x_i,\cdot),k^q(x_j,\cdot)\right\rangle_{P_{\mathcal{X}}}\right|=\left|\boldsymbol{e}_i^\top\boldsymbol{G}_{k^{p+q}}\boldsymbol{e}_j-\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-1}}{(m+n)^{p-1}}\boldsymbol{G}_{k^{q+1}}\boldsymbol{e}_j\right|\\
\leq&\sum_{t=1}^{p-1}\left|\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-t}}{(m+n)^{p-t}}\boldsymbol{G}_{k^{q+t}}\boldsymbol{e}_j-\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-t-1}}{(m+n)^{p-t-1}}\boldsymbol{G}_{k^{q+t+1}}\boldsymbol{e}_j\right|\\
=&\sum_{t=1}^{p-1}\left|\frac{1}{m+n}\sum_{l=1}^{m+n}\left[\boldsymbol{e}_i^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-t-1}\boldsymbol{v}_k\right](x_l)\left[\boldsymbol{e}_j^\top\boldsymbol{v}_{k^{q+t}}\right](x_l)\right.\\
&\left.-\left\langle\boldsymbol{e}_i^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-t-1}\boldsymbol{v}_k,\boldsymbol{e}_j^\top\boldsymbol{v}_{k^{q+t}}\right\rangle_{P_{\mathcal{X}}}\right|\\
\leq&\sum_{t=1}^{p-1}\lambda_1^{q+t-1}\hat{\lambda}_1^{p-t-1}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right)\leq(p-1)\lambda_{\max}^{p+q-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right).
\end{aligned}$$

Thus, we have

$$\left|k_{s^2}(x_i,x_j)-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_\mathcal{X}}\right|=\sum_{p,q=1}^{\infty}\left|\pi_p\pi_q\left(k^{p+q}(x_i,x_j)-\left\langle\hat{k}^p(x_i,\cdot),k^q(x_j,\cdot)\right\rangle_{P_\mathcal{X}}\right)\right|$$
$$\leq\sum_{p,q=1}^{\infty}\pi_p\pi_q(p-1)\lambda_{\max}^{p+q-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right).$$

Similarly, we can show that:

$$\begin{aligned}
&\left|\left\langle\hat{k}^p(x_i,\cdot),\hat{k}^q(x_j,\cdot)\right\rangle_{P_\mathcal{X}}-\left\langle\hat{k}^p(x_i,\cdot),k^q(x_j,\cdot)\right\rangle_{P_\mathcal{X}}\right|\\
=&\left|\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-1}}{(m+n)^{p-1}}\boldsymbol{G}_{k^2}\frac{\boldsymbol{G}_k^{q-1}}{(m+n)^{q-1}}\boldsymbol{e}_j-\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-1}}{(m+n)^{p-1}}\boldsymbol{G}_{k^{q+1}}\boldsymbol{e}_j\right|\\
\leq&\sum_{t=1}^{q-1}\left|\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-1}}{(m+n)^{p-1}}\boldsymbol{G}_{k^{t+1}}\frac{\boldsymbol{G}_k^{q-t}}{(m+n)^{q-t}}\boldsymbol{e}_j-\boldsymbol{e}_i^\top\frac{\boldsymbol{G}_k^{p-1}}{(m+n)^{p-1}}\boldsymbol{G}_{k^{t+2}}\frac{\boldsymbol{G}_k^{q-t-1}}{(m+n)^{q-t-1}}\boldsymbol{e}_j\right|\\
=&\sum_{t=1}^{q-1}\left|\frac{1}{m+n}\sum_{l=1}^{m+n}\left[\boldsymbol{e}_i^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-1}\boldsymbol{v}_{k^{t+1}}\right](x_l)\left[\boldsymbol{e}_j^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{q-t-1}\boldsymbol{v}_k\right](x_l)\right.\\
&\left.-\left\langle\boldsymbol{e}_i^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{p-1}\boldsymbol{v}_{k^{t+1}},\boldsymbol{e}_j^\top\left(\frac{\boldsymbol{G}_k}{m+n}\right)^{q-t-1}\boldsymbol{v}_k\right\rangle_{P_\mathcal{X}}\right|\\
\leq&\sum_{t=1}^{q-1}\lambda_1^t\hat{\lambda}_1^{p+q-t-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right)\leq(q-1)\lambda_{\max}^{p+q-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right),
\end{aligned}$$

which implies that

$$\begin{aligned}
&\left|\langle\hat{k}_s(x_i,\cdot),\hat{k}_s(x_j,\cdot)\rangle_{P_\mathcal{X}}-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_\mathcal{X}}\right|\\
=&\sum_{p,q=1}^{\infty}\left|\pi_p\pi_q\left(\left\langle\hat{k}^p(x_i,\cdot),\hat{k}^q(x_j,\cdot)\right\rangle_{P_\mathcal{X}}-\left\langle\hat{k}^p(x_i,\cdot),k^q(x_j,\cdot)\right\rangle_{P_\mathcal{X}}\right)\right|\\
\leq&\sum_{p,q=1}^{\infty}\pi_p\pi_q(q-1)\lambda_{\max}^{p+q-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right).
\end{aligned}$$

Combining the above inequalities, we obtain

$$\begin{aligned}
&\left|k_{s^2}(x_i,x_j)-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_\mathcal{X}}\right|+\left|\langle\hat{k}_s(x_i,\cdot),\hat{k}_s(x_j,\cdot)\rangle_{P_\mathcal{X}}-\langle\hat{k}_s(x_i,\cdot),k_s(x_j,\cdot)\rangle_{P_\mathcal{X}}\right|\\
\leq&\sum_{p,q=1}^{\infty}\pi_p\pi_q(p+q-2)\lambda_{\max}^{p+q-2}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right)\\
=&\lambda_{\max}\,\nabla_\lambda\left(\frac{s(\lambda)^2}{\lambda^2}\right)\Big|_{\lambda=\lambda_{\max}}\frac{\kappa^4}{\sqrt{m+n}}\left(2+\sqrt{2\log\frac{1}{\delta}}\right),
\end{aligned}$$

so we get the result by expanding the derivative. ■

We now prove Theorem 5.28.

Proof Define $\boldsymbol{v}_{k_s,n}(x) \in \mathbb{R}^n$ such that $\boldsymbol{v}_{k_s,n}(x)[i] = k_s(x, x_i)$. Define $\boldsymbol{v}_{\hat{k}_s,n}(x)$ similarly. Recall the formulas $\tilde{f} = \tilde{\boldsymbol{\alpha}}^\top \boldsymbol{v}_{k_s,n}$ and $\hat{f} = \hat{\boldsymbol{\alpha}}^\top \boldsymbol{v}_{\hat{k}_s,n}$. Define $f^\dagger := \hat{\boldsymbol{\alpha}}^\top \boldsymbol{v}_{k_s,n}$. Since $\boldsymbol{G}_{\hat{k}_s,n}$ is *p.s.d.*, we can see that $\|\hat{\boldsymbol{\alpha}}\|_2 \le \frac{\|\boldsymbol{y}\|_2}{n\beta_n}$, and $\|\hat{\boldsymbol{\alpha}}\|_1 \le \sqrt{n}\|\hat{\boldsymbol{\alpha}}\|_2$. So by Corollary D.7, we have

$$\begin{aligned}
\left\|\hat{f} - f^\dagger\right\|_{P_\mathcal{X}}^2 &= \hat{\boldsymbol{\alpha}}^\top \left\langle \boldsymbol{v}_{\hat{k}_s,n} - \boldsymbol{v}_{k_s,n}, \boldsymbol{v}_{\hat{k}_s,n} - \boldsymbol{v}_{k_s,n} \right\rangle_{P_\mathcal{X}} \hat{\boldsymbol{\alpha}} \\
&= \hat{\boldsymbol{\alpha}}^\top \Big(\langle \hat{k}_s(x_i,\cdot), \hat{k}_s(x_j,\cdot)\rangle_{P_\mathcal{X}} + k_{s^2}(x_i,x_j) - 2\langle \hat{k}_s(x_i,\cdot), k_s(x_j,\cdot)\rangle_{P_\mathcal{X}} \Big) \hat{\boldsymbol{\alpha}} \\
&\le 2s(\lambda_{\max})\, \nabla_\lambda \left(\frac{s(\lambda)}{\lambda} \right)\Big|_{\lambda=\lambda_{\max}} \frac{\beta_n^{-2}\kappa^4}{\sqrt{m+n}} \left(2 + \sqrt{2\log\frac{1}{\delta}} \right) \frac{\|\boldsymbol{y}\|_2^2}{n}.
\end{aligned}$$

By the definitions of $\tilde{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}$, we can also see that:

$$(\boldsymbol{G}_{k_s,n} + n\beta_n \boldsymbol{I}_n)(\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}}) = \Big(\boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \Big) \hat{\boldsymbol{\alpha}}. \tag{D.2}$$

Note that $\left\| \boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \right\|_2 \le n \left\| \boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \right\|_{\max}$. Here $\|\boldsymbol{M}\|_{\max} = \max|\boldsymbol{M}[i,j]|$. Thus, we have

$$\begin{aligned}
\left\|\tilde{f} - f^\dagger\right\|_{\mathcal{H}_{k_s}}^2 &= (\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}})^\top \boldsymbol{G}_{k_s,n} (\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}}) \\
&= (\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}})^\top \Big(\boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \Big) \hat{\boldsymbol{\alpha}} - n\beta_n (\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}})^\top (\hat{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}}) \\
&\le \|\hat{\boldsymbol{\alpha}}\|_2 \left\| \boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \right\|_2 \|\hat{\boldsymbol{\alpha}}\|_2 + \|\tilde{\boldsymbol{\alpha}}\|_2 \left\| \boldsymbol{G}_{k_s,n} - \boldsymbol{G}_{\hat{k}_s,n} \right\|_2 \|\hat{\boldsymbol{\alpha}}\|_2 - 0 \\
&\le 2\, \nabla_\lambda \left(\frac{s(\lambda)}{\lambda} \right)\Big|_{\lambda=\lambda_{\max}} \frac{\beta_n^{-2}\kappa^4}{\sqrt{m+n}} \left(2 + \sqrt{2\log\frac{1}{\delta}} \right) \frac{\|\boldsymbol{y}\|_2^2}{n}.
\end{aligned}$$

And note that we have $\left\|\tilde{f} - f^\dagger\right\|_{P_\mathcal{X}}^2 \le s(\lambda_1) \left\|\tilde{f} - f^\dagger\right\|_{\mathcal{H}_{k_s}}^2 \le s(\lambda_{\max}) \left\|\tilde{f} - f^\dagger\right\|_{\mathcal{H}_{k_s}}^2$. Thus,

$$\begin{aligned}
\left\|\hat{f} - \tilde{f}\right\|_{P_\mathcal{X}}^2 &\le 2 \left(\left\|\hat{f} - f^\dagger\right\|_{P_\mathcal{X}}^2 + \left\|\tilde{f} - f^\dagger\right\|_{P_\mathcal{X}}^2 \right) \\
&\le 8s(\lambda_{\max})\, \nabla_\lambda \left(\frac{s(\lambda)}{\lambda} \right)\Big|_{\lambda=\lambda_{\max}} \frac{\beta_n^{-2}\kappa^4}{\sqrt{m+n}} \left(2 + \sqrt{2\log\frac{1}{\delta}} \right) \frac{\|\boldsymbol{y}\|_2^2}{n},
\end{aligned}$$

as desired. ■

Appendix E

Proofs for Chapter 6

E.1 Proof of Theorem 6.4

We need the following classical result in convex optimization.

Theorem E.1 ([141], p. 16)**.** *Let f be a convex and L-smooth function (Definition 6.9) on $\mathcal{D} \subseteq \mathbb{R}^{d_X}$. Suppose it has a unique finite minimizer x^*. If one minimizes f with gradient descent $x_{t+1} = x_t - \eta \nabla f(x_t)$, staring from x_0 with a fixed learning rate $\eta \leq \frac{1}{L}$, then we have*

$$f(x_T) \leq f(x^*) + \frac{1}{\eta T}\|x_0 - x^*\|_2^2 \qquad \textit{for all } T > 0.$$

Now let us prove Theorem 6.4.

Proof Static GRW. We first prove the result for static GRW where $q_i^{(t)} = q_i > 0$ for all t. Let $q^* = \min_i q_i$. The minimization objective is $F(\theta) = \sum_{i=1}^n q_i (x_i^\top \theta - y_i)^2$, whose Hessian is $\nabla_\theta^2 F(\theta) = 2\sum_{i=1}^n q_i x_i x_i^\top$. Let $A = \sum_{i=1}^n \|x_i\|_2^2$. Since $q_i \in [0,1]$, for any unit vector $\boldsymbol{v} \in \mathbb{R}^{d_X}$, we have

$$\boldsymbol{v}^\top \nabla_\theta^2 F(\theta) \boldsymbol{v} = 2\sum_{i=1}^n q_i (x_i^\top \boldsymbol{v})^2 \leq 2\sum_{i=1}^n q_i \|x_i\|_2^2 \leq 2A,$$

which by Definition 6.9 implies that F is $2A$-smooth. Thus, we have

$$F(\theta_2) \leq F(\theta_1) + \langle \nabla_\theta F(\theta_1), \theta_2 - \theta_1 \rangle + A\|\theta_2 - \theta_1\|_2^2 \qquad \text{for all } \theta_1, \theta_2 \in \mathbb{R}^{d_X}. \tag{E.1}$$

Denote $g(\theta^{(t)}) = \boldsymbol{X}^\top \theta^{(t)} - \boldsymbol{Y} \in \mathbb{R}^n$. Let $\sqrt{\boldsymbol{Q}} = \operatorname{diag}\big(\sqrt{q_1}, \cdots, \sqrt{q_n}\big)$. Then, we have $F(\theta^{(t)}) = \big\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\big\|_2^2$, which implies that $\nabla F(\theta^{(t)}) = 2\boldsymbol{X}\boldsymbol{Q}g(\theta^{(t)})$. The update rule of static GRW with gradient descent is thus given by

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^n q_i x_i \big(f^{(t)}(x_i) - y_i\big) = \theta^{(t)} - \eta \boldsymbol{X}\boldsymbol{Q}g(\theta^{(t)}).$$

By Eqn. (E.1), we have

$$F(\theta^{(t+1)}) \leq F(\theta^{(t)}) - 2\eta g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{Q}g(\theta^{(t)}) + A\big\|\eta \boldsymbol{X}\boldsymbol{Q}g(\theta^{(t)})\big\|_2^2.$$

Since $x_1, \cdots, x_n$ are linearly independent, $\boldsymbol{X}^\top \boldsymbol{X}$ is a positive definite matrix. Let its smallest eigenvalue be $\lambda_{\min} > 0$. Note that $\big\|\boldsymbol{Q}g(\theta^{(t)})\big\|_2 \geq \sqrt{q^*}\big\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\big\|_2 = \sqrt{q^* F(\theta^{(t)})}$.

Thus, we have $g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{Q} g(\theta^{(t)}) \geq q^* \lambda_{\min} F(\theta^{(t)})$. So we have

$$\begin{aligned}
F(\theta^{(t+1)}) &\leq F(\theta^{(t)}) - 2\eta q^* \lambda_{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X}\sqrt{\boldsymbol{Q}} \right\|_2^2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 \\
&\leq F(\theta^{(t)}) - 2\eta q^* \lambda_{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X}\sqrt{\boldsymbol{Q}} \right\|_F^2 F(\theta^{(t)}) \\
&\leq F(\theta^{(t)}) - 2\eta q^* \lambda_{\min} F(\theta^{(t)}) + A\eta^2 \|\boldsymbol{X}\|_F^2 F(\theta^{(t)}) \\
&= \left(1 - 2\eta q^* \lambda_{\min} + A^2\eta^2\right) F(\theta^{(t)}).
\end{aligned}$$

Let $\eta_0 = \frac{q^* \lambda_{\min}}{A^2}$. For any $\eta \leq \eta_0$, we have $F(\theta^{(t+1)}) \leq (1 - \eta q^* \lambda_{\min}) F(\theta^{(t)})$ for all t, which implies that $F(\theta^{(t)})$ must converge to zero. Since every $q_i > 0$, this implies that the ERM risk must converge to zero.

Dynamic GRW. By Assumption 6.3, for any $\epsilon > 0$, there exists t_ϵ such that for all $t \geq t_\epsilon$, we have $q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon)$ for all i. Let $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and smallest eigenvalues of $\boldsymbol{X}^\top \boldsymbol{X}$, where $\lambda_{\min} > 0$. Fix $\epsilon = \min\left\{ \frac{q^*}{3}, \frac{(q^* \lambda_{\min})^2}{12\lambda_{\max}^2} \right\}$. Then, t_ϵ is also fixed.

Denote $\boldsymbol{Q} = \operatorname{diag}(q_1, \cdots, q_n)$. When $t \geq t_\epsilon$, the update rule of dynamic GRW is $\theta^{(t+1)} = \theta^{(t)} - \eta \boldsymbol{X} \boldsymbol{Q}_\epsilon^{(t)} \left(\boldsymbol{X}^\top \theta^{(t)} - \boldsymbol{Y} \right)$. We use the subscript ϵ to indicate that $\left\| \boldsymbol{Q}_\epsilon^{(t)} - \boldsymbol{Q} \right\|_2 < \epsilon$. Because $q_i + \epsilon\sqrt{(q_i + 3\epsilon) q_i}$ and $q_i - \epsilon \geq \sqrt{(q_i - \epsilon) q_i}$ for all $\epsilon \leq \frac{q_i}{3}$, we can rewrite $\boldsymbol{Q}_\epsilon^{(t)}$ as $\boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \sqrt{\boldsymbol{Q}}$. So by Eqn. (E.1), we have

$$F(\theta^{(t+1)}) \leq F(\theta^{(t)}) - 2\eta g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) + A \left\| \eta \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2.$$

For all $\epsilon < \frac{q_i}{3}$, we have $\sqrt{q_i + 3\epsilon} - \sqrt{q_i} \leq \sqrt{3\epsilon}$ and $\sqrt{q_i} - \sqrt{q_i - 3\epsilon} \leq \sqrt{3\epsilon}$. Thus, we have

$$\begin{aligned}
&\left| g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X} \left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right) \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right| \\
\leq &\left\| \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right) \right\|_2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 \\
\leq &\left\| \sqrt{\boldsymbol{Q}} \right\|_2 \left\| \boldsymbol{X}^\top \boldsymbol{X} \right\|_2 \left\| \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right\|_2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 \\
\leq &\lambda_{\max} \sqrt{3\epsilon} F(\theta^{(t)}).
\end{aligned}$$

Since $g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{Q} g(\theta^{(t)}) \geq q^* \lambda_{\min} F(\theta^{(t)})$, and $\epsilon \leq \frac{(q^* \lambda_{\min})^2}{12\lambda_{\max}^2}$, we have

$$g(\theta^{(t)})^\top \boldsymbol{Q}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \geq \left(q^* \lambda_{\min} - \lambda_{\max} \sqrt{3\epsilon} \right) F(\theta^{(t)}) \geq \frac{1}{2} q^* \lambda_{\min} F(\theta^{(t)}).$$

Thus, for all $\epsilon \leq \frac{1}{3}$, we have

$$\begin{aligned}
F(\theta^{(t+1)}) &\leq F(\theta^{(t)}) - \eta q^* \lambda_{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \right\|_2^2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 \\
&\leq (1 - \eta q^* \lambda_{\min} + A^2\eta^2 (1 + 3\epsilon)) F(\theta^{(t)}) \\
&\leq (1 - \eta q^* \lambda_{\min} + 2A^2\eta^2) F(\theta^{(t)})
\end{aligned}$$

Let $\eta_0 = \frac{q^* \lambda_{\min}}{4A^2}$. For any $\eta \leq \eta_0$, we have $F(\theta^{(t+1)}) \leq (1 - \eta q^* \lambda_{\min}/2) F(\theta^{(t)})$ for all $t \geq t_\epsilon$, which implies that $\lim_{t \to \infty} F(\theta^{(t)}) = 0$. Thus, the ERM risk converges to 0. ■

E.2 Proof of Theorem 6.6

The proof of this theorem is largely based on the following result.

Lemma E.2 (Approximation Theorem)**.** *For a wide NN $f^{(t)}$ trained by any GRW satisfying Assumption 6.3 with the squared loss, let $f_{\text{lin}}^{(t)}(x) = f^{(0)}(x) + \left\langle \theta^{(t)} - \theta^{(0)}, \nabla_\theta f^{(0)}(x) \right\rangle$ be its linearized neural network trained by the same GRW (i.e. $q_i^{(t)}$ are the same for both networks for any i and t). Under the conditions of Theorem 6.6, with a sufficiently small learning rate, for any $\delta > 0$, there exist constants $\tilde{D} > 0$ and $C > 0$ such that as long as $\tilde{d} \geq \tilde{D}$, with probability at least $(1-\delta)$ over random initialization we have: for any test point $x \in \mathbb{R}^d$ such that $\|x\|_2 \leq 1$,*

$$\sup_{t \geq 0} \left| f_{\text{lin}}^{(t)}(x) - f^{(t)}(x) \right| \leq C \tilde{d}^{-1/4}$$

Proof We will use the following short-hand in the proof:

$$\begin{cases} g(\theta^{(t)}) = f^{(t)}(\boldsymbol{X}) - \boldsymbol{Y} \\ J(\theta^{(t)}) = \nabla_\theta f(\boldsymbol{X}; \theta^{(t)}) \in \mathbb{R}^{p \times n} \\ \Theta^{(t)} = J(\theta^{(t)})^\top J(\theta^{(t)}) \end{cases}$$

For any $\epsilon > 0$, there exists t_ϵ such that for all $t \geq t_\epsilon$ and all i, $q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon)$. Let $\boldsymbol{Q} = \text{diag}(q_1, \cdots, q_n)$. Similar to Appendix E.1, we can rewrite $\boldsymbol{Q}^{(t)} = \boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \sqrt{\boldsymbol{Q}}$. The update rule of GRW of wide NN is

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)}) \boldsymbol{Q}^{(t)} g(\theta^{(t)}),$$

and when $t \geq t_\epsilon$, this can be rewritten as

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right]. \tag{E.2}$$

Next, we need three lemmas.

Lemma E.3 ([147], Corollary 5.35)**.** *If $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ is a random matrix whose entries are independent standard normal random variables, then for every $t \geq 0$, with probability at least $1 - 2\exp(-t^2/2)$,*

$$\sqrt{p} - \sqrt{q} - t \leq \lambda_{\min}(\boldsymbol{A}) \leq \lambda_{\max}(\boldsymbol{A}) \leq \sqrt{p} + \sqrt{q} + t.$$

Lemma E.4 (Local Lipschitzness of the Jacobian)**.** *There exists a constant $M > 0$ such that for any $C_0 > 0$ and any $\delta \in (0,1)$, there exists a $\tilde{D}$ such that: If $\tilde{d} \geq \tilde{D}$, then with probability at least $(1-\delta)$ over random initialization, for any x such that $\|x\|_2 \leq 1$, we have*

$$\begin{cases} \left\| \nabla_\theta f(x;\theta) - \nabla_\theta f(x;\tilde{\theta}) \right\|_2 \leq \dfrac{M}{\sqrt[4]{\tilde{d}}} \left\| \theta - \tilde{\theta} \right\|_2 \\ \|\nabla_\theta f(x;\theta)\|_2 \leq M \\ \left\| J(\theta) - J(\tilde{\theta}) \right\|_F \leq \dfrac{M}{\sqrt[4]{\tilde{d}}} \left\| \theta - \tilde{\theta} \right\|_2 \\ \|J(\theta)\|_F \leq M \end{cases}, \qquad \forall \theta, \tilde{\theta} \in B(\theta^{(0)}, C_0), \tag{E.3}$$

where $B(\theta^{(0)}, R) = \left\{ \theta : \left\| \theta - \theta^{(0)} \right\|_2 < R \right\}$.

Proof By Lemma E.3, for any $\delta \in (0,1)$, there exists $\tilde{D} > 0$ and $M_1 > 0$ such that if $\tilde{d} \geq \tilde{D}$, then with probability at least $1-\delta$, the following holds for all θ such that $\left\|\theta - \theta^{(0)}\right\|_2 < C_0$:

$$\forall 0 \leq l \leq L-1 : \left\|\boldsymbol{W}^l\right\|_2 \leq 3\sqrt{\tilde{d}}; \quad \left\|\boldsymbol{W}^L\right\|_2 \leq C_0 \leq 3\tilde{d}^{1/4}; \quad \forall 0 \leq l \leq L : \left\|\beta \boldsymbol{b}^l\right\| \leq M_1\sqrt{\tilde{d}}.$$

With these inequalities, for any x such that $\|x\|_2 \leq 1$, we have

$$\begin{aligned}
\left\|\boldsymbol{h}^1\right\|_2 &= \left\|\frac{1}{\sqrt{d_0}}\boldsymbol{W}^0 x + \beta \boldsymbol{b}^0\right\|_2 \leq \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{W}^0\right\|_2 \|x\|_2 + \left\|\beta \boldsymbol{b}^0\right\|_2 \leq (\frac{3}{\sqrt{d_0}} + M_1)\sqrt{\tilde{d}}; \\
\left\|\boldsymbol{h}^{l+1}\right\|_2 &= \left\|\frac{1}{\sqrt{\tilde{d}}}\boldsymbol{W}^l \boldsymbol{x}^l + \beta \boldsymbol{b}^l\right\|_2 \leq \frac{1}{\sqrt{\tilde{d}}}\left\|\boldsymbol{W}^l\right\|_2 \left\|\boldsymbol{x}^l\right\|_2 + \left\|\beta \boldsymbol{b}^l\right\|_2 \qquad (\forall l \geq 1); \\
\left\|\boldsymbol{x}^l\right\|_2 &= \left\|\sigma(\boldsymbol{h}^l) - \sigma(\boldsymbol{0}^l) + \sigma(\boldsymbol{0}^l)\right\|_2 \leq L_0\left\|\boldsymbol{h}^l\right\|_2 + \sigma(0)\sqrt{\tilde{d}} \qquad (\forall l \geq 1).
\end{aligned}$$

Here, L_0 is the Lipschitz constant of σ and $\sigma(\boldsymbol{0}^l) = (\sigma(0), \cdots, \sigma(0)) \in \mathbb{R}^{d_l}$. Thus, we can prove by induction that there exists an $M_2 > 0$ such that $\left\|\boldsymbol{x}^l\right\|_2 \leq M_2\sqrt{\tilde{d}}$ and $\left\|\boldsymbol{h}^l\right\|_2 \leq M_2\sqrt{\tilde{d}}$ for all $l = 1, \cdots, L$.

Denote $\boldsymbol{\alpha}^l = \nabla_{\boldsymbol{h}^l} f(x) = \nabla_{\boldsymbol{h}^l} \boldsymbol{h}^{L+1}$. For all $1 \leq l \leq L$, we have $\boldsymbol{\alpha}^l = \operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l))\frac{\boldsymbol{W}^{l\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+1}$, $\boldsymbol{\alpha}^{L+1} = 1$ and $\left\|\boldsymbol{\alpha}^L\right\|_2 = \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^L))\frac{\boldsymbol{W}^{L\top}}{\sqrt{\tilde{d}}}\right\|_2 \leq \frac{3}{\sqrt[4]{\tilde{d}}}L_0$. Since σ is L_0-Lipschitz, we have $\dot{\sigma}(x) \leq L_0$ for all $x \in \mathbb{R}$. Thus, we can prove by induction that there exists an $M_3 > 1$ such that $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3/\sqrt[4]{\tilde{d}}$ for all $l = 1, \cdots, L$ (note that this is not true for $L+1$ because $\boldsymbol{\alpha}^{L+1} = 1$).

For $l = 0$, $\nabla_{\boldsymbol{W}^0} f(x) = \frac{1}{\sqrt{d_0}}\boldsymbol{x}^0 \boldsymbol{\alpha}^{1\top}$, so $\|\nabla_{\boldsymbol{W}^l} f(x)\|_2 \leq \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{x}^0\right\|_2 \left\|\boldsymbol{\alpha}^1\right\|_2 \leq \frac{1}{\sqrt{d_0}}M_3/\sqrt[4]{\tilde{d}}$. And for any $l = 1, \cdots, L$, $\nabla_{\boldsymbol{W}^l} f(x) = \frac{1}{\sqrt{\tilde{d}}}\boldsymbol{x}^l \boldsymbol{\alpha}^{l+1}$, so $\|\nabla_{\boldsymbol{W}^l} f(x)\|_2 \leq \frac{1}{\sqrt{\tilde{d}}}\left\|\boldsymbol{x}^l\right\|_2 \left\|\boldsymbol{\alpha}^{l+1}\right\|_2 \leq M_2 M_3$. (Note that if $M_3 > 1$, then $\left\|\boldsymbol{\alpha}^{L+1}\right\|_2 \leq M_3$; and since $\tilde{d} \geq 1$, there is $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3$ for $l \leq L$.) Moreover, for $l = 0, \cdots, L$, $\nabla_{\boldsymbol{b}^l} f(\boldsymbol{x}) = \beta \boldsymbol{\alpha}^{l+1}$, so $\|\nabla_{\boldsymbol{b}^l} f(x)\|_2 \leq \beta M_3$. Thus, there exists an $M_4 > 0$, such that $\|\nabla_\theta f(x)\|_2 \leq M_4/\sqrt{n}$. And since $\|\boldsymbol{x}_i\|_2 \leq 1$ for all i, so $\|J(\theta)\|_F \leq M_4$.

Next, we consider the difference in $\nabla_\theta f(x)$ between θ and $\tilde{\theta}$. Let $\tilde{f}, \tilde{\boldsymbol{W}}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{h}}, \tilde{\boldsymbol{\alpha}}$ be the function and the values corresponding to $\tilde{\theta}$. We have

$$\begin{aligned}
\left\|\boldsymbol{h}^1 - \tilde{\boldsymbol{h}}^1\right\|_2 &= \left\|\frac{1}{\sqrt{d_0}}(\boldsymbol{W}^0 - \tilde{\boldsymbol{W}}^0)x + \beta(\boldsymbol{b}^0 - \tilde{\boldsymbol{b}}^0)\right\|_2 \\
&\leq \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{W}^0 - \tilde{\boldsymbol{W}}^0\right\|_2 \|x\|_2 + \beta\left\|\boldsymbol{b}^0 - \tilde{\boldsymbol{b}}^0\right\|_2 \leq \left(\frac{1}{\sqrt{d_0}} + \beta\right)\left\|\theta - \tilde{\theta}\right\|_2 \\
\left\|\boldsymbol{h}^{l+1} - \tilde{\boldsymbol{h}}^{l+1}\right\|_2 &= \left\|\frac{1}{\sqrt{\tilde{d}}}\boldsymbol{W}^l(\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l) + \frac{1}{\sqrt{\tilde{d}}}(\boldsymbol{W}^l - \tilde{\boldsymbol{W}}^l)\tilde{\boldsymbol{x}}^l + \beta(\boldsymbol{b}^l - \tilde{\boldsymbol{b}}^l)\right\|_2 \\
&\leq \frac{1}{\sqrt{\tilde{d}}}\left\|\boldsymbol{W}^l\right\|_2 \left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 + \frac{1}{\sqrt{\tilde{d}}}\left\|\boldsymbol{W}^l - \tilde{\boldsymbol{W}}^l\right\|_2 \left\|\tilde{\boldsymbol{x}}^l\right\|_2 + \beta\left\|\boldsymbol{b}^l - \tilde{\boldsymbol{b}}^l\right\|_2 \\
&\leq 3\left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 + (M_2 + \beta)\left\|\theta - \tilde{\theta}\right\|_2 \qquad (\forall l \geq 1) \\
\left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 &= \left\|\sigma(\boldsymbol{h}^l) - \sigma(\tilde{\boldsymbol{h}}^l)\right\|_2 \leq L_0\left\|\boldsymbol{h}^l - \tilde{\boldsymbol{h}}^l\right\|_2 \qquad (\forall l \geq 1).
\end{aligned}$$

Thus, we can prove by induction that there exists an $M_5 > 0$ such that $\left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 \leq M_5\left\|\theta - \tilde{\theta}\right\|_2$ for all l.

For $\boldsymbol{\alpha}^l$, we have $\boldsymbol{\alpha}^{L+1} = \tilde{\boldsymbol{\alpha}}^{L+1} = 1$, and for all $l \geq 1$,

$$\begin{aligned}
\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 &= \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l))\frac{\boldsymbol{W}^{l\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+1} - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^l))\frac{\tilde{\boldsymbol{W}}^{l\top}}{\sqrt{\tilde{d}}}\tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\leq \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l))\frac{\boldsymbol{W}^{l\top}}{\sqrt{\tilde{d}}}(\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1})\right\|_2 + \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l))\frac{(\boldsymbol{W}^l - \tilde{\boldsymbol{W}}^l)^\top}{\sqrt{\tilde{d}}}\tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\qquad + \left\|\operatorname{diag}((\dot{\sigma}(\boldsymbol{h}^l) - \dot{\sigma}(\tilde{\boldsymbol{h}}^l)))\frac{\tilde{\boldsymbol{W}}^{l\top}}{\sqrt{\tilde{d}}}\tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\leq 3L_0\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 + \left(M_3L_0\tilde{d}^{-1/2} + 3M_3M_5L_1\tilde{d}^{-1/4}\right)\left\|\theta - \tilde{\theta}\right\|_2,
\end{aligned} \tag{E.4}$$

where L_1 is the Lipschitz constant of $\dot{\sigma}$. In particular, for $l = L$, though $\tilde{\boldsymbol{\alpha}}^{L+1} = 1$, since $\left\|\tilde{\boldsymbol{W}}^L\right\|_2 \leq 3\tilde{d}^{1/4}$, Eqn. (E.4) is still true. Thus, we can prove by induction that there exists an $M_6 > 0$ such that $\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 \leq \frac{M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$ for all $l \geq 1$ (note that this is also true for $l = L+1$).

Thus, for all $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$, and any x such that $\|x\|_2 \leq 1$, we have

$$\begin{aligned}
\left\|\nabla_{\boldsymbol{W}^0} f(x) - \nabla_{\tilde{\boldsymbol{W}}^0}\tilde{f}(x)\right\|_2 &= \frac{1}{\sqrt{d_0}}\left\|x\boldsymbol{\alpha}^{1\top} - x\tilde{\boldsymbol{\alpha}}^{1\top}\right\|_2 \\
&\leq \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{\alpha}^1 - \tilde{\boldsymbol{\alpha}}^1\right\|_2 \\
&\leq \frac{1}{\sqrt{d_0}}\frac{M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2;
\end{aligned}$$

and for $l = 1, \cdots, L$, we have

$$\begin{aligned}
\left\|\nabla_{\boldsymbol{W}^l} f(x) - \nabla_{\tilde{\boldsymbol{W}}^l}\tilde{f}(x)\right\|_2 &= \frac{1}{\sqrt{\tilde{d}}}\left\|x^l\boldsymbol{\alpha}^{l+1\top} - \tilde{x}^l\tilde{\boldsymbol{\alpha}}^{l+1\top}\right\|_2 \\
&\leq \frac{1}{\sqrt{\tilde{d}}}\left(\left\|\boldsymbol{x}^l\right\|_2\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 + \left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2\left\|\tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2\right) \\
&\leq \left(\frac{M_2M_6}{\sqrt[4]{\tilde{d}}} + \frac{M_5M_3}{\sqrt{\tilde{d}}}\right)\left\|\theta - \tilde{\theta}\right\|_2.
\end{aligned}$$

Moreover, for any $l = 0, \cdots, L$, we have

$$\left\|\nabla_{b^l} f(x) - \nabla_{\tilde{b}^l}\tilde{f}(x)\right\|_2 = \beta\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \leq \frac{\beta M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2.$$

Combining all the above, we can see that there exists a constant $M_7 > 0$ such that $\left\|\nabla_\theta f(x) - \nabla_{\tilde{\theta}}\tilde{f}(x)\right\|_2 \leq \frac{M_7}{\sqrt{n}\cdot\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$, so that $\left\|J(\theta) - J(\tilde{\theta})\right\|_F \leq \frac{M_7}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$. ■

Lemma E.5. *There exist constants $M > 0$ and $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0]$, $\eta \leq \eta^*$ and any $\delta > 0$, there exist $R_0 > 0$, $\tilde{D} > 0$ and $B > 1$ such that for any $\tilde{d} \geq \tilde{D}$, the following (i) and (ii) hold with probability at least $(1-\delta)$ over random initialization when applying gradient descent with learning rate η:*

(*i*) *For all $t \le t_\epsilon$, there is*

$$\left\|g(\theta^{(t)})\right\|_2 \le B^t R_0 \tag{E.5}$$

$$\sum_{j=1}^{t} \left\|\theta^{(j)} - \theta^{(j-1)}\right\|_2 \le \eta M R_0 \sum_{j=1}^{t} B^{j-1} < \frac{M B^{t_\epsilon} R_0}{B-1} \tag{E.6}$$

(*ii*) *For all $t \ge t_\epsilon$, we have*

$$\left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \le \left(1 - \frac{\eta q^* \lambda_{\min}}{3}\right)^{t-t_\epsilon} B^{t_\epsilon} R_0 \tag{E.7}$$

$$\begin{aligned}\sum_{j=t_\epsilon+1}^{t} \left\|\theta^{(j)} - \theta^{(j-1)}\right\|_2 &\le \eta\sqrt{1+3\epsilon} M B^{t_\epsilon} R_0 \sum_{j=t_\epsilon+1}^{t} \left(1 - \frac{\eta q^* \lambda_{\min}}{3}\right)^{j-t_\epsilon} \\ &< \frac{3\sqrt{1+3\epsilon} M B^{t_\epsilon} R_0}{q^* \lambda_{\min}}\end{aligned} \tag{E.8}$$

Proof Note that for any x, $f^{(0)}(x) = \beta \boldsymbol{b}^L$ where $\boldsymbol{b}^L$ is sampled from the standard Gaussian distribution. Thus, for any $\delta > 0$, there exists a constant R_0 such that with probability at least $(1 - \delta/3)$ over random initialization, we have $\left\|g(\theta^{(0)})\right\|_2 < R_0$. And by Lemma 6.5, there exists $D_2 \ge 0$ such that for any $\tilde{d} \ge D_2$, with probability at least $(1 - \delta/3)$, we have $\left\|\Theta - \Theta^{(0)}\right\| \le \frac{q^* \lambda_{\min}}{3}$.

Let M be the constant in Lemma E.4. Let $\epsilon_0 = \frac{(q^* \lambda_{\min})^2}{108 M^4}$, $B = 1 + \eta^* M^2$, and $C_0 = \frac{M B^{t_\epsilon} R_0}{B-1} + \frac{3\sqrt{1+3\epsilon} M B^{t_\epsilon} R_0}{q^* \lambda_{\min}}$. By Lemma E.4, there exists $D_1 > 0$ such that with probability at least $(1 - \delta/3)$, for any $\tilde{d} \ge D_1$, Eqn. (E.3) is true for all $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$. By union bound, with probability at least $1 - \delta$, all the three above inequalities holds.

Let us prove Eqns. (E.5) and (E.6) by induction. They are obviously true for $t = 0$. Suppose they are true for t. Then, for $t + 1$, we have

$$\begin{aligned}\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2 &\le \eta \left\|J(\theta^{(t)}) \boldsymbol{Q}^{(t)}\right\|_2 \left\|g(\theta^{(t)})\right\|_2 \le \eta \left\|J(\theta^{(t)}) \boldsymbol{Q}^{(t)}\right\|_F \left\|g(\theta^{(t)})\right\|_2 \\ &\le \eta \left\|J(\theta^{(t)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 \le M \eta B^t R_0,\end{aligned}$$

which means that Eqn. (E.6) is also true for $t + 1$. In terms of Eqn. (E.5), we have

$$\begin{aligned}\left\|g(\theta^{(t+1)})\right\|_2 &= \left\|g(\theta^{(t+1)}) - g(\theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &= \left\|J(\tilde{\theta}^{(t)})^\top (\theta^{(t+1)} - \theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &= \left\|-\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) \boldsymbol{Q}^{(t)} g(\theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &\le \left\|\boldsymbol{I} - \eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) \boldsymbol{Q}^{(t)}\right\|_2 \left\|g(\theta^{(t)})\right\|_2 \\ &\le \left(1 + \left\|\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) \boldsymbol{Q}^{(t)}\right\|_2\right) \left\|g(\theta^{(t)})\right\|_2 \\ &\le \left(1 + \eta \left\|J(\tilde{\theta}^{(t)})\right\|_F \left\|J(\theta^{(t)})\right\|_F\right) \left\|g(\theta^{(t)})\right\|_2 \\ &\le (1 + \eta^* M^2) \left\|g(\theta^{(t)})\right\|_2 \le B^{t+1} R_0.\end{aligned}$$

Hence, Eqns. (E.5) and (E.6) are true for all $t \le t_\epsilon$, which implies that $\left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t_\epsilon)})\right\|_2 \le \left\|g(\theta^{(t_\epsilon)})\right\|_2 \le B^{t_\epsilon} R_0$. Thus, Eqn. (E.7) is true for $t = t_\epsilon$. And Eqn. (E.8) is obviously true for $t = t_\epsilon$.

Next, let us prove Eqns. (E.7) and (E.8) by induction. Suppose they are true for t. By Eqn. (E.2), for $t+1$ we have

$$\begin{aligned}
\left\|\theta^{(t+1)}-\theta^{(t)}\right\|_2 &\le \eta\left\|J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&\le \eta\left\|J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_F\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&\le \eta\sqrt{1+3\epsilon}\left\|J(\theta^{(t)})\right\|_F\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&\le M\eta\sqrt{1+3\epsilon}\left(1-\frac{\eta q^*\lambda_{\min}}{3}\right)^{t-t_\epsilon}B^{t_\epsilon}R_0,
\end{aligned}$$

which implies that Eqn. (E.8) holds for $t+1$. In terms of Eqn. (E.7), we have

$$\begin{aligned}
\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t+1)})\right\|_2 &= \left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t+1)})-\sqrt{\boldsymbol{Q}}g(\theta^{(t)})+\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&= \left\|\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top(\theta^{(t+1)}-\theta^{(t)})+\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&= \left\|-\eta\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)})+\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&\le \left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2\\
&\le \left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2\left(1-\frac{\eta q^*\lambda_{\min}}{3}\right)^t R_0,
\end{aligned}$$

where $\tilde\theta^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(t+1)}$. Now we prove that

$$\left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \le 1-\frac{\eta q^*\lambda_{\min}}{3}. \tag{E.9}$$

For any unit vector $\boldsymbol{v}\in\mathbb{R}^n$, we have

$$\boldsymbol{v}^\top(\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}})\boldsymbol{v} = 1-\eta\boldsymbol{v}^\top\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\boldsymbol{v},$$

where $\left\|\sqrt{\boldsymbol{Q}}\boldsymbol{v}\right\|_2\in[\sqrt{q^*},1]$, so for any $\eta\le\eta^*$, $\boldsymbol{v}^\top(\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}})\boldsymbol{v}\in[0,1-\eta\lambda_{\min}q^*]$, which implies that $\left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2\le 1-\eta\lambda_{\min}q^*$. Thus, we have

$$\begin{aligned}
&\left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}}\right\|_2\\
\le&\left\|\boldsymbol{I}-\eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2+\eta\left\|\sqrt{\boldsymbol{Q}}(\Theta-\Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_2+\eta\left\|\sqrt{\boldsymbol{Q}}(J(\theta^{(0)})^\top J(\theta^{(0)})-J(\tilde\theta^{(t)})^\top J(\theta^{(t)}))\sqrt{\boldsymbol{Q}}\right\|_2\\
\le&1-\eta\lambda_{\min}q^*+\eta\left\|\sqrt{\boldsymbol{Q}}(\Theta-\Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_F+\eta\left\|\sqrt{\boldsymbol{Q}}(J(\theta^{(0)})^\top J(\theta^{(0)})-J(\tilde\theta^{(t)})^\top J(\theta^{(t)}))\sqrt{\boldsymbol{Q}}\right\|_F\\
\le&1-\eta\lambda_{\min}q^*+\eta\left\|\Theta-\Theta^{(0)}\right\|_F+\eta\left\|J(\theta^{(0)})^\top J(\theta^{(0)})-J(\tilde\theta^{(t)})^\top J(\theta^{(t)})\right\|_F\\
\le&1-\eta\lambda_{\min}q^*+\frac{\eta q^*\lambda_{\min}}{3}+\frac{\eta M^2}{\sqrt[4]{\tilde d}}\left(\left\|\theta^{(t)}-\theta^{(0)}\right\|_2+\left\|\tilde\theta^{(t)}-\theta^{(0)}\right\|_2\right)\le 1-\frac{\eta q^*\lambda_{\min}}{2}
\end{aligned}$$

for all $\tilde{d} \geq \max\left\{D_1, D_2, \left(\frac{12M^2C_0}{q^*\lambda_{\min}}\right)^4\right\}$. This implies that

$$\begin{aligned}
&\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \\
\leq\ & 1 - \frac{\eta q^*\lambda_{\min}}{2} + \left\|\eta\sqrt{\boldsymbol{Q}}J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}}\right)\right\|_2 \\
\leq\ & 1 - \frac{\eta q^*\lambda_{\min}}{2} + \eta M^2\sqrt{3\epsilon} \leq 1 - \frac{\eta q^*\lambda_{\min}}{3}
\end{aligned}$$

holds for all $\epsilon \leq \epsilon_0$. Thus, Eqn. (E.7) holds for $t+1$. ■

Now let us return to the proof of Lemma E.2. Choose and fix an ϵ such that $\epsilon < \min\{\epsilon_0, \frac{1}{3}\left(\frac{q^*\lambda_{\min}}{3\lambda_{\max}+q^*\lambda_{\min}}\right)^2\}$, where ϵ_0 is given by Theorem E.5. Then, t_ϵ is also fixed. There exists $\tilde{D} \geq 0$ such that for any $\tilde{d} \geq \tilde{D}$, with probability at least $(1-\delta)$, the inequalities in Lemmas E.4 and E.5 hold, and $\left\|\Theta - \Theta^{(0)}\right\|_F \leq \frac{q^*\lambda_{\min}}{3}$. This implies that

$$\left\|\Theta^{(0)}\right\|_2 \leq \|\Theta\|_2 + \left\|\Theta - \Theta^{(0)}\right\|_F \leq \lambda_{\max} + \frac{q^*\lambda_{\min}}{3}.$$

We still denote $B = 1 + \eta^* M^2$ and $C_0 = \frac{MB^{t_\epsilon}R_0}{B-1} + \frac{3\sqrt{1+3\epsilon}MB^{t_\epsilon}R_0}{q^*\lambda_{\min}}$. Lemma E.5 guarantees that for all t, we have $\theta^{(t)} \in B(\theta^{(0)}, C_0)$. Thus, we have

$$\begin{aligned}
\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}}\right\|_2 &\leq \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2 + \eta\left\|\sqrt{\boldsymbol{Q}}(\Theta - \Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_2 \\
&\leq 1 - \eta\lambda_{\min}q^* + \frac{\eta q^*\lambda_{\min}}{3} = 1 - \frac{2\eta q^*\lambda_{\min}}{3}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 &\leq \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}}\right\|_2 + \left\|\eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}}\right)\right\|_2 \\
&\leq 1 - \frac{2\eta q^*\lambda_{\min}}{3} + \eta(\lambda_{\max} + \frac{q^*\lambda_{\min}}{3})\sqrt{3\epsilon}.
\end{aligned}$$

Thus, for all $\epsilon < \frac{1}{3}\left(\frac{q^*\lambda_{\min}}{3\lambda_{\max}+q^*\lambda_{\min}}\right)^2$, we have

$$\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \leq 1 - \frac{\eta q^*\lambda_{\min}}{3}. \tag{E.10}$$

The update rule of the GRW for the linearized neural network is:

$$\theta_{\text{lin}}^{(t+1)} = \theta_{\text{lin}}^{(t)} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}) \tag{E.11}$$

where we use the subscript "lin" to denote the linearized neural network, and with a slight abuse of notion denote $g_{\text{lin}}(\theta^{(t)}) = g(\theta_{\text{lin}}^{(t)})$.

First, let us consider the training data $\boldsymbol{X}$. Denote $\Delta_t = g_{\text{lin}}(\theta^{(t)}) - g(\theta^{(t)})$. We have

$$\begin{cases}
g_{\text{lin}}(\theta^{(t+1)}) - g_{\text{lin}}(\theta^{(t)}) = -\eta J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}) \\
g(\theta^{(t+1)}) - g(\theta^{(t)}) = -\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)})
\end{cases}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(t+1)}$. Thus, we have

$$\begin{aligned}\Delta_{t+1} - \Delta_t =& \eta \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \\ & - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \Delta_t.\end{aligned}$$

By Lemma E.4, we have

$$\begin{aligned}& \left\| J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right\|_F \\ \leq & \left\| \left(J(\tilde{\theta}^{(t)}) - J(\theta^{(0)}) \right)^\top J(\theta^{(t)}) \right\|_F + \left\| J(\theta^{(0)})^\top \left(J(\theta^{(t)}) - J(\theta^{(0)}) \right) \right\|_F \\ \leq & 2M^2 C_0 \tilde{d}^{-1/4},\end{aligned}$$

which implies that for all $t < t_\epsilon$, we have

$$\begin{aligned}& \|\Delta_{t+1}\|_2 \\ \leq & \left\| \left[\boldsymbol{I} - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \right] \Delta_t \right\|_2 + \left\| \eta \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \right\|_2 \\ \leq & \left\| \boldsymbol{I} - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \right\|_F \|\Delta_t\|_2 + \eta \left\| J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right\|_F \left\| g(\theta^{(t)}) \right\|_2 \\ \leq & (1 + \eta M^2) \|\Delta_t\|_2 + 2\eta M^2 C_0 B^t R_0 \tilde{d}^{-1/4} \\ \leq & B \|\Delta_t\|_2 + 2\eta M^2 C_0 B^t R_0 \tilde{d}^{-1/4}.\end{aligned}$$

This implies that

$$B^{-(t+1)} \|\Delta_{t+1}\|_2 \leq B^{-t} \|\Delta_t\|_2 + 2\eta M^2 C_0 B^{-1} R_0 \tilde{d}^{-1/4}.$$

Since $\Delta_0 = 0$, it follows that for all $t \leq t_\epsilon$, we have

$$\|\Delta_t\|_2 \leq 2t\eta M^2 C_0 B^{t-1} R_0 \tilde{d}^{-1/4},$$

and in particular, we have

$$\left\| \sqrt{\boldsymbol{Q}} \Delta_{t_\epsilon} \right\|_2 \leq \|\Delta_{t_\epsilon}\|_2 \leq 2t_\epsilon \eta M^2 C_0 B^{t_\epsilon - 1} R_0 \tilde{d}^{-1/4}.$$

For $t \geq t_\epsilon$, by Eqn. (E.2), we have

$$\begin{aligned}\sqrt{\boldsymbol{Q}} \Delta_{t+1} - \sqrt{\boldsymbol{Q}} \Delta_t =& \eta \sqrt{\boldsymbol{Q}} \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right] \\ & - \eta \sqrt{\boldsymbol{Q}} J(\theta^{(0)})^\top J(\theta^{(0)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} \Delta_t \right].\end{aligned}$$

Let $\boldsymbol{A} = \boldsymbol{I} - \eta \sqrt{\boldsymbol{Q}} J(\theta^{(0)})^\top J(\theta^{(0)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} = \boldsymbol{I} - \eta \sqrt{\boldsymbol{Q}} \Theta^{(0)} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}$. Then, we have

$$\sqrt{\boldsymbol{Q}} \Delta_{t+1} = \boldsymbol{A} \sqrt{\boldsymbol{Q}} \Delta_t + \eta \sqrt{\boldsymbol{Q}} \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left(\sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right).$$

Let $\gamma = 1 - \frac{\eta q^* \lambda_{\min}}{3} < 1$. Combining the above with Lemma E.5 and Eqn. (E.10), we have

$$\begin{aligned}& \left\| \sqrt{\boldsymbol{Q}} \Delta_{t+1} \right\|_2 \\ \leq & \|\boldsymbol{A}\|_2 \left\| \sqrt{\boldsymbol{Q}} \Delta_t \right\|_2 + \eta \left\| \sqrt{\boldsymbol{Q}} \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \right\|_2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2 \\ \leq & \gamma \left\| \sqrt{\boldsymbol{Q}} \Delta_t \right\|_2 + \eta \left\| J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right\|_F \sqrt{1 + 3\epsilon} \gamma^{t - t_\epsilon} B^{t_\epsilon} R_0 \\ \leq & \gamma \left\| \sqrt{\boldsymbol{Q}} \Delta_t \right\|_2 + 2\eta M^2 C_0 \sqrt{1 + 3\epsilon} \gamma^{t - t_\epsilon} B^{t_\epsilon} R_0 \tilde{d}^{-1/4},\end{aligned}$$

which implies that

$$\gamma^{-(t+1)}\left\|\sqrt{\boldsymbol{Q}}\Delta_{t+1}\right\|_2 \le \gamma^{-t}\left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 + 2\eta M^2 C_0\sqrt{1+3\epsilon}\gamma^{-1-t_\epsilon}B^{t_\epsilon}R_0\tilde{d}^{-1/4}.$$

Next, we consider an arbitrary test point x such that $\|x\|_2 \le 1$. Denote $\delta_t = f_{\text{lin}}^{(t)}(x) - f^{(t)}(x)$. Then, we have

$$\begin{cases} f_{\text{lin}}^{(t+1)}(x) - f_{\text{lin}}^{(t)}(x) = -\eta\nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}); \\ f^{(t+1)}(x) - f^{(t)}(x) = -\eta\nabla_\theta f(x;\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)}). \end{cases}$$

Therefore, we have

$$\begin{aligned}\delta_{t+1} - \delta_t =& \eta\left[\nabla_\theta f(x;\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - \nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\right]\boldsymbol{Q}^{(t)}g(\theta^{(t)}) \\ & - \eta\nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}\Delta_t.\end{aligned}$$

For $t \le t_\epsilon$, we have

$$\begin{aligned}\|\delta_t\|_2 \le& \eta\sum_{s=0}^{t-1}\left\|\left[\nabla_\theta f(x;\tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\right]\boldsymbol{Q}^{(s)}\right\|_2\left\|g(\theta^{(s)})\right\|_2 \\ & + \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(s)}\right\|_2\|\Delta_s\|_2 \\ \le& \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(x;\tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\right\|_F\left\|g(\theta^{(s)})\right\|_2 \\ & + \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(x;\theta^{(0)})\right\|_2\left\|J(\theta^{(0)})\right\|_F\|\Delta_s\|_2 \\ \le& 2\eta M^2 C_0\tilde{d}^{-1/4}\sum_{s=0}^{t-1}B^s R_0 + \eta M^2\sum_{s=0}^{t-1}(2s\eta M^2 C_0 B^{s-1}R_0\tilde{d}^{-1/4}).\end{aligned}$$

Hence, there exists a constant C_1 such that $\|\delta_{t_\epsilon}\|_2 \le C_1\tilde{d}^{-1/4}$. Then, for $t > t_\epsilon$, we have

$$\begin{aligned}\|\delta_t\|_2 - \|\delta_{t_\epsilon}\|_2 \le& \eta\sum_{s=t_\epsilon}^{t-1}\left\|\left[\nabla_\theta f(x;\tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\right]\sqrt{\boldsymbol{Q}_{3\epsilon}^{(s)}}\right\|_2\left\|\sqrt{\boldsymbol{Q}}g(\theta^{(s)})\right\|_2 \\ & + \eta\sum_{s=t_\epsilon}^{t-1}\left\|\nabla_\theta f(x;\theta^{(0)})^\top J(\theta^{(0)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(s)}}\right\|_2\left\|\sqrt{\boldsymbol{Q}}\Delta_s\right\|_2 \\ \le& 2\eta M^2 C_0\tilde{d}^{-1/4}\sqrt{1+3\epsilon}\sum_{s=t_\epsilon}^{t-1}\gamma^{s-t_\epsilon}B^{t_\epsilon}R_0 \\ & + \eta M^2\sqrt{1+3\epsilon}\sum_{s=t_\epsilon}^{t-1}\left(2\gamma^{s-t_\epsilon}\eta M^2 C_0 B^{t_\epsilon}R_0\left[t_\epsilon B^{-1} + \sqrt{1+3\epsilon}\gamma^{-1}(s-t_\epsilon)\right]\tilde{d}^{-1/4}\right).\end{aligned}$$

Note that $\sum_{t=0}^{\infty}t\gamma^t$ is finite as long as $\gamma \in (0,1)$. Therefore, there is a constant C such that for any t, $\|\delta_t\|_2 \le C\tilde{d}^{-1/4}$ with probability at least $(1-\delta)$ for any $\tilde{d} \ge \tilde{D}$. ■

Now, let us finish the proof of Theorem 6.6.

Proof Consider the linearized neural network $f_{\text{lin}}^{(t)}(x)$ defined in Eqn. (6.4). If we view $\left\{\nabla_\theta f^{(0)}(x_i)\right\}_{i=1}^n$ as inputs and $\left\{y_i - f^{(0)}(x_i) + \left\langle\theta^{(0)}, \nabla_\theta f^{(0)}(x_i)\right\rangle\right\}_{i=1}^n$ as the targets, then the linearized neural network is a linear model. Thus by Theorem 6.4, we have the following corollary.

Corollary E.6. *If $\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)$ are linearly independent, then there exists $\eta_0 > 0$ such that when $f_{\text{lin}}^{(t)}(x)$ is trained with any GRW method that satisfies Assumption 6.3, $\theta^{(t)}$ converges to a unique θ^* that does not depend on the sample weights $q_i^{(t)}$.*

Let $\eta_1 = \min\{\eta_0, \eta^*\}$, where η_0 is given by Corollary E.6. Let $f_{\text{lin}}^{(t)}(x)$ and $f_{\text{linERM}}^{(t)}(x)$ be the linearized neural networks of $f^{(t)}(x)$ and $f_{\text{ERM}}^{(t)}(x)$ (which are two networks trained with GRW and ERM), respectively. By Lemma E.2, for any $\delta \in (0,1)$, there exists $\tilde{D} > 0$ and a constant C such that with probability at least $1-\delta$, the following holds:

$$\begin{cases} \sup\limits_{t\geq 0}\left|f_{\text{lin}}^{(t)}(x) - f^{(t)}(x)\right| \leq C\tilde{d}^{-1/4}; \\ \sup\limits_{t\geq 0}\left|f_{\text{linERM}}^{(t)}(x) - f_{\text{ERM}}^{(t)}(x)\right| \leq C\tilde{d}^{-1/4}. \end{cases}$$

By Corollary E.6, we have

$$\lim_{t\to\infty}\left|f_{\text{lin}}^{(t)}(x) - f_{\text{linERM}}^{(t)}(x)\right| = 0.$$

Summing the above yields

$$\limsup_{t\to\infty}\left|f^{(t)}(x) - f_{\text{ERM}}^{(t)}(x)\right| \leq 2C\tilde{d}^{-1/4},$$

as desired. ■

E.3 Proof of Theorem 6.7

Similar to Theorem 6.6, proving this result needs a slightly different approximation theorem. We start with two necessary propositions.

Proposition E.7 ([77], Proposition 1)**.** *If σ is Lipschitz and $d_l \to \infty$ for $l = 1, \cdots, L$ sequentially, then for all $l = 1, \cdots, L$, the distribution of a single element of $\boldsymbol{h}^l$ converges in probability to a zero-mean Gaussian process of covariance Σ^l that is defined recursively by:*

$$\begin{aligned} \Sigma^1(x, x') &= \frac{1}{d_0} x^\top x' + \beta^2; \\ \Sigma^l(x, x') &= \mathbb{E}_f[\sigma(f(x))\sigma(f(x'))] + \beta^2, \end{aligned}$$

where f is sampled from a zero-mean Gaussian process of covariance $\Sigma^{(l-1)}$.

Proposition E.8. *For any positive definite symmetric matrix $\boldsymbol{H} \in \mathbb{R}^{n\times n}$, denote its largest and smallest eigenvalues by $\lambda_{\max}$ and $\lambda_{\min}$. Then, for any positive semi-definite diagonal matrix $\boldsymbol{Q} = \text{diag}(q_1, \cdots, q_n)$, $\boldsymbol{HQ}$ has n eigenvalues that all lie in $[\min_i q_i \cdot \lambda_{\min}, \max_i q_i \cdot \lambda_{\max}]$.*

Proof $\boldsymbol{H}$ is a positive definite symmetric matrix, so there exists $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{H} = \boldsymbol{A}^\top \boldsymbol{A}$, and $\boldsymbol{A}$ is full-rank. First, any eigenvalue of $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top$ is also an eigenvalue of $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$, because for any eigenvalue λ of $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top$ we have some $\boldsymbol{v} \neq 0$ such that $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top\boldsymbol{v} = \lambda\boldsymbol{v}$. Multiplying both sides by $\boldsymbol{A}^\top$ on the left yields $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}(\boldsymbol{A}^\top\boldsymbol{v}) = \lambda(\boldsymbol{A}^\top\boldsymbol{v})$ which implies that λ is also an eigenvalue of $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$ because $\boldsymbol{A}^\top\boldsymbol{v} \neq 0$ as $\lambda\boldsymbol{v} \neq 0$.

Second, by condition we know that the eigenvalues of $\boldsymbol{A}^\top\boldsymbol{A}$ are all in $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min} > 0$, which implies for any unit vector $\boldsymbol{v}$, $\boldsymbol{v}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{v} \in [\lambda_{\min}, \lambda_{\max}]$, which is equivalent to $\|\boldsymbol{A}\boldsymbol{v}\|_2 \in [\sqrt{\lambda_{\min}}, \sqrt{\lambda_{\max}}]$. Thus, $\boldsymbol{v}^\top\boldsymbol{A}^\top\boldsymbol{Q}\boldsymbol{A}\boldsymbol{v} \in [\lambda_{\min}\min_i q_i, \lambda_{\max}\max_i q_i]$, which implies that the eigenvalues of $\boldsymbol{A}^\top\boldsymbol{Q}\boldsymbol{A}$ are all in $[\lambda_{\min}\min_i q_i, \lambda_{\max}\max_i q_i]$.

Thus, the eigenvalues of $\boldsymbol{H}\boldsymbol{Q} = \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$ are all in $[\lambda_{\min}\min_i q_i, \lambda_{\max}\max_i q_i]$. ■

Lemma E.9 (Approximation Theorem for Regularized GRW)**.** *For a wide fully-connected neural network f, denote $J(\theta) = \nabla_\theta f(\boldsymbol{X};\theta) \in \mathbb{R}^{p\times n}$ and $g(\theta) = \nabla_{\hat{y}}\ell(f(\boldsymbol{X};\theta), \boldsymbol{Y}) \in \mathbb{R}^n$. Given that the loss function ℓ satisfies: $\nabla_\theta g(\theta) = J(\theta)\boldsymbol{U}(\theta)$ for any θ, and $\boldsymbol{U}(\theta)$ is a positive semi-definite diagonal matrix whose elements are uniformly bounded, we have: for any GRW that minimizes the regularized weighted empirical risk Eqn.* (6.5) *with a sufficiently small learning rate η, there is: for a sufficiently large $\tilde{d}$, with high probability over random initialization, on any test point x such that $\|x\|_2 \leq 1$, we have*

$$\sup_{t\geq 0}\left|f_{\text{lin}reg}^{(t)}(x) - f_{reg}^{(t)}(x)\right| \leq C\tilde{d}^{-1/4}, \tag{E.12}$$

where both $f_{\text{lin}reg}^{(t)}$ and $f_{reg}^{(t)}$ are trained by the same regularized GRW and start from the same initial point.

Proof Without loss of generality, assume that all elements of $\boldsymbol{U}(\theta)$ are in $[0,1]$ for all θ, and set $\eta \leq (\mu + \lambda_{\min} + \lambda_{\max})^{-1}$. If the elements of $\boldsymbol{U}(\theta)$ are bounded by $[0, C]$, then we can set $\eta \leq (\mu + C\lambda_{\min} + C\lambda_{\max})^{-1}$ and prove the result in the same way.

With L_2 penalty, the update rule of the GRW for the neural network is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)}) - \eta\mu(\theta^{(t)} - \theta^{(0)}). \tag{E.13}$$

And the update rule for the linearized neural network is:

$$\theta_{\text{lin}}^{(t+1)} = \theta_{\text{lin}}^{(t)} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}g(\theta_{\text{lin}}^{(t)}) - \eta\mu(\theta_{\text{lin}}^{(t)} - \theta^{(0)}). \tag{E.14}$$

By Proposition E.7, $f(x;\theta)$ converges in probability to a zero-mean Gaussian process. Thus, for any $\delta > 0$, there exists a constant $R_0 > 0$ such that with probability at least $(1-\delta/3)$, $\left\|g(\theta^{(0)})\right\|_2 < R_0$. Let M be given by Lemma E.4. Denote $A = \eta M R_0$, and let $C_0 = \frac{4A}{\eta\mu}$ be given by Lemma E.4. Note that Lemma E.4 only depends on the network structure, but does not depend on the update rule, so we can still use the lemma here. By Lemma E.4, there exists D_1 such that for all $\tilde{d} \geq D_1$, with probability at least $(1-\delta/3)$, Eqn. (E.3) holds.

Similar to the proof of Proposition E.8, we can show that for any $\tilde{\theta}$, all the non-zero eigenvalues of $J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})J(\theta^{(0)})^\top$ are also eigenvalues of $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})$. This is because for any eigenvalue $\lambda \neq 0$ such that $J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})J(\theta^{(0)})^\top\boldsymbol{v} = \lambda\boldsymbol{v}$ for some $\boldsymbol{v}$, we must have $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})(J(\theta^{(0)})^\top\boldsymbol{v}) = \lambda(J(\theta^{(0)})^\top\boldsymbol{v})$, and $J(\theta^{(0)})^\top\boldsymbol{v} \neq 0$ since $\lambda\boldsymbol{v} \neq 0$, so λ is also an eigenvalue of $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})$. On the other hand, by Lemma 6.5, $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})$ converges in probability to $\Theta\boldsymbol{Q}^{(t)}U(\tilde{\theta})$ whose eigenvalues are all in $[0, \lambda_{\max}]$ by Proposition E.8. Hence, there exists D_2 such that for all

$\tilde{d} \geq D_2$, with probability at least $(1-\delta/3)$, the eigenvalues of $J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta})J(\theta^{(0)})^\top$ are all in $[0, \lambda_{\max}+\lambda_{\min}]$ for all t.

By union bound, with probability at least $1-\delta$, all the above "$(1-\delta/3)$" statements are true. Now we prove that there exists D_0 such that for all $\tilde{d} \geq D_0$, $\sup_{t\geq 0}\left\|\theta^{(t)}-\theta^{(0)}\right\|_2$ is bounded with high probability. Denote $a_t = \theta^{(t)}-\theta^{(0)}$. By Eqn. (E.13), we have

$$\begin{aligned} a_{t+1} =& (1-\eta\mu)a_t - \eta[J(\theta^{(t)})-J(\theta^{(0)})]\boldsymbol{Q}^{(t)}g(\theta^{(t)}) \\ & - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}[g(\theta^{(t)})-g(\theta^{(0)})] - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}g(\theta^{(0)}), \end{aligned}$$

which implies that

$$\begin{aligned} \|a_{t+1}\|_2 \leq & \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\tilde{\theta}^{(t)})^\top\right\|_2 \|a_t\|_2 \\ & + \eta\left\|J(\theta^{(t)})-J(\theta^{(0)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 + \eta\left\|J(\theta^{(0)})\right\|_F \left\|g(\theta^{(0)})\right\|_2, \end{aligned} \tag{E.15}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(0)}$. Our choice of η ensures that $\eta\mu < 1$.

Next, we prove by induction that $\|a_t\|_2 < C_0$. It is true for $t=0$. Suppose $\|a_t\|_2 < C_0$, and consider a_{t+1}. Let us look at the three terms on the right-hand side of Eqn. (E.15). For the first term, we have

$$\begin{aligned} & \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\tilde{\theta}^{(t)})^\top\right\|_2 \\ \leq & (1-\eta\mu)\left\|\boldsymbol{I} - \frac{\eta}{1-\eta\mu}J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\theta^{(0)})^\top\right\|_2 \\ & + \eta\left\|J(\theta^{(0)})\right\|_F \left\|J(\tilde{\theta}^{(t)})-J(\theta^{(0)})\right\|_F. \end{aligned}$$

Since $\eta/(1-\eta\mu) \leq (\lambda_{\min}+\lambda_{\max})^{-1}$ by our choice of η, we have

$$\left\|\boldsymbol{I} - \frac{\eta}{1-\eta\mu}J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\theta^{(0)})^\top\right\|_2 \leq 1.$$

On the other hand, since $\|a_t\|_2 < C_0$, we have $\left\|J(\theta^{(0)})\right\|_F \left\|J(\tilde{\theta}^{(t)})-J(\theta^{(0)})\right\|_F \leq \frac{M^2}{\sqrt[4]{\tilde{d}}}C_0$ by Eqn. (E.3). Therefore, there exists D_3 such that for all $\tilde{d} \geq D_3$,

$$\left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\tilde{\theta}^{(t)})^\top\right\|_2 \leq 1 - \frac{\eta\mu}{2}. \tag{E.16}$$

For the second term, we have

$$\begin{aligned} \left\|g(\theta^{(t)})\right\|_2 \leq & \left\|g(\theta^{(t)})-g(\theta^{(0)})\right\|_2 + \left\|g(\theta^{(0)})\right\|_2 \\ \leq & \left\|J(\tilde{\theta}^{(t)})\right\|_2 \left\|U(\tilde{\theta}^{(t)})\right\|_2 \left\|\theta^{(t)}-\theta^{(0)}\right\|_2 + R_0 \leq MC_0 + R_0. \end{aligned} \tag{E.17}$$

For the third term, we have

$$\eta\left\|J(\theta^{(0)})\right\|_F \left\|g(\theta^{(0)})\right\|_2 \leq \eta M R_0 = A.$$

Thus, we have

$$\|a_{t+1}\|_2 \leq \left(1-\frac{\eta\mu}{2}\right)\|a_t\|_2 + \frac{\eta M(MC_0+R_0)}{\sqrt[4]{\tilde{d}}} + A.$$

Thus, there exists D_4 such that for all $\tilde{d} \geq D_4$, $\|a_{t+1}\|_2 \leq \left(1 - \frac{\eta\mu}{2}\right)\|a_t\|_2 + 2A$. This shows that if $\|a_t\|_2 < C_0$ is true, then $\|a_{t+1}\|_2 < C_0$ will also be true.

In conclusion, for all $\tilde{d} \geq D_0 = \max\{D_1, D_2, D_3, D_4\}$, $\left\|\theta^{(t)} - \theta^{(0)}\right\|_2 < C_0$ is true for all t. This also implies that for $C_1 = MC_0 + R_0$, we have $\left\|g(\theta^{(t)})\right\|_2 \leq C_1$ for all t by Eqn. (E.17). Similarly, we can prove that $\|\theta_{\text{lin}}^{(t)} - \theta^{(0)}\|_2 < C_0$ for all t.

Second, let $\Delta_t = \theta_{\text{lin}}^{(t)} - \theta^{(t)}$. Then, we have

$$\Delta_{t+1} - \Delta_t = \eta(J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)}) - J(\theta^{(0)})\boldsymbol{Q}^{(t)}g(\theta_{\text{lin}}^{(t)}) - \mu\Delta_t),$$

which implies that

$$\Delta_{t+1} = \left[(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\tilde{\theta}^{(t)})^\top\right]\Delta_t + \eta(J(\theta^{(t)}) - J(\theta^{(0)}))\boldsymbol{Q}^{(t)}g(\theta^{(t)}),$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta_{\text{lin}}^{(t)}$. By Eqn. (E.16), with probability at least $(1-\delta)$ for all $\tilde{d} \geq D_0$, we have

$$\begin{aligned}\|\Delta_{t+1}\|_2 &\leq \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}U(\tilde{\theta}^{(t)})J(\tilde{\theta}^{(t)})^\top\right\|_2 \|\Delta_t\|_2 + \eta\left\|J(\theta^{(t)}) - J(\theta^{(0)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 \\ &\leq \left(1 - \frac{\eta\mu}{2}\right)\|\Delta_t\|_2 + \eta\frac{M}{\sqrt[4]{\tilde{d}}}C_0C_1.\end{aligned}$$

Again, as $\Delta_0 = 0$, we can prove by induction that for all t,

$$\|\Delta_t\|_2 < \frac{2MC_0C_1}{\mu}\tilde{d}^{-1/4}.$$

For any test point x such that $\|x\|_2 \leq 1$, we have

$$\begin{aligned}\left|f_{\text{reg}}^{(t)}(x) - f_{\text{linreg}}^{(t)}(x)\right| &= \left|f(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta_{\text{lin}}^{(t)})\right| \\ &\leq \left|f(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta^{(t)})\right| + \left|f_{\text{lin}}(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta_{\text{lin}}^{(t)})\right| \\ &\leq \left|f(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta^{(t)})\right| + \left\|\nabla_\theta f(x;\theta^{(0)})\right\|_2 \left\|\theta^{(t)} - \theta_{\text{lin}}^{(t)}\right\|_2 \\ &\leq \left|f(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta^{(t)})\right| + M\|\Delta_t\|_2.\end{aligned}$$

For the first term, note that

$$\begin{cases} f(x;\theta^{(t)}) - f(x;\theta^{(0)}) = \nabla_\theta f(x;\tilde{\theta}^{(t)})(\theta^{(t)} - \theta^{(0)}); \\ f_{\text{lin}}(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta^{(0)}) = \nabla_\theta f(x;\theta^{(0)})(\theta^{(t)} - \theta^{(0)}), \end{cases}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(0)}$. Since $f(x;\theta^{(0)}) = f_{\text{lin}}(x;\theta^{(0)})$,

$$\left|f(x;\theta^{(t)}) - f_{\text{lin}}(x;\theta^{(t)})\right| \leq \left\|\nabla_\theta f(x;\tilde{\theta}^{(t)}) - \nabla_\theta f(x;\theta^{(0)})\right\|_2 \left\|\theta^{(t)} - \theta^{(0)}\right\|_2 \leq \frac{M}{\sqrt[4]{\tilde{d}}}C_0^2.$$

Thus, for all $\tilde{d} \geq D_0$, with probability at least $(1-\delta)$ for all t and all x,

$$\left|f_{\text{reg}}^{(t)}(x) - f_{\text{linreg}}^{(t)}(x)\right| \leq \left(MC_0^2 + \frac{2M^2C_0C_1}{\mu}\right)\tilde{d}^{-1/4} = O(\tilde{d}^{-1/4}),$$

which proves the lemma. ■

Lemma E.10. *Suppose there exists $M_0 > 0$ such that $\left\|\nabla_\theta f^{(0)}(x)\right\|_2 \le M_0$ for all x within the unit ball. If the gradients $\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)$ are linearly independent, and the empirical training risk of $f^{(t)}_{\text{linreg}}$ satisfies $\limsup_{t\to\infty} \hat{\mathcal{R}}(f^{(t)}_{\text{linreg}}) < \epsilon$ for some $\epsilon > 0$, then for any x within the unit ball, we have*

$$\limsup_{t\to\infty} \left| f^{(t)}_{\text{linreg}}(x) - f^{(t)}_{\text{linERM}}(x) \right| = O(\sqrt{\epsilon}).$$

Proof First, for all t we have $\theta^{(t)} - \theta^{(0)} \in \text{span}\{\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n)\}$. Let θ^* be the interpolator in $\text{span}(\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n))$, then the empirical risk of θ is $\frac{1}{2n}\sum_{i=1}^n \langle \theta - \theta^*, \nabla_\theta f^{(0)}(x_i)\rangle^2 = \frac{1}{2n}\left\|\nabla_\theta f^{(0)}(\boldsymbol{X})^\top(\theta - \theta^*)\right\|_2^2$. Thus, there exists $T > 0$ such that

$$\left\|\nabla_\theta f^{(0)}(\boldsymbol{X})^\top(\theta^{(t)} - \theta^*)\right\|_2^2 \le 2n\epsilon \qquad \text{for all } t \ge T.$$

Let the smallest singular value of $\frac{1}{\sqrt{n}}\nabla_\theta f^{(0)}(\boldsymbol{X})$ be $s_{\min}$. Then, we have $s_{\min} > 0$. Note that the column space of $\nabla_\theta f^{(0)}(\boldsymbol{X})$ is exactly $\text{span}(\nabla_\theta f^{(0)}(x_1), \cdots, \nabla_\theta f^{(0)}(x_n))$. Define $\boldsymbol{H} \in \mathbb{R}^{p\times n}$ such that its columns form an orthonormal basis of this subspace, then there exists $\boldsymbol{G} \in R^{n\times n}$ such that $\nabla_\theta f^{(0)}(\boldsymbol{X}) = \boldsymbol{H}\boldsymbol{G}$, and the smallest singular value of $\frac{1}{\sqrt{n}}\boldsymbol{G}$ is also $s_{\min}$. Since $\theta^{(t)} - \theta^{(0)}$ is also in this subspace, there exists $\boldsymbol{v} \in \mathbb{R}^n$ such that $\theta^{(t)} - \theta^* = \boldsymbol{H}\boldsymbol{v}$. Then we have $\sqrt{2n\epsilon} \ge \left\|\boldsymbol{G}^\top\boldsymbol{H}^\top\boldsymbol{H}\boldsymbol{v}\right\|_2 = \left\|\boldsymbol{G}^\top\boldsymbol{v}\right\|_2$. Thus, $\|\boldsymbol{v}\|_2 \le \frac{\sqrt{2\epsilon}}{s_{\min}}$, which implies that

$$\left\|\theta^{(t)} - \theta^*\right\|_2 \le \frac{\sqrt{2\epsilon}}{s_{\min}}.$$

We have already proved in the previous results that if we minimize the unregularized risk with ERM, then θ always converges to the interpolator θ^*. So for any $t \ge T$ and any test point x such that $\|x\|_2 \le 1$, we have

$$|f^{(t)}_{\text{linreg}}(x) - f^{(t)}_{\text{linERM}}(x)| = |\langle \theta^{(t)} - \theta^*, \nabla_\theta f^{(0)}(x)\rangle| \le \frac{M_0\sqrt{2\epsilon}}{s_{\min}},$$

as desired. ■

Now we prove Theorem 6.7.

Proof Given that $\hat{\mathcal{R}}(f^{(t)}_{\text{linreg}}) < \epsilon$ for sufficiently large t, Lemma E.9 implies that

$$\left|\hat{\mathcal{R}}(f^{(t)}_{\text{linreg}}) - \hat{\mathcal{R}}(f^{(t)}_{\text{reg}})\right| = O(\tilde{d}^{-1/4}\sqrt{\epsilon} + \tilde{d}^{-1/2}).$$

Thus, for a fixed ϵ, there exists $D > 0$ such that for all $d \ge D$, for sufficiently large t,

$$\hat{\mathcal{R}}(f^{(t)}_{\text{reg}}) < \epsilon \Rightarrow \hat{\mathcal{R}}(f^{(t)}_{\text{linreg}}) < 2\epsilon.$$

By Lemma E.2 and Lemma E.9, we have

$$\begin{cases} \sup_{t\ge 0} \left| f^{(t)}_{\text{linERM}}(x) - f^{(t)}_{\text{ERM}}(x) \right| = O(\tilde{d}^{-1/4}); \\ \sup_{t\ge 0} \left| f^{(t)}_{\text{linreg}}(x) - f^{(t)}_{\text{reg}}(x) \right| = O(\tilde{d}^{-1/4}). \end{cases}$$

Combining these with Lemma E.10 yields

$$\limsup_{t\to\infty} \left| f^{(t)}_{\text{reg}}(x) - f^{(t)}_{\text{ERM}}(x) \right| = O(\tilde{d}^{-1/4} + \sqrt{\epsilon}).$$

Letting $\tilde{d} \to \infty$ leads to the result we need. ■

Remark E.11. *One might wonder whether $\|\nabla_\theta f^{(0)}(x)\|_2$ will diverge as $\tilde{d} \to \infty$. In fact, in Lemma E.4, we have proved that there exists a constant M such that with high probability, for any $\tilde{d}$ there is $\|\nabla_\theta f^{(0)}(x)\|_2 \le M$ for any x such that $\|x\|_2 \le 1$. Therefore, it is fine to suppose that there exists such an M_0.*

E.4 Proof of Theorem 6.8

Proof First, we show that $\hat{\theta}_{\text{MM}}$ is unique. Suppose both θ_1 and θ_2 maximize $\min_i y_i \cdot \langle \theta, \boldsymbol{x}_i \rangle$ and $\|\theta_1\|_2 = \|\theta_2\|_2 = 1$. If $\theta_1 \neq \theta_2$, then we define $\theta = (\theta_1 + \theta_2)/2$ and $\theta_0 = \theta/\|\theta\|_2$. Obviously, $\|\theta\|_2 < 1$, and for all $i \in [n]$ there is $y_i\langle \theta, \boldsymbol{x}_i \rangle = (y_i\langle \theta_1, \boldsymbol{x}_i \rangle + y_i\langle \theta_2, \boldsymbol{x}_i \rangle)/2$. Thus, we have $y_i\langle \theta_0, \boldsymbol{x}_i \rangle > \min\{y_i\langle \theta_1, \boldsymbol{x}_i \rangle, y_i\langle \theta_2, \boldsymbol{x}_i \rangle\}$, which implies that $\min_i y_i\langle \theta_0, \boldsymbol{x}_i \rangle > \min\{\min_i y_i\langle \theta_1, \boldsymbol{x}_i \rangle, \min_i y_i\langle \theta_2, \boldsymbol{x}_i \rangle\}$, which contradicts the fact that θ_1, θ_2 are max-margin classifiers.

Without loss of generality, let $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)$ be the samples with the smallest margin for $\boldsymbol{u}$, that is $\arg\min_i y_i\langle \boldsymbol{u}, \boldsymbol{x}_i \rangle = \{1, \cdots, m\}$. Denote $\gamma = \min_i y_i\langle \boldsymbol{u}, \boldsymbol{x}_i \rangle$; then, $\gamma > 0$ since the training error converges to zero. Note that for the logistic loss, if $y_i\langle \theta, \boldsymbol{x}_i \rangle < y_j\langle \theta, \boldsymbol{x}_j \rangle$, then for any $M > 0$, there exists an $R_M > 0$ such that for all $R \ge R_M$, there is $\frac{\nabla_\theta \ell(\langle R\theta, \boldsymbol{x}_i \rangle, y_i)}{\nabla_\theta \ell(\langle R\theta, \boldsymbol{x}_j \rangle, y_j)} > M$. Since the training error converges to zero, we have $\left\|\theta^{(t)}\right\|_2 \to \infty$. So when t is sufficiently large, the impact of $(\boldsymbol{x}_j, y_j)$ on $\theta^{(t)}$ for $j > m$ is an infinitesimal compared to $j \le m$ since $\liminf_{t\to\infty} q_j^{(t)} > 0$. Thus, we must have $\boldsymbol{u} \in \text{span}\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$.

Let $\boldsymbol{u} = \alpha_1 y_1 \boldsymbol{x}_1 + \cdots + \alpha_m y_m \boldsymbol{x}_m$. Now we show that $\alpha_i \ge 0$ for all $i = 1, \cdots, m$. For a sufficiently large t, there is $\theta^{(t+1)} - \theta^{(t)} \approx \eta \sum_{i=1}^m \frac{q_i^{(t)} \exp(y_i \cdot \langle \theta^{(t)}, \boldsymbol{x}_i \rangle)}{1+\exp(y_i \cdot \langle \theta^{(t)}, \boldsymbol{x}_i \rangle)} y_i \boldsymbol{x}_i$. Since $\left\|\theta^{(t)}\right\| \to \infty$, for all $i \in [m]$ we have $\alpha_i \propto \lim_{T\to\infty} \sum_{t=T_0}^T \frac{q_i^{(t)} \exp(y_i \cdot \langle \theta^{(t)}, \boldsymbol{x}_i \rangle)}{1+\exp(y_i \cdot \langle \theta^{(t)}, \boldsymbol{x}_i \rangle)} := \lim_{T\to\infty} \alpha_i(T)$, where T_0 is sufficiently large. Here the notion $\alpha_i \propto \lim_{T\to\infty} \alpha_i(T)$ means that $\lim_{T\to\infty} \frac{\alpha_i(T)}{\alpha_j(T)} = \frac{\alpha_i}{\alpha_j}$ for any pair of i, j and $\alpha_j \neq 0$. Note that each term in the sum is non-negative. This implies that $\alpha_1, \cdots, \alpha_m$ have the same sign. Meanwhile, $\sum_{i=1}^m \alpha_i \gamma = \sum_{i=1}^m \alpha_i y_i \cdot \langle \boldsymbol{u}, \boldsymbol{x}_i \rangle = \langle \boldsymbol{u}, \boldsymbol{u} \rangle > 0$. Thus $\alpha_i \ge 0$ for all $i \in [m]$ and at least one is positive. Now suppose $\boldsymbol{u} \neq \hat{\theta}_{\text{MM}}$, which means that γ is smaller than the margin of $\hat{\theta}_{\text{MM}}$. Then, for all $i = 1, \cdots, m$, there is $y_i \cdot \langle \boldsymbol{u}, \boldsymbol{x}_i \rangle < y_i \cdot \langle \hat{\theta}_{\text{MM}}, \boldsymbol{x}_i \rangle$. This implies that $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = \sum_{i=1}^m \alpha_i y_i \cdot \langle \boldsymbol{u}, \boldsymbol{x}_i \rangle < \sum_{i=1}^m \alpha_i y_i \cdot \langle \hat{\theta}_{\text{MM}}, \boldsymbol{x}_i \rangle = \langle \hat{\theta}_{\text{MM}}, \boldsymbol{u} \rangle$, which is a contradiction. Thus, $\boldsymbol{u} = \hat{\theta}_{\text{MM}}$. ■

E.5 Proof of Theorem 6.10

Proof Denote the largest and smallest eigenvalues of $\boldsymbol{X}^\top \boldsymbol{X}$ by $\lambda^{\max}$ and $\lambda^{\min}$, and by condition we have $\lambda^{\min} > 0$. Let $\epsilon = \min\{\frac{q^*}{3}, \frac{(q^*\lambda^{\min})^2}{192\lambda^{\max 2}}\}$. Then, similar to the proof in Appendix E.1, there exists t_ϵ such that for all $t \ge t_\epsilon$ and all i, $q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon)$. Denote $\boldsymbol{Q} = \text{diag}(q_1, \cdots, q_n)$, then for all $t \ge t_\epsilon$, $\boldsymbol{Q}^{(t)} := \boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}$, where we use the subscript ϵ to indicate that $\left\|\boldsymbol{Q}_\epsilon^{(t)} - \boldsymbol{Q}\right\|_2 < \epsilon$.

First, we prove that $F(\theta)$ is L-smooth if $\|x_i\|_2 \le 1$ for all i. The gradient of F is

$$\nabla F(\theta) = \sum_{i=1}^n q_i \nabla_{\hat{y}} \ell(\langle \theta, \boldsymbol{x}_i \rangle, y_i) \boldsymbol{x}_i.$$

Since $\ell(\hat{y}, y)$ is L-smooth in $\hat{y}$, for any θ_1, θ_2 and any i, we have

$$\begin{aligned}
&\ell(\langle \theta_2, \boldsymbol{x}_i \rangle, y_i) - \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \\
\leq\ & \nabla_{\hat{y}} \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \cdot (\langle \theta_2, \boldsymbol{x}_i \rangle - \langle \theta_1, \boldsymbol{x}_i \rangle) + \frac{L}{2} (\langle \theta_2, \boldsymbol{x}_i \rangle - \langle \theta_1, \boldsymbol{x}_i \rangle)^2 \\
=\ & \langle \nabla_{\hat{y}} \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \cdot \boldsymbol{x}_i, \theta_2 - \theta_1 \rangle + \frac{L}{2} (\langle \theta_2 - \theta_1, \boldsymbol{x}_i \rangle)^2 \\
\leq\ & \langle \nabla_{\hat{y}} \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \cdot \boldsymbol{x}_i, \theta_2 - \theta_1 \rangle + \frac{L}{2} \left\| \theta_2 - \theta_1 \right\|_2^2 .
\end{aligned}$$

Thus, we have

$$\begin{aligned}
F(\theta_2) - F(\theta_1) =& \sum_{i=1}^{n} q_i \left[\ell(\langle \theta_2, \boldsymbol{x}_i \rangle, y_i) - \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \right] \\
\leq& \sum_{i=1}^{n} q_i \langle \nabla_{\hat{y}} \ell(\langle \theta_1, \boldsymbol{x}_i \rangle, y_i) \cdot \boldsymbol{x}_i, \theta_2 - \theta_1 \rangle + \frac{L}{2} \sum_{i=1}^{n} q_i \left\| \theta_2 - \theta_1 \right\|_2^2 \\
=& \langle \nabla F(\theta_1), \theta_2 - \theta_1 \rangle + \frac{L}{2} \left\| \theta_2 - \theta_1 \right\|_2^2 ,
\end{aligned}$$

which implies that $F(\theta)$ is L-smooth.

Denote $\tilde{g}(\theta) = \nabla_{\hat{y}} \ell(f(\boldsymbol{X}; \theta), \boldsymbol{Y}) \in \mathbb{R}^n$, then $\nabla F(\theta^{(t)}) = \boldsymbol{X}\boldsymbol{Q}\tilde{g}(\theta^{(t)})$, and the update rule is

$$\theta^{(t+1)} = \theta^{(t)} - \eta \boldsymbol{X}\boldsymbol{Q}^{(t)} \tilde{g}(\theta^{(t)}). \tag{E.18}$$

So by Definition 6.9, we have

$$F(\theta^{(t+1)}) \leq F(\theta^{(t)}) - \eta \langle \boldsymbol{X}\boldsymbol{Q}\tilde{g}(\theta^{(t)}), \boldsymbol{X}\boldsymbol{Q}^{(t)} \tilde{g}(\theta^{(t)}) \rangle + \frac{\eta^2 L}{2} \left\| \boldsymbol{X}\boldsymbol{Q}^{(t)} \tilde{g}(\theta^{(t)}) \right\|_2^2 . \tag{E.19}$$

Let $\eta_1 = \frac{q^* \lambda^{\min}}{2L(1+3\epsilon)\lambda^{\max}}$. Similar to the proof in Appendix E.1, we can prove that for all $\eta \leq \eta_1$, and for all $t \geq t_\epsilon$, we have

$$\begin{aligned}
F(\theta^{(t+1)}) &\leq F(\theta^{(t)}) - \frac{\eta q^* \lambda^{\min}}{2} \left\| \sqrt{\boldsymbol{Q}} \tilde{g}(\theta^{(t)}) \right\|_2^2 + \frac{\eta^2 L}{2} \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \right\|_2^2 \left\| \sqrt{\boldsymbol{Q}} \tilde{g}(\theta^{(t)}) \right\|_2^2 \\
&\leq F(\theta^{(t)}) - \frac{\eta q^* \lambda^{\min}}{2} \left\| \sqrt{\boldsymbol{Q}} \tilde{g}(\theta^{(t)}) \right\|_2^2 + \frac{\eta^2 L}{2} \left\| \boldsymbol{X} \right\|_2^2 (1+3\epsilon) \left\| \sqrt{\boldsymbol{Q}} \tilde{g}(\theta^{(t)}) \right\|_2^2 \\
&\leq F(\theta^{(t)}) - \frac{\eta q^* \lambda^{\min}}{4} \left\| \sqrt{\boldsymbol{Q}} \tilde{g}(\theta^{(t)}) \right\|_2^2 \\
&\leq F(\theta^{(t)}) - \frac{\eta q^{*2} \lambda^{\min}}{4} \left\| \tilde{g}(\theta^{(t)}) \right\|_2^2 .
\end{aligned}$$

This implies that $F(\theta^{(t)})$ is monotonically non-increasing. Since $F(\theta) \geq 0$, $F(\theta^{(t)})$ must converge as $t \to \infty$, and we need to prove that it converges to 0. Suppose that $F(\theta^{(t)})$ does not converge to 0, then there exists a constant $C > 0$ such that $F(\theta^{(t)}) \geq 2C$ for all t. On the other hand, it is easy to see that there exists θ^* such that $\ell(\langle \theta^*, \boldsymbol{x}_i \rangle, y_i) < C$ for all i. The above inequality also implies that $\left\| \tilde{g}(\theta^{(t)}) \right\|_2 \to 0$ as $t \to \infty$ because we must have $F(\theta^{(t)}) - F(\theta^{(t+1)}) \to 0$.

Note that by Eqn. (E.18), we have

$$\left\| \theta^{(t+1)} - \theta^* \right\|_2^2 = \left\| \theta^{(t)} - \theta^* \right\|_2^2 + 2\eta \langle \boldsymbol{X}\boldsymbol{Q}^{(t)} \tilde{g}(\theta^{(t)}), \theta^* - \theta^{(t)} \rangle + \eta^2 \left\| \boldsymbol{X}\boldsymbol{Q}^{(t)} \tilde{g}(\theta^{(t)}) \right\|_2^2 .$$

Define $F_t(\theta) = \sum_{i=1}^n q_i^{(t)} \ell(\langle \theta, \boldsymbol{x}_i \rangle, y_i)$. F_t is convex because ℓ is convex and $q_i^{(t)}$ are non-negative, and $\nabla F_t(\theta^{(t)}) = \boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)})$. Convexity guarantees that $F_t(\boldsymbol{y}) \geq F_t(\boldsymbol{x}) + \langle \nabla F_t(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$, so for all t we have

$$\langle \boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)}), \theta^* - \theta^{(t)} \rangle \leq F_t(\theta^*) - F_t(\theta^{(t)}) \leq F_t(\theta^*) - \frac{2}{3}F(\theta^{(t)}) \leq C - \frac{4C}{3} = -\frac{C}{3},$$

because $q_i^{(t)} \geq q_i - \epsilon \geq \frac{2}{3}q_i$ and $\sum_{i=1}^n q_i^{(t)} = 1$. Since $\left\|\tilde{g}(\theta^{(t)})\right\|_2 \to 0$, there exists $T > 0$ such that for all $t \geq T$ and all $\eta \leq \eta_0$,

$$\left\|\theta^{(t+1)} - \theta^*\right\|_2^2 \leq \left\|\theta^{(t)} - \theta^*\right\|_2^2 - \frac{\eta C}{3},$$

which means that $\left\|\theta^{(t)} - \theta^*\right\|_2^2 \to -\infty$ because $\frac{\eta C}{3}$ is a positive constant. This is a contradiction! Thus, $F(\theta^{(t)})$ must converge to 0, which is result (i).

(i) immediately implies (ii) because ℓ is strictly decreasing to 0 by the condition.

Now let us prove (iii). First of all, the uniqueness of θ_R can be easily proved from the convexity of $F(\theta)$. The condition implies that $y_i\langle \theta_R, \boldsymbol{x}_i \rangle > 0$, i.e. θ_R must classify all training samples correctly. If there are two different minimizers θ_R and θ'_R in whose norm is at most R, then consider $\theta''_R = \frac{1}{2}(\theta_R + \theta'_R)$. By the convexity of F, we know that θ''_R must also be a minimizer, and $\|\theta''_R\|_2 < R$. Thus, $F(\frac{R}{\|\theta''_R\|_2}\theta''_R) < F(\theta''_R)$ and $\|\frac{R}{\|\theta''_R\|_2}\theta''_R\|_2 = R$, which contradicts with the fact that θ''_R is a minimizer.

To prove the rest of (iii), look at Eqn. (E.19). On one hand, for all $t \geq t_\epsilon$, we have

$$\left|\langle \boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)}), \boldsymbol{X}(\boldsymbol{Q}^{(t)} - \boldsymbol{Q})\tilde{g}(\theta^{(t)}) \rangle\right| \leq \lambda^{\max}\sqrt{3\epsilon}\left\|\sqrt{\boldsymbol{Q}^{(t)}}\tilde{g}(\theta^{(t)})\right\|_2^2.$$

Since we chose $\epsilon = \min\{\frac{q^*}{3}, \frac{(q^*\lambda^{\min})^2}{192\lambda^{\max 2}}\}$, this inequality implies that

$$\begin{aligned}\left\|\nabla F_t(\theta^{(t)})\right\|_2^2 = \left\|\boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)})\right\|_2^2 &\geq \lambda^{\min}\left\|\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)})\right\|_2^2 \geq \lambda^{\min}(q^* - \epsilon)\left\|\sqrt{\boldsymbol{Q}^{(t)}}\tilde{g}(\theta^{(t)})\right\|_2^2 \\ &\geq \frac{\lambda^{\min}q^*}{2}\left\|\sqrt{\boldsymbol{Q}^{(t)}}\tilde{g}(\theta^{(t)})\right\|_2^2 \geq 4\left|\langle \boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)}), \boldsymbol{X}(\boldsymbol{Q}^{(t)} - \boldsymbol{Q})\tilde{g}(\theta^{(t)}) \rangle\right|.\end{aligned}$$

On the other hand, if $\eta \leq \eta_2 = \frac{1}{2L}$, then we have

$$\frac{\eta^2 L}{2}\left\|\boldsymbol{X}\boldsymbol{Q}^{(t)}\tilde{g}(\theta^{(t)})\right\|_2^2 \leq \frac{\eta}{4}\left\|\nabla F_t(\theta^{(t)})\right\|_2^2.$$

Combining the above with Eqn. (E.19), we get

$$F(\theta^{(t+1)}) - F(\theta^{(t)}) \leq -\frac{\eta}{2}\left\|\nabla F_t(\theta^{(t)})\right\|_2^2.$$

Denote $\boldsymbol{u} = \lim_{R\to\infty}\frac{\theta_R}{\|\theta_R\|_2}$. Similar to Lemma 9 in [79], we can prove that: for any $\alpha > 0$, there exists a constant $\rho(\alpha) > 0$ such that for any θ subject to $\|\theta\|_2 \geq \rho(\alpha)$, the following holds for all t:

$$F_t((1+\alpha)\|\theta\|_2\boldsymbol{u}) \leq F_t(\theta).$$

Let $t_\alpha \geq t_\epsilon$ satisfy that for all $t \geq t_\alpha$, $\|\theta^{(t)}\|_2 \geq \max\{\rho(\alpha), 1\}$. By the convexity of F_t, for all $t \geq t_\alpha$, we have

$$\langle \nabla F_t(\theta^{(t)}), \theta^{(t)} - (1+\alpha)\|\theta^{(t)}\|_2\boldsymbol{u} \rangle \geq F_t(\theta^{(t)}) - F_t((1+\alpha)\|\theta^{(t)}\|_2\boldsymbol{u}) \geq 0. \tag{E.20}$$

Thus, we have

$$\begin{aligned}\langle \theta^{(t+1)} - \theta^{(t)}, \boldsymbol{u} \rangle &= \langle -\eta \nabla F_t(\theta^{(t)}), \boldsymbol{u} \rangle \\ &\geq \langle -\eta \nabla F_t(\theta^{(t)}), \theta^{(t)} \rangle \frac{1}{(1+\alpha)\|\theta^{(t)}\|_2} \\ &= \langle \theta^{(t+1)} - \theta^{(t)}, \theta^{(t)} \rangle \frac{1}{(1+\alpha)\|\theta^{(t)}\|_2} \\ &= \left(\frac{1}{2} \left\|\theta^{(t+1)}\right\|_2^2 - \frac{1}{2} \left\|\theta^{(t)}\right\|_2^2 - \frac{1}{2} \left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2^2 \right) \frac{1}{(1+\alpha)\|\theta^{(t)}\|_2}.\end{aligned} \tag{E.21}$$

By $\frac{1}{2}(\|\theta^{(t+1)}\|_2 - \|\theta^{(t)}\|_2)^2 \geq 0$, we have $(\frac{1}{2} \left\|\theta^{(t+1)}\right\|_2^2 - \frac{1}{2} \left\|\theta^{(t)}\right\|_2^2)/\|\theta^{(t)}\|_2 \geq \left\|\theta^{(t+1)}\right\|_2 - \left\|\theta^{(t)}\right\|_2$. Moreover, by Eqn. (E.20), we have

$$\frac{\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2^2}{2(1+\alpha)\|\theta^{(t)}\|_2} \leq \frac{\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2^2}{2} = \frac{\eta^2 \left\|\nabla F_t(\theta^{(t)})\right\|_2^2}{2} \leq \eta \left(F(\theta^{(t)}) - F(\theta^{(t+1)}) \right).$$

Summing up Eqn. (E.21) from $t = t_\alpha$ to $t-1$ yields

$$\langle \theta^{(t)} - \theta^{(t_\alpha)}, \boldsymbol{u} \rangle \geq \frac{\left\|\theta^{(t)}\right\|_2 - \left\|\theta^{(t_\alpha)}\right\|_2}{1+\alpha} + \eta \left(F(\theta^{(t)}) - F(\theta^{(t_\alpha)}) \right) \geq \frac{\left\|\theta^{(t)}\right\|_2 - \left\|\theta^{(t_\alpha)}\right\|_2}{1+\alpha} - \eta F(\theta^{(t_\alpha)}),$$

which implies that

$$\left\langle \frac{\theta^{(t)}}{\left\|\theta^{(t)}\right\|_2}, \boldsymbol{u} \right\rangle \geq \frac{1}{1+\alpha} + \frac{1}{\left\|\theta^{(t)}\right\|_2} \left(\langle \theta^{(t_\alpha)}, \boldsymbol{u} \rangle - \frac{\|\theta^{(t_\alpha)}\|_2}{1+\alpha} - \eta F(\theta^{(t_\alpha)}) \right).$$

Since $\lim_{t\to\infty} \|\theta^{(t)}\|_2 = \infty$, we have

$$\liminf_{t\to\infty} \left\langle \frac{\theta^{(t)}}{\left\|\theta^{(t)}\right\|_2}, \boldsymbol{u} \right\rangle \geq \frac{1}{1+\alpha}.$$

Since α is arbitrary, we must have $\lim_{t\to\infty} \frac{\theta^{(t)}}{\left\|\theta^{(t)}\right\|_2} = \boldsymbol{u}$ as long as $\eta \leq \min\{\eta_1, \eta_2\}$. ■

E.6 Analysis of the Logistic Loss

Here, we show that the logistic loss satisfies all the conditions in Theorem 6.10, and $\lim_{R\to\infty} \frac{\theta_R}{R} = \hat{\theta}_{\text{MM}}$.

First, for the logistic loss we have $\nabla^2_{\hat{y}} \ell(\hat{y}, y) = \frac{y^2}{e^{y\hat{y}} + e^{-y\hat{y}} + 2} \leq \max_i \frac{y_i^2}{4}$, so ℓ is smooth.

Second, let us analyze $\lim_{R\to\infty} \frac{\theta_R}{R}$. For the logistic loss, it is easy to show that for any $\hat{\theta}' \neq \hat{\theta}_{\text{MM}}$, there exists an $R(\hat{\theta}') > 0$ and an $\delta(\hat{\theta}') > 0$ such that $F(R \cdot \theta) > F(R \cdot \hat{\theta}_{\text{MM}})$ for all $R \geq R(\hat{\theta}')$ and $\theta \in B(\hat{\theta}', \delta(\hat{\theta}'))$. Let $S = \{\theta : \|\theta\|_2 = 1\}$. For any $\epsilon > 0$, $S - B(\hat{\theta}_{\text{MM}}, \epsilon)$ is a compact set. And for any $\theta \in S - B(\hat{\theta}_{\text{MM}}, \epsilon)$, there exist $R(\theta)$ and $\delta(\theta)$ as defined above. Thus, there must exist $\theta_1, \cdots, \theta_m \in S - B(\hat{\theta}_{\text{MM}}, \epsilon)$ such that $S - B(\hat{\theta}_{\text{MM}}, \epsilon) \subseteq \cup_{i=1}^m B(\theta_i, \delta(\theta_i))$. Let $R(\epsilon) = \max\{R(\theta_1), \cdots, R(\theta_m)\}$, then for all $R \geq R(\epsilon)$ and all $\theta \in S - B(\hat{\theta}_{\text{MM}}, \epsilon)$, $F(R \cdot \theta) > F(R \cdot \hat{\theta}_{\text{MM}})$, which means that $\frac{\theta_R}{R} \in B(\hat{\theta}_{\text{MM}}, \epsilon)$ for all $R \geq R(\epsilon)$. Therefore, $\lim_{R\to\infty} \frac{\theta_R}{R}$ exists and is equal to $\hat{\theta}_{\text{MM}}$.

E.7 Proof of Theorem 6.11

Proof Let M_0 be the bound of $\left\|\nabla_\theta f^{(0)}(x)\right\|_2$. We first consider the regularized linearized neural network $f_{\text{linreg}}^{(t)}$. By Proposition E.7, $f^{(0)}(\boldsymbol{x})$ is sampled from a zero-mean Gaussian process, so there exists a constant $M > 0$ such that $|f^{(0)}(\boldsymbol{x}_i)| < M$ for all i with high probability. Define

$$F(\theta) = \sum_{i=1}^{n} q_i \ell(\langle \theta, \nabla_\theta f^{(0)}(\boldsymbol{x}_i)\rangle + f^{(0)}(\boldsymbol{x}_i), y_i).$$

Denote $\tilde{\theta}_R = \arg\min_\theta\{F(R\cdot\theta) : \|\theta\|_2 \le 1\}$. when the linearized neural network is trained by GRW satisfying Assumption 6.3 with regularization, since this is convex optimization and the objective function is smooth, we can prove that with a sufficiently small learning rate, as $t \to \infty$, $\theta^{(t)} \to R\cdot\tilde{\theta_R}+\theta^{(0)}$ where $R = \lim_{t\to\infty} \|\theta^{(t)}-\theta^{(0)}\|_2$ (which is the minimizer). And define

$$\gamma = \min_{i=1,\cdots,n} y_i \cdot \langle \hat{\theta}_{\text{MM}}, \nabla_\theta f^{(0)}(x_i)\rangle.$$

First, we derive the lower bound of R. By Lemma E.9, with a sufficiently large $\tilde{d}$, with high probability $\hat{\mathcal{R}}(f_{\text{reg}}^{(t)}) < \epsilon$ implies $\hat{\mathcal{R}}(f_{\text{linreg}}^{(t)}) < 2\epsilon$. By the convexity of ℓ, we have

$$\begin{aligned}
2\epsilon &> \frac{1}{n}\sum_{i=1}^{n} \ell(\langle R\tilde{\theta}_R, x_i\rangle + f^{(0)}(x_i), y_i) \ge \log\left(1 + \exp\left(-\frac{1}{n}\sum_{i=1}^{n}(\langle R\tilde{\theta}_R, \boldsymbol{x}_i\rangle + f^{(0)}(x_i))y_i\right)\right) \\
&\ge \log\left(1 + \exp\left(-\frac{1}{n}\sum_{i=1}^{n} R\langle\tilde{\theta}_R, x_i\rangle y_i - M\right)\right),
\end{aligned}$$

which implies that $R = \Omega(-\log 2\epsilon)$ for all $\epsilon \in (0, \frac{1}{4})$.

Denote $\delta = \|\hat{\theta}_{\text{MM}} - \tilde{\theta}_R\|_2$. Let $\theta' = \frac{\hat{\theta}_{\text{MM}}+\tilde{\theta}_R}{2}$, then we can see that $\|\theta'\|_2 = \sqrt{1-\frac{\delta^2}{4}}$. Let $\tilde{\theta}' = \frac{\theta'}{\|\theta'\|_2}$. By the definition of $\hat{\theta}_{\text{MM}}$, there exists j such that $y_j \cdot \langle\tilde{\theta}', \nabla_\theta f^{(0)}(\boldsymbol{x}_j)\rangle \le \gamma$, which implies that

$$y_j \cdot \left\langle \frac{\hat{\theta}_{\text{MM}} + \tilde{\theta}_R}{2} \frac{1}{\sqrt{1-\frac{\delta^2}{4}}}, \nabla_\theta f^{(0)}(\boldsymbol{x}_j)\right\rangle \le \gamma.$$

Thus, we have

$$\begin{aligned}
y_j \cdot \langle\tilde{\theta}_R, \nabla_\theta f^{(0)}(x_j)\rangle &\le 2\sqrt{1-\frac{\delta^2}{4}}\gamma - y_j \cdot \langle\hat{\theta}_{\text{MM}}, \nabla_\theta f^{(0)}(\boldsymbol{x}_j)\rangle \\
&\le \left(2\sqrt{1-\frac{\delta^2}{4}} - 1\right)\gamma \\
&\le \left(2(1-\frac{\delta^2}{8}) - 1\right)\gamma \qquad (\text{since } \sqrt{1-x} \le 1-\frac{x}{2}) \\
&= (1-\frac{\delta^2}{4})\gamma.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&q_j \log(1 + \exp(-y_j \cdot \langle R\cdot\tilde{\theta}_R, \nabla_\theta f^{(0)}(\boldsymbol{x}_j)\rangle - M)) \le F(R\cdot\tilde{\theta}_R) \\
&\le F(R\cdot\hat{\theta}_{\text{MM}}) \le \log(1 + \exp(-R\gamma + M)),
\end{aligned}$$

which implies that

$$q^* \log\left(1+\exp\left(-(1-\frac{\delta^2}{4})R\gamma - M\right)\right) \le \log(1+\exp(-R\gamma+M)).$$

Thus, we have

$$1+\exp(-R\gamma+M) \ge \left(1+\exp\left(-(1-\frac{\delta^2}{4})R\gamma - M\right)\right)^{q^*} \ge 1+q^*\exp\left(-(1-\frac{\delta^2}{4})R\gamma - M\right),$$

which is equivalent to

$$-R\gamma + M \ge -(1-\frac{\delta^2}{4})R\gamma - M + \log(q^*).$$

From this, we conclude that

$$\delta = O(R^{-1/2}) = O((-\log 2\epsilon)^{-1/2}).$$

So for any test point x such that $\left\|\nabla_\theta f^{(0)}(x)\right\| \le M_0$, we have

$$\left|\left\langle \hat{\theta}_{\text{MM}} - \tilde{\theta}_R, \nabla_\theta f^{(0)}(x)\right\rangle\right| \le \delta M_0 = O((-\log 2\epsilon)^{-1/2}).$$

Combining this with Lemma E.9, with high probability, we have

$$\limsup_{t\to\infty} |R\cdot f_{\text{MM}}(x) - f_{\text{reg}}^{(t)}(x)| = O(R\cdot(-\log 2\epsilon)^{-1/2} + \tilde{d}^{-1/4}).$$

Hence, there exists a constant $C > 0$ such that: As $\tilde{d}\to\infty$, with high probability, for all $\epsilon \in (0, \frac{1}{4})$, if $|f_{\text{MM}}(x)| > C\cdot(-\log 2\epsilon)^{-1/2}$, then $f_{\text{reg}}^{(t)}(x)$ will have the same sign as $f_{\text{MM}}(x)$ for a sufficiently large t. Note that this C only depends on n, q^*, γ, M and M_0, so it is a constant independent of ϵ. ■

E.8 Proof of Proposition 6.16

Proof We have

$$\begin{aligned}
\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta; P_{\text{train}}) &= \inf_{P'}\left\{\mathcal{R}_{D_\beta,\rho}(\theta;P') : \exists \tilde{P}' \text{ s.t. } P_{\text{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'\right\} \\
&= \inf_{P',\eta}\left\{c_\beta(\rho)\mathbb{E}_{P'}[(\ell(\theta;Z)-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \eta\right\} \\
&= \inf_{\eta}\left\{c_\beta(\rho)\inf_{P'}\{[\int_{\mathbb{R}_+} P'((\ell(\theta;Z)-\eta)_+^{\beta_*} > u)du]^{\frac{1}{\beta_*}}\} + \eta\right\}.
\end{aligned} \tag{E.22}$$

Since $P_{\text{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'$, for all $\ell_0 \in \mathbb{R}$, we have

$$P'(\ell(\theta;Z)\le\ell_0) \le \min\left\{1, \frac{1}{1-\epsilon}P_{\text{train}}(\ell(\theta;Z)\le\ell_0)\right\},$$

and we can show that there exists a $P^* = P'$ that attains the equality for all ℓ_0. This is because $P_{\text{train}}(\ell(\theta;z))$ is a continuous function of z for any fixed θ since both ℓ and P_{train}

are continuous, so there exists an ℓ^* such that $P_{\text{train}}(\ell(\theta;Z) > \ell^*) = \epsilon$. Hence, we can define

$$P^*(z) = \begin{cases} \frac{1}{1-\epsilon} P_{\text{train}}(z) & , \ell(\theta;z) \le \ell^*; \\ 0 & , \ell(\theta;z) > \ell^*. \end{cases}$$

For this P^*, we have $\int_{\mathcal{X}\times\mathcal{Y}} P^*(z)dz = \frac{1}{1-\epsilon}\int_{\ell(\theta;z)<\ell^*} P_{\text{train}}(z)dz = \frac{1}{1-\epsilon}P_{\text{train}}(\ell(\theta;Z) < \ell^*) = 1$ because $P_{\text{train}}(\ell(\theta;Z) = \ell^*) = 0$, which means that P^* is a proper probability density function.

Let $v = u^{\frac{1}{\beta_*}}$. Plugging $P^*(\ell(\theta;Z) \le \ell_0) = \min\left\{1, \frac{1}{1-\epsilon}P_{\text{train}}(\ell(\theta;Z) \le \ell_0)\right\}$ into Eqn. (E.22) produces

$$\begin{aligned}
\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta;P_{\text{train}}) &= \inf_\eta \left\{ c_\beta(\rho) \left[\int_{\mathbb{R}_+} [1 - P^*((\ell(\theta;Z)-\eta)_+^{\beta_*} \le v^{\beta_*})] dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\} \\
&= \inf_\eta \left\{ c_\beta(\rho) \left[\int_{\mathbb{R}_+} [1 - \frac{1}{1-\epsilon} P_{\text{train}}(\ell(\theta;Z) \le \eta + v)]_+ dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\} \\
&= \inf_\eta \left\{ c_\beta(\rho) \left[\int_0^{(\ell^*-\eta)_+} \frac{1}{1-\epsilon} [(1-\epsilon) - P_{\text{train}}(\ell(\theta;Z) \le \eta + v)]_+ dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\}.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&\mathbb{E}_{P_{\text{train}}}[(\ell-\eta)_+^{\beta_*} \mid P_{Z'\sim P_{\text{train}}}(\ell(\theta;Z') > \ell(\theta;Z)) \ge \epsilon] \\
=&\frac{1}{1-\epsilon}\int_0^{\ell^*} (u-\eta)_+^{\beta_*} d(P_{\text{train}}(\ell \le u)) \\
=&\frac{1}{1-\epsilon}\left\{ \left[(u-\eta)_+^{\beta_*} P_{\text{train}}(\ell \le u) \right]_0^{\ell^*} - \int_0^{\ell^*} P_{\text{train}}(\ell \le u) d((u-\eta)_+^{\beta_*}) \right\} \\
=&\frac{1}{1-\epsilon}\left\{ (\ell^*-\eta)_+^{\beta_*}(1-\epsilon) - \int_0^{\ell^*} P_{\text{train}}(\ell \le u) d((u-\eta)_+^{\beta_*}) \right\} \\
=&\frac{1}{1-\epsilon}\left\{ \int_0^{(\ell^*-\eta)_+} (1-\epsilon) dv^{\beta_*} - \int_0^{(\ell^*-\eta)_+} P_{\text{train}}(\ell \le \eta + w) dw^{\beta_*} \right\},
\end{aligned}$$

where $w = (u-\eta)_+$. This completes the proof. ■

Remark E.12. *We can prove a similar dual formula even if P_{train} is not continuous. For any P_{train}, there exists an ℓ^* such that $P_{\text{train}}(\ell(\theta;Z) > \ell^*) \le \epsilon$ and $P_{\text{train}}(\ell(\theta;Z) < \ell^*) \le 1-\epsilon$. If $P_{\text{train}}(\ell(\theta;Z) = \ell^*) = 0$, then we still define P^* the same as in the above proof. If $P_{\text{train}}(\ell(\theta;Z) = \ell^*) > 0$, then we define*

$$P^*(z) = \begin{cases} \frac{1}{1-\epsilon} P_{\text{train}}(z) & , \ell(\theta;z) < \ell^*; \\ \left[1 - \frac{1}{1-\epsilon} P_{\text{train}}(\ell(\theta;Z) < \ell^*)\right] / P_{\text{train}}(\ell(\theta;Z) = \ell^*) & , \ell(\theta;z) = \ell^*; \\ 0 & , \ell(\theta;z) > \ell^*, \end{cases}$$

with which the dual formula becomes

$$\begin{aligned}
&\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta;P_{\text{train}}) \\
&\quad= \inf_\eta \{ c_\beta(\rho) (\frac{P_{\text{train}}(\ell < \ell^*)}{1-\epsilon} \mathbb{E}_Z[(\ell(\theta;Z)-\eta)_+^{\beta_*} \mid P_{Z'}(\ell(\theta;Z') > \ell(\theta;Z)) > \epsilon] \\
&\qquad + \frac{1 - P_{\text{train}}(\ell < \ell^*)}{1-\epsilon} (\ell^* - \eta)_+^{\beta_*})^{\frac{1}{\beta_*}} + \eta \}.
\end{aligned}$$

E.9 Proof of Theorem 6.17

The proof relies on the following key technical lemma.

Lemma E.13. *For any distributions P, P', non-negative loss function $l(\cdot, Z)$ and $1 \le \beta_* < 2k$, such that $\mathbb{E}_P[l(\theta, Z)^{2k}] < \infty$, we have*

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} \le$$
$$\mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \mathbb{E}_P[(l(\theta,Z)-\eta)_+^{2k}]^{\frac{1}{2k}} \mathrm{TV}(P,P')^{\left(\frac{1}{\beta_*}-\frac{1}{2k}\right)} \beta_*^{-\frac{1}{2k}} \cdot \left(\frac{2k}{2k-\beta_*}\right)^{\frac{1}{\beta_*}}$$

Proof By the definition of the total variation distance, we have $P(\ell(\theta;Z) > u) - P'(\ell(\theta;Z') > u) \le \mathrm{TV}(P,P')$ for all $u \ge 0$. Let $s_{2k} := \mathbb{E}[(\ell-\eta)_+^{2k}]^{\frac{1}{2k}}$. By Markov's inequality and the non-negativity of ℓ, we have

$$P(\ell - \eta > u) \le \frac{\mathbb{E}[(\ell-\eta)_+^{2k}]}{u^{2k}} := (\frac{s_{2k}}{u})^{2k} \qquad \text{for all } \eta \ge 0. \tag{E.23}$$

Using integration by parts, we have

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] = \int_\eta^\infty \beta_*(t-\eta)^{(\beta_*-1)} P(\ell \ge t)dt = \int_0^\infty \beta_* u^{(\beta_*-1)} P(\ell-\eta \ge u)du.$$

This implies that

$$\begin{aligned}\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] - \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}] &= \int_0^\infty \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\ge u) - P'(\ell-\eta\ge u)\right) du \\ &= \left(\int_0^M + \int_M^\infty\right)\left(\beta_* u^{(\beta_*-1)}\left(P(\ell-\eta\ge u) - P'(\ell-\eta\ge u)\right)du\right).\end{aligned}$$

Here, M is a positive parameter whose value will be determined later. For the first integral, we have

$$\begin{aligned}\int_0^M \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\ge u) - P'(\ell-\eta\ge u)\right) du &\le \int_0^M \beta_* u^{(\beta_*-1)} \mathrm{TV}(P,P')du \\ &= M^{\beta_*}\mathrm{TV}(P,P').\end{aligned}$$

For the second integral, by Eqn. (E.23), we have

$$\begin{aligned}\int_M^\infty \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\ge u) - P'(\ell-\eta\ge u)\right) du &\le \int_M^\infty \beta_* u^{(\beta_*-1)} P(\ell-\eta\ge u)du \\ &\le \int_M^\infty \beta_* u^{(\beta_*-1)} \left(\frac{s_{2k}}{u}\right)^{2k} \\ &= \frac{s_{2k}^{2k}}{2k-\beta_*}\cdot\frac{1}{M^{2k-\beta_*}}.\end{aligned}$$

Therefore, by setting $M = s_{2k}(\mathrm{TV}(P,P')\beta_*)^{-1/2k}$ which minimizes the sum of two terms, we have

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] - \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}] \le$$
$$\inf_{M>0}\left(M^{\beta_*}\mathrm{TV}(P,P') + \frac{s_{2k}^{2k}}{2k-\beta_*}\cdot\frac{1}{M^{2k-\beta_*}}\right) = s_{2k}^{\beta_*}\mathrm{TV}(P,P')^{1-\frac{\beta_*}{2k}}\beta_*^{-\frac{\beta_*}{2k}}\cdot\frac{2k}{2k-\beta_*}.$$

Using the inequality $(A+B)^{\frac{1}{\beta_*}} \le A^{\frac{1}{\beta_*}} + B^{\frac{1}{\beta_*}}$ when $\beta_* \ge 1$, we have

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} \le \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + s_{2k}\mathrm{TV}(P,P')^{\left(\frac{1}{\beta_*}-\frac{1}{2k}\right)}\beta_*^{-\frac{1}{2k}} \cdot \left(\frac{2k}{2k-\beta_*}\right)^{\frac{1}{\beta_*}},$$

as desired. ■

Now we prove Theorem 6.17.

Proof By Lemma E.13, for any P' such that $\mathrm{TV}(P,P') \le \frac{\epsilon}{1-\epsilon}$, we have

$$\mathrm{CVaR}_\alpha(\theta;P) - \mathrm{CVaR}_\alpha(\theta;P') \le 2\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}.$$

By Corollary 6.13, if $\mathcal{R}_{\max}(\theta;P) > 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}$, then $\mathrm{CVaR}_\alpha(\theta;P) > 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}$, which implies that

$$\frac{\mathrm{CVaR}_\alpha(\theta;P')}{\mathcal{R}_{\max}(\theta;P)} \ge \frac{\mathrm{CVaR}_\alpha(\theta;P')}{\mathrm{CVaR}_\alpha(\theta;P)} = 1 - \frac{\delta}{\mathrm{CVaR}_\alpha(\theta;P)} \ge 1 - \frac{2\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}}{3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}} = \frac{1}{3}$$

holds for any P' such that $\mathrm{TV}(P,P') \le \frac{\epsilon}{1-\epsilon}$. By Lemma 6.15, taking the infimum over P' yields the first inequality of Eqn. (6.10). And by Corollary 6.13, we have $D_{\chi^2,\rho}(\theta;P') \ge \mathrm{CVaR}_\alpha(\theta;P')$ for all θ and P'. This combined with the above inequality yields the second inequality of Eqn. (6.10). ■

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, et al. Gpt-4 technical report. *OpenAI Blog*, 2023. 1

[2] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018. 1.5

[3] Kumar K Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. \alpha-req : Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17626–17638. Curran Associates, Inc., 2022. 3.3

[4] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2022. (iii)

[5] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*, 35:26671–26685, 2022. 1.5

[6] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. 2.1, 2.1, 2.2

[7] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000. 3.2

[8] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 1.2, 1.5

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[10] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural computation*, 16(10):2197–2219, 2004. 1.5

[11] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 1.5

[12] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88, 2007. 6

[13] Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66:259–294, 2007. 5.2

[14] Su Lin Blodgett, Lisa Green, and Brendan O'Connor. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas, November 2016. Association for Computational Linguistics. 6.1

[15] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1, 3

[16] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500, 2019. 6.4

[17] Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971. 6.4

[18] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer, 2011. D.5

[19] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881. PMLR, 09–15 Jun 2019. 6.2

[20] Vivien Cabannes, Bobak Kiani, Randall Balestriero, Yann Lecun, and Alberto Bietti. The SSL interplay: Augmentations, inductive bias, and generalization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3252–3298. PMLR, 23–29 Jul 2023. 1.5, 5.1

[21] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv: 2401.10774*, 2024. 4

[22] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016. 1.1, 2, 4, (ii)

[23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 1.4, 4.1

[24] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021. 2.1

[25] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2.4

[26] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. 1.14

[27] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. 4

[28] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. 2, 7

[29] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 1.2, 5.3

[30] Noel Cressie and Timothy RC Read. Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(3):440–464, 1984. 6.4

[31] Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International conference on machine learning*, pages 1528–1537. PMLR, 2019. 1.5

[32] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007. 2

[33] Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4976–4992. PMLR, 17–23 Jul 2022. 2.5

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 1, 1.2, 2.1, 4, 6.4

[35] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1.2, 5.1

[36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2.6

[37] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. 1

[38] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. 6.1, 6.4

[39] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 7

[40] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*,

2019. 5.5

[41] Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *The Journal of Machine Learning Research*, 21(1):8464–8501, 2020. 5.2, 5.2, 5.4, 5.27, D.7

[42] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 4.2

[43] Marco Fumero, Marco Pegoraro, Valentino Maiorca, Francesco Locatello, and Emanuele Rodolà. Latent functional maps: a spectral framework for representation alignment. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66178–66203. Curran Associates, Inc., 2024. 1.5

[44] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023. 2.4

[45] Gemini Team. Gemini: A Family of Highly Capable Multimodal Models. *arXiv e-prints*, page arXiv:2312.11805, December 2023. 1

[46] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 1.2

[47] Mehmet Gönen and Ethem Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(64):2211–2268, 2011. 4.2

[48] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2.4

[49] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 4.4

[50] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 2.1

[51] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. 6.1, 6.4

[52] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018. 6.2

[53] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incen-

tivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1.1, 2.3, 7

[54] Gerald J Hahn and William Q Meeker. *Statistical intervals: a guide for practitioners*, volume 92. John Wiley & Sons, 2011. 5.1

[55] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020. 2.4

[56] Jeff Z HaoChen and Tengyu Ma. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022. 1.5

[57] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021. 1.14, 1.5, 2, 2.1

[58] Jeff Z. HaoChen, Colin Wei, Ananya Kumar, and Tengyu Ma. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1.5

[59] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In Jennifer Dy and Andreas Krause, editors, *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1929–1938, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 6.4

[60] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2.6, 6.4

[62] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. 1.5

[63] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 2.3

[64] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2.4

[65] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023. 4.4

[66] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. 1

[67] Dirk Hovy and Anders Søgaard. Tagging performance correlates with author age. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*, pages 483–488, 2015. 6.1

[68] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in*

neural information processing systems, 19:601–608, 2006. 6

[69] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025. 3

[70] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR, 13–18 Jul 2020. 1.5

[71] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 6.4

[72] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv:1608.08614*, 2016. 1, 1.5

[73] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proc. International Conference on Machine Learning*, Vienna, Austria, July 2024. 1, 1.5, 2.6, 2.6, 2.6

[74] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. 1

[75] Francesco Insulla, Shuo Huang, and Lorenzo Rosasco. Towards a learning theory of representation alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. 1.5

[76] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 4

[77] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 6.2, E.7

[78] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 1.1, 7

[79] Ziwei Ji, Miroslav Dudík, Robert E. Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2109–2136. PMLR, 09–12 Jul 2020. 6.2, E.5

[80] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. 1.5, 2.6

[81] Daniel D. Johnson, Ayoub El Hanchi, and Chris J. Maddison. Contrastive learning can find an optimal basis for approximately view-invariant functions. In *The Eleventh International Conference on Learning Representations*, 2023. 1.3, 1.5

[82] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna

Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021. 7

[83] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011. 1.1

[84] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 2.6

[85] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 4.1

[86] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 2, 2.5, 7

[87] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. 2.4

[88] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 18–24 Jul 2021. 6, 6.4

[89] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019. 1, 1.5, 2.6, 2.6

[90] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008. 1.5

[91] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 1.2

[92] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1, 1.5

[93] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 1.5

[94] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9(1):3–3, 2016. 6.3

[95] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3.3

[96] Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021. 1.5

[97] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empow-

ers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024. 7

[98] Paul Pu Liang. Foundations of multisensory artificial intelligence. *arXiv preprint arXiv:2404.18976*, 2024. 7

[99] Bingbin Liu, Daniel Hsu, Pradeep Ravikumar, and Andrej Risteski. Masked prediction tasks: a parameter identifiability view. *Advances in Neural Information Processing Systems*, 2022. 1.5

[100] Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 1.5

[101] Qingshan Liu, Hanqing Lu, and Songde Ma. Improving kernel fisher discriminant analysis for face recognition. *IEEE transactions on circuits and systems for video technology*, 14(1):42–49, 2004. 3.2

[102] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. 1

[103] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 6.4

[104] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 2.5

[105] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36, 2023. (document), 4.2, 4.4

[106] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pages 41–48. Ieee, 1999. 3.2

[107] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1.5

[108] Youssef Mroueh, Stephen Voinea, and Tomaso A Poggio. Learning with group invariant features: A kernel perspective. *Advances in neural information processing systems*, 28, 2015. 1.5

[109] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[110] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, December 2022. 1, 1, 2.4, 7

[111] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 6

[112] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. 1, 2.1

[113] David Pfau, Stig Petersen, Ashish Agarwal, David G. T. Barrett, and Kimberly L. Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations*, 2019. 2.2

[114] Ashwini Pokle, Jinjin Tian, Yuchen Li, and Andrej Risteski. Contrasting the landscape of contrastive and non-contrastive learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 8592–8618. PMLR, 2022. 1.5

[115] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009. 6

[116] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1.2, 2.2

[117] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 1, 1, 1.2, 4

[118] Anant Raj, Abhishek Kumar, Youssef Mroueh, Tom Fletcher, and Bernhard Schölkopf. Local group invariant representations via orbit embeddings. In *Artificial Intelligence and Statistics*, pages 1225–1235. PMLR, 2017. 1.5

[119] Lewis Fry Richardson. Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 210(459-470):307–357, 1911. 5.4

[120] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 1.2

[121] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. 6.2, 6.1, 6.2

[122] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pages 19250–19286. PMLR, 2022. 1.5, 5.1

[123] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pages 5628–5637. PMLR, 2019. 1.5

[124] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 1.2, 1.5, 2, 5.3, 5.3

[125] John Shawe-Taylor, Christopher KI Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the gram matrix and the generalization error of kernel-pca. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005. 2.5, 2.5, 5.29

[126] Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 19847–19878. PMLR, 2022. 1.5

[127] Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR, 2019. 6.4

[128] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 2.1, 2.5, 6, 6.1

[129] Ravid Shwartz-Ziv, Randall Balestriero, Kenji Kawaguchi, Tim GJ Rudner, and Yann LeCun. An information theory perspective on variance-invariance-covariance regularization. *Advances in Neural Information Processing Systems*, 36:33965–33998, 2023. 1.5

[130] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. 5.1

[131] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1.5

[132] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2.4

[133] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018. 6.2

[134] Ilya Sutskever. Test of time award talk: Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 2024. 1, 1.1

[135] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007. 3.3

[136] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 1.5

[137] Rachael Tatman. Gender and dialect bias in youtube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59, 2017. 6.1

[138] Christos Thrampoulidis, Ganesh Ramachandra Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *Advances in Neural Information Processing Systems*, 35:27225–27238, 2022. 2.1

[139] Yuandong Tian. Deep contrastive learning is provably (almost) principal component analysis. *Advances in Neural Information Processing Systems*, 2022. 1.5

[140] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10268–10278. PMLR, 2021. 1.5

[141] Ryan Tibshirani. Gradient descent. Lecture Notes, 2019. Available at: `https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/grad-descent.pdf`. E.1

[142] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *J. Mach. Learn. Res.*, 22:281–1, 2021. 1.5

[143] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021. 1.5

[144] Boris Van Breugel and Mihaela Van Der Schaar. Position: Why tabular foundation models should be a research priority. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 48976–48993. PMLR, 21–27 Jul 2024. 1

[145] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 1.5

[146] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. 2.5, 4.4

[147] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010. E.3

[148] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. 5.2, D.3

[149] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *BLACKBOXNLP@EMNLP*, 2018. 5.1

[150] Ke Alexander Wang, Niladri Shekhar Chatterji, Saminul Haque, and Tatsunori Hashimoto. Is importance weighting incompatible with interpolating classifiers? In *International Conference on Learning Representations*, 2022. 6.2

[151] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 6

[152] Yifei Wang, Qi Zhang, Tianqi Du, Jiansheng Yang, Zhouchen Lin, and Yisen Wang. A message passing perspective on learning dynamics of contrastive learning. In

The Eleventh International Conference on Learning Representations, 2023. 1.5

[153] Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170, 2021. 1.5

[154] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. Survey Certification. 1

[155] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1.1, 7

[156] Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11112–11122. PMLR, 2021. 1.5

[157] Zixin Wen and Yuanzhi Li. The mechanism of prediction head in non-contrastive self-supervised learning. *Advances in Neural Information Processing Systems*, 2022. 1.5

[158] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2023. 5.1

[159] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15802–15812, 2023. 2.4

[160] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 1.5

[161] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 1.5

[162] Da Xu, Yuting Ye, and Chuanwei Ruan. Understanding the role of importance weighting for deep learning. In *International Conference on Learning Representations*, 2021. 6.2

[163] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 2.1

[164] Runtian Zhai, Chen Dan, J Zico Kolter, and Pradeep Kumar Ravikumar. Understanding why generalized reweighting does not improve over ERM. In *The Eleventh International Conference on Learning Representations*, 2023. 6.1

[165] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Doro: Distribu-

tional and outlier robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12345–12355. PMLR, 18–24 Jul 2021. 6.1

[166] Runtian Zhai, Chen Dan, Arun Suggala, J Zico Kolter, and Pradeep Kumar Ravikumar. Boosted CVar classification. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 6.1

[167] Runtian Zhai, Bingbin Liu, Andrej Risteski, Zico Kolter, and Pradeep Ravikumar. Understanding augmentation-based self-supervised representation learning via rkhs approximation and regression. In *International Conference on Learning Representations*, 2024. 1.5, 5.1, 5.2

[168] Runtian Zhai, Rattana Pukdee, Roger Jin, Maria Florina Balcan, and Pradeep Kumar Ravikumar. Spectrally transformed kernel regression. In *The Twelfth International Conference on Learning Representations*, 2024. 1.5, 5.3

[169] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 1.2, 1.5

[170] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003. 5.3, 5.5

[171] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. In *CMU CALD tech report CMU-CALD-02-107*, 2002. 5.3