NONPARAMETRIC PERSPECTIVE OF DEEP LEARNING

by

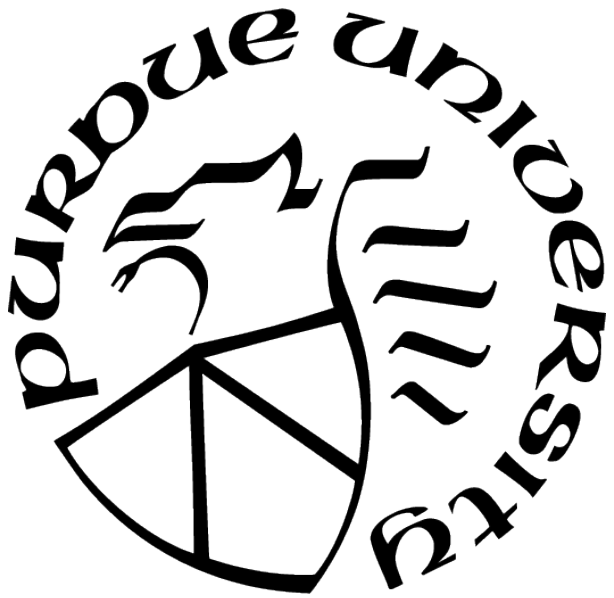
Tianyang Hu

A dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Statistics

West Lafayette, Indiana

December 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Guang Cheng, Chair

Department of Statistics

Dr. Faming Liang

Department of Statistics

Dr. Xiao Wang

Department of Statistics

Dr. Bruno Ribeiro

Department of Computer Science

Approved by:

Dr. Jun Xie

To my beloved family and friends

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Guang Cheng for his continuous support during my Ph.D. study and research. When I first got into the area of deep learning, it is his brilliant guidance that helped me to find my niche in the vast and fast-growing field. As my advisor, Prof. Cheng provided me with not only valuable suggestions for new research topics and guidance for pursuing them, but also plenty of great opportunities for attending academic conferences, connecting with potential collaborators, etc. Throughout my Ph.D. study, Prof Cheng always inspired me to be a rigorous and independent researcher.

I deeply appreciate the guidance from my committee members: Prof. Xiao Wang, Prof. Faming Liang, and Prof. Bruno Ribeiro, for their insightful comments and questions that help me complete my dissertation. I would also like to acknowledge my collaborators, Prof. Zuofeng Shang from NJIT, Dr. Ruiqi Liu from Texas Tech, Dr. Wenjia Wang, and Cong Lin from SAMSI. Without them, my research wouldn't have gone this far and I want to thank them for their time and efforts in our collaboration.

To my dear friends, thank you for all the support over the years and it's a great pleasure spending quality time with you during my time at Purdue. I want to express my great appreciation to my fellow students, Botao Hao, Jiasen Yang, Cheng Li, Yixuan Qiu, Jincheng Bai, Meimei Liu, Boqian Zhang, Yao Chen, Jiapeng Liu, Xiaodong Huang, Wenbin Zhu, Fan Wu, Hao Xin, Zhanyu Wang, Yue Xing, Mao Ye, Xiang Lyu, among many others. Special thanks to my girlfriend, Hanxi Sun, for all the support, encouragement and always being there when I needed the most.

Finally, I would like to thank my family for their selfless support of my study, both financially and spiritually, especially my mother. To my mom, with all my heart, I am extremely grateful to you for all the unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

ABBREVIATIONS

DL	Deep learning
DNN	Deep neural network
CNN	Convolutional neural network
ONN	Overparametrized neural network
ReLU	Rectified linear unit
i.i.d.	Independent and identically distributed
RKHS	Reproducing kernel Hilbert space
NTK	Neural tangent kernel
GD	Gradient descent
CV	Cross validation
ES	Early stopping

ABSTRACT

Models built with deep neural network (DNN) can handle complicated real-world data extremely well, seemingly without suffering from the curse of dimensionality or the non-convex optimization. To contribute to the theoretical understanding of deep learning, this work studies the nonparametric perspective of DNNs by considering the following questions: (1) What is the underlying estimation problem and what are the most appropriate data assumptions? (2) What is the corresponding optimal convergence rate and does the curse of dimensionality occur? (3) Is the optimal rate achievable for DNN estimators and is there any optimization guarantee? These questions are investigated on two of the most fundamental problems — regression and classification. Specifically, *statistical optimality* of DNN estimators is established under various settings with special focuses on the *curse of dimensionality* and *optimization guarantee*.

In the classic binary classification problem, statistical optimal convergence rates that suffer less from the curse of dimensionality are established under two settings: (1) Under the smooth boundary assumption [1], I show that DNN classifiers with proper architectures can benefit from the compositional smoothness structure [2] underlying the high dimensional data in the sense that the optimal convergence rates only depend on some effective dimension d^*, potentially much smaller than the data dimension d. (2) Under a novel teacher-student framework that assumes the Bayes classifier to be expressed as ReLU neural networks, I obtain a dimension-free rate of convergence $O(n^{-2/3})$ for DNN classifiers, which is also proven optimal.

1. INTRODUCTION

Deep learning has shown outstanding empirical successes and demonstrates superior performance in many standard machine learning tasks, such as image classification [3]–[5], generative modeling [6], [7], etc. Various benchmark scores have been drastically improved by the introduction of deep neural networks [4]. One major surprise of deep learning methods is their high representation power and accurate predictive performance in analyzing massive and high-dimensional datasets. Despite common accusations of being a black box with no theoretical guarantee, DNNs tend to achieve higher accuracy than other classical methods in various prediction tasks, which attracts plenty of interests from researchers. In contrast to the huge empirical success, little is yet settled from the theoretical side why DNN outperforms other methods. Without enough understanding, practical use of deep learning models could be inefficient and unreliable. To this end, there are mainly three aspects of theoretical deep learning.

Approximation DNN as a function space has great flexibility and capacity. For different structures and activation functions, what kind of functions can DNN efficiently approximate? [8] shows that even a single hidden layer neural network can approximate continuous functions on compact subsets of $\mathbb{R}^n$ arbitrarily well, as long as the number of neurons is large enough. [9] considers the universal approximation property when the width of the network is fixed and investigates the minimal width such that DNN can approximate continuous functions on unit cube arbitrarily well with increasing depth. In particular, [9] shows that DNN with Rectified Linear Units (ReLU) activation of width $d+1$ can approximate any continuous d-dimensional convex function arbitrarily well. [10] show that there exist certain functions representable by a ReLU DNN such that for any ReLU DNN with at fewer layers, it will require exponentially many more total nodes to represent. Optimality has also been established when representing smooth functions. To approximate d-variate, β-time differentiable functions to error ϵ measured in $\|\cdot\|_\infty$ norm, [11] show that DNN needs $SL \asymp O(\epsilon^{-\frac{d}{\beta}})$, where S is the number of nonzero weights and L is the depth.

Optimization The optimization in training DNN is highly non-convex. However, simple gradient-based methods such as stochastic gradient descent (SGD) works fairly well in practice. [12] propose Adam, the adaptive learning rate optimization algorithm that's been designed specifically for training DNNs. Many researchers have been studying the loss surface of DNN optimization [13]–[15] and convergence properties of certain gradient-based algorithms [16]–[19]. [20] proves that under certain assumptions, optimizing the squared loss of DNN has no poor local minima that every local minima is a global minima and every critical point is either a global minimum or a saddle point. [21], [22] show that adding one exponential neuron in the DNN can eliminate all bad local minimums. [17] specifically consider training one-hidden-layer ReLU neural network with GD and show that as long as the network is heavily overparametrized and initialized closely to zero, the training loss converges to zero as training step increases. [23] introduced neural tangent kernel (NTK) to characterize the convergence behavior of infinitely wide DNNs and it inspired numerous follow-ups [24]–[26].

Generalization Advances in optimization assure that we can efficiently minimize the empirical risk. But how close is the empirical risk minimizer to the population counterpart? Generalization error bound quantifies the gap between training error and population error. In learning theory, the generalization bound is directly linked to the complexity measurement of the model [27]. Various generalization error bounds in deep learning are developed using the PAC-Bayesian framework [28], [29] and Rademacher complexity [17], [30]–[32]. It's empirically observed that DNNs have great generalization ability and overparametrization tends to help with generalization. The model generalizes well even when training data is interpolated and the prediction error keeps decreasing after training error reaches zero [33], [34]. Among others, [35]–[37] link overparametrization to good generalization behavior and [38]–[40] study the effect of implicit or explicit regularizations on generalization.

1.1 The Nonparametric Perspective

The aforementioned theories are not perfect in characterizing the performance of DNNs. On one hand, despite the huge empirical success, deep learning is not better than traditional methods in every task. In turn, the success of DL should not only be contributed to the

effectiveness of DNNs, but also those specific tasks themselves, e.g., the data structures, noise level, etc. The approximation capacity of DNNs, the flexibility of the architecture, the adaptivity to specific tasks all contribute to deep learning's empirical success. However, the optimization/generalization perspective mainly depends on properties of the DNNs but not the data distributions or the tasks at hand. To illustrate, [20] shows that under mild assumptions, i.e., full rank, distinct eigenvalues in training data matrices, every local minimum is a global minimum for deep linear network. [16] prove that under some regularity conditions, gradient descent (GD) provably optimizes overparametrized neural networks. A more comprehensive understanding of deep learning can be developed by incorporating the underlying data assumptions into the analysis of DNNs.

On the other hand, the generalization error bounds mostly depend on the complexity of the DNN family used, often independent of data. Typical complexity measures include VC-dimension [41], number of parameters, norm or margin based complexities [30], [31], [33], [42]. However, almost all generalization error bounds are vacuous [43] and often doesn't reflect the actual generalization performance. [44] carried out large scale of empirical studies and showed that theoretical bounds doesn't correlate well with practice. The current generalization error bounds are not tight enough and sharper tools are needed.

To this end, statistics has a lot to offer, especially the nonparametric estimation perspective, where **task-specific** and **statistical optimal** results can be derived. The nonparametric perspective views the supervised or unsupervised learning tasks as estimation problems. By making specific assumptions about the data, the corresponding optimal rate of convergence can be established and we can sharply characterize the performance of different estimation methods. Together with sharp characterizations of the DNNs, this nonparametric perspective provides another angle to understand why models built with neural networks handle large-scale, high dimensional data extremely well. Specifically, we want to answer the following questions for the tasks DL excels at:

- What is the estimation problem and what are the most appropriate data assumptions?
- What is the corresponding optimal convergence rate and does *curse of dimensionality* occur?

- Is the optimal rate achievable for DNN estimators? If so, are there any algorithmic or optimization guarantees?

From the nonparametric perspective, an estimation method is said to have **statistical optimality** if it achieves the above optimal rate of convergence, indicating that it performs the best in the worst possible scenario. The current gold standard in deep learning community is empirical performance, which depends on too many aspects, e.g. DNN structure, initialization, step size, tuning parameters, etc. and doesn't provide a fair assessment of the estimation method at its core. Statistical optimality, on the other hand, focuses on asymptotic behaviours of the estimator in the specific estimation problem and can provide clearer, more quantitative characterizations of the methods. Comparing to the typical theoretical DL approaches, the proposed nonparametric perspective provides new insights and the key differences are highlighted in Table 1.1.

Table 1.1. Nonparametric perspective from Statistics v.s. Optimization/Generalization perspective. The modified check mark means not quite, in between yes (✓) and no (✗). There are data assumptions when analyzing optimization/generalization but they are not as thorough as those from the nonparametric estimation perspective. In turn, theories from optimization/generalization are not as strong, e.g., the generalization error bound can be tighter and tighter but no optimality can be established.

	Nonparametric	Optimization/Generalization
Ground Truth Assumption	✓	✓́
Theoretical Guarantee	✓	✓́
Optimality	✓	✗

Studying the nonparametric perspective of deep learning can produce **sharp** characterization of the performance of DNN models and offer fair comparisons between different models. As a different angle, the nonparametric perspective compliments the other research areas revolving DNNs. To summarize, this thesis views DNNs as flexible nonparametric estimation tools and investigates whether DNN based methods can achieve statistical optimal rates in popular tasks of deep learning. Under various settings, affirmative answers are given with

special focuses on the *curse of dimensionality* in Section 2 and *optimization guarantee* in Section 3.

Acknowledgment Section 2.2 and Section 3 is based on my own preprints, [45], [46] respectively.

1.2 Preliminary

Notations Bold letters denote vectors and regular letters denote scalars. For any function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$, denote $\|f\|_\infty = \sup_{\boldsymbol{x}\in\mathcal{X}} |f(\boldsymbol{x})|$ and $\|f\|_p = (\int_{\mathcal{X}} |f(\boldsymbol{x})|^p d\boldsymbol{x})^{1/p}$. For any vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_p$ denotes its p-norm, for $1 \le p \le \infty$. L_p and l_p are used to distinguish function norms and vector norms. For two given sequences $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$ of real numbers, we write $a_n \lesssim b_n$ if there exists a constant $C > 0$ such that $a_n \le Cb_n$ for all sufficiently large n. Let $\Omega(\cdot)$ be the counterpart of $O(\cdot)$ that $a_n = \Omega(b_n)$ means $a_n \gtrsim b_n$. Further, $a_n = \tilde{O}(b_n)$ and $a_n = \tilde{\Omega}(b_n)$ are used to hide the $\log n$ terms. Similarly, $\bar{O}(\cdot)$ and $\bar{\Omega}(\cdot)$ are used to indicate there are specific requirements for the multiplicative constants. We write $a_n \asymp b_n$ if $a_n \lesssim b_n$ and $a_n \gtrsim b_n$. Let $\lambda_{\min}(\boldsymbol{A})$ be the minimum eigenvalue of a symmetric matrix $\boldsymbol{A}$. We use $\mathbb{I}$ to denote the indicator function and $\boldsymbol{I}_d$ to denote the $d \times d$ identity matrix. $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and $\text{poly}(t_1, t_2, \ldots)$ denotes some polynomial function with arguments $t_1, t_2, \ldots$.

Neural Network Setup We consider DNNs with Rectified Linear Unit (ReLU) activation $\sigma(x) = \max\{0, x\}$. For a L-hidden-layer neural network, denote the weight matrices and bias vectors in each layer to be $W_1, W_2, \cdots, W_L$ and $b_1, b_2, \cdots, b_L$. We denote $\Theta = ((\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}))_{l=1,\ldots,L+1}$ to be the parameter set including all weights and biases. For the given Θ, let $|\Theta|$ be the number of layers in Θ. Let $N_{\max}(\Theta)$ be the maximum number of nodes, that is, $f(\cdot|\Theta)$ has at most $N_{\max}(\Theta)$ nodes at each layer. We define $\|\Theta\|_0$ as the number of nonzero parameters in Θ,

$$\|\Theta\|_0 = \sum_{l=1}^{L+1} \left(\|\text{vec}(\boldsymbol{W}^{(l)})\|_0 + \|\boldsymbol{b}^{(l)}\|_0 \right),$$

where $\text{vec}(\boldsymbol{W}^{(l)})$ transforms the matrix $\boldsymbol{W}^{(l)}$ into the corresponding vector by concatenating the column vectors. Similarly, we define $\|\Theta\|_\infty$ as the largest absolute value of the parameters in Θ,

$$\|\Theta\|_\infty = \max\left\{\max_{1\le l\le L+1}\|\text{vec}(\boldsymbol{W}^{(l)})\|_\infty, \max_{1\le l\le L+1}\|\boldsymbol{b}^{(l)}\|_\infty\right\}.$$

For a given n, let $\mathcal{F}_n$ be

$$\begin{aligned}\mathcal{F}_n &= \mathcal{F}^{\text{DNN}}(L_n, N_n, S_n, B_n, F_n)\\ &= \Big\{f(\boldsymbol{x}|\Theta) : |\Theta| \le L_n, N_{\max}(\Theta) \le N_n, \|\Theta\|_0 \le S_n,\\ &\qquad\qquad \|\Theta\|_\infty \le B_n, \|f(\cdot|\Theta)\|_\infty \le F_n\Big\}.\end{aligned}$$

Smoothness of Functions A function has Hölder smoothness index β if all partial derivatives up to order $\lfloor\beta\rfloor$ exist and are bounded, and the partial derivatives of order $\lfloor\beta\rfloor$ are $\beta - \lfloor\beta\rfloor$ Lipschitz. The ball of β-Hölder functions with radius R is then defined as

$$\begin{aligned}\mathcal{H}_d^\beta(R) = \Big\{&f : \mathbb{R}^d \to \mathbb{R} :\\ &\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta}\|\partial^{\boldsymbol{\alpha}} f\|_\infty + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor}\sup_{\substack{\boldsymbol{x},\boldsymbol{y}\in D\\ \boldsymbol{x}\neq\boldsymbol{y}}}\frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|}{|\boldsymbol{x}-\boldsymbol{y}|_\infty^{\beta-\lfloor\beta\rfloor}} \le R\Big\},\end{aligned} \tag{1.1}$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1}\ldots\partial^{\alpha_r}$ with $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_r) \in \mathbb{N}^r$ and $|\boldsymbol{\alpha}| := |\boldsymbol{\alpha}|_1$.

1.3 Nonparametric Regression

Suppose we observe data $\{(\boldsymbol{x}_{\text{i}}, y_{\text{i}})\}_{\text{i}=1}^n$, given by

$$y_{\text{i}} = f^*(\boldsymbol{x}_{\text{i}}) + \epsilon_{\text{i}}, \tag{1.2}$$

where f^* is the ground truth, $\boldsymbol{x}_{\text{i}} \in \mathbb{R}^d$, and ϵ_{i}'s are i.i.d. random noises with mean 0 and finite variance σ^2. The goal is construct an estimator $\hat{f}$ from data such that the L_2 estimation error $\|\hat{f} - f^*\|_2$ is small. From the nonparametric perspective, we want to know how fast does the error converge to zero as sample size grows. Note that the L_2 convergence rate critically depends on the assumptions of the true function, e.g., linearity, smoothness, boundedness,

etc., based on which minimax lower bounds are established [47]. In nonparametric statistics, [48] shows that when f^* is d-variate and β-time differentiable, the optimal rate of convergence for the L_2 estimation error is $n^{-\beta/(2\beta+d)}$. Many popular methods such as kernel methods, Gaussian process, splines, etc., achieve this rate.

DNNs in Nonparametric Regression Statistical optimality of DNN estimators has only been recently established. [2] considers using ReLU DNN in regression where the ground truth f^* in (1.2) is the composition of several smooth functions. Under the compositional smoothness assumption, the author proves that the DNN estimator (sparsely connected) from empirical risk minimization achieves the minimax optimal convergence rate up to a $\log(n)$ factor. Following the same setting, [49] later improved the convergence rate and removed the $\log(n)$ factor by considering B-spline, whose eigenvalues are known to have balanced orders. [50] investigate another compositional structure for the ground truth called generalized hierarchical interaction model, which is defined sequentially via smooth functions. Optimal convergence rate are given for the constructed DNN estimator, which is also structured and sparsely connected. If using fully connected DNN, [51] show that in general, the optimal rate cannot be achieved anymore. To showcase the advantage of DNN in regression, [52] consider learning a certain class of non-smooth functions, where ReLU DNNs are almost optimal while some of the popular models, linear estimators, e.g. kernel methods, splines, Gaussian processes, etc., do not attain the same rate. Statistical optimality has also been established in regression on manifolds [53]–[55].

1.4 Binary Classification

Classification is fundamentally different from regression due to the combinatorial nature of the class labels. Consider binary classification with a feature vector $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ and a label $y \in \{-1, 1\}$. Assume $\boldsymbol{x}|y = 1 \sim p(\boldsymbol{x}), \boldsymbol{x}|y = -1 \sim q(\boldsymbol{x})$ where p and q are two bounded densities on $\mathcal{X}$ w.r.t. some base measure $\mathbb{Q}$. If p, q have disjoint support, we say the data distribution or the classification problem is *separable*. For simplicity, let $\mathbb{Q}$ be the Lebesgue measure, positive and negative labels are equally likely to appear, i.e., labels are balanced. Denote classifiers to be $C : \mathcal{X} \to \{-1, 1\}$ and let $\mathcal{C}$ be a class of classifiers. The objective of

classification is to find the optimal classifier (called the Bayes classifier) C^*, which is defined as

$$C^* = \operatorname*{argmin}_{C \in \mathcal{C}} R(C) := \operatorname*{argmin}_{C \in \mathcal{C}} \mathbb{E}\left[\mathbb{I}\{C(\boldsymbol{x}) \neq y\}\right].$$

Using p, q, the Bayes classifier can be written as $C^*(\boldsymbol{x}) = \text{sign}(p(\boldsymbol{x}) - q(\boldsymbol{x}))$. We can estimate C^* from the training data by minimization the empirical risk. That is, we estimate C^* using $\hat{C}$, where

$$\hat{C}_n = \operatorname*{argmin}_{C \in \mathcal{C}_n} R_n(C) := \operatorname*{argmin}_{C \in \mathcal{C}_n} \sum_{\text{i}=1}^{n} \mathbb{I}\{C(\boldsymbol{x}_\text{i}) \neq y_\text{i}\}/n,$$

where $\mathcal{C}_n$ is a given class of classifiers depending on the sample size n. In practice, $\hat{C}$ is not computationally feasible because minimizing the empirical risk with the 0-1 loss over $\mathcal{C}_n$ is NP hard [56]. An alternative approach is to replace the 0-1 loss with other computationally easier losses so-called *surrogate losses*, e.g. logistic loss ($\phi(z) = \log(1 + \exp(-z))$), hinge loss ($\phi(z) = (1 - z)_+ = \max\{1 - z, 0\}$), etc. In addition, instead of a class of classifiers $\mathcal{C}_n$, we consider a class of real-valued functions $\mathcal{F}_n$. For a given surrogate loss ϕ, we estimate $\hat{f}$ by minimizing the surrogate empirical risk (or empirical ϕ-risk)

$$R_{\phi,n}(f) = \sum_{\text{i}=1}^{n} \phi(y_\text{i} f(\boldsymbol{x}_\text{i}))/n$$

on $\mathcal{F}_n$, and construct a classifier by $\hat{C}(\boldsymbol{x}) = \text{sign}(\hat{f}(\boldsymbol{x}))$. Accordingly, define an optimal f_ϕ^* as

$$f_\phi^* = \operatorname*{argmin}_{f \in \mathcal{F}} R_\phi(f),$$

where $R_\phi(f) := \mathbb{E}R_{\phi,n}(f)$ is the population risk. Given that $C(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$, with a slight abuse of notation, we write $R(C)$ and $R(f)$ interchangeably. A classifier C is evaluated by its excess risk defined as the difference of the population risk between C and the Bayes optimal classifier C^* that

$$\mathcal{E}(C, C^*) = R(C) - R(C^*) \quad \text{or} \quad \mathcal{E}_\phi(C, C^*) = R_\phi(C) - R_\phi(C^*).$$

Classification problem can be seen as **nonparametric estimation of sets**. The Bayes classifier C^* corresponds to the optimal decision region $G^* := \{\boldsymbol{x} \in \mathcal{X}, p(\boldsymbol{x}) - q(\boldsymbol{x}) \geq 0\}$. The set estimate $\hat{G} = \{\boldsymbol{x} \in \mathcal{X}, \hat{f}(\boldsymbol{x}) \geq 0\}$ can be constructed through deep neural network classifiers $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ trained using either 0-1 loss or surrogate losses. For set estimation, we define two distances over sets. The first one is the usual symmetric difference of sets: for any $G_1, G_2 \subset \mathbb{R}^d$,

$$d_{\triangle}(G_1, G_2) = \mathbb{Q}(G_1 \triangle G_2) = \mathbb{Q}\left((G_1 \backslash G_2) \cup (G_2 \backslash G_1)\right).$$

The second one is induced by densities p and q, which has deep connections to the 0-1 loss: for any $G_1, G_2 \subset \mathbb{R}^d$,

$$d_{p,q}(G_1, G_2) = \int_{G_1 \triangle G_2} |p(\boldsymbol{x}) - q(\boldsymbol{x})| \mathbb{Q}(d\boldsymbol{x}).$$

There are two key factors governing the rate of convergence in classification:

- The complexity of the set $\mathcal{G}^*$ where the optimal G^* resides.
- How concentrated the data are around the decision boundary;

For the first factor, *bracketing entropy* is often used to measure the complexity of a collection of subsets $\mathcal{G}$ in $\mathbb{R}^d$. For any $\delta > 0$, the bracketing number $\mathcal{N}_B(\delta, \mathcal{G}, d_{\triangle})$ is the minimal number of set pairs $(U_{\mathrm{j}}, V_{\mathrm{j}})$ such that

(a) For each j, $U_{\mathrm{j}} \subset V_{\mathrm{j}}$ and $d_{\triangle}(U_{\mathrm{j}}, V_{\mathrm{j}}) \leq \delta$;

(b) For any $G \in \mathcal{G}$, there exists a pair $(U_{\mathrm{j}}, V_{\mathrm{j}})$ such that $U_{\mathrm{j}} \subset G \subset V_{\mathrm{j}}$.

Simply denote $\mathcal{N}_B(\delta) = \mathcal{N}_B(\delta, \mathcal{G}, d_{\triangle})$ if no confusion arises. The bracketing entropy is defined as $H_B(\delta) = \log \mathcal{N}_B(\delta, \mathcal{G}, d_{\triangle})$. In statistics literature, one of the most common assumptions on the complexity is called smooth boundary fragments [1], [57]. The set $\mathcal{G}^*$ is assumed to be

$$\mathcal{G}_\beta := \{\boldsymbol{x} \in \mathbb{R}^d : h(\boldsymbol{x}_{-d}) - x_d \geq 0, h \in \mathcal{H}^\beta_{d-1}(R)\}, \tag{1.3}$$

where $\boldsymbol{x}_{-d} = (x_1, \cdots, x_{d-1})$ and $\mathcal{H}^\beta_{d-1}(R)$ is as defined in (1.1). It has been shown that such set of sets satisfies

$$H_B(\delta, \mathcal{G}_\beta, d_{\triangle}) \leq A\delta^{-\frac{d-1}{\beta}}.$$

For the second factor, the following *Tsybakov noise condition* [1] quantifies how close p and q are:

(N) There exists constant $c > 0$ and $\kappa \in [0, \infty]$ such that for any $0 \le t \le T$

$$\mathbb{Q}\left(\{\boldsymbol{x} : |p(\boldsymbol{x}) - q(\boldsymbol{x})| \le t\}\right) \le ct^{\kappa}.$$

The parameter $\kappa > 0$ is referred to as the *noise exponent*. The bigger the κ, the less concentrated the data are around the decision boundary and hence the easier the classification. In the extreme case that p, q have different supports, κ can be arbitrarily large (∞) and the classification is easy. To another extreme where $\mathbb{Q}\{\boldsymbol{x} \in \mathcal{X} : p(\boldsymbol{x}) = q(\boldsymbol{x})\} > 0$, there exists a region where different classes are indistinguishable. In this case, $\kappa = 0$ and the classification is hard in that region. Under the smooth boundary fragment assumption (smoothness β) and the Tsybakov noise condition (noise exponent κ), [1] shows that the optimal rate of convergence for the 0-1 loss excess risk is

$$\inf_{C \in \mathcal{C}} \sup_{G^* \in \mathcal{G}_\beta} \mathcal{E}(C, C^*) = \Omega\left(n^{-\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)\kappa}}\right), \tag{1.4}$$

where $\mathcal{C}$ is any classifier family.

DNN in Classification Convergence rate of DNN classifiers has also been investigated. [58] derive fast convergence rates of ReLU DNN classifiers learned using the hinge loss. Under the smooth boundary fragment assumption (1.3) and Tsybakov noise condition (N), the empirical hinge loss minimizer

$$\hat{f}_{\phi,n} = \operatorname*{argmin}_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{\mathrm{i}=1}^{n} \phi(y_{\mathrm{i}} f(\boldsymbol{x}_{\mathrm{i}})),$$

within some DNN family with carefully selected L_n, N_n, S_n, B_n, and F_n satisfies

$$\sup_{C^* \in \mathcal{G}_\beta} \mathbb{E}\left[\mathcal{E}(\hat{f}_{\phi,n}, C^*)\right] \lesssim \left(\frac{\log^3 n}{n}\right)^{\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)(\kappa+1)}},$$

which is almost optimal comparing to the minimax lower bound (1.4). Inspired by the success of convolutional neural network (CNN) in image classification, [59] analyze classifiers based on CNNs and show that under suitable assumptions on the smoothness and structure of the conditional probability, the convergence rate is fast and independent of the dimension of the data. However, no statistical optimality is established.

2. STATISTICAL OPTIMALITY THAT BREAKS THE CURSE OF DIMENSIONALITY

With the introduction of convolutional neural network [3] and residual neural network (ResNet) [5], various benchmarks in computer vision have been revolutionized and neural network based methods have achieved better-than-human performance [60]. For instance, AlexNet [3] and its variants [61], [62] have demonstrated superior performance in ImageNet data [63], [64], where the data dimension is huge, i.e., each image has pixel size 256×256 and hence is an 65536-dimensional vector. This is quite surprising given that neither structural model assumptions, such as additive or sparsity structure, are imposed, nor explicit dimension reduction steps, such as LASSO, are incorporated in deep learning methods. Traditional statistical thinking sounds an alarm when facing such high-dimension data as the "curse of dimensionality" usually prevents nonparametric classification achieving fast convergence rates. In this chapter, we attempt to provide theoretical explanations for the empirical success of deep neural networks in (especially high dimensional) classification, beyond the existing statistical theories.

In the context of nonparametric regression, similar investigations have been recently carried out. Among others [49], [65]–[69], [2] showed that deep ReLU neural networks can achieve minimax rate of convergence when the underlying regression function possesses a certain compositional smooth structure; [50], [70] showed a similar result by considering an alternative hierarchical interaction models. However, for classification tasks, there are few similar results. Classification and regression are fundamentally different due to the discrete nature of class labels. Specifically, in nonparametric regression, we are interested in recovering the whole underlying function while in classification, the focus is on the nonparametric estimation of sets corresponding to different classes, i.e., the decision boundaries. As a result, it is well known that many established results on regression cannot be directly translated to classification.

The goal of this section is hence to fill this gap by investigating how well neural network based classifiers can perform in theory and further provide a theoretical explanation for the "break-the-curse-of-dimensionality" phenomenon. Recall the optimal convergence rate

in (1.4) and note that curse of dimensionality does occur in this bound. As d gets larger, the rate becomes extremely slow. In ultra-high dimension, a natural assumption to make is that the true classifier does have some low-dimensional structure. To this end, following the road map of our nonparametric perspective, two settings with different data assumptions are investigated.

2.1 Smooth Boundary Fragments with Compositional Structure

As a starting point, we adopt the compositional smoothness assumption [2] with effective dimension d^* and effective smoothness β^{**} in the smooth boundary fragment setting (2.1) and investigate the rate of convergence of the excess risk.

Compositional Smooth Function Assume h in (2.1) is of the compositional form in [2] such that

$$h = g_l \circ g_{l-1} \circ \ldots \circ g_1 \circ q_0, \tag{2.1}$$

where $g_{\mathrm{i}} : [a_{\mathrm{i}}, b_{\mathrm{i}}]^{d_{\mathrm{i}}} \to [a_{\mathrm{i}+1}, b_{\mathrm{i}+1}]^{d_{\mathrm{i}+1}}$. Denote components of g_{i} by $\{g_{\mathrm{ij}}\}_{\mathrm{j}=1}^{d_{\mathrm{i}+1}}$ and let t_{i} be the maximal number of variables g_{ij}'s depend on. Thus, each g_{ij} is a t_{i}-variate function. It's further assumed that each function g_{ij} shares the same Hölder smoothness β_{i}. Since g_{ij} is also t_{i}-variate, $g_{\mathrm{ij}} \in \mathcal{C}_{t_{\mathrm{i}}}^{\beta_{\mathrm{i}}}([a_{\mathrm{i}}, b_{\mathrm{i}}]^{t_{\mathrm{i}}}, M_{\mathrm{i}})$ and the underlying function space becomes

$$\mathcal{H}\Big(l, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, R\Big) := \Big\{ h = g_l \circ \ldots \circ q_0 \,:\, g_{\mathrm{i}} = (g_{\mathrm{ij}})_{\mathrm{j}} : [a_{\mathrm{i}}, b_{\mathrm{i}}]^{d_{\mathrm{i}}} \to [a_{\mathrm{i}+1}, b_{\mathrm{i}+1}]^{d_{\mathrm{i}+1}}, \\ g_{\mathrm{ij}} \in \mathcal{C}_{t_{\mathrm{i}}}^{\beta_{\mathrm{i}}}\Big([a_{\mathrm{i}}, b_{\mathrm{i}}]^{t_{\mathrm{i}}}, R\Big), \text{ for some } |a_{\mathrm{i}}|, |b_{\mathrm{i}}| \le R \Big\},$$

with $\boldsymbol{d} := (d_0, \ldots, d_{q+1})$, $\boldsymbol{t} := (t_0, \ldots, t_q)$, $\boldsymbol{\beta} := (\beta_0, \ldots, \beta_q)$. Denote

$$\beta_{\mathrm{i}}^* := \beta_{\mathrm{i}} \prod_{l=\mathrm{i}+1}^{q} \min\{\beta_l, 1\} \quad \text{and} \quad \phi_n = \max_{\mathrm{i}=0,1,\cdots,q} n^{-\frac{2\beta_{\mathrm{i}}^*}{2\beta_{\mathrm{i}}^* + t_{\mathrm{i}}}} := n^{-\frac{2\beta^{**}}{2\beta^{**}+d^*}}.$$

In the above formula, β_{i}^* describes the effective smoothness for each layer of functions and the overall effective smoothness and dimension are denoted as β^{**} and d^*. For ease of notation, denote $\mathcal{H}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, R)$ as $\mathcal{H}(d^*, \beta^{**})$. Note that ϕ_n is proven to be the best possible rate from

regression with L_2 loss [2]. If $h(\boldsymbol{x})$ is a general $(d-1)$-dimensional smooth function with Hölder smoothness β, then $q = 0, \beta^{**} = \beta, d^* = d-1$.

Having defined the compositional structure in the decision boundary, denote the corresponding classifiers to be $\mathcal{C}(d^*, \beta^{**})$ where

$$\mathcal{C}(d^*, \beta^{**}) := \{\text{sign}(h(\boldsymbol{x}_{-d}) - x_d) : h \in \mathcal{H}(d^*, \beta^{**})\}.$$

Functions in $\mathcal{C}(d^*, \beta^{**})$ are not continuous but with discrete values. The next lemma establishes the DNN approximation result for functions in $\mathcal{C}(d^*, \beta^{**})$.

Lemma 2.1.1 *For any $\epsilon > 0, p \in \mathbb{N}^+$ and $C(\boldsymbol{x}) \in \mathcal{C}(d^*, \beta^{**})$, there exists a neural network f_C with layers at most $O(\log n + \log_2(1/\epsilon))$ and non-zero weights at most $O(\epsilon^{-d^*p/\beta^{**}} \log n + \log_2(1/\epsilon))$ such that*

$$\|C(\boldsymbol{x}) - f_C(\boldsymbol{x})\|_p \leq 2\epsilon.$$

Lemma 2.1.1 demonstrates the expressive power of DNN at approximating discrete functions and shows DNN can potentially recover the Bayes classifier arbitrarily well given large enough size. To further characterize how fast is the convergence rate, we introduce the following novel margin condition, which is a finer version of the Tsybakov noise condition (N).

2.1.1 Localized Margin Condition

Existing results fail to establish the statistical optimality of DNN classifiers in the smooth boundary fragment setting (2.1) while methods like sieve estimators can achieve the optimal rate of convergence [1]. Comparing the rates, the sub-optimality comes from the noise exponent term κ. To this end, instead of the classical Tsybakov noise condition (N), we propose to consider a localized, finer-grained margin condition that allows the separation between two classes to change along the decision boundary.

Without loss of generality, let $\mathcal{X} = [0,1]^d$. Let the optimal decision region associated with C^* be $G^* = \{\boldsymbol{x} \in [0,1]^d : h^*(\boldsymbol{x}_{-d}) - x_d \geq 0\}$ for some $h^* \in \mathcal{H}(d^*, \beta^{**})$. Denote the decision

boundary to be $\partial G^* := \{\boldsymbol{x} \in [0,1]^d : h^*(\boldsymbol{x}_{-d}) = x_d\}$. Without loss of generality, assume $\mathbb{Q}(\partial G^*) = 0$. For every point in the decision boundary $\boldsymbol{x} \in \partial G^*$, define

$$m_{\boldsymbol{x}_{-d}}(t) := |p((\boldsymbol{x}_{-d}, h^*(\boldsymbol{x}_{-d}) + t)) - q((\boldsymbol{x}_{-d}, h^*(\boldsymbol{x}_{-d}) + t))|,$$

which captures the how $|p(\boldsymbol{x}) - q(\boldsymbol{x})|$ changes along the direction of x_d on each point of the decision boundary. For ease of notation, we write $m_{\boldsymbol{x}_{-d}}(t)$ and $m_{\boldsymbol{x}}(t)$ when no confusion raises. Notice that $m_{\boldsymbol{x}}(0) = 0$ by definition. Further define for any $\boldsymbol{x} \in \partial G^*$,

$$K(\boldsymbol{x}) = \sup\{k \geq 0 : \lim_{t \to 0} \frac{m_{\boldsymbol{x}}(t)}{t^{1/k}} > 0\},$$

which characterizes the margin condition locally at $\boldsymbol{x}_{-d}$, i.e. how separated are p and q on each point of the decision boundary, along the direction of x_d. Similar to the κ in (N), the bigger the $K(\boldsymbol{x})$, the more separated are the two densities and the easier the classification problem locally at $\boldsymbol{x}$. Since ∂G^* is of measure zero, we know $K(\boldsymbol{x})$ is non-negative. The proposed localized margin condition is specified in the following.

(M1) There exists $\epsilon_0 > 0$ small enough and a constant $0 < C_{\epsilon_0} < \infty$ such that for all $\boldsymbol{x} \in \partial G^*$ and any $0 < t < \epsilon_0$,

$$\frac{1}{C_{\epsilon_0}} \leq \frac{m_{\boldsymbol{x}}(t)}{t^{1/K(\boldsymbol{x})}} \leq C_{\epsilon_0}.$$

(M2) $K(\boldsymbol{x})$ is α-Holder continuous for some $0 < \alpha \leq 1$, i.e. there exists constant C_K such that for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \partial G^*$,

$$|K(\boldsymbol{x}_1) - K(\boldsymbol{x}_2)| \leq C_K \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^{\alpha}.$$

(M1) and (M2) together provide a finer characterization of the margin condition. $K(\boldsymbol{x})$ specifies the separation at each point of the decision boundary and (M2) characterizes along the x_d dimension the smoothness of $K(\boldsymbol{x})$. Note that by Tsybakov noise condition (N) with exponent κ implies that $\kappa \leq \inf_{\boldsymbol{x} \in \partial G^*} K(\boldsymbol{x})$. The following lemma shows that (M1) also implies (N).

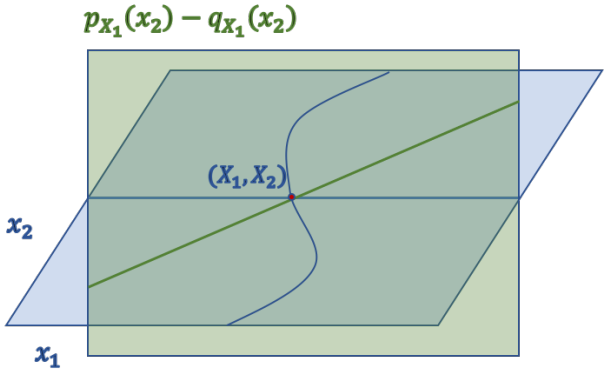


Figure 2.1. Illustration of the localized margin condition in the $d = 2$ case. Data are in the blue plane and the curved blue line is the decision boundary. Fix some X_2, along the x_1 direction (solid blue line), the density difference $p - q$ is plotted in the green plane as the green solid line. (M1) defines the noise exponent $K(X_1)$ locally at X_1. If $p - q$ is linear as shown, $K(X_1) = 1$.

Lemma 2.1.2 *If $\kappa^- = \inf_{\boldsymbol{x}\in\partial G^*} K(\boldsymbol{x})$, then condition (M1) implies Tsybakov noise condition (N) holds with $\kappa = \kappa^-$ and $T = \epsilon_0^{1/\kappa}/C_{\epsilon_0}$.*

Since we are considering a new condition on the separation along the decision boundary, the corresponding lower bound needs to be re-established, which is the goal of the next theorem.

Theorem 2.1.3 *Assume conditions (M1), (M2) with noise exponent $\kappa = \inf_{\boldsymbol{x}\in\partial G^*} K(\boldsymbol{x})$ and the composition structure (2.1) of the boundary function. Then, the excess risk has the following lower bound for any classifier*

$$\inf_{\hat{f}\in\mathcal{F}} \sup_{C^*\in\mathcal{C}(d^*,\beta^{**})} \mathbb{E}[\mathcal{E}(\hat{f}, C^*)] \gtrsim \left(\frac{1}{n}\right)^{\frac{\beta^{**}(\kappa+1)}{\beta^{**}(\kappa+2)+d^*\kappa}},$$

where $\mathcal{F}$ is an arbitrary function class.

Theorem 2.1.3 proves the optimal rate of convergence under the compositional assumption and localized margin condition (M1), (M2). Interestingly, the rate is adaptive to the optimal rate of convergence (1.4) under (N) established in [1]. On one hand, this lower bound is determined by the infimum of the localized noise exponent $K(\boldsymbol{x})$, which plays similar roles to the original κ. On the other hand, the rate only depends on the effective smoothness β^{**} and effective dimension d^*.

Next we investigate whether DNN classifiers can achieve this optimal rate under the proposed localized margin condition.

2.1.2 Localized Convergence Analysis

Defining the function $K(\boldsymbol{x})$ that describes the local margin condition enables us to consider local convergence behaviours. If $K(\boldsymbol{x}) \equiv \kappa$, then our localized margin condition produces the same results as those under (N). However, if $K(\boldsymbol{x}) = \infty$ in region A and $K(\boldsymbol{x}) = 0$ in region B, the classification problem is much easier at region A and the convergence rate should mainly depend on region B, i.e., locations with smaller $K(\boldsymbol{x})$ are the bottlenecks for the convergence rate. To justify this intuition, we conduct the following localized analysis.

Choose $M \in \mathbb{N}$ and divide $[0,1]^d$ along the $\boldsymbol{x}_{-d}$ dimensions into disjoint equal-sized grids

$$[0,1]^d = \bigcup_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}=1}^{M} D_{(\mathrm{j}_1,\cdots,\mathrm{j}_{d-1})},$$

where $D_{(\mathrm{j}_1,\cdots,\mathrm{j}_{d-1})} := \{\boldsymbol{x} \in [0,1]^d : x_1 \in [\frac{\mathrm{j}_1-1}{M}, \frac{\mathrm{j}_1}{M}), \cdots, x_{d-1} \in [\frac{\mathrm{j}_{d-1}-1}{M}, \frac{\mathrm{j}_{d-1}}{M}]\}$. For ease of notation, let $\mathrm{j}_{-d} = (\mathrm{j}_1, \cdots, \mathrm{j}_{d-1})$ and $\bar{\boldsymbol{x}}_{\mathrm{j}_{-d}}$ be the corresponding grid point. Denote J_M as all M^{d-1} combinations of j_{-d}'s described above. Correspondingly, divide the dataset as $\mathcal{D} = \cup_{\mathrm{j}_{-d} \in J_M} \mathcal{D}_{\mathrm{j}_{-d}}$ where $\mathcal{D}_{\mathrm{j}_{-d}} = \{(\boldsymbol{x}, y) : \boldsymbol{x} \in D_{\mathrm{j}_{-d}}\}$. Similarly, the 0-1 loss can be decomposed as

$$\begin{aligned} d_{p,q}(\hat{G}_n, G^*) &= \int_{\hat{G}_n \triangle G^*} |p(\boldsymbol{x}) - q(\boldsymbol{x})| d\boldsymbol{x} \\ &= \sum_{\mathrm{j}_{-d} \in J_M} \int_{(\hat{G}_n \triangle G^*) \cap D_{\mathrm{j}_{-d}}} |p(\boldsymbol{x}) - q(\boldsymbol{x})| d\boldsymbol{x} \\ &:= \sum_{\mathrm{j}_{-d} \in J_M} d_{\mathrm{j}_{-d}}(\hat{G}_n, G^*). \end{aligned}$$

Let the empirical 0-1 loss be $R_n(f) = \frac{1}{n}\sum_{\mathrm{i}=1}^n \mathbb{I}\{f(\boldsymbol{x}_\mathrm{i}) y_\mathrm{i} < 0\}$, which can also be decomposed into M^{d-1} parts that $R_n(G) = \sum_{\mathrm{j}_{-d} \in J_M} R_{n,\mathrm{j}_{-d}}$ where $R_{n,\mathrm{j}_{-d}} = \frac{1}{|\mathcal{D}_{\mathrm{j}_{-d}}|}\sum_{\mathrm{i}=1}^n \mathbb{I}\{\boldsymbol{x}_\mathrm{i} \in D_{\mathrm{j}_{-d}} : f(\boldsymbol{x}_\mathrm{i}) y_\mathrm{i} < 0\}$.

Recall the compositional smoothness assumption on h^* that it has an effective dimension d^* and effective smoothness β^{**}. Consider using a DNN family $\tilde{\mathcal{F}}_n$ to approximate h^*. By Lemma 2.1.1, for any $\epsilon > 0$, there exists a neural network $\tilde{f}_n \in \tilde{\mathcal{F}}_n$ with $\tilde{L}_n = O(\log n)$ layers

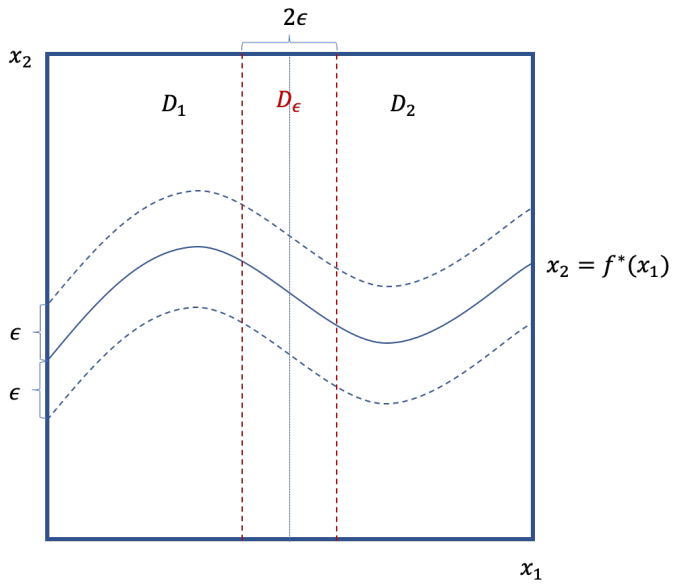


Figure 2.2. Illustration of region D_ϵ in $d = 2, M = 1$ case.

and $\tilde{S}_n = O(\epsilon^{-d^*/\beta^{**}} \log n)$ non-zero weights such that $\|\tilde{f}_n - h^*\|_\infty \leq \epsilon$. The size of $\tilde{f}_n$ is jointly determined by $\epsilon, d^*, \beta^{**}$ and n.

Now we focus on each $D_{\mathrm{j}_{-d}}$. Similarly to that in the whole region $[0,1]^d$, we have the following lemma.

Lemma 2.1.4 *Under assumption (M1), further assume for some* $\mathrm{j}_{-d} \in J_M$, $\kappa^- \leq K(\boldsymbol{x}) \leq \kappa^+$ *for all* $\boldsymbol{x} \in D_{\mathrm{j}_{-d}}$. *Let the empirical 0-1 loss minimizer be*

$$\hat{f}_{n,\mathrm{j}_{-d}} := \operatorname*{argmin}_{f \in \tilde{\mathcal{F}}_n} R_{n,\mathrm{j}_{-d}}(f).$$

Then the 0-1 loss excess risk satisfies

$$\sup_{h^* \in \mathcal{H}(d^*,\beta^{**})} \mathbb{E}(R_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}) - R_{\mathrm{j}_{-d}}(h^*)) = O\left(n^{-\frac{(\kappa^-+1)\beta^{**}}{(\kappa^-+2)\beta^{**}+\left(\frac{\kappa^-+1}{\kappa^++1}\right)d^*\kappa^+}}\right).$$

Remark 2.1.5 (Local Convergence Rate) *Our local convergence rate is very similar to the established one under the original Tsybakov noise condition (N). On one hand, the bottleneck is indeed the minimum of* $K(\boldsymbol{x})$ *in that region and* κ^- *plays the same role as* κ

in (N). On the other hand, the extra term in the denominator $(\kappa^- + 1)/(\kappa^+ + 1)$ *reveals the source of the sub-optimality of existing results. If no assumption is made on* κ^+*, then the best rate possible reduces to that in [58]. However, if* $\kappa^+ \approx \kappa^-$*, optimal rate can be attained.*

Next, we proceed from a localized convergence analysis to the global one and evaluate the overall convergence rate.

2.1.3 Construction of the Global Estimator

As illustrated in Figure 2.2, $\tilde{f}_n$ is inside the 2ϵ-band centering at h^*. Let

$$D_\epsilon = \{\boldsymbol{x} \in [0,1]^d : \|\boldsymbol{x}_{-d} - \bar{\boldsymbol{x}}_{\mathrm{j}_{-d}}\|_2 \leq \epsilon, \mathrm{j}_{-d} \in J_M\}$$

and define event $E_\epsilon := \{\boldsymbol{x}_\mathrm{i} \notin D_\epsilon : \forall \mathrm{i} = 1, 2, \ldots, n\}$. Since $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are both bounded densities (be c_0) and h^* is Holder smooth with finite radius, there exists some constant c_1 depending on c_0 and the radius such that

$$\mathbb{P}(x \in D_\epsilon) \leq c_0 \mathbb{Q}(D_\epsilon) \leq c_1 (M\epsilon)^d.$$

Therefore, if we choose M such that $nM^d\epsilon^d \to 0$ as $n \to \infty$, then

$$\mathbb{P}(E_\epsilon) \geq (1 - c_1(M\epsilon)^d)^n \to 1.$$

In the remaining of the analysis, we assume E_ϵ happens.

For any $f_n \in \tilde{\mathcal{F}}_n$, we make modifications and further construct $f^+_{n,\mathrm{j}_{-d}}$ that satisfies the following properties:

(P1) On $D_{\mathrm{j}_{-d}} \backslash D_\epsilon$, $f^+_{n,\mathrm{j}_{-d}} = f_n$;

(P2) Outside $D_{\mathrm{j}_{-d}}$, $f^+_{n,\mathrm{j}_{-d}} = 0$;

(P3) $f^+_{n,\mathrm{j}_{-d}} \in \tilde{\mathcal{F}}^+_n$ where $\tilde{\mathcal{F}}^+_n$ is slightly larger than $\tilde{\mathcal{F}}_n$ with $\tilde{L}^+_n = \tilde{L}_n + O(1)$ layers and $\tilde{S}^+_n = 2\tilde{S}_n + O(1)$ number of nonzero weights.

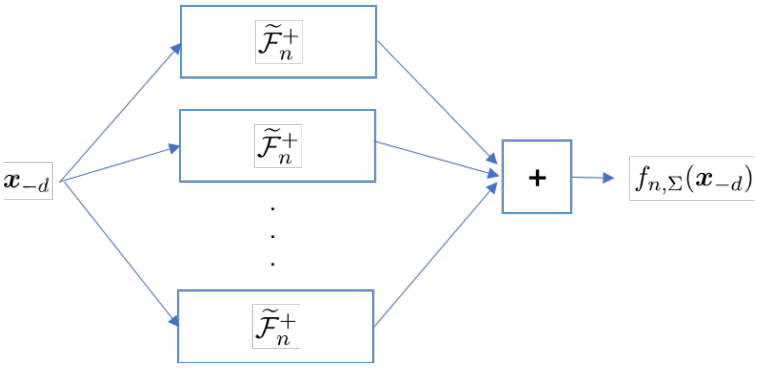


Figure 2.3. Illustration of the estimator DNN family $\mathcal{F}_n$.

The construction details and verification of (P1) to (P3) are deferred to Section 2.4.4. Let's proceed with the properties of $f^+_{n,\mathrm{j}_{-d}}$ and $\tilde{\mathcal{F}}_n^+$. By (P2), $f^+_{n,\mathrm{j}_{-d}}$ is zero outside $D_{\mathrm{j}_{-d}}$ and we can combine them together to define

$$f_{n,\Sigma}(\boldsymbol{x}_{-d}) = \sum_{\mathrm{j}_{-d}\in J_M} f^+_{n,\mathrm{j}_{-d}}(\boldsymbol{x}_{-d}). \tag{2.2}$$

Easy to see that $f_{n,\Sigma}(\boldsymbol{x})$ is still a ReLU network and let the overall DNN estimator to be of this form. Correspondingly, define such structured DNN family to be $\mathcal{F}_n$, which is $\tilde{\mathcal{F}}_n$ stacked in parallel M^{d-1} times. See Figure 2.3 for illustration.

Denote the overall empirical minimizer within $\mathcal{F}_n$ to be

$$\hat{f}_n := \operatorname*{argmin}_{f\in\mathcal{F}_n} R_n(f). \tag{2.3}$$

Due to the formulation of $\mathcal{F}_n$, $\hat{f}_n$ can be written in form of (2.2) as $\hat{f}_n = \sum_{\mathrm{j}_{-d}\in J_M} \hat{f}_{n,\mathrm{j}_{-d}}$. Under event E_ϵ, we have that for any $\mathrm{j}_{-d} \in J_M$,

$$R_{n,\mathrm{j}_{-d}}(\hat{f}_n) = R_{n,\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}) = \min_{f\in\tilde{\mathcal{F}}_n} R_{n,\mathrm{j}_{-d}}(f) \le R_{n,\mathrm{j}_{-d}}(\tilde{f}_n). \tag{2.4}$$

The second equality is guaranteed by event E_ϵ and property (P1). The last inequality is due to empirical risk minimization and the fact that $\tilde{f}_n \in \tilde{\mathcal{F}}_n$. (2.4) indicates that the global empirical minimizer within $\mathcal{F}_n$ also gives rise to the empirical minimizer locally within each $D_{\mathrm{j}_{-d}}$.

2.1.4 Optimal Rate of Convergence

Now we are ready to state the statistical optimality result of the DNN classifier $\hat{f}_n$ as in (2.3). Let $\rho = d^*/\beta^{**}$.

Theorem 2.1.6 *Under the compositional smoothness assumption (2.1), let* $\kappa = \min_{\boldsymbol{x}\in[0,1]^d} K(\boldsymbol{x})$. *Assume (M1), (M2),* $\rho < d^*$ *and* $n = \Omega(\epsilon_0^{-(1+\rho)})$. *Then with probability at least* $\exp(n^{\frac{\rho-d^*+1}{\rho+1.1}})$, *which goes to 1 as* $n \to \infty$, *the 0-1 excess risk for the empirical 0-1 loss minimizer satisfies*

$$\sup_{C^*\in\mathcal{C}(d^*,\beta^{**})} \mathbb{E}(R(\hat{f}_n) - R(C^*)) = \tilde{O}\left(n^{-\frac{\beta^{**}(\kappa+1)}{\beta^{**}(\kappa+2)+d^*\kappa}}\right).$$

Theorem 2.1.6 establishes the statistical optimality of DNN classifiers under the compositional smooth fragment assumption. The convergence rate only depends on the effective dimension d^*, which can be potentially much smaller than d. To further illustrate its power, we consider a special case where $h(\boldsymbol{x})$ is a $(d-1)$-dimensional additive function that

$$h(\boldsymbol{x}_{-\mathrm{i}}) = \sum_{\mathrm{i}\neq\mathrm{j}} h_{\mathrm{i}}(x_{\mathrm{i}}) = g_1 \circ q_0, \tag{2.5}$$

where $q_0(x_1, \cdots, x_{d-1}) = (h_1(x_1), \cdots, h_{d-1}(x_{d-1}))$ and $g_1(x_1, \cdots, x_{d-1}) = x_1 + \cdots + x_{d-1}$. In this case, $q = 1, \boldsymbol{d} = (d-1, d-1), \boldsymbol{t} = (1, d-1)$. Under the assumption that each $h_{\mathrm{i}}(\boldsymbol{x})$ has Hölder smoothness β, then $\boldsymbol{\beta} = (\beta, \infty)$ and the convergence rate under the additive structure is $\tilde{O}\left(n^{-\frac{\beta(\kappa+1)}{\beta(\kappa+2)+\kappa}}\right)$.

Remark 2.1.7 (Structured DNN) *The constructed DNN classifier* $\hat{f}_n$ *in Theorem 2.1.6 has special structures and is sparsely connected as illustrated in Figure 2.3. In order to have more practical impact, we want our DNN estimators to be as general as possible. However, such a structural requirement is not uncommon in nonparametric study of deep learning where almost all DNN estimators constructed with special structures [2], [50], [52]. In particular, [51] show that in regression, the optimal rate cannot be achieved generally by fully connected neural networks.*

We have shown that DNNs classifiers can indeed benefit from the compositional structure and statistical optimality has been established that breaks the curse of dimensionality.

However, the optimal rates in this section are still not dimension-free and the effective dimension d^* is hard to evaluate in practice. In the next section, we propose to study DNN classifiers in a teacher-student setting where the traditional smoothness assumption is no longer present.

2.2 Teacher-Student Framework for Classification

The teacher-student framework has originated from statistical mechanics [71]–[73] and recently gained increasing interest [74]–[77]. In this setup, one neural network, called student net, is trained on data generated by another neural network, called teacher net. While worst-case analysis for arbitrary data distributions may not be suitable for real structured dataset, adopting this framework can facilitate the understanding of how deep neural networks work as it provides an explicit target function with bounded complexity. Furthermore, assuming the target classifier to be a teacher network of an explicit architecture may provide insights on what specific architecture of the student classifier is needed to achieve an optimal excess risk. At the same time, by comparing the two networks, both optimization and generalization can be handled more elegantly. Existing works on how well student network can learn from the teacher mostly focus on regression problems and study how the student network evolves during training from computational aspects, e.g., [76], [78]–[81]. Still, there is a lack of statistical understanding in this important direction, particularly on classification aspects.

In this section, we consider the teacher-student framework where the optimal decision region is defined by ReLU neural networks. Recall that the Bayes classifier C^* is defined via the optimal decision region $G^* := \{\boldsymbol{x} \in \mathcal{X}, p(\boldsymbol{x}) - q(\boldsymbol{x}) \geq 0\}$. The set estimate $\hat{G} = \{\boldsymbol{x} \in \mathcal{X}, \hat{f}(\boldsymbol{x}) \geq 0\}$ can be constructed through deep neural network classifiers $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ trained using either 0-1 loss or surrogate loss. Accordingly, a natural teacher network assumption is that $p(\boldsymbol{x}) - q(\boldsymbol{x})$ can be expressed by some neural network $f_n^* \in \mathcal{F}_n^*$. Here, the underlying densities are indexed by n, but such an assumption is not uncommon in high-dimensional statistics, where population quantities may depend on the sample size n, e.g., [82]. This setting is closely related to the classical smooth boundary assumption considered in the Section 2.1. The teacher-student network setting is more general as it does not impose any special structures on the decision boundary. Moreover, by the universal approximation

property [8], [10], [83], the teacher network can sufficiently approximate any continuous function given large enough size.

In the proposed teacher-student setting, an un-improvable rate of convergence is derived as $\tilde{O}(n^{-2/3})$ for the excess risk of the empirical 0-1 loss minimizer, given that the student network is deeper and larger than the teacher network (unless the teacher network has a limited capacity in some sense to be specified later). When data are separable, the rate improves to $\tilde{O}(n^{-1})$. Furthermore, we extend our analysis to a specific surrogate loss, i.e., hinge loss, and show that the convergence rate remains the same (up to higher order logarithmic terms) while allowing *deeper* student and teacher nets. The obtained sharp risk bounds may explain the empirical success of deep neural networks in high-dimensional classification as the data dimension d only appears in the $\log(n)$ terms. Our main technical novelty is the nontrivial entropy calculation for nonparametric set estimation based on combinatorial analysis of ReLU neural networks.

2.2.1 Training with 0-1 Loss

For the theoretical purpose, we first focus on DNN classifiers trained with the empirical 0-1 loss. Denote

$$\hat{f}_n = \operatorname*{argmin}_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{\mathrm{i}=1}^{n} \mathbb{I}\{y_{\mathrm{i}} f(\boldsymbol{x}_{\mathrm{i}}) < 0\},$$

given a certain DNN family $\mathcal{F}_n$.

It is important to control the complexity of the underlying classification problem. Otherwise, the student network would not be able to recover the Bayes classifier [84] with sufficient accuracy. To this end, we impose the following teacher network assumptions on $(p(\boldsymbol{x}) - q(\boldsymbol{x}))$:

(A1) p, q have compact supports.

(A2) $p(\boldsymbol{x}) - q(\boldsymbol{x})$ is representable by some teacher ReLU DNN $f_n^* \in \mathcal{F}_n^*$ with

$$S_n^* = O\left(n^{\alpha}\right), \quad L_n^* = \mathrm{poly}(\log n), \quad N_n^*, B_n^* = \mathrm{poly}(n),$$

for some constant $0 < \alpha < 1$.

(A3) For any $n \in \mathbb{N}$, there exists $c_n, 1/T_n = \text{poly}(\log n)$, such that for all $0 \leq t \leq T_n$,

$$\mathbb{Q}\{\boldsymbol{x} \in \mathcal{X} : |f_n^*(\boldsymbol{x})| \leq t\} \leq c_n t$$

Assumption (A3) characterizes how concentrated the data are around the decision boundary, which can be seen as an extension to the classical Tsybakov noise condition [1]. The difference is that in our case, the underlying densities are indexed by sample size and thus c_n and T_n are allowed to vary with n. Assumption (A3) is not unrealistic as we will show that it holds with high probability if the teacher network is random as stated in the following Theorem.

Theorem 2.2.1 *Let f_n^* be the teacher network with structures specified in assumption (A2). Suppose that all weights of f_n^* are i.i.d. with any continuous distribution, e.g. Gaussian, truncated Gaussian, etc.. Then, with probability at least $1-\delta$, assumption (A3) holds with $c_n, 1/T_n \leq A(\delta)(\log n)^{m^* d^2 L_n^{*2}}$ where $A(\delta)$ is some constant depending on δ.*

The following theorem characterizes how well the student network of proper size can learn from the teacher in terms of the excess risk.

Theorem 2.2.2 *Under the teacher assumptions (A1) through (A3), denote all such (p, q) pairs to be $\mathcal{P}_n^*$ and let the corresponding Bayes classifier be C_n^*. Let $\mathcal{F}_n$ be a student ReLU DNN family with $N_n = O[(\log n)^m]$ and $L_n = O(1)$ for some $m \geq m_*$ and assume the student network is larger than the teacher network in the sense that $L_n \geq L_n^*, S_n \geq S_n^*, N_n \geq N_n^*, B_n \geq B_n^*$. Then the excess risk for $\hat{f}_n \in \mathcal{F}_n$ satisfies*

$$\sup_{(p,q)\in\mathcal{P}_n^*} \mathbb{E}[\mathcal{E}(\hat{f}_n, C_n^*)] = \tilde{O}_d\left(n^{-\frac{2}{3}}\right),$$

where $\tilde{O}_d$ hides the $\log n$ terms, which depends on d.

The dependence on the dimension d is in the order of $O[(\log n)^{d^2}]$. The reason for the dimension dependence is rooted in the teacher network assumption. The complexity of the classification problem, measured by how complicated are the sets created by the teacher, $\{\boldsymbol{x} \in \Omega : f(\boldsymbol{x}) \geq 0, f^* \in \mathcal{F}^*\}$, grows with dimension in a exponential fashion. If we change

the teacher assumption that $f^*(\boldsymbol{x}) = h^*(x_1, \cdots, x_{d-1}) - x_d$ where h^* is a neural network, then the dependence on the dimension can be overcome.

We further argue that under the present setting, the rate $n^{-2/3}$ in Theorem 2.2.2 cannot be further improved.

Theorem 2.2.3 *Under the same assumptions of p, q as in Theorem 2.2.2 that $(p, q) \in \tilde{\mathcal{F}}_n^*$. Let $\tilde{\mathcal{F}}_n$ be an arbitrary function space, then*

$$\inf_{\tilde{f}_n \in \tilde{\mathcal{F}}_n} \sup_{(p,q) \in \tilde{\mathcal{F}}_n^*} \mathbb{E}[\mathcal{E}(f_n, f_n^*)] = \Omega_d\left(n^{-\frac{2}{3}}\right),$$

where Ω_d hides the dependence on d.

Theorem 2.2.3 shows that the convergence rate achieved by the empirical 0-1 loss minimizer cannot be further improved (up to a logarithmic term). If p and q have disjoint supports, i.e. separable, which could be true in some image data, the rate improves to n^{-1}, as stated in the following corollary. This rate improvement is not surprising since the classification task becomes much easier for separable data.

Corollary 2.2.3.1 *Under the same setting as in Theorem 2.2.2, if we further assume p, q have disjoint supports, then the rate of convergence of the empirical 0-1 loss minimizer improves to*

$$\inf_{f_n \in \mathcal{F}_n} \sup_{(p,q) \in \tilde{\mathcal{F}}_n^*} \mathbb{E}[\mathcal{E}(f_n, f_n^*)] \asymp \tilde{O}\left(\frac{1}{n}\right).$$

Remark 2.2.4 (Disjoint Support) *Given that data are separable, [85] derived the excess risk bound as $O(D \log n/n)$ (under a smooth loss) where D is the VC-subgraph-dimension of the estimation family. Additionally, separability implies that the noise exponent κ in Tsybakov noise condition [1], [86] can be arbitrarily large, which also gives $O(1/n)$ rate under the "boundary fragments" assumption.*

Remark 2.2.5 (Connections to the Classical Setting) *The optimal risk bound under the smooth boundary fragments assumption [1] is $O(n^{-\beta(\kappa+1)/[\beta(\kappa+2)+(d-1)\kappa]})$. Interestingly, this rate coincides with our rate when $\kappa = 1$ and $\beta \to \infty$ (up to a logarithmic factor). If we further allow $\kappa \to \infty$ (corresponding to separable data), the classical rate above recovers $\tilde{O}(n^{-1})$ (up to a logarithmic factor).*

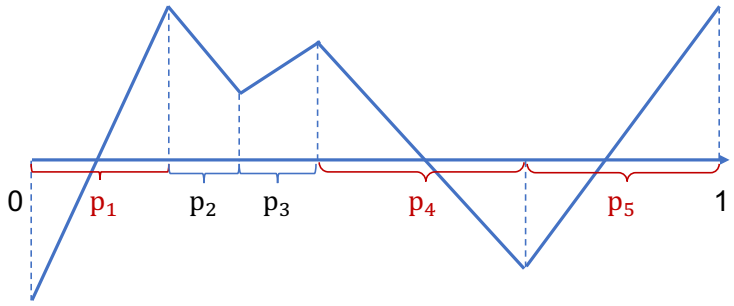


Figure 2.4. Example of a ReLU DNN function in $[0,1]$. There are 5 pieces $p_1, p_2, \ldots, p_5$ and among them, only p_1, p_4, p_5 cross value 0 (horizontal line). There are 3 active pieces in this example and they are colored red.

The imposed relation between the teacher and student nets in Theorem 2.2.2 is referred to as "over-realization" in [76], [79], [87]: at each layer, the number of student nodes is larger than that of teacher nodes given the same depth. In other words, the student network is larger than the teacher in order to obtain zero approximation error. On the other hand, such a requirement is not necessary as long as the corresponding Bayes classifier is not too complicated. A ReLU neural network is a continuous piecewise linear function, i.e. its domain can be divided into connected regions (pieces) within where the function is linear. If the ReLU neural network crosses 0 on one piece, we call that piece as being *active* (see Figure 2.4 for an illustration). One way to measure the complexity of the teacher network is the number of active pieces. The following Corollary says that the teacher network can be much larger and deeper as long as the number of active pieces are in a logarithmic order with respect to n.

Corollary 2.2.5.1 *The same result in Theorem 2.2.2 holds when the teacher network is larger than the student network, i.e.* $L_n \leq L_n^*, S_n \leq S_n^*, N_n \leq N_n^*, B_n \leq B_n^*$ *, given that the total number of active pieces in the teacher network is of the following order*

$$o\left(\left(\prod_{l=1}^{L_n-1}\left\lfloor\frac{n_l}{d}\right\rfloor^d\right)\sum_{\mathrm{j}=0}^{d}\binom{n_{L_n}}{\mathrm{j}}\right), \tag{2.6}$$

where $n_1, \cdots, n_{L_n}$ *are the width of each hidden layer of the student network.*

The number of active pieces is the key quantity in controlling the complexity of the optimal set G^*. The expression in (2.6) comes from the lower bound developed by [88] on the maximum number of linear pieces for a ReLU neural network (Lemma 2.4.13). This lower bound is determined by the structure of the student network. If the number of active pieces of the teacher network is on this order, i.e. within the capacity of the student, then the corresponding optimal set can still be recovered by an even smaller student network, which ensures zero approximation error. Since the student network in consideration satisfies $N_n = O[(\log n)^m]$, the required order for the number of active pieces is in the order of $o[(\log n)^{mdL_n}]$.

2.2.2 Training with Surrogate Loss

In this section, we consider deep classifiers trained under the hinge loss $\phi(z) = (1-z)_+ = \max\{1-z, 0\}$. This kind of surrogate loss has been widely used for "maximum-margin" classification, most notably for support vector machines [89]. An desirable property of hinge loss is that its optimal classifier coincides with that under 0-1 loss [90], i.e. $f_\phi^*(\boldsymbol{x}) = C^*(\boldsymbol{x})$. Hence, a lot of arguments for 0-1 loss can be easily carried over. Additionally, minimizing the sample average of an appropriately behaved loss function has a regularizing effect [56]. It is thus possible to obtain uniform upper bounds on the risk of a function that minimizes the empirical average of the loss ϕ, even for rich classes that no such upper bounds are possible for the minimizer of the empirical average of the 0–1 loss.

Under the surrogate loss, our requirement on the size of the teacher network is relaxed from (A2) as follows:

(A2$_\phi$) $p(\boldsymbol{x}) - q(\boldsymbol{x})$ is representable by some teacher ReLU DNN $f_n^* \in \mathcal{F}_n^*$ with

$$N_n^* = O[(\log n)^{m_*}], \quad L_n^* = O(\log n), \quad B_n^*, F_n^* = O(\sqrt{n})$$

for some $m_* \geq 1$.

The following theorem says that the same un-improvable rate can be obtained for the empirical hinge loss minimizer $\hat{f}_{\phi,n} \in \mathcal{F}_n$.

Theorem 2.2.6 *Suppose the underlying densities p and q satisfy assumptions (A1), ($A2_\phi$), (A3) and denote all such (p, q) pairs as $\tilde{\mathcal{F}}_n^*$. Let $\mathcal{F}_n$ be a student ReLU DNN family with $L_n = O(\log n), N_n = O[(\log n)^m]$ and $B_n, F_n = O(\log n)$ for some $m \geq m_*$. Assume the student network is larger than the teacher network, i.e., $L_n \geq L_n^*, S_n \geq S_n^*, N_n \geq N_n^*, B_n \geq B_n^*, F_n \geq F_n^*$. Then the excess risk for $\hat{f}_{\phi,n} \in \mathcal{F}_n$ satisfies*

$$\sup_{(p,q)\in\tilde{\mathcal{F}}_n^*} \mathbb{E}[\mathcal{E}(\hat{f}_{\phi,n}, C_n^*)] \asymp \tilde{O}_d\left(n^{-\frac{2}{3}}\right)$$

Similarly, results in Corollary 2.2.3.1 and 2.2.5.1 hold for the empirical hinge loss minimizer. Specifically, when p, q are disjoint, the convergence rate of excess risk improves to n^{-1}, and all conclusions hold when the teacher network is larger but with bounded active pieces.

Remark 2.2.7 (Network Depth) *Training with surrogate loss such as hinge loss, unlike 0-1 loss, doesn't involve any hard thresholding, i.e. $\mathbb{I}\{yf(\boldsymbol{x}) < 0\}$. As a result, to control the complexity of the student network, Lemma 2.4.15 is used instead of Lemma 2.4.14, which allows us to use deeper neural networks ($L_n = O(\log n)$) for both the student and teacher network.*

Statistical optimality is established in the proposed teacher-student classification setting. As long as the teacher network is not too large, the convergence rate is dimension-free, which may provide one theoretical explanation for the empirical successes of deep neural networks in high dimensional classification, particularly for structured data. For image data, one consensus researchers have is that high-resolution images are not actually high-dimensional data. The pixels close to each other tend to be highly correlated. Such local connectivity greatly reduces the actual dimension of images. However, there is no consensus on which is the most appropriate low-dimensional assumption for images. Considering the general teacher-student network setting provides great flexibility. To illustrate, by considering a CNN as the teacher, it automatically assumes the local connectivity of pixels and accounts for their spatial correlations. CNN is also a type of DNN with convolutional sparse structures and our theorems apply to CNNs as well.

2.3 Discussion

In this section, we obtain optimal rates of convergence for DNN classifiers in both the smooth boundary fragment and teacher-student setting. Through our localized analysis, we are able to improve the existing convergence rate to optimal and prove that DNN classifiers can benefit from the compositional structure of the data and adapt to its effective dimension d^*. The dimension dependence is further removed in our teacher-student classification setting where student network can achieve dimension-agnostic rate of $\tilde{O}(n^{-2/3})$.

The results under the smooth boundary setting can be further improved if we can relax the structural requirement on the DNN classifier or the local margin condition itself. In the teacher-student setting, the results for training under 0-1 loss only hold for student networks with $O(1)$ layers and the assumption that $f_n^* \in \mathcal{F}_n$, i.e. zero approximation, is required. In the future, we aim to relax these two constraints and provide more comprehensive analysis of the teacher-student network. Additionally, we would like to explore other type of neural networks such as convolutional neural network and residual neural network, which are both very successful at image classification. Another direction is to consider the more general improper learning scenario where the Bayes classifier is not necessarily in the student neural network. Further investigation under the teacher-student network setting may facilitate a better understanding of how deep neural network works and shed light on its empirical success especially in high-dimensional image classification.

2.4 Technical Proofs

Since we are estimating the optimal decision boundary via deep ReLU neural network, the proof can be broken down to two parts. The first part is to develop efficient approximation of the piecewise constant Bayes classifier and the second part is to control the stochastic error from empirical estimation.

2.4.1 Proof of Lemmas in Section 2.1

To address the approximation error, let's consider a more general setting than the defined $\mathcal{C}(d^*, \beta^{**})$. Let $\mathcal{H}$ be some smooth function space from $\mathbb{R}^{d-1} \to \mathbb{R}$. For $h \in \mathcal{H}$, we define the horizon function to be

$$\Psi_{h,\mathrm{i}} := \mathbb{I}\{h(\boldsymbol{x}_{-\mathrm{i}}) \geq x_{\mathrm{i}}\}.$$

Each horizon function is a $\{0,1\}$-function defined via a smooth function h. We further define the corresponding support to be

$$I_{h,\mathrm{i}} = \{\boldsymbol{x} \in \mathcal{X} : \Psi_{h,\mathrm{i}}(\boldsymbol{x}) = 1\}.$$

Intersection of K such support defines all the pieces A that

$$\mathcal{A}(\mathcal{H}, K) = \{A \subset \mathcal{X} : A = \bigcap_{k=1}^{K} I_{h_k,\mathrm{j}_k}, h_k \in \mathcal{H}, \mathrm{j}_k = 1, \cdots, d\}. \tag{2.7}$$

Let $\mathcal{C}(\mathcal{H}, K, T)$ be the set of classifiers of the form

$$C(\boldsymbol{x}) = 2\sum_{t=1}^{T} \mathbb{I}_{A_t}(\boldsymbol{x}) - 1,$$

where $A_1, \cdots, A_T \in \mathcal{A}(\mathcal{H}, K)$ are disjoint. Then $\mathcal{C}(\mathcal{H}, K, T)$ defines a family of classifiers with smooth boundaries and the smoothness is determined by $\mathcal{H}$. $\mathcal{C}(d^*, \beta^{**})$ is a special case of $\mathcal{C}(\mathcal{H}, K, T)$ with $\mathcal{H} = \mathcal{H}(d^*, \beta^{**})$, $K = T = 1$.

Before the proof, we present some lemmas.

Lemma 2.4.1 *[Approximation Part of Theorem 1 in [2]] Consider the d-variate nonparametric regression model for composite regression function f_0 in the class $\mathcal{H}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, R)$. There exists $\tilde{f}_n$ in the network class $\mathcal{F}_n^{\mathrm{DNN}}(L_n, N_n, S_n, B_n, F_n)$ with $L_n \lesssim \log_2 n$, $B_n = 1$, $F_n \geq \max(R, 1)$,*

$$N_n \lesssim \max_{\mathrm{i}=0,\cdots,q} n^{\frac{t_{\mathrm{i}}}{2\beta_{\mathrm{i}}^* + t_{\mathrm{i}}}}, \quad S_n \lesssim \max_{\mathrm{i}=0,\cdots,q} n^{\frac{t_{\mathrm{i}}}{2\beta_{\mathrm{i}}^* + t_{\mathrm{i}}}} \log n,$$

such that

$$\|\hat{f}_n - f_0\|_\infty^2 \lesssim \phi_n.$$

Lemma 2.4.2 *[Lemma A.2 in [91]] Let $1 < d \in \mathbb{N}$ and $H(\boldsymbol{x}) := \mathbb{I}_{[0,\infty)\times\mathbb{R}^{d-1}}(\boldsymbol{x})$. For every $\epsilon > 0$ there exists a neural network f_H^{DNN} with 2 layers and 5 nonzero weights (only taking values from $\{-1, 1, 1/\epsilon\}$), such that $0 \le f_H^{\mathrm{DNN}}(\boldsymbol{x}) \le 1$ and*

$$|H(\boldsymbol{x}) - f_H^{\mathrm{DNN}}(\boldsymbol{x})| \le \mathbb{I}_{[0,\epsilon)\times\mathbb{R}^{d-1}}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

Moreover,

$$\|H(\boldsymbol{x}) - f_H^{\mathrm{DNN}}(\boldsymbol{x})\|_{L^p([-1/2,1/2]^d)} \le \epsilon^{1/p}.$$

Lemma 2.4.3 (Lemma A.4 in [91]) *Let $d, \ell \in \mathbb{N}$ be arbitrary. Then, there are constants $s = s(d) \in \mathbb{N}$, $c = c(d,\ell) \in \mathbb{N}$, and $L = L(d,\ell) \in \mathbb{N}$ such that $L \le (1 + \lceil \log_2 d \rceil) \cdot (10 + \frac{\ell}{d})$ with the following property: For any $\varepsilon \in (0, \frac{1}{2})$, there is a ReLU neural network $f_\varepsilon^{\mathrm{DNN}}$ with d-dimensional input and one-dimensional output, with at most L layers, and with at most $c \cdot \varepsilon^{-d/\ell}$ nonzero, (s,ε)-quantized weights, and such that $f_\varepsilon^{\mathrm{DNN}}$ satisfies*

$$|f_\varepsilon^{\mathrm{DNN}}(\boldsymbol{x}) - \prod_{\mathrm{i}=1}^{d} x_{\mathrm{i}}| \le \varepsilon \quad \text{for all } \boldsymbol{x} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^d. \tag{2.8}$$

The following lemma quantifies the approximation of indicator of a single basis.

Lemma 2.4.4 *Assume $h \in \mathcal{G}(d^*, \beta^{**})$ with the compositional structure. For any $\epsilon > 0$ and $\mathrm{i} = 1, \cdots, d$, there exists a neural network $f_{\Psi_{h,\mathrm{i}}}^{\mathrm{DNN}}$ with $2 + \log n$ layers and number of nonzero weights $s \lesssim \epsilon^{-d^*/\beta^{**}} \log n$ such that*

$$\|\Psi_{h,\mathrm{i}} - f_{\Psi_{h,\mathrm{i}}}^{\mathrm{DNN}}\|_p^p \le 2\epsilon.$$

Proof Without loss of generality, consider $\mathrm{i} = 1$ and let the target function be $H \circ \tilde{h}$ where $H(\boldsymbol{x}) = \mathbb{I}_{[0,\infty)\times\mathbb{R}^{d-1}}$ is the Heaviside function and

$$\tilde{h}(\boldsymbol{x}) = (h(x_2, \cdots, x_d) - x_1, x_2, \cdots, x_d).$$

By lemma 2.4.1, for any ϵ_1, there exist a neural network with at most $\log n$ layers and $O(\epsilon_1^{-d^*/\beta^{**}} \log n)$ non-zero weights that

$$\|h - f_h^{\text{DNN}}\|_\infty \leq \epsilon_1.$$

By lemma 2.4.2, for any ϵ_2, there exist a neural network with 2 layers and 5 nonzero weights such that

$$|H(\boldsymbol{x}) - f_H^{\text{DNN}}(\boldsymbol{x})| \leq \mathbb{I}_{[0,\epsilon_2]\times\mathbb{R}^{d-1}}(\boldsymbol{x}).$$

Construct the neural network estimator to be $f_H^{\text{DNN}} \circ f_{\tilde{h}}^{\text{DNN}}$. Then,

$$\begin{aligned}
&\|H \circ \tilde{h} - f_H^{\text{DNN}} \circ f_{\tilde{h}}^{\text{DNN}}\|_p \\
&\leq \|H \circ \tilde{h} - H \circ f_{\tilde{h}}^{\text{DNN}}\|_p + \|H \circ f_{\tilde{h}}^{\text{DNN}} - f_H^{\text{DNN}} \circ f_{\tilde{h}}^{\text{DNN}}\|_p.
\end{aligned}$$

For the first term, note that the difference is 1 only under two cases, one being $h(x_2, \cdots, x_d) - x_1 \geq 0$ and $f_h^{\text{DNN}}(x_2, \cdots, x_d) - x_1 < 0$ and the other one being $h(x_2, \cdots, x_d) - x_1 < 0$ and $f_h^{\text{DNN}}(x_2, \cdots, x_d) - x_1 \geq 0$. Combining both cases, we have

$$h(x_2, \cdots, x_d) \wedge f_h^{\text{DNN}}(x_2, \cdots, x_d) < x_1 \leq h(x_2, \cdots, x_d) \vee f_h^{\text{DNN}}(x_2, \cdots, x_d).$$

Thus

$$\begin{aligned}
\|H \circ \tilde{h} - H \circ f_{\tilde{h}}^{\text{DNN}}\|_p^p &\leq \int \left| h(x_2, \cdots, x_d) - f_h^{\text{DNN}}(x_2, \cdots, x_d) \right| d\boldsymbol{x}_{-1} \\
&\leq \|h(\boldsymbol{x}) - f_h^{\text{DNN}}(\boldsymbol{x})\|_1 \\
&\leq \epsilon_1.
\end{aligned}$$

For the second term,

$$\begin{aligned}
\|H \circ f_{\tilde{h}}^{\text{DNN}} - f_H^{\text{DNN}} \circ f_{\tilde{h}}^{\text{DNN}}\|_p^p &\leq \int \mathbb{I}_{[0,\epsilon_2]\times\mathbb{R}^{d-1}} \left(f_{\tilde{h}}^{\text{DNN}}(\boldsymbol{x}) \right) dx \\
&\leq \int \int \mathbb{I}_{\{0 \leq x_1 + f_h^{\text{DNN}}(x_2, \cdots, x_d) \leq \epsilon_2\}}(x_1) dx_1 d\boldsymbol{x}_{-1} \\
&\leq \epsilon_2.
\end{aligned}$$

By choosing $\epsilon_1 = \epsilon_2 = \epsilon$ yields this lemma. ■

Proof of Lemma 2.1.1

Proof We first consider approximation of the indicator function of a single piece A_1, which is the product of K basis indicator denoted by $\Psi_1, \cdots, \Psi_K$. That is

$$\mathbb{I}_{A_1} = \prod_{k=1}^{K} \Psi_k.$$

By lemma 2.4.4, for any $\epsilon_3 > 0$ there exist neural networks $f_{\Psi_\mathrm{i}}^{\mathrm{DNN}}$ with $O(\log n)$ layers and $\epsilon_3^{-d^*p/\beta^{**}}$ such that

$$\|\Psi_\mathrm{i} - f_{\Psi_\mathrm{i}}^{\mathrm{DNN}}\|_p \leq \epsilon_3, \quad \mathrm{i} = 1, 2, \cdots, K.$$

By lemma 2.4.3, for any $\epsilon_4 > 0$, we can construct neural network $f_{\prod}^{\mathrm{DNN}}$ with at most $(5+\log_2(K^2/\epsilon_4))\lceil\log_2(K)\rceil$ layers and $36K^2(5+\log_2(K^2/\epsilon_4))\lceil\log_2(K)\rceil$ nonzero weights, such that

$$\|\prod_{k=1}^{K} x_k - f_{\prod}^{\mathrm{DNN}}(x_1, \cdots, x_K)\|_\infty \leq \epsilon_4.$$

Construct our neural network function to be $f_{\prod}^{\mathrm{DNN}}(f_{\Psi_1,\cdots,f_{\Psi_K}^{\mathrm{DNN}}}^{\mathrm{DNN}})$, which has layers at most

$$2 + \log n + (5 + \log_2(K^2/\epsilon_4))\lceil\log_2(K)\rceil + 1 \lesssim \log n + \log_2 K\,(\log_2 K + \log_2(1/\epsilon_4))$$

and non-zero weights at most

$$\begin{aligned} &CKT\epsilon_3^{-d^*p/\beta^{**}} \log n + 36K^2T(5 + \log_2(K^2/\epsilon_4))\lceil\log_2(K)\rceil \\ \lesssim &KT\epsilon_3^{-d^*p/\beta^{**}} \log n + K^2T \log_2 K(\log K + \log_2(1/\epsilon_4)). \end{aligned}$$

Thus,

$$\begin{aligned} &\|\prod_{k=1}^{K} \Psi_k - f_{\prod}^{\mathrm{DNN}}(f_{\Psi_1}^{\mathrm{DNN}}, \cdots, f_{\Psi_K}^{\mathrm{DNN}})\|_p \\ \leq &\|\prod_{k=1}^{K} \Psi_k - \prod_{k=1}^{K} f_{\Psi_k}^{\mathrm{DNN}}\|_p + \|\prod_{k=1}^{K} f_{\Psi_k}^{\mathrm{DNN}} - f_{\prod}^{\mathrm{DNN}}(f_{\Psi_1}^{\mathrm{DNN}}, \cdots, f_{\Psi_K}^{\mathrm{DNN}})\|_p. \end{aligned}$$

For the first term, since all Ψ_k and $f^{\text{DNN}}_{\Psi_k}$ are functions between 0 and 1,

$$\begin{aligned}
&\|\prod_{k=1}^{K}\Psi_k - \prod_{k=1}^{K} f^{\text{DNN}}_{\Psi_k}\|_p \\
&\leq \|\prod_{k=1}^{K}\Psi_k - f^{\text{DNN}}_{\Psi_1}\prod_{k=2}^{K}\Psi_k\|_p + \|f^{\text{DNN}}_{\Psi_1}\prod_{k=2}^{K}\Psi_k - \prod_{k=1}^{K} f^{\text{DNN}}_{\Psi_k}\|_p \\
&\leq \|(\Psi_1 - f^{\text{DNN}}_{\Psi_1})\prod_{k=2}^{K}\Psi_k\|_p + \|f^{\text{DNN}}_{\Psi_1}(\prod_{k=2}^{K}\Psi_k - \prod_{k=2}^{K} f^{\text{DNN}}_{\Psi_k})\|_p \\
&\leq \|\Psi_1 - f^{\text{DNN}}_{\Psi_1}\|_p + \|\prod_{k=2}^{K}\Psi_k - \prod_{k=2}^{K} f^{\text{DNN}}_{\Psi_k}\|_p \\
&\leq \cdots \leq \sum_{k=1}^{K}\|\Psi_k - f^{\text{DNN}}_{\Psi_k}\|_p \leq K\epsilon_3.
\end{aligned}$$

For the second term, since $0 \leq f^{\text{DNN}}_{\Psi_k}(\boldsymbol{x}) \leq 1$ for all $k = 1, \cdots, K$,

$$\begin{aligned}
&\|\prod_{k=1}^{K} f^{\text{DNN}}_{\Psi_k} - f^{\text{DNN}}_{\prod}(f^{\text{DNN}}_{\Psi_1}, \cdots, f^{\text{DNN}}_{\Psi_K})\|_p \\
\leq &\|\prod_{k=1}^{K} f^{\text{DNN}}_{\Psi_k} - f^{\text{DNN}}_{\prod}(f^{\text{DNN}}_{\Psi_1}, \cdots, f^{\text{DNN}}_{\Psi_K})\|_\infty \leq \epsilon_4.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\sum_{t=1}^{T}\mathbb{I}_{A_t} - \sum_{t=1}^{T} f^{\text{DNN}}_{\mathbb{I}_{A_t}}\|_p &\leq \sum_{t=1}^{T}\|\mathbb{I}_{A_t} - f^{\text{DNN}}_{\mathbb{I}_{A_t}}\|_p \\
&\leq T(K\epsilon_3 + \epsilon_4).
\end{aligned}$$

Choosing $\epsilon_3 = \epsilon_4 = \epsilon$ yields the lemma. ■

Proof of Lemma 2.1.2

Proof By definition, we can write

$$\begin{aligned}
\mathbb{Q}(\boldsymbol{x} : |p(\boldsymbol{x}) - q(\boldsymbol{x})| \leq t) &= \int_{\boldsymbol{x}_{-d}} \int_{x_d} \mathbb{I}\{|p(\boldsymbol{x}) - q(\boldsymbol{x})| \leq t\} dx_d d\boldsymbol{x}_{-d} \\
&= \int_{\boldsymbol{x}_{-d}} \int_{x_d} \mathbb{I}\{m_{\boldsymbol{x}_{-d}}(u) \leq t\} du d\boldsymbol{x}_{-d} \\
&\leq \int_{\boldsymbol{x}_{-d}} \int_{x_d} \mathbb{I}\{\frac{u^{1/K(\boldsymbol{x})}}{C_{\epsilon_0}} \leq t\} du d\boldsymbol{x}_{-d} \\
&\leq \int_{\boldsymbol{x}_{-d}} (C_{\epsilon_0} t)^{K(\boldsymbol{x})} d\boldsymbol{x}_{-d} \\
&\leq C t^{\kappa},
\end{aligned}$$

where C is a constant depending on C_{ϵ_0}, c, d. ■

2.4.2 Proof of Theorem 2.1.3

The lower bound result comes from estimation of sets in the discriminative analysis [1] where two sets of independent samples $\mathcal{X}^+ = \{\boldsymbol{x}_1^+, \cdots, \boldsymbol{x}_n^+\}$ and $\boldsymbol{x}^- = \{\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_m^-\}$ with unknown densities p or q respectively (w.r.t. a σ-finite measure Q) are given. The goal is to predict whether a new sample $\boldsymbol{x}$ is coming from f or g with a discrimination decision rule defined by a set $G \subset \mathbb{R}^d$ that we attribute $\boldsymbol{x}$ to p if $\boldsymbol{x} \in G$ and to q otherwise. Let the Bayes risk to be

$$R(G) = \frac{1}{2}\left(\int_{G^c} p(\boldsymbol{x}) Q(d\boldsymbol{x}) + \int_G q(\boldsymbol{x}) Q(d\boldsymbol{x})\right).$$

Denote $G^* = \{\boldsymbol{x} : p(\boldsymbol{x}) \geq q(\boldsymbol{x})\}$ to be the Bayes risk minimizer. Let $\tilde{G}_{m,n}$ be an empirical rule based on observations. The excess risk can be expressed as $R(\tilde{G}_{m,n}) - R(G^*) = \frac{1}{2} d_{p,q}(\tilde{G}_{m,n}, G^*)$. In the following, we establish how fast can the excess risk go to zero under the smooth boundary fragment setting with compositional smoothness assumption.

For positive constants c_1, c_2, η_0, κ and for a σ-finite measure Q, consider densities p, q on $\mathbb{R}^d$ w.r.t. Q and define class $\mathcal{F}$ of paired densities to be

$$\mathcal{F}_{\mathcal{G}} = \{(p,q) : Q\{\boldsymbol{x} \in \mathcal{X} : |p(\boldsymbol{x}) - q(\boldsymbol{x})| \leq \eta\} \leq c_2 \eta^{\kappa} \text{ for } 0 \leq \eta \leq \eta_0,$$
$$\{\boldsymbol{x} \in \mathcal{X} : p(\boldsymbol{x}) \geq q(\boldsymbol{x})\} \in \mathcal{G}, p(\boldsymbol{x}), q(\boldsymbol{x}) \leq c_1 \text{ for } x \in \mathcal{X}\}.$$

Now let the base measure Q be the Lebesgue measure $\mathbb{Q}$ and recall $d_{\triangle}(G_1, G_2) = \mathbb{Q}(G_1 \triangle G_2)$. The following lemma establishes the connection between $d_{\triangle}$ and $d_{p,q}$.

Lemma 2.4.5 (Lemma 2 in [1]) *There exists a constant $c(\kappa)$ depending on κ such that for Lebesgue measurable subsets G_1 and G_2 of $\mathcal{X}$ and for $(p,q) \in \mathcal{F}_{\mathcal{G}}$,*

$$c(\kappa) d_{\triangle}^{(1+\kappa)/\kappa}(G_1, G_2) \leq d_{p,q}(G_1, G_2) \leq 2c_1 d_{\triangle}(G_1, G_2).$$

Lemma 2.4.6 *Let $\mathcal{X} = [0,1]^d$ and Q be the Lebesgue measure on $\mathcal{X}$. Consider*

$$\mathcal{G}_h = \{(x_1, \cdots, x_d) \in \mathcal{X} : 0 \leq x_d \leq h(x_1, \cdots, x_{d-1}), h \in \mathcal{H}(d^*, \beta^{**})\}$$

and $\mathcal{F}_{\mathcal{G}}$ with $\mathcal{G} = \mathcal{G}_h$. Then

$$\liminf_{n\to\infty} \inf_{\hat{G}_{m,n}} \sup_{(p,q)\in\mathcal{F}_{\mathcal{G}_h}} (n \wedge m)^{\frac{\beta^{**}(\kappa+1)}{\beta^{**}(\kappa+2)+d^*\kappa}} \mathbb{E}_{p,q}[d_{p,q}(\tilde{G}_{m,n}, G^*)] > 0.$$

Proof of Theorem 2.1.3

Proof Without loss of generality, assume $n \leq m$ so we mainly focus on $\mathcal{X}^+$. Consider the subset of $\mathcal{F}_{\mathcal{G}_h}$ that contains all pairs (p, q_0), where q_0 is fixed and f belongs to a finite class of densities $\mathcal{F}_1$ that will be defined later. Then,

$$\begin{aligned}\sup_{(p,q)\in\mathcal{F}_{\mathcal{G}_h}} \mathbb{E}_{p,q} d_{\triangle}(\tilde{G}_{m,n}, G^*) &\geq \sup_{(p,q_0): f\in\mathcal{F}_1} \mathbb{E}_{p,q} d_{\triangle}(\tilde{G}_{m,n}, G^*) \\ &\geq \mathbb{E}_{q_0}\left[\frac{1}{|\mathcal{F}_1|}\sum_{f\in\mathcal{F}_1} \mathbb{E}_p[d_{\triangle}(\tilde{G}_{m,n}, G^*)|y_1, \cdots, y_m]\right],\end{aligned}$$

where $\mathbb{E}_p$ and $\mathbb{E}_{q_0}$ denotes the expectations w.r.t. the distributions of $(x_1, \cdots, x_n)$ and $(y_1, \cdots, y_m)$ when the underlying densities are p and q_0.

Recall the compositional assumption (2.1) and let

$$\mathrm{i}^* \in \underset{\mathrm{i}=0,1,\cdots,q}{\operatorname{argmax}} n^{-\frac{2\beta_{\mathrm{i}}^*}{2\beta_{\mathrm{i}}^* + t_{\mathrm{i}}}} \quad \text{and} \quad \beta^* = \beta_{\mathrm{i}^*}.$$

Further denote $B = \prod_{l=\mathrm{i}^*+1}^{q} (\beta_l \wedge 1)$ and then $\beta^{**} = \beta^* B$. For simplicity, we give the proof for the case $d^* = t_{\mathrm{i}^*} = 1$, that is the effective dimension of the smooth boundaries is 1 instead of $d-1$. For this case, let $\phi \in \mathcal{C}_1^{\beta^*}(\mathbb{R}, 1)$ be a real-valued function supported on $[-1, 1]$ with $\phi(t) \geq 0$ for all t, $\max \phi(t) = 1$ and $\phi(0) = 1$. For $x = (x_1, \cdots, x_d) \in [0,1]^d$, define

$$\begin{aligned} q_0(\boldsymbol{x}) =& (1 - \eta_0 - b_1)\mathbb{I}_{\{0 < x_2 < \frac{1}{2}\}} + \mathbb{I}_{\{\frac{1}{2} \leq x_2 < \frac{1}{2} + (\tau M^{-\beta^*})^B\}} \\ &+ (1 + \eta_0 + b_2)\mathbb{I}_{\{\frac{1}{2} + (\tau M^{-\beta^*})^B \leq x_2 \leq 1\}} \end{aligned}$$

where $M \geq 2$ is an integer to be specified later and $\tau \in (0,1)$ is a constant. $b_1 = (\tau M^{-\beta^*}/c_2)^{B/\kappa}$ and $b_2 > 0$ is chosen such that q_0 integrates to 1. For $\mathrm{j} = 1, 2, \cdots, M$ and $t \in [0,1]$, let

$$\psi_{\mathrm{j}}(t) = \tau M^{-\beta^*} \phi\left(M\left[t - \frac{\mathrm{j}-1}{M}\right]\right).$$

Note that ψ_{j} is only supported on $[\frac{\mathrm{j}-1}{M}, \frac{\mathrm{j}}{M}]$. For vectors $\omega = (\omega_1, \cdots, \omega_M)$ with elements $w_{\mathrm{j}} \in \{0,1\}$, define

$$b_\omega(t) = \sum_{\mathrm{j}=1}^{M} \omega_{\mathrm{j}} \psi_{\mathrm{j}}(t).$$

Now we construct functions in $\mathcal{H}(d^*, \beta^{**})$. For $\mathrm{i} < \mathrm{i}^*$, let $q_{\mathrm{i}}(\boldsymbol{x}) := (x_1, \cdots, x_{d_{\mathrm{i}}})^\intercal$. For $\mathrm{i} = \mathrm{i}^*$ define $q_{\mathrm{i}^*,\omega}(\boldsymbol{x}) = (b_\omega(x_1), 0, \cdots, 0)^\intercal$. For $\mathrm{i} > \mathrm{i}^*$, set $q_{\mathrm{i}}(\boldsymbol{x}) := (x_1^{\beta_{\mathrm{i}} \wedge 1}, 0, \cdots, 0)^\intercal$.

$$\tilde{b}_\omega(\boldsymbol{x}) = q_l \circ \cdots \circ q_{\mathrm{i}^*+1} \circ q_{\mathrm{i}^*,\omega} \circ q_{\mathrm{i}^*-1} \circ \cdots \circ q_0(\boldsymbol{x}) = b_\omega(x_1)^B.$$

Notice that $\tilde{b}_\omega(\boldsymbol{x}) \leq (\tau M^{-\beta^*})^B$. Let $\Omega = \{0,1\}^M$. Define

$$p_\omega(\boldsymbol{x}) = 1 + \left[\frac{\frac{1}{2} + (\tau M^{-\beta^*})^B - x_2}{c_2}\right]^{1/\kappa} \mathbb{I}_{\{\frac{1}{2} \leq x_2 \leq \frac{1}{2} + \tilde{b}_\omega(\boldsymbol{x})\}} - b_3(\omega)\mathbb{I}_{\{\frac{1}{2} + \tilde{b}_\omega(\boldsymbol{x}) < x_2 \leq 1\}},$$

where $b_3(\omega) > 0$ is chosen such that $p_\omega(x)$ integrates to 1. Note that both $q_0(\boldsymbol{x})$ and $p_\omega(\boldsymbol{x})$ are d-dimensional densities even though they seem to only depend on x_1 and x_2. Other entries follow independent uniform distribution on $[0, 1]$ and don't show on the density formulas.

Set $\mathcal{F}_1 = \{p_\omega : \omega \in \Omega\}$ and we will show that $(p_\omega, q_0) \in \mathcal{F}_{\mathcal{G}_h}$ for all $\omega \in \Omega$. To this end, we need to verify that

(a) $p_\omega(\boldsymbol{x}) \leq c_1$ for $x \in K$;

(b) $\{\boldsymbol{x} \in \mathcal{X} : p_\omega(\boldsymbol{x}) \geq q_0(\boldsymbol{x})\} \in \mathcal{G}_h$;

(c) $Q\{\boldsymbol{x} \in \mathcal{X} : |p_\omega(\boldsymbol{x}) - q_0(\boldsymbol{x})| \leq \eta\} \leq c_2\eta^\kappa$ for all $0 < \eta < \eta_0$.

For (a), since p_ω integrates to 1,

$$\begin{aligned} b_3(\omega) &\leq \max_{\{\frac{1}{2} \leq x_2 \leq \frac{1}{2} + \tilde{b}_\omega(\boldsymbol{x})\}} \left[\frac{\frac{1}{2} + (\tau M^{-\beta^*})^B - x_2}{c_2}\right]^{1/\kappa} \\ &\leq \left[\frac{2\tau^B M^{-\beta^{**}}}{c_2}\right]^{1/\kappa} = O(M^{-\beta^{**}/\kappa}). \end{aligned}$$

Thus, $p_\omega(\boldsymbol{x}) \leq c_1$ for c_1 and M large enough.

(b) is satisfied since

$$\{\boldsymbol{x} : p_\omega(\boldsymbol{x}) \geq q_0(\boldsymbol{x})\} = \{\boldsymbol{x} : 0 \leq x_2 \leq \frac{1}{2} + \tilde{b}_\omega(x_1)\},$$

and by construction, $\tilde{b}_\omega(\boldsymbol{x}) \in \mathcal{H}(d^*, \beta^{**})$ for τ small enough.

(c) follows that

$$\begin{aligned} & Q\{\boldsymbol{x} \in \mathcal{X} : |p_\omega(\boldsymbol{x}) - q_0(\boldsymbol{x})| \leq \eta\} \\ \leq & Q\{\boldsymbol{x} \in \mathcal{X} : \frac{1}{2} \leq x_2 \leq \frac{1}{2} + (\tau M^{-\beta^*})^B, \left[\frac{1/2 + (\tau M^{-\beta^*})^B - x_2}{c_2}\right]^{1/\kappa} \leq \eta\} \\ \leq & Q\{\boldsymbol{x} \in \mathcal{X} : \frac{1}{2} + (\tau M^{-\beta^*})^B - c_2\eta^\kappa \leq x_2 \leq \frac{1}{2} + (\tau M^{-\beta^*})^B\} \\ \leq & c_2\eta^\kappa. \end{aligned}$$

After verifying $(p_\omega, q_0) \in \mathcal{F}_{\mathcal{G}_h}$ for all $\omega \in \Omega$, we now establish how fast can S go to zero where

$$S := \frac{1}{|\mathcal{F}_1|} \sum_{p \in \mathcal{F}_1} \mathbb{E}_p[d_\triangle(\tilde{G}_{m,n}, G^*)|y_1, \cdots, y_m].$$

To this end, we use the Assouad's lemma stated in [57] which is adapted to the estimation of sets.

For $\mathrm{j} = 1, \cdots, M$ and for a vector $\omega = (\omega_1, \cdots, \omega_M)$, we write

$$\omega_{\mathrm{j}0} = (\omega_1, \cdots, \omega_{\mathrm{j}-1}, 0, \omega_{\mathrm{j}+1}, \cdots, \omega_M),$$
$$\omega_{\mathrm{j}1} = (\omega_1, \cdots, \omega_{\mathrm{j}-1}, 1, \omega_{\mathrm{j}+1}, \cdots, \omega_M).$$

For $\mathrm{i} = 0$ and $\mathrm{i} = 1$, let P_{ji} be the probability measure corresponding to the distribution of $x_1, \cdots, x_n$ when the underlying density is $p_{\omega_{\mathrm{ji}}}$. Denote the expectation w.r.t. P_{ji} as $\mathbb{E}_{\mathrm{ji}}$. Let

$$\begin{aligned}
\mathcal{D}_{\mathrm{j}} &= \{\boldsymbol{x} \in \mathcal{X} : \frac{1}{2} + \tilde{b}_{\omega_{\mathrm{j}0}}(\boldsymbol{x}) < x_2 \leq \frac{1}{2} + \tilde{b}_{\omega_{\mathrm{j}1}}(\boldsymbol{x})\} \\
&= \{\boldsymbol{x} \in \mathcal{X} : b_{\omega_{\mathrm{j}0}}(x_1) < \left(x_2 - \frac{1}{2}\right)^{1/B} \leq b_{\omega_{\mathrm{j}1}}(x_1)\} \\
&= \{\boldsymbol{x} \in \mathcal{X} : b_{\omega_{\mathrm{j}0}}(x_1) < \left(x_2 - \frac{1}{2}\right)^{1/B} \leq b_{\omega_{\mathrm{j}0}}(x_1) + \psi_{\mathrm{j}}(x_1)\}.
\end{aligned}$$

Then

$$\begin{aligned}
S &\geq \frac{1}{2} \sum_{\mathrm{j}=1}^{M} Q\{\mathcal{D}_{\mathrm{j}}\} \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq \frac{1}{2} \sum_{\mathrm{j}=1}^{M} \int_0^1 \psi_{\mathrm{j}}(x_1)^B dx_1 \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq \frac{1}{2} \sum_{\mathrm{j}=1}^{M} \tau^B M^{-\beta^{**}} \int \phi(Mt)^B dt \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq \frac{1}{4} \sum_{\mathrm{j}=1}^{M} \tau^B M^{-\beta^{**}} \int \phi(Mt)^B dt \left[1 - H^2(P_{10}, P_{11})/2\right]^n
\end{aligned}$$

where $H(\cdot, \cdot)$ denotes the Hellinger distance.

$$
\begin{aligned}
H^2(P_{10}, P_{11}) &= \int \left[\sqrt{p_{\omega_{10}}(\boldsymbol{x})} - \sqrt{p_{\omega_{11}}(\boldsymbol{x})}\right]^2 d\boldsymbol{x} \\
&\leq \int_0^1 \left\{ \int_{\frac{1}{2}}^{\frac{1}{2}+\psi_1(x_1)^B} \left[1 - \sqrt{1 + \left(\frac{\frac{1}{2} + \tau^B M^{-\beta^{**}} - x_2}{c_2}\right)^{1/\kappa}}\right]^2 dx_2 \right. \\
&\quad \left. + \int_{\frac{1}{2}}^1 \left[\sqrt{1 - b_3(\omega_{10})} - \sqrt{1 - b_3(\omega_{11})}\right]^2 dx_2 \right\} dx_1 \\
&\leq \int_0^1 \int_{(\tau M^{-\beta^*})^B - \psi_1(x_1)^B}^{(\tau M^{-\beta^*})^B} \left[1 - \sqrt{1 + \left(\frac{v}{c_2}\right)^{1/\kappa}}\right]^2 dv dx_1 \\
&\quad + |b_3(\omega_{10}) - b_3(\omega_{11})|^2.
\end{aligned}
$$

For the first term,

$$
\begin{aligned}
&\int_0^1 \int_{\tau^B M^{-\beta^{**}} - \psi_1(x_1)^B}^{\tau^B M^{-\beta^{**}}} \left[1 - \sqrt{1 + \left(\frac{v}{c_2}\right)^{1/\kappa}}\right]^2 dv dx_1 \\
&\leq \int_0^1 \int_{\tau^B M^{-\beta^{**}} - \psi_1(x_1)^B}^{\tau^B M^{-\beta^{**}}} \left(\frac{v}{c_2}\right)^{2/\kappa} dv dx_1 \\
&\leq \frac{\kappa c_2^{-2/\kappa}}{\kappa + 2} \int_0^1 \left(\tau^B M^{-\beta^{**}}\right)^{1+2/\kappa} - \left(\tau^B M^{-\beta^{**}} - \psi_1(x_1)^B\right)^{1+2/\kappa} dx_1 \\
&\leq \frac{\kappa c_2^{-2/\kappa}}{\kappa + 2} \left(\tau^B M^{-\beta^{**}}\right)^{1+2/\kappa} \int \left(1 - (1 - \phi(Mt)^B)^{1+2/\kappa}\right) dt \\
&= O\left(M^{-\beta^{**}(1+2/\kappa)-1}\right).
\end{aligned}
$$

On the other hand,

$$
\int_0^1 \int_{1/2}^{1/2+b_\omega(x_1)^B} \left[\frac{\frac{1}{2} + \tau^B M^{-\beta^{**}} - x_2}{c_2}\right]^{1/\kappa} dx_2 dx_1 = b_3(\omega) \left[\frac{1}{2} - b_\omega(x_1)^B\right]
$$

yields

$$\begin{aligned}
b_3(\omega_{11}) &= \frac{1}{\frac{1}{2} - b_{\omega_{11}}(x_1)^B} \int_0^1 \int_{1/2}^{1/2+b_{\omega_{11}}(x_1)^B} \left[\frac{\frac{1}{2} + \tau^B M^{-\beta^{**}} - x_2}{c_2}\right]^{1/\kappa} dx_2 dx_1 \\
&\leq \frac{M c_2^{-1/\kappa}}{\frac{1}{2} - \tau^B M^{-\beta^{**}}} \int_0^1 \int_{\tau^B M^{-\beta^{**}}(1-\phi(Mx_1))}^{\tau^B M^{-\beta^{**}}} u^{1/\kappa} du dx_1 \\
&= \frac{M c_2^{-1/\kappa} \tau^B}{(\frac{1}{2} - \tau^B M^{-\beta^{**}})(1+1/\kappa)} M^{-\beta^{**}(1+1/\kappa)} \int (1 - (1 - \phi(Mt)^B)^{1+1/\kappa}) dt \\
&\leq \frac{c_2^{-1/\kappa} \tau^B}{(\frac{1}{2} - \tau^B M^{-\beta^{**}})(1+1/\kappa)} M^{-\beta^{**}(1+1/\kappa)} \\
&= O(M^{-\beta^{**}(1+1/\kappa)}).
\end{aligned}$$

Hence $|b_3(\omega_{11}) - b_3(\omega_{10})| = O(M^{-\beta^{**}(1+1/\kappa)-1})$ and we have

$$\begin{aligned}
H^2(P_{10}, P_{11}) &= O\left(M^{-\beta^{**}(1+2/\kappa)-1} \vee M^{-\beta^{**}(2+2/\kappa)-2}\right) \\
&= O\left(M^{-\beta^{**}(1+2/\kappa)-1}\right).
\end{aligned}$$

Now choose M as the smallest integer that is larger or equal to

$$n^{\frac{\kappa}{(2+\kappa)\beta^{**}+\kappa}}.$$

Then we have $H^2(P_{10}, P_{11}) \leq C^* n^{-1}(1 + o(1))$ for some constant C^* depending only on κ, c_2, τ, ϕ and

$$\int \min\{dP_{\text{j1}}, dP_{\text{j0}}\} \geq \frac{1}{2}\left[1 - \frac{C^*}{2} n^{-1}(1 + o(1))\right]^n \geq C_1^*$$

for n large enough and C_1^* is another constant. Thus for n large enough,

$$S \geq \frac{1}{2} C_1^* \tau^B M^{-\beta^{**}} \int \phi(t) dt \geq C_2^* n^{-\frac{\kappa\beta^{**}}{(2+\kappa)\beta^{**}+\kappa}}.$$

The constant C_2^* only depends on κ, c_2, τ and ϕ.

Combining all the results so far yields that

$$\liminf_{n\to\infty} \inf_{\tilde{G}_{m,n}} \sup_{(p,q)\in\mathcal{F}_{\mathcal{G}_h}} (n \wedge m)^{\frac{\beta^{**}\kappa}{\beta^{**}(\kappa+2)+d^*\kappa}} \mathbb{E}_{p,q}[d_\triangle(\tilde{G}_{m,n}, G^*)] > 0$$

holds when $d^* = 1$. Using Lemma 2.4.5, we have

$$\liminf_{n\to\infty} \inf_{\tilde{G}_{m,n}} \sup_{(p,q)\in\mathcal{F}_{\mathcal{G}_h}} (n \wedge m)^{\frac{\beta^{**}(\kappa+1)}{\beta^{**}(\kappa+2)+d^*\kappa}} \mathbb{E}_{p,q}[d_{p,q}(\tilde{G}_{m,n}, G^*)] > 0.$$

■

2.4.3 Proof of Theorem 2.1.6

Let $G_f := \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}_d) - x_d \geq 0\}$. Then we have the following lemma characterizing the relationship between $d_\triangle$ and $d_{p,q}$.

Lemma 2.4.7 *Under assumption (M1), further assume on some $D \subset \mathcal{X}$, $0 < \kappa^- \leq K(\boldsymbol{x})$ for all $\boldsymbol{x} \in D$. For any set $G = G_f \subset D$ satisfying $\|f - h^*\|_\infty \leq \epsilon_0$, the following inequality holds*

$$d_\triangle(G, G^*)^{\frac{\kappa^-+1}{\kappa^-}} \lesssim d_{p,q}(G, G^*).$$

Proof Let $\delta(\boldsymbol{x}_{-d}) := |f(\boldsymbol{x}_{-d}) - h^*(\boldsymbol{x}_{-d})| \leq \epsilon_0$. Consider $G \triangle G^*$ in dimension x_d and $\boldsymbol{x}_{-d}$ separately and write $G \triangle G^* = ((G \triangle G^*)_{-d}, (G \triangle G^*)_d)$. Then

$$\begin{aligned} d_\triangle(G, G^*) &= \int_{(G\triangle G^*)_{-d}} \int_{(G\triangle G^*)_d} dx_d d\boldsymbol{x}_{-d} \\ &= \int_{(G\triangle G^*)_{-d}} \delta(\boldsymbol{x}_{-d}) d\boldsymbol{x}_{-d}. \end{aligned}$$

Applying assumption (M1) and Jensen's inequality yields

$$\begin{aligned} d_{p,q}(G, G^*) &= \int_{G\triangle G^*} |p(\boldsymbol{x}) - q(\boldsymbol{x})| d\boldsymbol{x} \\ &= \int_{(G\triangle G^*)_{-d}} \int_0^{\delta(\boldsymbol{x}_{-d})} m_{\boldsymbol{x}}(t) dt d\boldsymbol{x}_{-d} \\ &\geq \int_{(G\triangle G^*)_{-d}} \int_0^{\delta(\boldsymbol{x}_{-d})} \frac{1}{C_{\epsilon_0}} t^{1/\kappa^-} dt d\boldsymbol{x}_{-d} \\ &\geq \frac{1}{C_{\epsilon_0}(1 + 1/\kappa^-)} \int_{(G\triangle G^*)_{-d}} \delta(\boldsymbol{x}_{-d})^{\frac{\kappa^-+1}{\kappa^-}} d\boldsymbol{x}_{-d} \\ &\geq \frac{1}{C_{\epsilon_0}(1 + 1/\kappa^-)} d_\triangle(G, G^*)^{\frac{\kappa^-+1}{\kappa^-}}. \end{aligned}$$

■

Following the notations of [92] and [86]. Let $v_n(h) = \sqrt{n}\int h(\boldsymbol{x})d(P_n - P)$ where P denotes the data distribution, i.e. $\boldsymbol{x} \sim P$ and P_n denotes the empirical distribution of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$.

Lemma 2.4.8 (Theorem 5.11 in [92]) *For some function space $\mathcal{H}$ with $\sup_{h\in\mathcal{H}} \|h(\boldsymbol{x})\|_\infty \leq K$ and $\sup_{h\in\mathcal{H}} \|h(\boldsymbol{x})\|_{L_2(P)} \leq R$ where P is the distribution of $\boldsymbol{x}$. Take $a > 0$ satisfying (1) $a \leq C_1\sqrt{n}R^2/K$; (2) $a \leq 8\sqrt{n}R$;*

$$(3) \quad a \geq C_0 \left(\int_{a/64\sqrt{n}}^{R} H_B^{1/2}(u, \mathcal{F}, L_2(P))du \vee R \right);$$

and (4) $C_0^2 \geq C^2(C_1 + 1)$. Then

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} \left| \sqrt{n} \int h d(P_n - P) \right| \geq a \right) \leq C \exp\left(-\frac{a^2}{C^2(C_1+1)R^2} \right),$$

where P_n is the empirical counterpart of P.

The next lemma investigates the modulus of continuity of the empirical process. It's similar to Lemma 5.13 in [92] but with a key difference in the entropy assumption (2.9), where the entropy bound contains n.

Lemma 2.4.9 *For a probability measure P, let $\mathcal{H}_n$ be a class of uniformly bounded (by 1) functions h in $L_2(P)$ depending on n. Suppose that the δ-entropy with bracketing satisfies for all $0 < \delta < 1$ small enough, the inequality*

$$H_B(\delta, \mathcal{H}_n, L_2(P)) \leq A_n \log(1/\delta), \tag{2.9}$$

where $0 < A_n = o(n)$. Let h_{0n} be a fixed element in $\mathcal{H}_n$. Let $\mathcal{H}_n(\delta) = \{h_n \in \mathcal{H}_n : \|h_n - h_{0n}\|_{L_2(P)} \leq \delta\}$. Then there exist constants $D_1 > 0, D_2 > 0$ such that for a sequence of i.i.d. random variables $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ with probability distribution P, it holds that for all T large enough,

$$\begin{aligned} &\mathbb{P}\left(\sup_{h_n \in \mathcal{H}_n(\sqrt{A_n/n})} \left| \int (h_n - h_{0n}) d(P_n - P) \right| \geq T\frac{A_n}{n} \right) \\ &\leq C \exp\left(-\frac{TA_n}{8C^2} \right) \end{aligned}$$

and for n large enough,

$$\mathbb{P}\left(\sup_{\substack{h_n\in\mathcal{H}_n;\\ \|h_n-h_{0n}\|>\sqrt{A_n/n}}}\frac{|v_n(h_n)-v_n(h_{0n})|}{A_n^{1/2}\|h_n-h_{n0}\|}>D_1x\right)$$
$$\leq D_2\mathrm{e}^{-A_nx}$$

for all $x\geq 1$.

Proof The main tool for the proof is Lemma 2.4.8. Replace $\mathcal{H}$ with $\mathcal{H}_n(\delta)$ in Lemma 2.4.8 and take $K=4$, $R=\sqrt{2}\delta$ and $a=\frac{1}{2}C_1A_n^{1/2}\delta$, with $C_1=2\sqrt{2}C_0$. Then (1) is satisfied if

$$\delta\geq\sqrt{\frac{A_n}{n}}, \tag{2.10}$$

under which, (2) and (3) is trivially satisfied when n is large enough. Choosing C_0 sufficiently large will ensure (4). Thus, for all δ satisfying (2.10), we have

$$\mathbb{P}\left(\sup_{h_n\in\mathcal{H}_n(\delta)}\left|\sqrt{n}\int(h_n-h_{0n})d(P_n-P)\right|\geq\frac{C_1}{2}{A_n}^{1/2}\delta\right)$$
$$\leq C\exp\left(-\frac{C_1A_n}{16C^2}\right).$$

Let $B=\min\{b>1:2^{-b}\leq\sqrt{A_n/n}\}$ and apply the peeling device. Then,

$$\mathbb{P}\left(\sup_{\substack{h_n\in\mathcal{H}_n;\\ \|h_n-h_{0n}\|>\sqrt{A_n/n}}}\frac{|\sqrt{n}\int(h_n-h_{n0})d(P_n-P)|}{A_n^{1/2}\|h_n-h_{n0}\|}\geq\frac{C_1}{2}\right)$$
$$\leq\sum_{b=0}^{B}\mathbb{P}\left(\sup_{h_n\in\mathcal{H}_n(2^{-b})}\left|\sqrt{n}\int(h_n-h_{n0})d(P_n-P)\right|\geq\frac{C_1}{2}{A_n}^{1/2}(2^{-b})\right)$$
$$\leq\sum_{b=0}^{B}C\exp\left(-\frac{C_1A_n}{16C^2}\right)\leq 2C(\log n)\exp\left(-\frac{C_1A_n}{16C^2}\right),$$

if C_1A_n is sufficiently large. ■

Proof of Lemma 2.1.4

Proof For ease of notation, we will write G_f and its defining function f interchangeably. For any $\epsilon > 0$, by construction, we can find $\tilde{f}_n \in \tilde{\mathcal{F}}_n$ such that $\|\tilde{f}_n - h^*\|_\infty \leq \epsilon$. The 0-1 loss can be bounded as

$$\begin{aligned} d_{\mathrm{j}_{-d}}(\tilde{f}_n, h^*) &= \int_{D_{\mathrm{j}_{-d}}: G_{\tilde{f}_{n,\mathrm{j}_{-d}}} \triangle G_{h^*}} |p(\boldsymbol{x}) - q(\boldsymbol{x})| d\boldsymbol{x} \\ &\leq \int_{D_{\mathrm{j}_{-d}}} \int_0^\epsilon m_{\boldsymbol{x}}(t) dt d\boldsymbol{x}_{-d} \\ &\leq C_{\epsilon_0} \int_{D_{\mathrm{j}_{-d}}} \int_0^\epsilon t^{1/K(\boldsymbol{x})} dt d\boldsymbol{x}_{-d} \\ &\leq \frac{C_{\epsilon_0}}{M^{d-1}(1 + 1/\kappa^+)} \epsilon^{\frac{\kappa^+ + 1}{\kappa^+}}. \end{aligned}$$

Since $\hat{f}_{n,\mathrm{j}_{-d}}$ is the empirical risk minimizer within $\tilde{\mathcal{F}}_n$, we have $R_{n,\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}) \leq R_{n,\mathrm{j}_{-d}}(\tilde{f}_n)$. Therefore,

$$\begin{aligned} d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) \leq & d_{\mathrm{j}_{-d}}(\tilde{f}_n, h^*) + [R_{n,\mathrm{j}_{-d}}(\tilde{f}_n) - R_{n,\mathrm{j}_{-d}}(h^*) - d_{\mathrm{j}_{-d}}(\tilde{f}_n, h^*)] \\ & + [R_{n,\mathrm{j}_{-d}}(h^*) - R_{n,\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}) + d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)] \\ :\leq & \frac{C_{\epsilon_0}}{M^{d-1}(1 + 1/\kappa^+)} \epsilon^{\frac{\kappa^+ + 1}{\kappa^+}} + I(\tilde{f}_n, h^*) + I(\hat{f}_{n,\mathrm{j}_{-d}}, h^*). \end{aligned}$$

For $I(\tilde{f}_n, h^*)$, by Lemma 2.4.9, we have

$$\begin{aligned} I(\tilde{f}_n, h^*) \leq & \sup_{\substack{f \in \tilde{\mathcal{F}}_n: \|f - h^*\|_1 \\ \leq \sqrt{A_n/n}}} \left| R_{n,\mathrm{j}_{-d}}(f) - R_{n,\mathrm{j}_{-d}}(h^*) - d_{\mathrm{j}_{-d}}(f, h^*) \right| + \\ & \sqrt{\frac{A_n d_\triangle(\tilde{f}_n, h^*)}{n}} \sup_{\substack{f \in \tilde{\mathcal{F}}_n: \|f - h^*\|_1 \\ > \sqrt{A_n/n}}} \frac{\sqrt{n} \left| R_{n,\mathrm{j}_{-d}}(f) - R_{n,\mathrm{j}_{-d}}(h^*) - d_{\mathrm{j}_{-d}}(f, h^*) \right|}{\sqrt{A_n d_\triangle(f, h^*)}} \\ = & O_{\mathbb{P}}\left(\frac{A_n}{n}\right) + \sqrt{\frac{A_n d_\triangle(\tilde{f}_n, h^*)}{n}} O_{\mathbb{P}}(1), \end{aligned}$$

where A_n is from the assumption (2.9). Similarly for $I(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)$, we have

$$I(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) = O_{\mathbb{P}}\left(\frac{A_n}{n}\right) + \sqrt{\frac{A_n d_{\triangle}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)}{n}}\, O_{\mathbb{P}}(1).$$

By construction, $d_{\triangle}(\tilde{f}_n, h^*) \leq \epsilon$. Hence

$$\begin{aligned} d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) \leq & \frac{C_{\epsilon_0}}{M^{d-1}(1+1/\kappa^+)} \epsilon^{\frac{\kappa^+ + 1}{\kappa^+}} + O_{\mathbb{P}}\left(\frac{A_n}{n}\right) + \\ & \sqrt{\frac{A_n\left(d_{\triangle}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) + \epsilon\right)}{n}}\, O_{\mathbb{P}}(1). \end{aligned}$$

The last term dominates the second term. Omitting the approximation error, i.e. $\epsilon^{\frac{\kappa^+ + 1}{\kappa^+}} \lesssim \sqrt{\frac{A_n}{n}} d_{\triangle}^{1/2}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)$, by Lemma 2.4.7 we have

$$\begin{aligned} d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) &\leq \sqrt{\frac{A_n}{n}} d_{\triangle}^{1/2}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)\, O_{\mathbb{P}}(1) \\ &\leq \sqrt{\frac{A_n}{n}} d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*)^{\frac{\kappa^-}{2(\kappa^- + 1)}}\, O_{\mathbb{P}}(1), \end{aligned}$$

which simplifies to

$$d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) = O_{\mathbb{P}}\left(\frac{A_n}{n}\right)^{\frac{\kappa^- + 1}{\kappa^- + 2}}.$$

From Lemma 2.1.1, we know that $A_n = O(\epsilon^{-\rho} \log n)$. Balancing the approximation error and the empirical error by choosing

$$\epsilon = O\left(n^{-\frac{\kappa^+(\kappa^- + 1)}{(\kappa^- + 2)(\kappa^+ + 1) + \rho\kappa^+(\kappa^- + 1)}}\right)$$

yields

$$\mathbb{E}\, d_{\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}, h^*) = O\left(\frac{1}{n}\right)^{\frac{\kappa^- + 1}{\kappa^- + 2 + \kappa^+ \rho\left(\frac{\kappa^- + 1}{\kappa^+ + 1}\right)}}.$$

■

Proof of Theorem 2.1.6

Proof Choose $\epsilon = n^{-1/(1+\rho)}$, $M = \log n$. Notice that $nM^d\epsilon^d \to 0$, i.e., $\mathbb{P}(E_\epsilon) \to 1$ as $n \to \infty$ as long as $\rho < d-1$. Assumption $n = \Omega(\epsilon_0^{-(1+\rho)})$ implies that the approximation error ϵ can be smaller than ϵ_0. Let

$$\kappa^-_{\mathrm{j}_{-d}} := \min_{\boldsymbol{x} \in D_{\mathrm{j}_{-d}}} K(\boldsymbol{x}) \quad \text{and} \quad \kappa^+_{\mathrm{j}_{-d}} := \max_{\boldsymbol{x} \in D_{\mathrm{j}_{-d}}} K(\boldsymbol{x}).$$

Since $R_{n,\mathrm{j}_{-d}}(\hat{f}_n) = R_{n,\mathrm{j}_{-d}}(\hat{f}_{n,\mathrm{j}_{-d}}) \leq R_{n,\mathrm{j}_{-d}}(\tilde{f}_n)$ for any $\mathrm{j}_{-d} \in J_M$ as in (2.4), Lemma 2.1.4 yields that

$$\sup_{h^* \in \mathcal{F}(d^*,\beta^{**})} \mathbb{E}(R_{\mathrm{j}_{-d}}(\hat{f}_n) - R_{\mathrm{j}_{-d}}(h^*)) \lesssim \left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\left(\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^+_{\mathrm{j}_{-d}}+1}\right)\rho\kappa^+_{\mathrm{j}_{-d}}}}.$$

Then, the overall 0-1 loss excess risk can be decomposed as

$$\begin{aligned}
\sup_{h^* \in \mathcal{F}(d^*,\beta^{**})} \mathbb{E}(R(\hat{f}_n) - R(h^*)) &\leq \sum_{\mathrm{j}_{-d} \in J_M} \sup_{h^* \in \mathcal{F}(d^*,\beta^{**})} \mathbb{E}(R_{\mathrm{j}_{-d}}(\hat{f}_n) - R_{\mathrm{j}_{-d}}(h^*)) \\
&\lesssim \sum_{\mathrm{j}_{-d} \in J_M} \left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\left(\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^+_{\mathrm{j}_{-d}}+1}\right)\rho\kappa^+_{\mathrm{j}_{-d}}}}.
\end{aligned}$$

By assumption (M2), we can write for any $\mathrm{j}_{-d} \in J_M$ that

$$\begin{aligned}
\left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\left(\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^+_{\mathrm{j}_{-d}}+1}\right)\rho\kappa^+_{\mathrm{j}_{-d}}}} &= \left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}}+\rho\frac{\kappa^+_{\mathrm{j}_{-d}}-\kappa^-_{\mathrm{j}_{-d}}}{\kappa^+_{\mathrm{j}_{-d}}+1}}} \\
&\leq \left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}}+\rho C_K(\sqrt{d}/M)^\alpha}} \\
&= \left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}}}+\frac{(\kappa^-_{\mathrm{j}_{-d}}+1)\rho C_K(\sqrt{d}/M)^\alpha}{(\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}}+\rho C_K(\sqrt{d}/M)^\alpha)(\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}})}} \\
&= O\left(\frac{1}{n}\right)^{\frac{\kappa^-_{\mathrm{j}_{-d}}+1}{\kappa^-_{\mathrm{j}_{-d}}+2+\rho\kappa^-_{\mathrm{j}_{-d}}}}.
\end{aligned}$$

The last equality follows from the fact that $M = \log n$ and $n^{-1/\log n} = O(1)$. Since κ is defined as the overall minimum, under E_ϵ, we have

$$\begin{aligned}\sup_{h^* \in \mathcal{H}(d^*, \beta^{**})} \mathbb{E}(R(\hat{f}_n) - R(h^*)) &\lesssim \sum_{\mathrm{j}_{-d} \in J_M} \left(\frac{1}{n}\right)^{\frac{\kappa_{\mathrm{j}}^- + 1}{\kappa_{\mathrm{j}}^- + 2 + \rho\kappa_{\mathrm{j}}^-}} \\ &= O\left(n^{-\frac{(\kappa+1)\beta^{**}}{(\kappa+2)\beta^{**}+\kappa d^*}} (\log n)^{d-1}\right).\end{aligned}$$

■

2.4.4 Proof of Properties (P1) to (P3)

Let's first consider the $d = 2$ case and focus on some region $D_{\mathrm{i}} = \{(x_1, x_2) \in [0,1]^2 : x_1 \in (a, b)\}$ with $b - a > 2\epsilon$. Let $f_n \in \tilde{\mathcal{F}}$ be any DNN. Define three continuous piecewise linear functions

$$g(x_1) = \begin{cases} x_1 & \text{if } a \le x_1 \le b \\ a & \text{if } x_1 < a \\ b & \text{if } x_1 > b \end{cases}, \quad c(x_1) = \begin{cases} f_n(0) & \text{if } a + \epsilon \le x_1 \le b - \epsilon \\ 0 & \text{if } x_1 < a \text{ or } x_1 > b \\ \text{linear transition} & \text{else} \end{cases}$$

and

$$h(x_1) = \begin{cases} 0 & \text{if } a + \epsilon \le x_1 \le b - \epsilon \\ a & \text{if } x_1 < a \\ b & \text{if } x_1 > b \\ \text{linear transition} & \text{else.} \end{cases}$$

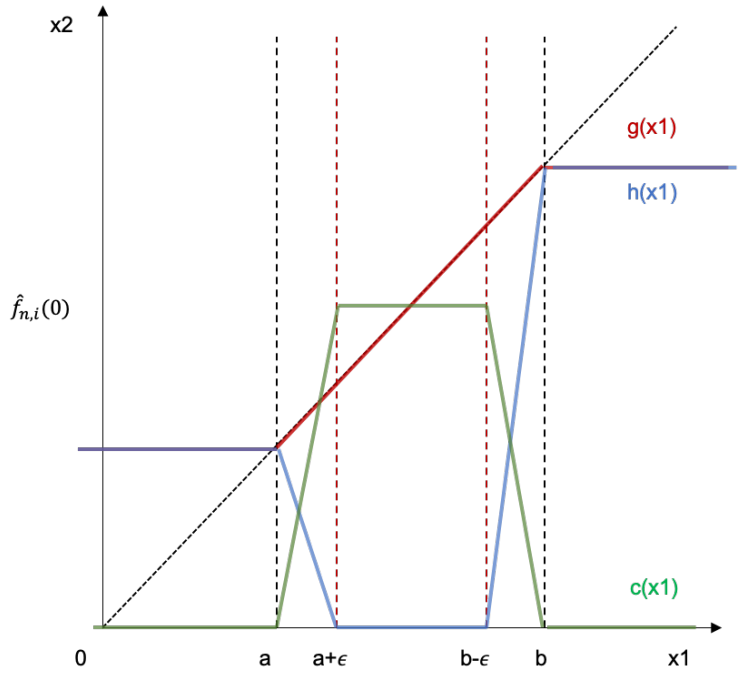


Figure 2.5. Illustration of the constructed functions g, h, c in $d = 2$ case.

Linear transition means linking the end points with a line segment. The constructed piecewise linear functions are illustrated in Figure 2.5. Let $f^+_{n,\mathrm{i}}(x_1) := f_n(g(x_1)) - f_n(h(x_1)) + c(x_1)$. Then, it's easy to verify that

$$f^+_{n,\mathrm{i}}(x_1) = \begin{cases} f_n(x_1) & \text{if } a + \epsilon \le x_1 \le b - \epsilon \\ 0 & \text{if } x_1 < a \text{ or } x_1 > b \\ \text{piecewise linear} & \text{else.} \end{cases}$$

Therefore, (P1) and (P2) hold and we move to evaluate (P3). The constructed g, h, c are all piecewise linear functions with at most 5 pieces. By Theorem 2.2 in [10], they can all be represented by two-layer ReLU neural networks with width at most 5. $f^+_{n,\mathrm{i}}(x_1)$ is constructed by composition and addition of ReLU networks, which correspond to stacking more layers and expanding the width respectively. Easy to see that $f^+_{n,\mathrm{i}}(x_1)$ satisfies (P3).

In the $d > 2$ case, we can make similar constructions. Consider some region $D_{\mathrm{j}_{-d}}$ and denote $D^{\circ}_{\mathrm{j}_{-d}} := D_{\mathrm{j}_{-d}} \backslash D_{\epsilon}$. For each of the dimensions $x_1, \ldots, x_{d-1}$, we can define $g_{\mathrm{i}}(x_{\mathrm{i}}), h_{\mathrm{i}}(x_{\mathrm{i}}), c_{\mathrm{i}}(x_{\mathrm{i}})$ separately as in the $d = 2$ case. Let $g(\boldsymbol{x}_{-d}) = (g_1(x_1), \ldots, g_{d-1}(x_{d-1}))$, $h(\boldsymbol{x}_{-d}) = (h_1(x_1), \ldots, h_{d-1}(x_{d-1}))$, $c(\boldsymbol{x}_{-d}) = (c_1(x_1), \ldots, c_{d-1}(x_{d-1}))$ and $f^{+}_{n,\mathrm{j}_{-d}} = (f_n \circ g - f_n \circ h + c)$. Then, it's easy to verify that

$$f^{+}_{n,\mathrm{j}_{-d}}(\boldsymbol{x}_{-d}) = \begin{cases} f_n(\boldsymbol{x}_{-d}) & \text{if } \boldsymbol{x}_{-d} \in D^{\circ}_{\mathrm{j}_{-d}} \\ 0 & \text{if } \boldsymbol{x}_{-d} \notin D_{\mathrm{j}_{-d}} \\ \text{piecewise linear} & \text{else.} \end{cases}$$

Thus, (P1) and (P2) hold. For (P3), notice that $g(\boldsymbol{x}_{-d})$ can be viewed as a ReLU neural network with the same depth as $g_{\mathrm{i}}(x_{\mathrm{i}})$ but $(d-1)$-times the width.

2.4.5 Proof of Lemmas in Section 2.2

We first present some preliminary lemmas. Corresponding to assumption (A3), we define (N_n) as an extension to the classical Tsybakov noise condition (N).

(N_n) There exists $c_n > 0$ depending on n and $\kappa \in [0, \infty]$ such that for any $0 \leq t \leq T_n$

$$\mathbb{P}\left(\{\boldsymbol{x} : |p_n(\boldsymbol{x}) - q_n(\boldsymbol{x})| \leq t\}\right) \leq c_n t^{\kappa}.$$

Note that the (N) is a special case of (N_n) with T_n and c_n being absolute constant. The following lemma establishes the connection between $d_{\triangle}$ and $d_{p,q}$, which is adapted from Lemma 2 in [1] to our teacher network setting.

Lemma 2.4.10 *Assume (N_n) and p_n, q_n are bounded by $b_2 > 0$. Then, there exists absolute constants $b_1(\kappa) > 0$ depending on κ such that for any Lebesgue measurable subsets G_1 and G_2 of $\mathcal{X}$,*

$$b_1(\kappa)\left(T_n \wedge c_n^{-1/\kappa}\right) d_{\triangle}^{(\kappa+1)/\kappa}(G_1, G_2) \leq d_{p_n,q_n}(G_1, G_2) \leq 2b_2 d_{\triangle}(G_1, G_2).$$

Proof The second inequality is trivial given that p, q are bounded by b_2. For the first inequality, since $\mathbb{Q}(|p_n - q_n| \leq t) \leq c_n t^\kappa$ for all $0 \leq t \leq T_n$, the boundedness of $\mathbb{Q}(\mathcal{X})$ implies that

$$\mathbb{Q}(|p_n - q_n| \leq t) \leq A_n t^\kappa, \quad \forall\ t > 0,$$

where $A_n = \left(\frac{\mathbb{Q}(\mathcal{X})}{T_n^\kappa} \vee c_n\right)$. Then,

$$\begin{aligned}
&d_{p_n,q_n}(G_1, G_2) \\
&\geq \int_{G_1 \triangle G_2} |p_n - q_n| \mathbb{I}\{|p_n - q_n| \geq \left(\frac{d_\triangle(G_1, G_2)}{2A_n}\right)^{1/\kappa}\} d\mathbb{Q} \\
&\geq \left(\frac{d_\triangle(G_1, G_2)}{2A_n}\right)^{1/\kappa} \left[\mathbb{Q}(G_1 \triangle G_2) - \mathbb{Q}(|p_n - q_n| < \left(\frac{d_\triangle(G_1, G_2)}{2A_n}\right)^{1/\kappa})\right] \\
&\geq \frac{d_\triangle(G_1, G_2)^{1+1/\kappa}}{(2A_n)^{1/\kappa}} - 1/2 \frac{d_\triangle(G_1, G_2)^{(\kappa+1)/\kappa}}{(2A_n)^{1/\kappa}} \\
&\geq \frac{2^{-(\kappa+1)/\kappa}}{A_n^{1/\kappa}} d_\triangle(G_1, G_2)^{(\kappa+1)/\kappa}.
\end{aligned}$$

■

Lemma 2.4.11 characterizes the complexity of a special collection of sets.

Lemma 2.4.11 *Let $\mathcal{X} = [0,1]^d$ and $\mathcal{G}$ be a collection of polyhedrons with at most S vertices in $\mathbb{R}^d$. Then the bracketing entropy of $\bar{\mathcal{G}} = \mathcal{G} \cap \mathcal{X}$ satisfies*

$$H_B(\delta, \bar{\mathcal{G}}, d_\triangle) = \log \mathcal{N}_B(\delta, \bar{\mathcal{G}}, d_\triangle) \lesssim d^2 S \log(d^{3/2} S/\delta)$$

Proof Let's first introduce some notations and terminologies. For any $\delta > 0$, let M_δ denote the smallest integer such that $M_\delta > 1/\delta$. Consider the set of lattice points $\boldsymbol{X}_\delta^d = \{(\mathrm{i}_1/M_\delta, \ldots, \mathrm{i}_d/M_\delta) : \mathrm{i}_1, \ldots, \mathrm{i}_d = 0, 1, \ldots, M_\delta\}$ which has cardinality $(M_\delta+1)^d$. Let $G(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$ denote a polyhedron with vertices $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s \in [0,1]^d$ where $s \leq S$. (the $\boldsymbol{x}_\mathrm{i}$'s are not necessarily distinct). Any convex polyhedron G in $\mathbb{R}^d$ is the intersection of multiple $(d-1)$-dimensional hyperplanes. If we move all such hyperplanes inwards (to the direction perpendicular to the hyperplanes) by a small distance δ, they produce another polyhedron, denoted $G_{-\delta}$, called as the δ-contraction of G. Note that $G_{-\delta}$ can be empty if δ is not small enough.

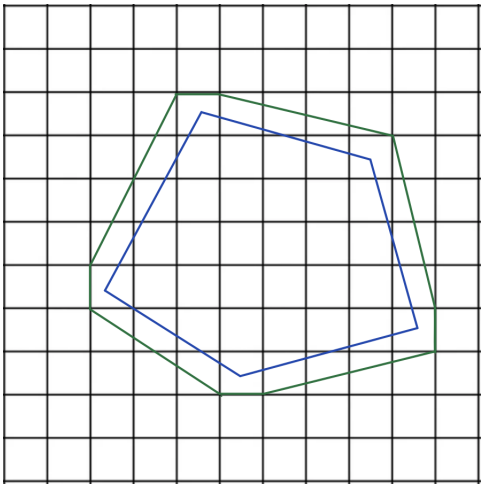


Figure 2.6. Grid in 2D and the outer cover (green) constructed for with grid points for a polygon (blue).

We prove the result for $d = 1$, in which $\bar{\mathcal{G}}$ is a collection of subintervals in $[0, 1]$. For any subinterval $[a, b] \subset [0, 1]$, there exist $x_{\mathrm{i}}, x_{\mathrm{j}} \in \boldsymbol{X}_\delta^1$ such that

$$x_{\mathrm{i}} \le a \le x_{\mathrm{i}+1}, \quad x_{\mathrm{j}} \le b \le x_{\mathrm{j}+1}.$$

(By convention, $[x_{\mathrm{i}}, x_{\mathrm{j}}]$ is empty if $x_{\mathrm{i}} > x_{\mathrm{j}}$.) Then $([x_{\mathrm{i}}, x_{\mathrm{j}+1}], [x_{\mathrm{i}+1}, x_{\mathrm{j}}])$ is a 2δ-bracket of $[a, b]$ since obviously

$$[x_{\mathrm{i}+1}, x_{\mathrm{j}}] \subset [a, b] \subset [x_{\mathrm{i}}, x_{\mathrm{j}+1}], \quad d_\triangle([x_{\mathrm{i}}, x_{\mathrm{j}+1}], [x_{\mathrm{i}+1}, x_{\mathrm{j}}]) \le 2\delta. \tag{2.11}$$

There are $\binom{M_\delta+1}{2}$ different choices of $[x_{\mathrm{i}}, x_{\mathrm{j}}]$, hence, $\binom{M_\delta+1}{2}$ different choices of the pairs $([x_{\mathrm{i}}, x_{\mathrm{j}+1}], [x_{\mathrm{i}+1}, x_{\mathrm{j}}])$. Any $[a, b] \subset [0, 1]$ can be 2δ bracketed by one of such pairs in the sense of (2.11). This shows that $H_B(2\delta) \le \log \binom{M_\delta+1}{2} \le 2 \log(1/\delta)$.

When $d \ge 2$, any $G \in \bar{\mathcal{G}}$ has at most S vertices, so $\bar{G} := G \cap [0, 1]^d$ has at most dS vertices where the factor d is due to the fact that each edge of G intersects at most d edges of $[0, 1]^d$ therefore creates at most dS vertices for $\bar{G}$. For any polygon $G(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$ where $s \le dS$, denote $G_{-\sqrt{d}\delta}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s) = G(\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_s^-)$. Each vertex must be in one of the grids in $\boldsymbol{X}_\delta^d$. It is easy to see that there exist $\boldsymbol{v}_1^1, \ldots, \boldsymbol{v}_1^d, \cdots, \boldsymbol{v}_s^1, \ldots, \boldsymbol{v}_s^d \in \boldsymbol{X}_\delta^d$, where $\boldsymbol{v}_{\mathrm{i}}^1, \ldots, \boldsymbol{v}_{\mathrm{i}}^d$ are in the same grid, such that

- $G(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s) \subset G(\boldsymbol{v}_1^1, \ldots, \boldsymbol{v}_1^d, \cdots, \boldsymbol{v}_s^1, \ldots, \boldsymbol{v}_s^d)$;
- $\|\boldsymbol{v}_{\mathrm{i}}^{\mathrm{j}} - \boldsymbol{x}_{\mathrm{i}}\|_2 \leq \sqrt{d}\delta$ for $\mathrm{i} = 1, 2 \cdots, s$ and $\mathrm{j} = 1, 2, \cdots, d$.

See Figure 2.6 for an illustration when $d = 2$.

Similarly for $G(\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_s^-)$, there exist $\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^d, \cdots, \boldsymbol{u}_s^1, \ldots, \boldsymbol{u}_s^d \in \boldsymbol{X}_\delta^d$ such that

- $G(\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_s^-) \subset G(\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^d, \cdots, \boldsymbol{u}_s^1, \ldots, \boldsymbol{u}_s^d)$;
- $\|\boldsymbol{u}_{\mathrm{i}}^{\mathrm{j}} - \boldsymbol{x}_{\mathrm{i}}^-\|_2 \leq \sqrt{d}\delta$ for $\mathrm{i} = 1, 2 \cdots, s$ and $\mathrm{j} = 1, 2, \cdots, d$.

By the definition of $G_{-\sqrt{d}\delta}$, we have $\|\boldsymbol{x}_{\mathrm{i}} - \boldsymbol{x}_{\mathrm{i}}^-\|_2 \geq \sqrt{d}\delta$. Thus $\|\boldsymbol{u}_{\mathrm{i}}^{\mathrm{j}} - \boldsymbol{x}_{\mathrm{i}}^-\|_2 \leq \sqrt{d}\delta$ implies $G(\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^d, \cdots, \boldsymbol{u}_s^1, \ldots, \boldsymbol{u}_s^d) \subset G(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$. On the other hand,

$$\begin{aligned}
&d_\triangle(G(\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^d, \cdots, \boldsymbol{u}_s^1, \ldots, \boldsymbol{u}_s^d), G(\boldsymbol{v}_1^1, \ldots, \boldsymbol{v}_1^d, \cdots, \boldsymbol{v}_s^1, \ldots, \boldsymbol{v}_s^d)) \\
&\leq d_\triangle(G_{+\sqrt{d}\delta}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s), G_{-\sqrt{d}\delta}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)) \\
&\leq s \cdot 2\sqrt{d}\delta,
\end{aligned}$$

where the term s is due to the fact that $G(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$ has at most $O(s)$ faces. Notice that

$$G(\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^d, \cdots, \boldsymbol{u}_s^1, \ldots, \boldsymbol{u}_s^d), G(\boldsymbol{v}_1^1, \ldots, \boldsymbol{v}_1^d, \cdots, \boldsymbol{v}_s^1, \ldots, \boldsymbol{v}_s^d) \in \bar{\mathcal{G}},$$

and $s \leq dS$. Thus, with at most $(M_\delta + 1)^{d^2 S}$ pairs of subsets in $\bar{\mathcal{G}}$, we can $2d^{3/2}S\delta$-bracket any $\bar{G} \in \bar{\mathcal{G}}$. Therefore,

$$\log \mathcal{N}_B((2d^{3/2}S\delta), \bar{\mathcal{G}}, d_\triangle) \lesssim \log\left((M_\delta + 1)^{d^2 S}\right),$$

which implies

$$\log \mathcal{N}_B(\delta, \bar{\mathcal{G}}, d_\triangle) \lesssim d^2 S \log(d^{3/2}S/\delta).$$

■

Lemma 2.4.12 (Theorem 1 in [93]) *Consider a deep ReLU network with L layers, n_l ReLU nodes at each layer l, and an input of dimension n_0. The maximal number of linear pieces of this neural network is at most*

$$\sum_{(\mathrm{j}_1,\cdots,\mathrm{j}_L)\in J}\prod_{l=1}^{L}\binom{n_l}{\mathrm{j}_l},$$

where $J = \{(\mathrm{j}_1,\ldots,\mathrm{j}_L) \in \mathbb{Z}^L : 0 \le \mathrm{j}_l \le \min\{n_0, n_1 - \mathrm{j}_1, \ldots, n_{l-1} - \mathrm{j}_{l-1}, n_l\}\ \forall l = 1,\ldots,L\}$. This bound is tight when $L = 1$. When $n_0 = O(1)$ and all layers have the same width N, we have the same best known asymptotic bound $O(N^{Ln_0})$ first presented in [94].

Consider a deep ReLU network with $n_0 = d$ inputs and L hidden layers of widths $n_{\mathrm{i}} \ge n_0$ for all $\mathrm{i} \in [L]$. The following lemma establishes a lower bound for the maximal number of linear pieces of deep ReLU networks:

Lemma 2.4.13 (Theorem 4 in [88]) *The maximal number of linear pieces of a ReLU network with n_0 input units, L hidden layers, and $n_{\mathrm{i}} \ge n_0$ rectifiers on the* i*-th layer, is lower bounded by*

$$\left(\prod_{\mathrm{i}=1}^{L-1}\left\lfloor\frac{n_{\mathrm{i}}}{n_0}\right\rfloor^{n_0}\right)\sum_{\mathrm{j}=0}^{n_0}\binom{n_L}{\mathrm{j}}.$$

2.4.6 Proof of Theorem 2.2.2

Lemma 2.4.14 *Let $\mathcal{F}$ be a class of ReLU neural networks, defined on $\mathcal{X} = [0,1]^d$, with at most L layers and N neurons per layer. Let $G^f = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \ge 0\}$ and $\mathcal{G}^{\mathcal{F}} = \{G^f : f \in \mathcal{F}\}$. Then the bracketing number of $\mathcal{G}^{\mathcal{F}}$ satisfies*

$$\log \mathcal{N}_B(\delta, \mathcal{G}^{\mathcal{F}}, d_\triangle) \lesssim N^{Ld^2} d^3 \left(Ld^2 \log(N) \vee \log(1/\delta)\right).$$

Proof The proof relies on Lemma 2.4.11 for which we need to control the number of vertexes of G^f based on the number of pieces (linear regions) of the ReLU neural network. Since

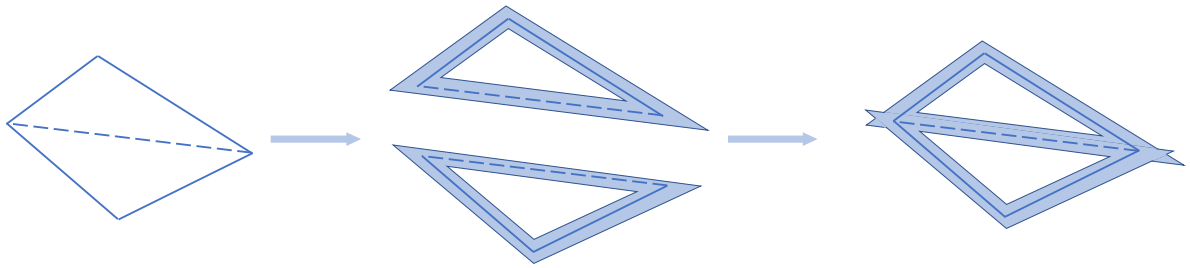


Figure 2.7. Demonstration of how a polygon in $d = 2$ case can be divided into basic triangles. The union of the two brackets form a bracket of the original polygon. The blue shade is the symmetric difference.

ReLU neural networks are piecewise linear, G^f is a collection of sets of polyhedrons. Define the subgraph of a function $f : \mathbb{R}^d \to \mathbb{R}$ to be the set of points in $\mathbb{R}^{d+1}$:

$$\text{sub}(f) = \{(\boldsymbol{x}, t) : f(\boldsymbol{x}) \geq t\}.$$

In this sense, $\text{sub}(f) \cap \{(\boldsymbol{x}, 0) : \boldsymbol{x} \in \mathcal{X}\} = \{(\boldsymbol{x}, 0) : \boldsymbol{x} \in G^f\}$, a slice of the subgraph. Denote all the pieces to be $p_1, p_2, \cdots, p_s$. Each piece is a d-dimensional polyhedron on which $f(x)$ is linear. To control the complexity of G^f, consider the most extreme case that the function crosses zero on each piece, i.e. for any $\mathrm{i} = 1, \ldots, s$, $\{(\boldsymbol{x}, f(\boldsymbol{x})) : \boldsymbol{x} \in p_\mathrm{i}\} \cap \{(\boldsymbol{x}, 0) : \boldsymbol{x} \in \mathcal{X}\} \neq \emptyset$. Each intersection resides in a $(d-1)$-dimensional hyperplane, e.g. dot for $d = 1$, line segment for $d = 2$ and so on. So the number of such $(d-1)$-dimensional hyperplanes in G^f is at most s.

A vertex of a polyhedron in $[0, 1]^d$ can be thought of as the intersection of at least d hyperplanes of dimension $d-1$. Thus, with at most s hyperplanes there are at most $\binom{s}{d} < s^d$ vertices in G^f. In order to apply Lemma 2.4.11, we break the collection of polyhedrons into the so-called *basic polyhedrons* each with $d+1$ vertices. For instance, the basic polyhedrons are intervals when $d = 2$, are triangles when $d = 3$, and so on.

A polyhedron G with at most s vertices can be divided into at most s disjoint basic polyhedrons $B_1, \ldots, B_s$. For instance, Figure 2.7 demonstrates the $d = 2$ case. Therefore, the bracketing number of the polyhedrons can be derived by bracketing the basic polyhedrons.

For a basic polyhedron B, denote its δ-bracketing pair to be $(U_{B,\delta}, V_{B,\delta})$, i.e., $U_{B,\delta} \subset B \subset V_{B,\delta}$. Then $(U_{G,\delta}, V_{G,\delta})$, defined as below

$$\begin{aligned} U_{G,\delta} =& U_{B_1,\delta} \cup U_{B_2,\delta} \cup \cdots \cup U_{B_s,\delta} \\ V_{G,\delta} =& V_{B_1,\delta} \cup V_{B_2,\delta} \cup \cdots \cup V_{B_s,\delta}, \end{aligned}$$

form a $(s\delta)$-bracket of G. Hence, the bracketing number of all polyhedrons is controlled by the s-th power of the bracketing number of all basic polyhedrons. Applying Lemma 2.4.12 we know $s = O(N^{Ld})$ and the number of vertices is at most $S = O(N^{Ld^2})$. Together with Lemma 2.4.11, we therefore get that

$$\log \mathcal{N}_B(S\delta, \mathcal{G}^{\mathcal{F}}, d_\triangle) \lesssim S(d+1)d^2 \log((d+1)d^{3/2}/\delta),$$

which implies

$$\begin{aligned} \log \mathcal{N}_B(\delta, \mathcal{G}^{\mathcal{F}}, d_\triangle) &\lesssim N^{Ld^2} d^3 \log(N^{Ld^2} d^3/\delta) \\ &\lesssim N^{Ld^2} d^3 \left(Ld^2 \log(N) \vee \log(1/\delta) \right). \end{aligned}$$

■

Lemma 2.4.14 is the main result for controlling the bracketing entropy of the estimation sets. Below we point out some key properties of this result and compare it to other entropy bounds of neural networks.

Exponential Dependence on Depth The bracketing entropy of $\mathcal{G}^{\mathcal{F}}$ developed in Lemma 2.4.14 is much larger than that of $\mathcal{F}$ itself with respect to $\|\cdot\|_\infty$, as described in Lemma 2.4.15. The main difference is the dependence on the number of layers L: the dependence is linear in Lemma 2.4.15 while exponential in Lemma 2.4.14. Thus, even though $\mathcal{G}^{\mathcal{F}}$ is a slice of the subgraph of $\mathcal{F}$, $\mathcal{G}^{\mathcal{F}}$ is much more complicated than $\mathcal{F}$ in term of entropy. We argue that this gap cannot be closed even in the special case $d = 1$.

A lower bound on the maximum number of linear pieces for a ReLU neural network is established in [88] (Lemma 2.4.13). Consider a 1-dimensional ReLU DNN function with L

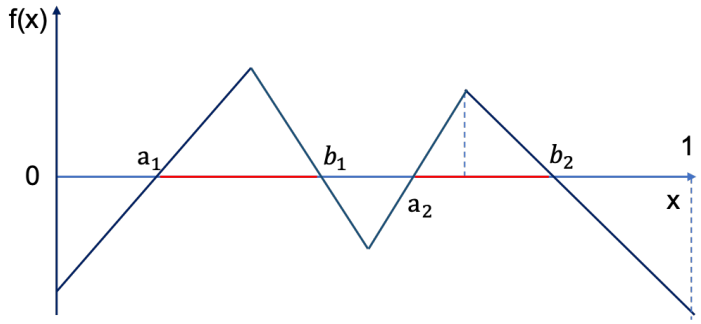


Figure 2.8. Example of a ReLU function in 1D. The induced set where $f > 0$ is colored red and it's a union of two intervals $(a_1, b_1), (a_2, b_2)$. All pieces cross 0 so there are all active.

layers and 2 nodes on each layer. Corollary 5 of [88] show that there exists some f with $s = \Omega(2^{L-1})$ pieces on $[0, 1]$. With scaling and shifting, assume that on each piece the linear function crosses 0. Then, G^f will be at least $\lfloor s/2 \rfloor = \Omega(2^{L-2})$ intervals. Denote these disjoint intervals to be $\{(a_i, b_i)\}_{i=1}^{\lfloor s/2 \rfloor}$. Since they are disjoint, to construct a δ-bracket of all the intervals, we need to δ-cover all the a_i's and b_i's. Similar to the grid argument from the proof of Lemma 2.4.11, we need at least

$$\binom{1/\delta}{s} = \Omega\left((1/\delta - s)^s\right)$$

different combinations of the s grid points. Hence the bracketing entropy must be in the order of

$$\log((1/\delta - s)^s) = 2^{L-2} \log(1/\delta).$$

The exponential dependence of depth L in the entropy stems from the fact that the number of linear regions of ReLU DNNs scales exponentially with L.

Independent of Weights Magnitude We also want to point out that the entropy of $\mathcal{G}^{\mathcal{F}}$ is not concerned with the magnitude of the neural network weights, in contrast to the bound in Lemma 2.4.15. This is because any scaling of the function doesn't change how it intercepts with zero. Hence, unlike $\mathcal{F}$, the entropy of $\mathcal{G}^{\mathcal{F}}$ doesn't depend on the weight maximum B.

The Use of ReLU Activation The reason why we can even bound the entropy of $\mathcal{G}^{\mathcal{F}}$ critically relies on the fact that we are considering the ReLU activation function. If we consider smooth nonlinear activation functions, e.g. hyperbolic tangent, sigmoid, instead of the order $\log(1/\delta)$, we can only get the entropy of a much larger order

$$H_B(\delta, \mathcal{G}^{\mathcal{F}}, d_{\triangle}) \leq A\delta^{-\alpha}$$

for some constant $A > 0$ and $\alpha > 0$. To see this, consider the case $d = 2$. Instead of polygons, which can be controlled by the vertices, the regions have smooth boundary and will require $O(1/\delta)$ many grid points to cover. Thus the covering number is of order

$$\binom{1/\delta^2}{1/\delta} = O\left(\left(\frac{1}{\delta}\right)^{2/\delta}\right).$$

Thus, the entropy is in a polynomial order of $1/\delta$.

To characterize the bracketing entropy in our teacher-student setting, as an intermediate step, we investigate the bracketing entropy with respect to d_{p_n,q_n}. As a direct outcome from Lemma 2.4.10, we can conclude that

$$H_B\left(b_1(\kappa)\left(T_n \wedge c_n^{-1/\kappa}\right)\delta^{\frac{\kappa+1}{\kappa}}, \mathcal{G}, d_{\triangle}\right) \leq H_B\left(\delta, \mathcal{G}, d_{p_n,q_n}\right). \tag{2.12}$$

To bound $H_B\left(\delta, \mathcal{G}, d_{p_n,q_n}\right)$, we construct the brackets of $\mathcal{G}$ using the δ-covering set of $\mathcal{F}$ with respect to $\|\cdot\|_\infty$. Let $\mathcal{N}$ and $H = \log(\mathcal{N})$ denote the covering number and entropy respectively. The following lemma establishes upper bounds on the L_∞ covering number of neural networks.

Lemma 2.4.15 *[Lemma 3 in [66]] For any $\delta > 0$, the covering number of $\mathcal{F}^{\mathrm{DNN}}(L, N, S, B)$ (in sup-norm) satisfies*

$$\begin{aligned}&\log \mathcal{N}(\delta, \mathcal{F}^{DNN}(L, N, S, B), ||\cdot||_\infty)\\ &\leq 2L(S+1)\log(\delta^{-1}(L+1)(N+1)(B \vee 1)).\end{aligned}$$

For any $f \in \mathcal{F}$, let $G_f := \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \geq 0\}$ and $\mathcal{G}_{\mathcal{F}} := \{G_f : f \in \mathcal{F}\}$. Now we state our bracketing entropy bound for $\mathcal{G}^*$, which is $\mathcal{G}_{\mathcal{F}_n^*}$ in our teacher student setting.

Lemma 2.4.16 *Let $\mathcal{F}_n^*$ denote the teacher DNN family $\mathcal{F}^{\mathrm{DNN}}(L, N, S, B)$. Under assumptions (A1) to (A3), we have*

$$H_B\left(\delta, \mathcal{G}_{\mathcal{F}_n^*}, d_{\triangle}\right) \leq cSL\log(\delta^{-1} \vee n),$$

where $c > 0$ is some constant independent of the neural network architecture.

Proof Let the δ-covering set of $\mathcal{F}_n^*$ with respect to L_∞ norm be $\bar{\mathcal{F}}_\delta$, i.e., $\forall f_n^* \in \mathcal{F}_n^*$, there exists $\bar{f}_\delta \in \bar{\mathcal{F}}_\delta$ such that $\|f_n^* - \bar{f}_\delta\|_\infty \leq \delta$. Denote $\bar{f}_{\delta-} := \bar{f}_\delta - \delta$ and $\bar{f}_{\delta+} := \bar{f}_\delta + \delta$. Construct bracketing set $\tilde{\mathcal{G}}_\delta := \{(G_{\bar{f}_{\delta-}}, G_{\bar{f}_{\delta+}}) : \bar{f}_\delta \in \bar{\mathcal{F}}_\delta\}$. Notice that $\bar{f}_{\delta-}(\boldsymbol{x}) \leq f_n^*(\boldsymbol{x}) \leq \bar{f}_{\delta+}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$, which indicates $G_{\bar{f}_{\delta-}} \subset G_f \subset G_{\bar{f}_{\delta+}}$, i.e., $\tilde{\mathcal{G}}_\delta$ is a bracketing set of $\mathcal{G}_{\mathcal{F}_n^*}$.

Next, we show that $d_{p_n,q_n}(G_{\bar{f}_{\delta-}}, G_{\bar{f}_{\delta+}}) \leq c_0\delta$ for any teacher network $f_n^* \in \mathcal{F}_n^*$, where c_0 is the Lebesgue measure of the support union of p_n and q_n, i.e., $c_0 = \mathbb{Q}(\mathrm{supp}(p_n) \cup \mathrm{supp}(q_n))$. By assumption (A1), $c_0 < \infty$. For any $\boldsymbol{x} \in G_{\bar{f}_{\delta-}} \triangle G_{\bar{f}_{\delta+}}$, by definition we have $f(\boldsymbol{x}) + \delta \geq 0$ and $f(\boldsymbol{x}) - \delta < 0$, which suggests $|f(\boldsymbol{x})| \leq \delta$. Recall the teacher network setting that $p_n - q_n \in \mathcal{F}_n^*$. Then, we can conclude

$$\begin{aligned} d_{p_n,q_n}(G_{\bar{f}_{\delta-}}, G_{\bar{f}_{\delta+}}) &= \int_{G_{\bar{f}_{\delta-}} \triangle G_{\bar{f}_{\delta+}}} |p_n - q_n| \\ &= \int_{G_{\bar{f}_{\delta-}} \triangle G_{\bar{f}_{\delta+}}} |f| \leq c_0 \cdot \delta. \end{aligned}$$

Therefore, $\tilde{\mathcal{G}}_\delta$ is a $c_0\delta$-bracketing set of $\mathcal{G}_{\mathcal{F}_n^*}$ and

$$H_B\left(c_0\delta,\ \mathcal{G}_{\mathcal{F}_n^*},\ d_{p_n,q_n}\right) \leq \log|\tilde{\mathcal{G}}_\delta| \leq H\left(\delta,\ \mathcal{F}_n^*,\ \|\cdot\|_\infty\right).$$

Applying (2.12) and Lemma 2.4.15 yields

$$
\begin{aligned}
&H_B\left(\delta, \mathcal{G}_{\mathcal{F}_n^*}, d_{\triangle}\right) \\
&\leq H\left(\left(\frac{c_0^{-\frac{\kappa+1}{\kappa}} \cdot \delta}{b_1(\kappa)\left(T_n \wedge c_n^{-1/\kappa}\right)}\right)^{\frac{\kappa}{\kappa+1}}, \mathcal{F}_n^*, \|\cdot\|_\infty\right) \\
&\leq \frac{2\kappa L(S+1)}{\kappa+1} \log\left(\left(\frac{b_1(\kappa)\left(T_n \wedge c_n^{-1/\kappa}\right)}{c_0^{-\frac{\kappa+1}{\kappa}} \cdot \delta}\right)(L+1)(N+1)(B \vee 1)\right)
\end{aligned}
$$

By assumption (A2) we have $\log(LNB) \lesssim \log n$ and assumption (A3) indicates $\kappa = 1$ and $\log(T_n c_n) = o(\log n)$. The proof is complete. ■

Next, we present some lemmas that can take advantage of the obtained entropy bound and eventually take us to the proof of the excess risk convergence rate. So far, the presented lemmas are only concerned with the general case, i.e. set G^*, p, q, etc. that does not depend on n. However, in our teacher-student framework, the optimal set G_n^* is indexed by n as it's determined by the teacher network $\mathcal{F}_n^*$. In the remaining part of the proof, we will consider specifically for our teacher network case.

Our goal in classification is to estimate G_n^* by $\hat{G}_n = \operatorname{argmin}_{G \in \mathcal{G}_n} R_n(G)$, where $\mathcal{G}_n$ is some collection of sets associated with the student network $\mathcal{F}_n$ and

$$
R_n(G) = \frac{1}{2n} \sum_{\mathrm{i}=1}^{n} \left(\mathbb{I}\{\boldsymbol{x}_\mathrm{i} \in G | y_\mathrm{i} = 1\}(\boldsymbol{x}) + \mathbb{I}\{\boldsymbol{x}_\mathrm{i} \notin G | y_\mathrm{i} = -1\}(\boldsymbol{x})\right).
$$

Similar to Theorem 1 in [1], we have the following lemma regarding the upper bound on the rate of convergence.

Lemma 2.4.17 *Suppose $0 < \mathbb{Q}(\mathcal{X}) < \infty$ and let $\mathcal{G}_n^*$ be a collection of subsets of $\mathcal{X} \subset \mathbb{R}^d$. Define*

$$
\begin{aligned}
\mathcal{D}_n^{\mathcal{G}_n^*} = \{(p_n, q_n) : \mathbb{Q}\{\boldsymbol{x} \in \mathcal{X} : |p_n(\boldsymbol{x}) - q_n(\boldsymbol{x})| \leq t\} \leq c_n t^\kappa \text{ for } 0 \leq t \leq T_n, \\
\{\boldsymbol{x} \in \mathcal{X} : p_n(\boldsymbol{x}) \geq q_n(\boldsymbol{x})\} \in \mathcal{G}_n^*, p_n(\boldsymbol{x}), q_n(\boldsymbol{x}) \leq b_2 \text{ for } x \in \mathcal{X}\},
\end{aligned} \tag{2.13}
$$

where b_2 is an absolute constant. Let $\mathcal{G}_n$ be another class of subsets satisfying $\mathcal{G}_n^* \subset \mathcal{G}_n$. Suppose there exist positive constants $A_n > 0$ depending on n such that for any $\delta > 0$ small enough,

$$H_B(\delta, \mathcal{G}_n, d_\triangle) \leq A_n \log(1/\delta). \tag{2.14}$$

Then we have

$$\lim_{n\to\infty} \sup_{(p_n,q_n)\in\mathcal{D}_n^{\mathcal{G}_n^*}} \left(\frac{A_n \log^2 n}{n}\right)^{-\frac{\kappa+1}{\kappa+2}} \left(T_n \wedge c_n^{-1/\kappa}\right)^{\frac{\kappa}{\kappa+2}} \mathbb{E}[d_{p_n,q_n}(\hat{G}_n, G_n^*)] < \infty. \tag{2.15}$$

Proof For $(p_n, q_n) \in \mathcal{F}_n^{G_n^*}$, let $G_n^* = \{\boldsymbol{x} \in \mathcal{X} : p_n(\boldsymbol{x}) \geq q_n(\boldsymbol{x})\}$. For a given set $G \in \mathcal{X}$, let $h_G(\boldsymbol{x}) = \mathbb{I}\{\boldsymbol{x} \in G\}$. In particular, let $h_n^* = h_{G_n^*}$. Let $\|h\|_p^2 = \int h^2(\boldsymbol{x}) p(\boldsymbol{x}) \mathbb{Q}(d\boldsymbol{x})$. Since both p_n and q_n are bounded,

$$\begin{aligned} \|h_{G_n} - h_n^*\|_p^2 &= \int_{G_n \triangle G_n^*} p_n(\boldsymbol{x}) \mathbb{Q}(d\boldsymbol{x}) \leq b_2 d_\triangle(G_n, G_n^*), \\ \|h_{G_n} - h_n^*\|_q^2 &= \int_{G_n \triangle G_n^*} q_n(\boldsymbol{x}) \mathbb{Q}(d\boldsymbol{x}) \leq b_2 d_\triangle(G_n, G_n^*). \end{aligned} \tag{2.16}$$

Consider the random variable

$$V_n = -\sqrt{n}\, \frac{R_n(\hat{G}_n) - R_n(G_n^*) - \mathbb{E}(R_n(\hat{G}_n) - R_n(G_n^*))}{\sqrt{A_n d_\triangle(G_n^*, \hat{G}_n)} \log(1/d_\triangle(G_n^*, \hat{G}_n))}.$$

Since $\mathcal{G}_n^* \subset \mathcal{G}_n$, we have $R_n(\hat{G}_n) \leq R_n(G_n^*)$. Thus

$$\frac{\sqrt{n}\, \mathbb{E}(R_n(\hat{G}_n) - R_n(G_n^*))}{\sqrt{A_n d_\triangle(G_n^*, \hat{G}_n)} \log(1/d_\triangle(G_n^*, \hat{G}_n))} \leq V_n. \tag{2.17}$$

Note that

$$\begin{aligned} R_n(G_n) - R_n(G_n^*) =& \frac{1}{2n} \sum_{\mathrm{i}=1}^n \mathbb{I}_{\{y_\mathrm{i}=1\}} (h_n^* - h_{G_n})(\boldsymbol{x}_\mathrm{i}) \\ &+ \frac{1}{2n} \sum_{\mathrm{i}=1}^n \mathbb{I}_{\{y_\mathrm{i}=-1\}} (h_{G_n} - h_n^*)(\boldsymbol{x}_\mathrm{i}). \end{aligned}$$

Then V_n can be written as

$$V_n = \frac{(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=1\}}(h_{\hat{G}_n} - h_n^*)(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=1\}}(h_{\hat{G}_n} - h_n^*)(\boldsymbol{x}))}{\sqrt{A_n d_\triangle(G_n^*, \hat{G}_n)/n \log(1/d_\triangle(G_n^*, \hat{G}_n))} + \frac{(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=-1\}}(h_n^* - h_{\hat{G}_n})(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=-1\}}(h_n^* - h_{\hat{G}_n})(\boldsymbol{x}))}{\sqrt{A_n d_\triangle(G_n^*, \hat{G}_n)/n \log(1/d_\triangle(G_n^*, \hat{G}_n))}}.$$

Consider the event $E_n = \{d_\triangle(G_n^*, \hat{G}_n) > \sqrt{A_n/n}\}$ and let $\tilde{\mathcal{G}}_n = \{G \in \mathcal{G}_n : d_\triangle(G, G_n^*) > \sqrt{A_n/n}\}$. If E_n holds, then

$$\begin{aligned}
V_n =& -\sqrt{n}\, \frac{R_n(\hat{G}_n) - R_n(G_n^*) - \mathbb{E}(R_n(\hat{G}_n) - R_n(G_n^*))}{\sqrt{A_n d_\triangle(G_n^*, \hat{G}_n)} \log(1/d_\triangle(G_n^*, \hat{G}_n))} \\
\leq & \sup_{G_n \in \tilde{\mathcal{G}}_n} \sqrt{n}\, \frac{R_n(G_n^*) - R_n(G_n) - \mathbb{E}(R_n(G_n) - R_n(G_n^*))}{\sqrt{A_n d_\triangle(G_n^*, G_n)} \log(1/d_\triangle(G_n^*, G_n))} \\
\leq & \sup_{G_n \in \tilde{\mathcal{G}}_n} \frac{|(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=1\}}(h_{G_n} - h_n^*)(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=1\}}(h_{G_n} - h_n^*)(\boldsymbol{x}))|}{\sqrt{A_n d_\triangle(G_n^*, G_n)/n} \log(1/d_\triangle(G_n^*, G_n))} + \\
& \sup_{G_n \in \tilde{\mathcal{G}}_n} \frac{|(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=-1\}}(h_{G_n} - h_n^*)(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=-1\}}(h_{G_n} - h_n^*)(\boldsymbol{x}))|}{\sqrt{A_n d_\triangle(G_n^*, G_n)/n} \log(1/d_\triangle(G_n^*, G_n))} \\
\leq & \sup_{h_n \in \mathcal{H}_n} \frac{|(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=1\}}(h_n - h_n^*)(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=1\}}(h_n - h_n^*)(\boldsymbol{x}))|}{2b_2^{-1/2}\sqrt{A_n/n}\|h_n - h_n^*\|_p \log(\sqrt{b_2}/\|h_n - h_n^*\|_p)} + \\
& \sup_{h_n \in \mathcal{H}_n} \frac{|(1/2n)\sum_{i=1}^n \mathbb{I}_{\{y_i=1\}}(h_n - h_n^*)(\boldsymbol{x}_i) - \mathbb{E}(\mathbb{I}_{\{y=1\}}(h_n - h_n^*)(\boldsymbol{x}))|}{2b_2^{-1/2}\sqrt{A_n/n}\|h_n - h_n^*\|_q \log(\sqrt{b_2}/\|h_n - h_n^*\|_q)},
\end{aligned}$$

where $\mathcal{H}_n = \{h_n(\boldsymbol{x}) = \mathbb{I}\{\boldsymbol{x} \in G_n\} : G_n \in \mathcal{G}_n\}$. The last inequality follow from the fact that $\sqrt{x}\log(1/x)$ is strictly increasing when $x < 1$. Notice that h_n's are uniformly bounded by 1 and the L_2 norm squared of $h_{G_1} - h_{G_2}$ is $d_\triangle(G_1, G_2)$. Applying Lemma 2.4.9, we have

$$\mathbb{E}[V_n \mathbb{I}(E_n)] \leq C \tag{2.18}$$

for some finite constant C. Now we use this inequality to prove the main result. From (2.17), we know that

$$d_{p_n,q_n}(\hat{G}_n, G_n^*) \leq V_n (A_n/n)^{1/2} d_\triangle(G_n^*, \hat{G}_n)^{1/2} \log(1/d_\triangle(G_n^*, \hat{G}_n)),$$

which, together with Lemma 2.4.10, yields that

$$d_{p_n,q_n}(\hat{G}_n, G_n^*) \lesssim V_n (A_n/n)^{1/2} \left(T_n \wedge c_n^{-1/\kappa}\right)^{-\frac{\kappa}{2(\kappa+1)}} d_{p_n,q_n}(\hat{G}_n, G_n^*)^{\frac{\kappa}{2(\kappa+1)}} \cdot [\log(1/d_{p_n,q_n}(\hat{G}_n, G_n^*)) + \log(b_1(\kappa)(T_n \wedge c_n^{-1/\kappa}))],$$

which simplifies to be

$$d_{p_n,q_n}(\hat{G}_n, G_n^*) \lesssim V_n^{\frac{2\kappa+2}{\kappa+2}} \left(\frac{A_n \log^2 n}{n}\right)^{\frac{\kappa+1}{\kappa+2}} \left(T_n \wedge c_n^{-1/\kappa}\right)^{-\frac{\kappa}{\kappa+2}}.$$

where we used the fact that $d_{p_n,q_n}(\hat{G}_n, G_n^*) \gtrsim 1/n$. Therefore, under E_n, (2.18) implies that

$$\mathbb{E}[d_{p_n,q_n}(\hat{G}_n, G_n^*)] \lesssim \left(\frac{A_n \log^2 n}{n}\right)^{\frac{\kappa+1}{\kappa+2}} \left(T_n \wedge c_n^{-1/\kappa}\right)^{-\frac{\kappa}{\kappa+2}}.$$

On the other hand, under E_n^c, we have

$$d_{\triangle}(\hat{G}_n, G_n^*) \leq \sqrt{A_n/n}.$$

By Lemma 2.4.10 we know $d_{p,q}(\hat{G}_n, G_n^*)$ is also bounded by $\sqrt{A_n/n}$. Since $(\kappa+1)/(\kappa+2) \leq 1$, the rate under E_n dominates and the proof is complete.

■

Proof of Theorem 2.2.2

Proof First, we verify that the Tsybakov noise condition holds for $\kappa = 1$ in our setting. The proof is based on the fact that a ReLU network is piecewise linear and the number of linear pieces is quantifiable. Assumption (A3) implies (N_n) with $c_n, 1/T_n = O(\log n)^{m^* d^2 L_n^*}$ and $\kappa = 1$. In the case where p, q have disjoint support, obviously κ can be arbitrarily large.

Next, we consider the bracketing number of $\mathcal{G}_n$ defined via $\mathcal{F}_n$ that $\mathcal{G}_n = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \geq 0, f \in \mathcal{F}_n\}$. From Lemma 2.4.14 we have

$$\log \mathcal{N}_B(\delta, \mathcal{G}_n, d_{\triangle}) \lesssim N^{Ld^2} d^2 \left(Ld^2 \log(N) \vee \log(1/\delta)\right).$$

Thus, $A_n = O(N_n)^{d^2 L_n}$ as in (2.14) if $\delta \ll 1/N$. Recall from assumption (A2) and (A3) that $L_n = O(1)$, $N_n = O(\log n)^m$ and $1/T_n, c_n = O(\log n)^{m^* d^2 L_n^*}$. Applying Lemma 2.4.17 with $\kappa = 1$ we have that the excess risk has upper bound

$$\begin{aligned}
&\sup_{(p,q)\in\tilde{\mathcal{F}}_n^*} \mathbb{E}[\mathcal{E}(\hat{f}_n, C_n^*)] \\
&\lesssim \left(\frac{A_n \log^2 n}{n}\right)^{\frac{2}{3}} \left(T_n^{-1} \wedge c_n\right)^{\frac{1}{3}} \\
&\lesssim \left(\frac{1}{n}\right)^{\frac{2}{3}} (\log n)^{\frac{2}{3}(md^2 L_n + 2) + \frac{1}{3} m^* d^2 L_n^*}.
\end{aligned}$$

Corollary 2.2.3.1 easily follows from the fact that p, q having disjoint support implies $\kappa = \infty$ in (N_n). ■

2.4.7 Proof of Theorem 2.2.3

We will show that the lower bound holds in special case that (1) assumption (A3) satisfies $c_n, 1/T_n$ being absolute constants that doesn't depend on n; (2) instead of general ReLU neural network $f_n^* \in \mathcal{F}^*$, we consider a special structure where f_n^* is linear in one of the dimensions, reminiscent of the "boundary fragment" assumption. In this special case, we are able to show the best possible convergence rate already matches that in Theorem 2.2.2. For ease of notation, we omit the subscript n and write p_n, q_n as p, q if no confusion arises.

Proof The proof is very similar to that of Theorem 2.1.3. For completeness, the full proof is shown. Without loss of generality, let $\mathcal{X} = [0,1]^d$. Consider the "boundary fragment" setting and let $\tilde{\mathcal{G}}_n$ be a set defined by a ReLU network family $\tilde{\mathcal{F}}_n$ containing functions from $\mathbb{R}^{d-1}$ to $\mathbb{R}$:

$$\tilde{\mathcal{G}}_n = \{(x_1, \cdots, x_d) \in [0,1]^d : 0 \le x_{\mathrm{j}} \le h(\boldsymbol{x}_{-\mathrm{j}}), h \in \tilde{\mathcal{F}}_n, \mathrm{j} = 1, \cdots, d\},$$

where $\boldsymbol{x}_{-\mathrm{j}} = (x_1, \cdots, x_{\mathrm{j}-1}, x_{\mathrm{j}}, \cdots, x_d)$. Notice that if $h(\boldsymbol{x}_{-\mathrm{j}})$ is a ReLU network on $\mathbb{R}^{d-1}$, then $\tilde{h}(\boldsymbol{x}) = h(\boldsymbol{x}_{-\mathrm{j}}) - x_{\mathrm{j}}$ is a ReLU network on $\mathbb{R}^d$. Thus $\tilde{\mathcal{G}}_n$ is a subset of $\mathcal{G}_n$ which corresponds to the student network that

$$\mathcal{G}_n = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) > 0, f \in \mathcal{F}_n\} \tag{2.19}$$

Let $\tilde{G}_n$ denote the empirical 0-1 loss minimizer over $\tilde{\mathcal{G}}_n$. To show the lower bound, consider the subset of $\mathcal{D}^{\tilde{\mathcal{G}}_n}$ (2.13) that contains all pairs like (p, q_0), where $p \in \mathcal{F}_1, q_0$ will be specified later. Then,

$$\begin{aligned}\sup_{(p,q)\in\mathcal{D}^{\tilde{\mathcal{G}}_n}} \mathbb{E}d_{\triangle}(\tilde{G}_n, G^*) &\geq \sup_{(p,q_0):p\in\mathcal{F}_1} \mathbb{E}d_{\triangle}(\tilde{G}_n, G^*) \\ &\geq \mathbb{E}_{q_0}\left[\frac{1}{|\mathcal{F}_1|}\sum_{p\in\mathcal{F}_1}\mathbb{E}_p[d_{\triangle}(\tilde{G}_n, G^*)|\mathcal{D}_{q_0}]\right],\end{aligned}$$

where $\mathcal{F}_1$ is a finite set to be specified later, p, q_0 are the underlying densities for the two labels and $\mathcal{D}_{q_0}$ denotes all the data generated from q_0.

For ease of presentation, we first give the proof for the case $d = 2$ and then extend to general d. Let $\phi(t)$ be a piecewise linear function supported on $[-1, 1]$ defined as

$$\phi(t) = \begin{cases} t+1 & -1 < t \leq 0, \\ -t+1 & 0 < t < 1, \\ 0 & |t| \geq 1. \end{cases}$$

Rewrite ϕ as $\phi(t) = \sigma(t+1) - \sigma(t) + \sigma(-t+1) - \sigma(-t) - 2$, which is a one hidden layer ReLU neural network with 11 non-zero weights that are either 1 or -1. For $\boldsymbol{x} = (x_1, x_2) \in [0, 1]^2$, define

$$\begin{aligned}q_0(\boldsymbol{x}) =&(1 - \eta_0 - b_1)\mathbb{I}\{0 \leq x_2 < 1/2\} + \mathbb{I}\{1/2 \leq x_2 < 1/2 + \mathrm{e}^{-M}\} \\ &+ (1 + \eta_0 + b_2)\mathbb{I}\{1/2 + \mathrm{e}^{-M} \leq x_2 \leq 1\},\end{aligned}$$

where $M \geq 2$ is an integer to be specified later. Let $b_1 = c_2^{-1/\kappa}\mathrm{e}^{-M/\kappa}$ and $b_2 > 0$ be chosen such that q_0 integrates to 1 (so q_0 is a valid probability density).

For $\mathrm{j} = 1, 2, \cdots, M$ and $t \in [0, 1]$, let

$$\psi_{\mathrm{j}}(t) = \mathrm{e}^{-M}\phi\left(M\left[t - \frac{\mathrm{j}-1}{M}\right]\right).$$

Note that ψ_{j} is only supported on $[\frac{\mathrm{j}-1}{M}, \frac{\mathrm{j}}{M}]$. For any vector $\omega = (\omega_1, \cdots, \omega_M) \in \Omega := \{0,1\}^M$, define

$$b_\omega(t) = \sum_{\mathrm{j}=1}^{M} \omega_{\mathrm{j}} \psi_{\mathrm{j}}(t),$$

and

$$\begin{aligned} p_\omega(\boldsymbol{x}) =& 1 + \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} \mathbb{I}\{1/2 \le x_2 \le 1/2 + b_\omega(x_1)\} \\ & - b_3(\omega)\mathbb{I}\{1/2 + b_\omega(x_1) < x_2 \le 1\}, \end{aligned}$$

where $b_3(\omega) > 0$ is a constant depending on ω chosen such that $p_\omega(x)$ integrates to 1. Let $\mathcal{F}_1 = \{p_\omega : \omega \in \Omega\}$ and we will show that $(p_\omega, q_0) \in \mathcal{D}^{\tilde{\mathcal{G}}_n}$ for all $\omega \in \Omega$.

To this end, we need to verify that

(a) $p_\omega(\boldsymbol{x}) \le c_1$ for $\boldsymbol{x} \in [0,1]^2$;

(b) $\{\boldsymbol{x} \in \mathcal{X} : p_\omega(\boldsymbol{x}) \ge q_0(\boldsymbol{x})\} \in \mathcal{G}_n$;

(c) $\mathbb{Q}\{\boldsymbol{x} \in \mathcal{X} : |p_\omega(\boldsymbol{x}) - q_0(\boldsymbol{x})| \le \eta\} \le c_2 \eta^\kappa$.

For (a), since p_ω integrates to 1,

$$b_3(\omega) \le \max_{\{1/2 \le x_2 \le 1/2 + b_\omega(x_1)\}} \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} = O(\mathrm{e}^{-M/\kappa}).$$

Thus, $p_\omega(\boldsymbol{x}) \le c_1$ for a large enough M and some absolute constant c_1. For (b), notice that

$$\begin{aligned} \{\boldsymbol{x} : p_\omega(\boldsymbol{x}) \ge q_0(\boldsymbol{x})\} &= \{\boldsymbol{x} : 0 \le x_2 \le 1/2 + b_\omega(x_1)\} \\ &= \{\boldsymbol{x} \in [0,1]^2 : b_\omega(x_1) - \sigma(x_2) + 1/2 \ge 0\} \in \mathcal{G}_n, \end{aligned}$$

where the last inclusion follows from the definition of $\mathcal{G}_n$ (2.19) and the fact that $b_\omega(x_1) - \sigma(x_2) + 1/2$ is a ReLU neural network with one hidden layer, whose width and number of non-zero weights are both $O(M)$. Later we will see that $M = O(\log n)$, and thus the

constructed neural network satisfies all the size constraints in Theorem 2.2.2. For (c), it follows that

$$\begin{aligned}
&Q\{\boldsymbol{x} \in \mathcal{X} : |p_\omega(\boldsymbol{x}) - q_0(\boldsymbol{x})| \leq \eta\} \\
\leq &Q\{\boldsymbol{x} \in \mathcal{X} : 1/2 \leq x_2 \leq 1/2 + \mathrm{e}^{-M}, \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} \leq \eta\} \\
\leq &Q\{\boldsymbol{x} \in \mathcal{X} : 1/2 + \mathrm{e}^{-M} - c_2\eta^\kappa \leq x_2 \leq 1/2 + \mathrm{e}^{-M}\} \\
\leq &c_2\eta^\kappa.
\end{aligned}$$

Since the above (a)-(c) hold and by the definition of $\mathcal{D}^{\tilde{\mathcal{G}}_n}$ (2.13), we conclude that $(p_\omega, q_0) \in \mathcal{D}^{\tilde{\mathcal{G}}_n}$ for all $\omega \in \Omega$. We next establish how fast $S := |\mathcal{F}_1|^{-1} \sum_{p \in \mathcal{F}_1} \mathbb{E}_p[d_\triangle(\tilde{G}_n, G^*)|\mathcal{D}_{q_0}]$ can converge to zero. To this end, we use the Assouad's lemma stated in [57] which is adapted to the estimation of sets.

For $\mathrm{j} = 1, \cdots, M$ and $\omega = (\omega_1, \cdots, \omega_M) \in \Omega$, let

$$\begin{aligned}
\omega_{\mathrm{j}0} &= (\omega_1, \cdots, \omega_{\mathrm{j}-1}, 0, \omega_{\mathrm{j}+1}, \cdots, \omega_M) \\
\omega_{\mathrm{j}1} &= (\omega_1, \cdots, \omega_{\mathrm{j}-1}, 1, \omega_{\mathrm{j}+1}, \cdots, \omega_M)
\end{aligned}$$

For $\mathrm{i} = 0$ and $\mathrm{i} = 1$, let P_{ji} be the probability measure corresponding to the distribution of $x_1, \cdots, x_n$ when the underlying density is $f_{\omega_{\mathrm{ji}}}$. Denote the expectation w.r.t. P_{ji} as $\mathbb{E}_{\mathrm{ji}}$. Let

$$\begin{aligned}
\mathcal{D}_{\mathrm{j}} &= \{\boldsymbol{x} \in \mathcal{X} : 1/2 + b_{\omega_{\mathrm{j}0}}(x_1) < x_2 \leq 1/2 + b_{\omega_{\mathrm{j}1}}(x_1)\} \\
&= \{\boldsymbol{x} \in \mathcal{X} : b_{\omega_{\mathrm{j}0}}(x_1) < x_2 - 1/2 \leq b_{\omega_{\mathrm{j}0}}(x_1) + \psi_{\mathrm{j}}(x_1)\}.
\end{aligned}$$

Then

$$
\begin{aligned}
S &\geq 1/2 \sum_{\mathrm{j}=1}^{M} \mathbb{Q}(\mathcal{D}_{\mathrm{j}}) \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq 1/2 \sum_{\mathrm{j}=1}^{M} \int_0^1 \psi_{\mathrm{j}}(x_1) dx_1 \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq 1/2 \sum_{\mathrm{j}=1}^{M} \mathrm{e}^{-M} \int \phi(Mt) dt \int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \\
&\geq \frac{1}{4} \sum_{\mathrm{j}=1}^{M} \mathrm{e}^{-M} \int \phi(Mt) dt \left[1 - H^2(P_{10}, P_{11})/2\right]^n,
\end{aligned}
$$

where $H(\cdot, \cdot)$ denotes the Hellinger distance. Then it holds that

$$
\begin{aligned}
H^2(P_{10}, P_{11}) &= \int \left[\sqrt{f_{\omega_{10}}(\boldsymbol{x})} - \sqrt{f_{\omega_{11}}(\boldsymbol{x})}\right]^2 d\boldsymbol{x} \\
&\leq \int_0^1 \left\{ \int_{1/2}^{1/2+\psi_1(x_1)} \left[1 - \sqrt{1 + \left(\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right)^{1/\kappa}}\right]^2 dx_2 \right. \\
&\quad \left. + \int_{1/2}^{1} \left[\sqrt{1 - b_3(\omega_{10})} - \sqrt{1 - b_3(\omega_{11})}\right]^2 dx_2 \right\} dx_1 \\
&\leq \int_0^1 \int_{\mathrm{e}^{-M} - \psi_1(x_1)}^{\mathrm{e}^{-M}} \left[1 - \sqrt{1 + \left(\frac{v}{c_2}\right)^{1/\kappa}}\right]^2 dv dx_1 \\
&\quad + |b_3(\omega_{10}) - b_3(\omega_{11})|^2.
\end{aligned}
$$

We will analyze the last two terms. For the first term,

$$
\begin{aligned}
&\int_0^1 \int_{\mathrm{e}^{-M} - \psi_1(x_1)}^{\mathrm{e}^{-M}} \left[1 - \sqrt{1 + \left(\frac{v}{c_2}\right)^{1/\kappa}}\right]^2 dv dx_1 \\
&\leq \int_0^1 \int_{\mathrm{e}^{-M} - \psi_1(x_1)}^{\mathrm{e}^{-M}} \left(\frac{v}{c_2}\right)^{2/\kappa} dv dx_1 \\
&\leq \frac{\kappa c_2^{-2/\kappa}}{\kappa + 2} \int_0^1 \left(\mathrm{e}^{-M}\right)^{1+2/\kappa} - \left(\mathrm{e}^{-M} - \psi_1(x_1)\right)^{1+2/\kappa} dx_1 \\
&\leq \frac{\kappa c_2^{-2/\kappa}}{\kappa + 2} \left(\mathrm{e}^{-M}\right)^{1+2/\kappa} \int \left(1 - (1 - \phi(Mt))^{1+2/\kappa}\right) dt \\
&= O\left(\frac{1}{M} \mathrm{e}^{-M(1+2/\kappa)}\right).
\end{aligned}
$$

For the second term, notice that

$$\int_0^1 \int_{1/2}^{1/2+b_\omega(x_1)} \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} dx_2 dx_1 = b_3(\omega)\left[1/2 - b_\omega(x_1)\right]$$

which yields

$$\begin{aligned}
b_3(\omega_{11}) &= \frac{1}{1/2 - b_{\omega_{11}}(x_1)} \int_0^1 \int_{1/2}^{1/2+b_{\omega_{11}}(x_1)} \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} dx_2 dx_1 \\
&\leq \frac{M c_2^{-1/\kappa}}{1/2 - \mathrm{e}^{-M}} \int_0^1 \int_{\mathrm{e}^{-M}(1-\phi(Mx_1))}^{\mathrm{e}^{-M}} u^{1/\kappa} du dx_1 \\
&= \frac{M c_2^{-1/\kappa}}{(1/2 - \mathrm{e}^{-M})(1+1/\kappa)} \mathrm{e}^{-M(1+1/\kappa)} \int (1 - (1-\phi(Mt))^{1+1/\kappa}) dt \\
&\leq \frac{c_2^{-1/\kappa}}{(1/2 - \mathrm{e}^{-M})(1+1/\kappa)} \mathrm{e}^{-M(1+1/\kappa)} \\
&= O\left(\mathrm{e}^{-M(1+1/\kappa)}\right).
\end{aligned}$$

Hence, $|b_3(\omega_{11}) - b_3(\omega_{10})| = O\left(\mathrm{e}^{-M(1+1/\kappa)}\right)$. Unifying the above, we have

$$\begin{aligned}
H^2(P_{10}, P_{11}) &= O\left(\frac{1}{M}\mathrm{e}^{-M(1+2/\kappa)} \vee \mathrm{e}^{-M(2+2/\kappa)}\right) \\
&= O\left(\frac{1}{M}\mathrm{e}^{-M(1+2/\kappa)}\right).
\end{aligned}$$

Now choose M as the smallest integer such that

$$M \geq \frac{\kappa}{\kappa+2} \log n.$$

Then we have $H^2(P_{10}, P_{11}) \leq C^* n^{-1}(1 + o(1))$ for some constant C^* depending only on κ, c_2, ϕ, and

$$\int \min\{dP_{\mathrm{j}1}, dP_{\mathrm{j}0}\} \geq 1/2\left[1 - \frac{C^*}{2} n^{-1}(1+o(1))\right]^n \geq C_1^*$$

for n large enough and C_1^* is another absolute constant depending only on C^*. Thus for n large enough,

$$S \geq \frac{1}{4} C_1^* \mathrm{e}^{-M} \int \phi(t) dt \geq C_2^* n^{-\frac{\kappa}{\kappa+2}},$$

in which the constant C_2^* only depends on κ, c_2 and ϕ.

Combining all the results so far we get that

$$\liminf_{n\to\infty} \inf_{\tilde{G}_n} \sup_{(p,q)\in\mathcal{D}^{\tilde{\mathcal{G}}_n}} n^{\frac{\kappa}{\kappa+2}} \mathbb{E}[d_{\triangle}(\tilde{G}_n, G^*)] > 0,$$

which holds when $d = 2$. Using Lemma 2.4.10, we have

$$\liminf_{n\to\infty} \inf_{\tilde{G}_n} \sup_{(p,q)\in\mathcal{D}^{\tilde{\mathcal{G}}_n}} n^{\frac{\kappa+1}{\kappa+2}} \mathbb{E}[d_{p,q}(\tilde{G}_n, G^*)] > 0.$$

Using the same argument as in the proof of Theorem 2.2.2, we get $\kappa = 1$, which will give us the rate $2/3$.

The proof for general d can be derived similarly. We treat the last dimension x_d as x_2 in the $d = 2$ case and treat $\boldsymbol{x}_{-d} := (x_1, \cdots, x_{d-1})$ as x_1 in the $d = 2$ case. Define

$$\begin{aligned} q_0(\boldsymbol{x}) =& (1 - \eta_0 - b_1)\mathbb{I}\{0 \le x_d < 1/2\} + \mathbb{I}\{1/2 \le x_d < 1/2 + \mathrm{e}^{-M}\} \\ &+ (1 + \eta_0 + b_2)\mathbb{I}\{1/2 + \mathrm{e}^{-M} \le x_d \le 1\}, \end{aligned}$$

and

$$\begin{aligned} p_\omega(\boldsymbol{x}) =& 1 + \left[\frac{1/2 + \mathrm{e}^{-M} - x_2}{c_2}\right]^{1/\kappa} \mathbb{I}\{1/2 \le x_d \le 1/2 + \boldsymbol{b}_\omega(\boldsymbol{x}_{-d})\} \\ &- b_3(\omega)\mathbb{I}\{1/2 + \boldsymbol{b}_\omega(\boldsymbol{x}_{-d}) < x_d \le 1\}, \end{aligned}$$

where $\boldsymbol{b}_\omega(\boldsymbol{x}_{-d})$ is constructed similarly as a shallow ReLU neural network that

$$\boldsymbol{b}_\omega(\boldsymbol{x}_{-d}) = \sum_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}=1}^{M} \omega_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}} \psi_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}}(\boldsymbol{x}_{-d}),$$

where $\omega_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}}$ are binary $0, 1$ variables and

$$\psi_{\mathrm{j}_1,\cdots,\mathrm{j}_{d-1}}(\boldsymbol{x}_{-d}) = \mathrm{e}^{-M} \boldsymbol{\phi}\left(M\left[\boldsymbol{x}_{-d} - \left(\frac{\mathrm{j}_1 - 1}{M}, \cdots, \frac{\mathrm{j}_{d-1} - 1}{M}\right)\right]\right),$$

where $\boldsymbol{\phi}(\cdot)$ is a shallow ReLU neural network with input dimension $d-1$ satisfying the following conditions:

- $\boldsymbol{\phi} = 0$ outside $[-1,1]^d$ and $\boldsymbol{\phi} \leq 1$ on $[-1,1]^d$;
- $\max_{\boldsymbol{x}_{-d} \in [-1,1]^d} \boldsymbol{\phi}(\boldsymbol{x}_{-d}) \leq 1$ and $\boldsymbol{\phi}(\mathbf{0}) = 1$.

Such a construction is similar to the "spike" function in [95] and it requires $O(d^2)$ non-zero weights. The rest of the proof follows the $d = 2$ case. ■

2.4.8 Proof of Theorem 2.2.6

One important observation to be used in the proof is that the Bayes classifier under hinge loss is the same as that under 0-1 loss, i.e. $f_\phi^*(\boldsymbol{x}) = C^*(\boldsymbol{x})$. To show the upper bound on excess risk convergence rate, we utilize the following lemma from [58]. Let $\eta(\boldsymbol{x})$ denote the conditional probability of label 1 that $\eta(\boldsymbol{x}) = \mathbb{P}(y = 1|\boldsymbol{x})$.

Lemma 2.4.18 *[Theorem 6 of [58]] Let ϕ be the hinge loss. Assume* (N) *with the noise exponent $\kappa \in [0, \infty]$, and that following conditions (C1) through (C4) hold.*

(C1) For a positive sequence $a_n = O(n^{-a_0})$ as $n \to \infty$ for some $a_0 > 0$, there exists a sequence of function classes $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ such that $\mathcal{E}_\phi(f_n, f_\phi^*) \leq a_n$ for some $f_n \in \mathcal{F}_n$.

(C2) There exists a real valued sequence $\{F_n\}_{n\in\mathbb{N}}$ with $F_n \gtrsim 1$ such that $\sup_{f\in\mathcal{F}_n} \|f\|_\infty \leq F_n$.

(C3) There exists a constant $\nu \in (0, 1]$ such that for any $f \in \mathcal{F}_n$ and any $n \in \mathbb{N}$,

$$\mathbb{E}\left[\left\{\phi(Yf(\boldsymbol{X})) - \phi(Yf_\phi^*(\boldsymbol{X}))\right\}^2\right] \leq C_2 F_n^{2-\nu}\{\mathcal{E}_\phi(f, f_\phi^*)\}^\nu$$

for a constant $C_2 > 0$ depending only on ϕ and $\eta(\cdot)$.

(C4) For a positive constant $C_3 > 0$, there exists a sequence $\{\delta_n\}_{n\in\mathbb{N}}$ such that

$$H_B(\delta_n, \mathcal{F}_n, \|\cdot\|_2) \leq C_3 n \left(\frac{\delta_n}{F_n}\right)^{2-\nu},$$

for $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ in (C1), $\{F_n\}_{n\in\mathbb{N}}$ in (C2), and ν in (C3).

Let $\epsilon_n^2 \asymp \max\{a_n, \delta_n\}$. *Assume that* $n^{1-\iota}(\epsilon_n^2/F_n)^{(\kappa+2)/(\kappa+1)} \gtrsim 1$ *for an arbitrarily small constant* $\iota > 0$. *Then, the empirical* ϕ*-risk minimizer* $\hat{f}_{\phi,n}$ *over* $\mathcal{F}_n$ *satisfies*

$$\mathbb{E}\left[\mathcal{E}(\hat{f}_{\phi,n}, C^*)\right] \lesssim \epsilon_n^2.$$

In Lemma 2.4.18, condition (C1) guarantees the approximation error of f_n to f_ϕ^* to be sufficiently small. For condition (C3), we introduce the following lemma, which is reminiscent of Lemma 2.4.10 in the sense that it characterizes the relationship between the $\mathcal{E}_\phi(f, f_\phi^*)$ and the some other distance measure between f and f_ϕ^*.

Lemma 2.4.19 (Lemma 6.1 of [96]) *Assume* (N) *with the Tsybakov noise exponent* $\kappa \in [0, \infty]$. *Assume* $\|f\|_\infty \le F$ *for any* $f \in \mathcal{F}$. *Under the hinge loss* ϕ, *for any* $f \in \mathcal{F}$,

$$\begin{aligned}&\mathbb{E}\left[\left(\phi(Yf(\boldsymbol{x})) - \phi(Yf_\phi^*(\boldsymbol{x}))\right)^2\right] \\ &\le C_{\eta,\kappa}(F+1)^{(\kappa+2)/(\kappa+1)}\left(\mathbb{E}\left[\phi(Yf(\boldsymbol{x})) - \phi(Yf_\phi^*(\boldsymbol{x}))\right]\right)^{\kappa/\kappa+1},\end{aligned}$$

where $C_{\eta,\kappa} = \left(\|(2\eta-1)^{-1}\|_{\kappa,\infty}^\kappa + 1\right)\mathbb{I}(\kappa > 0) + 1$ *and* $\|(2\eta-1)^{-1}\|_{\kappa,\infty}^\kappa$ *is defined by*

$$\|(2\eta-1)^{-1}\|_{\kappa,\infty}^\kappa = \sup_{t>0}\left(t^\kappa \Pr\left(\{\boldsymbol{x} : |(2\eta(\boldsymbol{x})-1)^{-1}| > t\}\right)\right).$$

Proof of Theorem 2.2.6

Proof The lower bound directly follows from Theorem 2.2.3, as the constructed ReLU neural network in the proof also satisfy assumption ($A2_\phi$).

For the upper bound on the convergence rate, we utilize Lemma 2.4.18 and check the conditions (C1) through (C4). Since the student network is larger than the teacher, (C1) and (C2) trivially hold with arbitrarily small a_n and $F_n = O(\log n)$ as assumed. To apply Lemma 2.4.19, notice that $C_{\eta,\kappa} = O(c_n) = O(\log n)^{m^* d^2 L_n^*}$ by assumption (A3) and $F = O(\log n)$, we have (C3) holds for $\nu = \kappa/(\kappa+1) + \epsilon_n$, where $\epsilon_n = (2 + m^* d^2 L_n^*)\log\log n/\log n$. The term ϵ_n is to deal with the fact that $C_{\eta,\kappa}$ can also diverge at an $O(\log n)^{m^* d^2 L_n^*}$ rate.

For (C4), by Lemma 2.4.15,

$$\begin{aligned}
&\log \mathcal{N}(\delta_n, \mathcal{F}^{\mathrm{DNN}}(L_n, N_n, S_n, B_n, F_n), \|\cdot\|_\infty) \\
&\leq 2L_n(S_n+1)\log\left(\delta_n^{-1}(L_n+1)(N_n+1)(B_n\vee 1)\right) \\
&\lesssim (\log n)^{2m+2}\log\left(\delta_n^{-1}\vee \log^m(n)\right).
\end{aligned}$$

Therefore, (2.4.18) implies that (C3) is satisfied if we choose δ_n with

$$\delta_n^{\frac{\kappa+2}{\kappa+1}} \gtrsim \frac{(\log n)^{2m+2+(\kappa+2)/(\kappa+1)+2+m^*d^2L_n^*+1}}{n},$$

which can be satisfied by choosing

$$\delta_n = \left(\frac{(\log n)^{2m+m^*d^2L_n^*+7}}{n}\right)^{\frac{\kappa+1}{\kappa+2}}.$$

Similar to the proof of Theorem 2.2.2, the Tsybakov exponent $\kappa = 1$. Thus, by Lemma 2.4.18 with $\epsilon_n^2 = \delta_n$, the proof of Theorem 2.2.6 is completed. ■

2.4.9 Proof of Theorem 2.2.1

In this section, Assumption (A3) will be examined in the setting that the teacher network f_n^* has random weights. We will argue that with probability at least $1-\delta$, f_n^* will satisfy assumption (A3) with $T_n = A(\delta)/(\log n)^{m^*d^2L_n}$ and $c_n = B(\delta)(\log n)^{m^*dL_n^*(L_n^*+1)}$, where $A(\delta), B(\delta)$ are constants depending only on δ and the distribution of the random weights, e.g. normal, truncated normal, etc. Hence, the results which assume Assumption (A3) will hold with high probability.

A Toy Case To illustrate the intuition, consider the case where $d = 1$ and f_n^* is the following one hidden-layer ReLU neural network

$$f_n^*(x) = \sum_{\mathrm{j}=1}^{N_n^*} w_{2\mathrm{j}}\sigma(w_{1\mathrm{j}}x + b_{\mathrm{j}}) + b, \quad x \in [0,1], \tag{2.20}$$

with $L_n^* = 1$, $N_n^* = O(\log n)$ and $w_{1\mathrm{j}}, w_{2\mathrm{j}}, b_{\mathrm{j}}, b$ are i.i.d. standard Gaussian. Since all the weights are almost surely nonzero, we omit the zero weight cases for the analysis. Let $p_{\mathrm{i}} = (u_{\mathrm{i}}, v_{\mathrm{i}})$, $\mathrm{i} = 1, 2, \ldots, s$, denote the active pieces of (2.20). By Lemma 2.4.12, we know that $s = O(\log n)$. For each p_{i}, define the following quantities:

1. k_{i} = the slope of $f_n^*(x)$ on $x \in p_{\mathrm{i}}$;

2. $t_{\mathrm{i}} = \max_{x \in p_{\mathrm{i}}} f_n^*(x) \wedge \max_{\boldsymbol{x} \in p_{\mathrm{i}}} -f_n^*(x)$.

See Figure 2.9 for an illustration. Then, assumption (A3) is satisfied if

$$\min_{\mathrm{i}}\{|k_{\mathrm{i}}|\} = \Omega(1/\log^2 n) \quad \text{and} \quad \min_{\mathrm{i}}\{t_{\mathrm{i}}\} = \Omega(1/\log n). \tag{2.21}$$

Next we will rigorously examine (2.21).

From (2.20), each k_{i} can be expressed as $\sum_{\mathrm{j} \in J} w_{1\mathrm{j}} w_{2\mathrm{j}}$ for some index set $J \subset \{1, 2, \cdots, N_n^*\}$. Notice f_n^* has at most $N_n^* + 1$ pieces and denote the corresponding index sets to be $J_0, J_1, \cdots, J_{N_n^*}$. As a result, $\min_{1 \le \mathrm{i} \le N_n^*}\{|k_{\mathrm{i}}|\} = \min_{J_0, \ldots, J_{N_n^*}}\{|\sum_{\mathrm{i} \in J_{\mathrm{i}}} w_{1\mathrm{j}} w_{2\mathrm{j}}|\}$. Since $w_{1\mathrm{j}}, w_{2\mathrm{j}}$ are i.i.d. standard Gaussian, we have

$$\begin{aligned}
&\mathbb{P}(\min_{0 \le \mathrm{i} \le N_n^*}\{|k_{\mathrm{i}}|\} < k) = \mathbb{P}(\min_{0 \le \mathrm{i} \le N_n^*}\{|\sum_{\mathrm{j} \in J_{\mathrm{i}}} w_{1\mathrm{j}} w_{2\mathrm{j}}|\} < k) \\
&\le \sum_{\mathrm{i}=0}^{N_n^*} \mathbb{P}\left(\left|\sum_{\mathrm{j} \in J_{\mathrm{i}}} w_{1\mathrm{j}} w_{2\mathrm{j}}\right| < k\right) \\
&\le (N_n^* + 1)\mathbb{P}\left(\sqrt{|w_{11} w_{21}|} < \sqrt{k}\right) \\
&\le 2(N_n^* + 1)\sqrt{k}.
\end{aligned}$$

By choosing $k = \left(\frac{\delta}{2(N_n^*+1)}\right)^2$, we have $\min_{1 \le \mathrm{i} \le N_n^*}\{|k_{\mathrm{i}}|\} = \Omega(1/\log^2 n)$ with probability at least $1 - \delta$.

On the other hand, for any $\mathrm{i} = 1, \ldots, s$, $t_{\mathrm{i}} = |f_n^*(x_{h_{\mathrm{i}}})|$ for some $h_{\mathrm{i}} \in \{1, \cdots, N_n^*\}$, where $x_{h_{\mathrm{i}}} = -b_{h_{\mathrm{i}}}/w_{1h_{\mathrm{i}}}$. Hence

$$\min_{1 \le \mathrm{i} \le s}\{t_{\mathrm{i}}\} \ge \min_{1 \le \mathrm{j} \le N_n^*}\{|f_n^*(x_{\mathrm{j}})|\}.$$

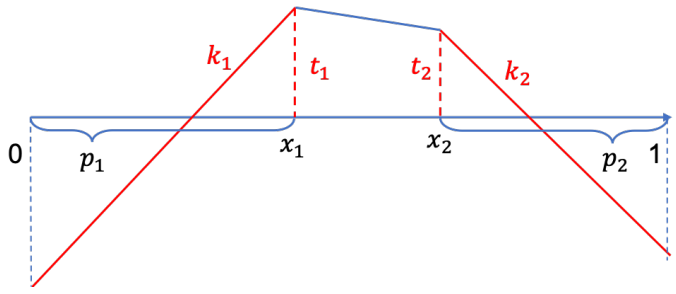


Figure 2.9. Example of a ReLU function in $[0, 1]$. There are two active pieces p_1, p_2. On each active piece, $t_{\mathrm{i}}.k_{\mathrm{i}}$ are illustrated in color red.

Let $W_1 = \{w_{1\mathrm{j}}, b_{\mathrm{j}}\}_{\mathrm{j}=1}^{N_n^*}$. Then, $f_n^*(x_{\mathrm{i}}) \,|\, W_1 \sim N(0, \sigma_{x_{\mathrm{i}}}^2)$, where $\sigma_{x_{\mathrm{i}}}^2$ has an expression of $\sum_{\mathrm{j}=1}^{N_n^*} \sigma(w_{1\mathrm{j}} x_{\mathrm{i}} + b_{\mathrm{j}})^2 + 1$. Hence, for any $t > 0$,

$$\begin{aligned}\mathbb{P}(\min_{\mathrm{i} \le N_n^*}\{|f_n^*(x_{\mathrm{i}})|\} < t \,|\, W_1) &\le \sum_{\mathrm{i}=1}^{N_n^*} \mathbb{P}(|f_n^*(x_{\mathrm{i}})| < t \,|\, W_1) \\ &= N_n^* \mathbb{P}(|f_n^*(x_{\mathrm{i}})| < t \,|\, W_1) \le N_n^* \left(\frac{t}{\sigma_{x_{\mathrm{i}}}}\right).\end{aligned}$$

Since $\sigma_{x_{\mathrm{i}}} \ge 1$, by taking $t = \delta / N_n^*$, we have that with probability at least $1 - \delta$, $\min_{\mathrm{i}}\{t_{\mathrm{i}}\} \ge t$ and $t = \Omega(1/\log n)$. Therefore, (2.21) holds with high probability, so that assumption (A3) holds by setting $1/c_n = \min_{\mathrm{i}}\{|k_{\mathrm{i}}|\}$ and $T_n = \min_{\mathrm{i}}\{t_{\mathrm{i}}\}$, which are both in the order of $\Omega(1/\log n)$.

General Case Now we consider the general case $d > 1$ and $L_n^* > 1$. The teacher network has an expression

$$f_n^*(\boldsymbol{x}) = \boldsymbol{W}^{(L_n^*+1)} \sigma_{(\boldsymbol{W}^{(L_n^*)}, \boldsymbol{b}^{(L_n^*)})} \circ \cdots \circ \sigma_{(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)})}(\boldsymbol{x}) + \boldsymbol{b}^{(L_n^*+1)}, \boldsymbol{x} \in [0, 1]^d.$$

Let $N_n^* = O(\log n)^{m^*}$. By Lemma 2.4.12, f_n^* has linear pieces $p_1, \ldots, p_s$ for $s = O(\log n)^{m^* L_n^* d}$. Let $\{\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{v_s}\}$ be the collection of vertices of $\{p_1, \ldots, p_s\}$. We call such $\boldsymbol{x}_{\mathrm{i}} \in \mathbb{R}^d$ a *piece vertex* and it's not the same as the vertex of $\{\boldsymbol{x} \in \mathcal{X} : f_n(x) \ge 0\}$, which is closely examined in the proof of Lemma 2.4.14. The following lemma states that $v_s = O(\log n)^{m^* L_n^* d^2}$.

Lemma 2.4.20 *Let f be a ReLU neural network with d-dimensional input, L hidden layers and width N for every layer. Then, $v_s = O(N)^{Ld^2}$.*

Proof Recall that $\boldsymbol{w}_{\mathrm{i}}^{(l)}$ and $b_{\mathrm{i}}^{(l)}$ for $\mathrm{i}=1,\ldots,N$, $1\le l\le L$ are the weight vectors and biases on the l-th hidden layer. For $\mathrm{i}=1,\ldots,N$, define

$$f_{\mathrm{i}}^{(l-1)}(\boldsymbol{x})=\boldsymbol{w}_{\mathrm{i}}^{(l)}\sigma_{(\boldsymbol{W}^{(l-1)},\boldsymbol{b}^{(l-1)})}\circ\cdots\circ\sigma_{(\boldsymbol{W}^{(1)},\boldsymbol{b}^{(1)})}(\boldsymbol{x})+b_{\mathrm{i}}^{(l)},$$

which maps $\mathbb{R}^d\to\mathbb{R}$. We can rewrite f as

$$f(\boldsymbol{x})=\sum_{\mathrm{i}=1}^{N}w_{\mathrm{i}}^{(L+1)}\sigma(f_{\mathrm{i}}^{(L-1)}(\boldsymbol{x}))+b^{(L+1)},\tag{2.22}$$

In other words, $f_{\mathrm{i}}^{(L-1)}(\boldsymbol{x})$ represents the inputs to the i-th ReLU unit in the last hidden layer of f and itself is an $(L-1)$-hidden-layer ReLU neural network.

The key idea of the proof is by induction. Notice that the piece vertices of f can only come from the following two ways: Type I: The piece vertices of $f_1^{(L-1)},f_2^{(L-1)},\ldots,f_N^{(L-1)}$, in whose local neighbourhoods, the ReLU units in the last layer doesn't change sign; Type II: By activations of the ReLU unit in the last layer. i.e. $f_{\mathrm{i}}^{(L-1)}(\boldsymbol{x})=0$ for some $\mathrm{i}=1,\ldots,N$. Let $V(l)$ be the maximum number of piece vertices of an l-hidden-layer ReLU neural network with width N and let $U(l)$ be the maximum number of Type II piece vertices created at layer l. Then for $1<l\le L$ we have

$$V(l)\le NV(l-1)+U(l).\tag{2.23}$$

For $U(l)$, the key is to connect the Type II piece vertices of f to the vertices of $\{\boldsymbol{x}\in\mathcal{X}:f_{\mathrm{i}}^{(L-1)}(\boldsymbol{x})\ge 0\}$, which has been extensively studied in Lemma 2.4.14. To this end, we define another quantity. On the i-th ReLU unit in the l-th hidden layer, let $R_{\mathrm{i}}^{(l)}:=\{\boldsymbol{x}\in\mathcal{X}:f_{\mathrm{i}}^{(l)}(\boldsymbol{x})=0\}$, which consists of $(d-1)$-dimensional hyperplane segments. To be specific, denote all the active pieces of $f_{\mathrm{i}}^{(l)}(\boldsymbol{x})$ to be $\{p_{\mathrm{ij}}^{(l)}:\mathrm{j}=1,\ldots,s_{\mathrm{i}}^{(l)}\}$, where $s_{\mathrm{i}}^{(l)}=O(N)^{(l-1)d}$ according to Lemma 2.4.12 for any $1\le\mathrm{i}\le N$. On each active piece $p_{\mathrm{ij}}^{(l)}$, denote

$$h_{\mathrm{ij}}^{(l)}=\{(\boldsymbol{x},f_{\mathrm{i}}^{(l)}(\boldsymbol{x})):\boldsymbol{x}\in p_{\mathrm{ij}}^{(l)}\}\cap\{(\boldsymbol{x},0):\boldsymbol{x}\in p_{\mathrm{ij}}^{(l)}\},$$

which is part of a $(d-1)$-dimensional hyperplane. Then we have $R_{\mathrm{i}}^{(l)} = \{h_{\mathrm{ij}}^{(l)} : \mathrm{j} = 1, \ldots, s_{\mathrm{i}}^{(l)}\}$, a collection of $(d-1)$-dimensional hyperplane segments. Let $R^{(l)} = \cup_{\mathrm{i}=1}^{N} R_{\mathrm{i}}^{(l)}$, which corresponds to the piece boundaries of f^{l+1}.

By definition, all Type II pieces vertices must reside in at least one of the the activation sets ($z = 0$ in $\sigma(z)$) of the ReLU units in the last layer. $R^{(L)}$ contains all such activation sets for the last hidden layer, i.e. for any $h \in R$, there exists $1 \leq \mathrm{i} \leq N$ such that $f_{\mathrm{i}}(\boldsymbol{x}) = 0,\ \forall \boldsymbol{x} \in h$. The Type II pieces vertices are jointly determined by such activation sets and the piece boundary of f_{i}'s (dimension $d-1$), i.e. $R_{\mathrm{i}}^{(L-2)}$. Therefore, the total number of such piece vertices can be bounded by

$$U(l) \leq \binom{\left|R^{(l-1)}\right| + \left|R^{(l-2)}\right|}{d} = O(N)^{(l-1)d^2+d},$$

where $\left|R^{(l)}\right|$ denotes the number of elements in $R^{(l)}$, which is bounded by $O(N)^{(l-1)d+1}$.

For $V(L)$, we first conclude that $V(1) = O(N^d)$. For a 1-hidden layer ReLU network, the decision boundary of every ReLU unit is a $(d-1)$-dimension hyperplane, i.e. $\{\boldsymbol{x} : \boldsymbol{w}_1\boldsymbol{x}+b_1 = 0\}$. The maximum number of piece vertices is bounded by $\binom{N}{d} = O(N^d)$. Then, (2.23) can be repeatedly broken down as

$$\begin{aligned}
V(L) &\leq NV(L-1) + U(L) \\
&\leq N^2V(L-2) + NU(L-1) + U(L) \\
&\leq \cdots \\
&\leq N^{L-1}V(1) + \sum_{l=0}^{L-1} N^l U(L-l) \\
&= O\left(N^{L-1+d}\right) + O\left(\sum_{l=0}^{L-1} N^{(L-l-1)d^2+d+l}\right) \\
&= O\left(N^{(L-1)d^2+d}\right) = O\left(N^{Ld^2}\right).
\end{aligned}$$

■

As an extension to the toy case, for any $1 \leq \mathrm{i} \leq s$, define

1. $k_{\mathrm{i}} = \min_{\mathrm{j}=1,\ldots,d} \left\{\left|\frac{\partial f_n^*(\boldsymbol{x})}{\partial x_{\mathrm{j}}}\right| : \boldsymbol{x} \in p_{\mathrm{i}}\right\}$;

2. $t_0 = \min_{1\le \mathrm{i}\le v_s}\{|f_n^*(\boldsymbol{x}_\mathrm{i})|\}$.

That is, k_i is the minimal absolute values of the directional derivatives of f_n^* on piece p_i. Assumption (A3) is satisfied if the following holds:

$$\min_{1\le \mathrm{i}\le s}\{k_\mathrm{i}\}, t_0 = \Omega(\log n)^{m^* d^2 L_n^{*2}}. \tag{2.24}$$

We will check (2.24). The partial derivative of $f_n^*(\boldsymbol{x})$ for $\boldsymbol{x}\in p_\mathrm{i}$ can be expressed as sum of the product of the random weights, i.e. $\sum_J \prod_{l=1}^{L_n^*+1} w_{J_l}^{(l)}$, where $w_{J_l}^{(l)}$ is an element from $\boldsymbol{W}^{(l)}$ and J is some collections of L_n^*+1 index pairs, e.g. $\{(\mathrm{i}_l,\mathrm{j}_l)\}_{l=1}^{L_n^*+1}$. There are s pieces and denote the corresponding index sets by $J_1, J_2, \cdots, J_s$. Then we have

$$\min_{1\le \mathrm{i}\le s}\{k_\mathrm{i}\} = \min_{1\le \mathrm{i}\le s}\left|\sum_{J=J_\mathrm{i}}\prod_{l=1}^{L_n^*+1} w_{J_l}^{(l)}\right|,$$

Since all the weights are i.i.d. from standard normal distribution, we have for any index set J that

$$\mathbb{P}\left(\left|\sum_{J}\prod_{l=1}^{L_n^*+1} w_{J_l}^{(l)}\right| < k\right) \le \mathbb{P}\left(\left|\prod_{l=1}^{L_n^*+1} w_{1,1}^{(l)}\right| < k\right).$$

Therefore,

$$\begin{aligned}
\mathbb{P}(\min_{1\le \mathrm{i}\le s}\{k_\mathrm{i}\} < k) &\le \sum_{J=J_1}^{J_s}\mathbb{P}\left(\left|\sum_{J}\prod_{l=1}^{L_n^*+1} w_{J_l}^{(l)}\right| \le k\right) \\
&\le s\ \mathbb{P}\left(\left|\prod_{l=1}^{L_n^*+1} w_{1,1}^{(l)}\right|^{1/(L_n^*+1)} < k^{1/(L_n^*+1)}\right) \\
&\lesssim s\ k^{1/(L_n^*+1)}.
\end{aligned}$$

By taking

$$k_0 = \Omega\left(\frac{\delta}{(N_n^*)^{L_n^* d}}\right)^{L_n^*+1},$$

we have that with probability at least $1-\delta$, $\min_{1\le \mathrm{i}\le s}\{k_\mathrm{i}\} \ge k_0$ and $k_0 = \Omega(1/\log n)^{m^* L_n (L_n^*+1) d}$.

On the other hand, for any t_{i}, there exist $\mathrm{j} = 1, \ldots, v_s$ such that $t_{\mathrm{i}} = f_n^*(\boldsymbol{x}_{\mathrm{j}})$. Hence

$$\min_{\mathrm{i}=1,\ldots,v_s} \{t_{\mathrm{i}}\} \geq \min_{\mathrm{j}=1,\ldots,v_s} \{|f_n^*(\boldsymbol{x}_{\mathrm{j}})|\}.$$

Let $\boldsymbol{W}_{-L_n^*} := \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}_{l=1}^{L_n^*}$. Then we have $f_n^*(\boldsymbol{x}_{\mathrm{j}}) \,|\, \boldsymbol{W}_{-L_n^*} \sim N(0, \sigma^2_{\boldsymbol{x}_{\mathrm{j}}})$, where $\sigma^2_{\boldsymbol{x}_{\mathrm{j}}}$ depends on $\boldsymbol{W}_{-L_n^*}$ and $\sigma^2_{\boldsymbol{x}_{\mathrm{j}}} \geq 1$ that

$$\sigma^2_{\boldsymbol{x}_{\mathrm{j}}} \,|\, \boldsymbol{W}_{-L_n^*} := \sum_{\mathrm{i}=1}^{N_{L_n^*}} \sigma^2_{\mathrm{i}}(\boldsymbol{x}_{\mathrm{j}}) + 1,$$

which is reminiscent of (2.22) and $N_{L_n^*}$ is the width of the last layer and $\sigma_{\mathrm{j}}(\cdot)$'s are outputs (post-activations) from the last layer given $\boldsymbol{W}_{-L_n^*}$. Therefore, for any $t > 0$, we have

$$\begin{aligned}
\mathbb{P}(\min_{1\leq \mathrm{j} \leq v_s} \{|f_n^*(\boldsymbol{x}_{\mathrm{j}})|\} < t \,|\, \boldsymbol{W}_{-L_n^*}) &\leq \sum_{\mathrm{j}=1}^{v_s} \mathbb{P}(|f_n^*(\boldsymbol{x}_{\mathrm{j}})| < t \,|\, \boldsymbol{W}_{-L_n^*}) \\
&= v_s \mathbb{P}(|f_n^*(\boldsymbol{x}_1)| < t \,|\, \boldsymbol{W}_{-L_n^*}) \\
&\leq v_s \left(\frac{t}{\sigma_{x_{\mathrm{i}}}}\right) \leq t(N_n^*)^{d^2 L_n^*}.
\end{aligned}$$

Thus by taking $t = \delta/(N_n^*)^{d^2 L_n^*}$, we have that with probability at least $1 - \delta$, $\min_{\mathrm{i}}\{t_{\mathrm{i}}\} \geq t$ and $t = \Omega(1/\log n)^{m^* d^2 L_n^*}$. Therefore, (2.24) holds. That is to say, when $d \geq 2$, with high probability, Assumption (A3) holds in which $c_n, 1/T_n = O(\log n)^{m^* d^2 L_n^{*2}}$.

Notice that the probability arguments used in this section don't rely on Gaussian distribution. As long as all weights are i.i.d. with distribution that doesn't have a point mass at 0, our claim holds.

3. STATISTICAL OPTIMALITY WITH ALGORITHMIC GUARANTEES

In the previous section, we have extended the nonparametric theory of deep learning by establishing statistical optimality of DNNs under various new settings. However, this type of results has two limitations. Firstly, they only apply to the empirical risk minimizer or some specially constructed DNNs without any algorithmic guarantee. Secondly, the theoretical analysis relies on delicate complexity control of the DNN family and cannot handle overparametrization, which is very common in practice. Therefore, statistical optimality without algorithmic guarantees are less helpful in understanding deep neural network models.

Recently, many efforts have been devoted to provable deep learning methods with algorithmic guarantees, particularly training overparametrized neural networks by gradient descent (GD) or other gradient-based optimization. It has been shown that with enough overparametrization, e.g., neural network width tends to infinity, training DNN resembles a kernel method with a specific kernel called as "neural tangent kernel" (NTK) [23]. In the NTK regime, GD can provably minimize the training error to zero in both regression [16], [17], [97], [98] and classification [99]–[101] settings. Corresponding generalization error bounds are developed to ensure prediction performance on unseen data. However, a closer inspection of these generalization results reveals that they only hold under the noiseless assumption, i.e., the response variable is deterministic given the explanatory variables. For overparametrized neural networks, the training loss can be minimized to zero so that the generalization error equals the population loss, which cannot be zero in the presence of noises. As random noises are ubiquitous in the real world, theoretical guarantees and provable learning algorithms that take into account of random noises are much needed in practice.

In contrast, classic nonparametric statistics literature demonstrate that in the presence of noises, the L_2 estimation error can still go to zero with possibly optimal rates as established in [48]. To further investigate how overparametrized neural networks trained via GD work and how well they can learn the underlying true function with noisy data, we consider the classic nonparametric regression setting (1.2). In this section, we consider neural network estimators $\hat{f}$ produced by overparametrized one-hidden-layer ReLU neural networks, where

the number of neurons can be much larger than the sample size, and investigate how fast the L_2 estimation error $\|\hat{f} - f^*\|_2$ converges to zero as sample size grows. The main contributions in this section are:

- We prove that overparametrized one-hidden-layer ReLU neural networks trained using GD do not recover the true function in the classic nonparametric regression setting (1.2), i.e., the L_2 estimation error is bounded away from zero as sample size goes to infinity. To predict well on unseen data, a delicate early stopping rule has to be deployed.

- We analyze the ℓ_2-regularized GD trajectory and show that the ℓ_2 penalty on network weights amounts to penalizing the reproducing kernel Hilbert space (induced by NTK) norm of the associated neural network. With ℓ_2 regularization, overparametrized neural network trained by GD resembles the solution of kernel ridge regression.

- We further prove that by adding proper ℓ_2 regularization, overparametrized neural network trained by GD achieves the *minimax-optimal* L_2 convergence rate $n^{-d/(4d-2)}$, in recovering the ground truth in (1.2).

The correspondence between overparametrized neural network trained by ℓ_2-regularized GD and kernel ridge regression is nontrivial and technically challenging. In spite of the well-established equivalence between NTK and infinite-width DNN trained by GD, there is a huge technical gap for finite-width overparametrized neural networks, especially when the training objective includes explicit regularization terms.

To sum up, this work broadens the current scope of the NTK literature and connects the recent advances in deep learning theory, e.g., analyzing the trajectory of GD updates, implicit bias of overparametrization, etc., to the classical results in nonparametric statistics. More specifically, our findings not only contribute to the theoretical (in particular, nonparametric) understanding of training overparametrized DNN on noisy data but also promotes the use of ℓ_2 penalty or weight decay in practice for better theoretical guarantees.

3.1 Overparametrized Neural Networks and Kernel Methods

Overparametrized neural networks trained by gradient descent can provably overfit any training data. As the width goes to infinity, training DNN under resembles kernel regression and the corresponding kernel is called Neural Tangent Kernel (NTK).

Neural Tangent Kernel The seminal paper [23] proves that the evolution of DNNs during training can be described by the so-called neural tangent kernel, which is central to characterize the convergence and generalization behaviors. [16], [17], [97] investigate specifically for one-hidden-layer ReLU neural networks and show explicitly that with enough overparametrization, the weight vectors and the corresponding NTK do not change much during GD training. Similar investigations have been done for other neural networks and other settings [98], [100]. Among others, [17], [102] provide generalization error bounds and provable learning scenarios, but only hold for noiseless data.

For noisy data, explicit regularizations have recently been considered in the NTK literature. [103] promote the ℓ_2 penalty when using NTK by showing that in a constructed classification example, sample efficiency can benefit from the regularization. [104] consider classification with noisy labels and propose to add ℓ_2 regularization to ensure robustness. However, their analyses only apply to the kernel estimator directly using NTK and only relate to infinite width neural networks, which greatly restricts the model class capacity. As pointed out before, bridging the technical gap between NTK and finite-width overparametrized neural networks is technically challenging when the training objective includes an ℓ_2 regularization term and we should not take it for granted. [105] demonstrate the similarity between the Laplace kernels and ReLU NTKs. However, in order for NTK to be a good characterization of neural network training, how wide is wide enough remains an active field of research [106]. In comparison, we directly analyze GD trajectories of training finite-width neural networks (with and without ℓ_2 regularization) and prove that the corresponding NTK solutions can be well-approximated after a polynomial number of GD iterations. To the best of our knowledge, we are among the first to rigorously establish the L_2 convergence rate for trained neural networks under noisy data. [107] recently provide similar convergence rate analysis by considering a particular

penalized stochastic gradient descent algorithm but they require the neural network width to be exponential with n.

Our algorithm-dependent statistical analysis bridges the gap between these two types of research. Based on the GD trajectories and the corresponding NTK, we are able to analyze the trained overparametrized neural networks within the nonparametric framework and show they can also achieve the optimal convergence rate with proper regularizations.

Neural Network Setup Consider the one-hidden-layer ReLU neural network family $\mathcal{F}$ with m nodes in the hidden layer, expressed as

$$f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\boldsymbol{w}_r^\top \boldsymbol{x}),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ denotes the input, $\boldsymbol{W} = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m) \in \mathbb{R}^{d\times m}$ is the weight matrix in the hidden layer, $\boldsymbol{a} = (a_1, \cdots, a_m)^\top \in \mathbb{R}^m$ is the weight vector in the output layer, $\sigma(z) = \max\{0, z\}$ is the rectified linear unit (ReLU). The initial values of the weights are independently generated from

$$\boldsymbol{w}_r(0) \ \sim \ N(\boldsymbol{0}, \tau^2 \boldsymbol{I}_m), \quad a_r \ \sim \ \text{unif}\{-1, 1\}, \quad \forall r \in [m].$$

When $m \gg n$, the neural network is highly overparametrized. As is usually assumed in the NTK literature [17], [104], [108], we consider data on the unit sphere $\mathbb{S}^{d-1}$, i.e., $\|\boldsymbol{x}_\mathrm{i}\|_2 = 1$ for any $\mathrm{i} \in [n]$. Throughout this work, we further assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are uniformly distributed on $\mathbb{S}^{d-1}$ so that $\mathbb{E}_{\boldsymbol{x}\sim\text{unif}(\mathbb{S}^{d-1})}(\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2$ and $\|f - f^*\|_2^2$ are equal up to a constant multiplier and thus will be used interchangeably.

Gradient Descent Let $\boldsymbol{y} = (y_1, \cdots, y_n)^\top$ and $\boldsymbol{\epsilon} = (\epsilon_1, \cdots, \epsilon_n)^\top$. Denote $u_\mathrm{i} = f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}_\mathrm{i})$ to be the network's prediction on $\boldsymbol{x}_\mathrm{i}$ and let $\boldsymbol{u} = (u_1, ..., u_n)^\top$. Without loss of generality, we consider fixing the second layer $\boldsymbol{a}$ after initialization and only training the first layer $\boldsymbol{W}$ by GD. Fixing the last layer is not a strong restriction since $a \cdot \sigma(z) = \text{sign}(a) \cdot \sigma(|a|z)$ and we can

always reparametrize the network to have all a_{i}'s to be either 1 or -1. Denote the empirical squared loss as $\Phi(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u}\|_2^2$. The gradient of $\Phi(\boldsymbol{W})$ w.r.t. $\boldsymbol{w}_r$ can be written as

$$\frac{\partial \Phi(\boldsymbol{W})}{\partial \boldsymbol{w}_r} = \frac{1}{\sqrt{m}} a_r \sum_{\mathrm{i}=1}^{n} (u_{\mathrm{i}} - y_{\mathrm{i}}) \mathbb{I}_{r,\mathrm{i}} \boldsymbol{x}_{\mathrm{i}}, \quad r \in [m],$$

where $\mathbb{I}_{r,\mathrm{i}} = \mathbb{I}\{\boldsymbol{w}_r^\top \boldsymbol{x}_{\mathrm{i}} \geq 0\}$. Then the GD update rule at the k-th iteration is given by

$$\boldsymbol{w}_r(k+1) = \boldsymbol{w}_r(k) - \eta \frac{\partial \Phi(\boldsymbol{W})}{\partial \boldsymbol{w}_r} \bigg|_{\boldsymbol{W} = \boldsymbol{W}(k)},$$

where $\eta > 0$ is the step size (a.k.a. learning rate). In the rest of this work, we use k to index variables at the k-th iteration, e.g., $u_{\mathrm{i}}(k) = f_{\boldsymbol{W}(k),\boldsymbol{a}}(\boldsymbol{x}_{\mathrm{i}})$, etc. Define $\mathbb{I}_{r,\mathrm{i}}(k) = \mathbb{I}\{\boldsymbol{w}_r(k)^\top \boldsymbol{x}_{\mathrm{i}} \geq 0\}$, $\boldsymbol{Z}(k) \in \mathbb{R}^{md \times n}$ that

$$\boldsymbol{Z}(k) = \frac{1}{\sqrt{m}} \begin{pmatrix} a_1 \mathbb{I}_{1,1}(k) \boldsymbol{x}_1 & \dots & a_1 \mathbb{I}_{1,n}(k) \boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ a_m \mathbb{I}_{m,1}(k) \boldsymbol{x}_1 & \dots & a_m \mathbb{I}_{m,n}(k) \boldsymbol{x}_n \end{pmatrix}$$

and $\boldsymbol{H}(k) = \boldsymbol{Z}(k)^\top \boldsymbol{Z}(k)$. It is shown that matrices $\boldsymbol{Z}(k)$ and $\boldsymbol{H}(k)$ are close to $\boldsymbol{Z}(0)$ and $\boldsymbol{H}(0)$, respectively for any k, when m is sufficiently large [17]. We can rewrite the GD update rule as

$$\mathrm{vec}(\boldsymbol{W}(k+1)) = \mathrm{vec}(\boldsymbol{W}(k)) - \eta \boldsymbol{Z}(k)(\boldsymbol{u}(k) - \boldsymbol{y}), \tag{3.1}$$

where $\mathrm{vec}(\boldsymbol{W}) = (\boldsymbol{w}_1^\top, \cdots, \boldsymbol{w}_m^\top)^\top \in \mathbb{R}^{md \times 1}$ is the vectorized weight matrix.

Kernel Ridge Regression with NTK The study of one-hidden-layer ReLU neural networks is closely related to the NTK defined as

$$\begin{aligned} h(\boldsymbol{s}, \boldsymbol{t}) =& \mathbb{E}_{\boldsymbol{w} \sim N(0, \boldsymbol{I}_d)} \left(\boldsymbol{s}^\top \boldsymbol{t} \, \mathbb{I}\{\boldsymbol{w}^\top \boldsymbol{s} \geq 0, \boldsymbol{w}^\top \boldsymbol{t} \geq 0\} \right) \\ =& \frac{\boldsymbol{s}^\top \boldsymbol{t} (\boldsymbol{\pi} - \arccos(\boldsymbol{s}^\top \boldsymbol{t}))}{2\boldsymbol{\pi}}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{s}, \boldsymbol{t}$ are d-dimensional vectors. It can be shown that h is positive definite on the unit sphere $\mathbb{S}^{d-1}$ [108]. Let the Mercer decomposition of h be $h(\boldsymbol{s}, \boldsymbol{t}) = \sum_{\mathrm{j}=0}^{\infty} \lambda_{\mathrm{j}} \varphi_{\mathrm{j}}(\boldsymbol{s}) \varphi_{\mathrm{j}}(\boldsymbol{t})$, where $\lambda_1 \geq \lambda_2 \geq ... \geq 0$ are the eigenvalues, and $\{\varphi_{\mathrm{j}}\}_{\mathrm{j}=1}^{\infty}$ is an orthonormal basis.

The following lemma states the decay rate of eigenvalues of the NTK associated with one-hidden-layer ReLU neural networks, as a key technical contribution of this work.

Lemma 3.1.1 *Let λ_{j} be the eigenvalues of NTK h defined above. Then we have $\lambda_{\mathrm{j}} \asymp \mathrm{j}^{-\frac{d}{d-1}}$.*

Let $\mathcal{N}$ denote the reproducing kernel Hilbert space (RKHS) generated by h on $\mathbb{S}^{d-1}$, equipped with norm $\|\cdot\|_{\mathcal{N}}$. For an unknown function $f^* \in \mathcal{N}$, the kernel ridge regression minimizes

$$\min_{f \in \mathcal{N}} \frac{1}{2} \sum_{\mathrm{i}=1}^{n} (y_{\mathrm{i}} - f(\boldsymbol{x}_{\mathrm{i}}))^2 + \frac{\mu}{2} \|f\|_{\mathcal{N}}^2, \tag{3.3}$$

where $\mu > 0$ is a tuning parameter controlling the regularization strength. The representer theorem says that the solution to (3.3) can be written as

$$\hat{f}(\boldsymbol{x}) = h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^{\infty} + \mu \boldsymbol{I}_n)^{-1} \boldsymbol{y} \tag{3.4}$$

for any point $\boldsymbol{x} \in \mathbb{R}^d$, where $h(\boldsymbol{x}, \boldsymbol{X}) = (h(\boldsymbol{x}, \boldsymbol{x}_1), ..., h(\boldsymbol{x}, \boldsymbol{x}_n)) \in \mathbb{R}^{1 \times n}$ and $\boldsymbol{H}^{\infty} = (h(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}}))_{n \times n}$ ($\boldsymbol{H}^{\infty}$ is usually called the NTK matrix). In the following theorem, we show that the function $\hat{f}$ is close to the true function f^* under the L_2 metric.

Theorem 3.1.2 *Let $\hat{f}$ be as in (3.4). By choosing $\mu \asymp n^{(d-1)/(2d-1)}$, we have*

$$\|\hat{f} - f^*\|_2^2 = O_{\mathbb{P}}\left(n^{-\frac{d}{2d-1}}\right), \quad \|\hat{f}\|_{\mathcal{N}}^2 = O_{\mathbb{P}}(1).$$

The proof of the convergence rate requires an accurate characterization of the complexity of $\mathcal{N}$, which is determined by the eigenvalues and eigenfunction expansion of the NTK h. If the eigenvalues decay at rate $\lambda_{\mathrm{j}} \asymp \mathrm{j}^{-2\nu}$, the corresponding minimax optimal rate is $n^{-2\nu/(2\nu+1)}$ [109], [110]. Building on the the eigenvalue decay rate established in Lemma 3.1.1, it can be shown that the L_2 estimation rate in Theorem 3.1.2 is minimax-optimal.

In the rest of this work, we assume that $f^* \in \mathcal{N}$.

3.2 Problems of Gradient Descent from the Nonparametric Perspective

In this section, we consider training overparametrized neural networks with the GD update rule (3.1). Among others, [16], [17] prove that as iteration $k \to \infty$, the training data are interpolated, achieving zero training loss. However, in the presence of noises, i.e., ϵ_i in (1.2), such an overfitting to the training data can be harmful for recovering the ground truth. The following theorem shows that if k is too small or too large, the L_2 estimation error of the trained neural network is bounded away from zero.

Theorem 3.2.1 *Fix a failure probability $\delta \in (0,1)$. Let λ_0 be the largest number that with probability at least $1-\delta$, $\lambda_{\min}(\boldsymbol{H}^\infty) \geq \lambda_0$. Suppose $m \geq \tau^{-2}\mathrm{poly}\left(n, \frac{1}{\lambda_0}, \frac{1}{\delta}\right)$, $\eta = \tilde{O}\left(\frac{\lambda_0}{n^2}\right)$, and $\tau = \tilde{O}\left(\frac{\lambda_0 \delta}{n}\right)$. For sufficiently large n, if the iteration $k = \tilde{\Omega}\left(\frac{\log n}{\eta \lambda_0}\right)$ or $k = \tilde{O}\left(\frac{1}{n\eta}\right)$, then with probability at least $1-2\delta$, we have*

$$\mathbb{E}_{\boldsymbol{\epsilon}} \| f_{\boldsymbol{W}(k),\boldsymbol{a}} - f^* \|_2^2 = \Omega(1).$$

The conditions on m, η, and τ have the same rates as those in Theorem 5.1 of [17], but the constants requirements are different. The probability $1-2\delta$ in Theorem 3.2.1 comes from the randomness of $\lambda_{\min}(\boldsymbol{H}^\infty)$ and $(\boldsymbol{W}(0), \boldsymbol{a})$.

Theorem 3.2.1 states that the estimation error for non-regularized one-hidden-layer neural networks is bounded away from zero by some constant if trained for too short or too long. The latter scenario indicates that overfitting is harmful in terms of the L_2 estimation error. Similar results have been shown in [111] for specifically designed overparametrized DNNs that is a linear combination of $\Omega(n^{10d^2})$ smaller neural networks, which is much more restrictive than ours.

In order to have low L_2 estimation errors, Theorem 3.2.1 implies that the iteration number k must satisfy $(\eta\lambda_0)^{-1} \log n \lesssim k \lesssim (n\eta)^{-1}$. However, deriving a precise order of k, which leads to the optimal rate of convergence, could be extremely challenging. Alternatively, we consider the infinite-width limit of one-hidden-layer ReLU networks, i.e., directly using the NTK (3.2) in kernel regression. This may shed some light on the optimal stopping time for practical overparametrized neural networks.

In kernel regression, the objective becomes

$$\min_{f\in\mathcal{N}} \frac{1}{2}\sum_{\mathrm{i}=1}^{n}(y_{\mathrm{i}} - f(\boldsymbol{x}_{\mathrm{i}}))^2, \tag{3.5}$$

whose solution can be explicitly expressed as $h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}$, by setting $\mu = 0$ in (3.4). However, inverting the kernel matrix can be computationally intensive. In practice, gradient-based methods are often applied to solve (3.5) [110]. The following theorem establishes estimation error results for the NTK estimators trained by GD, complementary to Theorem 3.2.1.

Theorem 3.2.2 *Consider using GD to optimize (3.5) with a sufficiently small step size η depending on n (but not on k). There exists a stopping time k^* depending on data, such that*

$$\mathbb{E}\|\hat{f}_{k^*} - f^*\|_2^2 = O\left(n^{-\frac{d}{2d-1}}\right),$$

where $\hat{f}_k$ is the predictor obtained at the k-th iteration. Moreover, if $k \to \infty$, the interpolated estimator $\hat{f}_\infty$ satisfies

$$\mathbb{E}\|\hat{f}_\infty - f^*\|_2^2 = \Omega(1).$$

To specify the optimal stopping time k^* in Theorem 3.2.2, we first introduce the local empirical Rademacher complexity defined as

$$\hat{\mathcal{R}}_{\boldsymbol{H}^{\infty}}(\varepsilon) := \left(\frac{1}{n}\sum_{\mathrm{i}=1}^{n}\min\left\{\hat{\lambda}_{\mathrm{i}}/n, \varepsilon^2\right\}\right)^{1/2},$$

which relies on the eigenvalues $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n > 0$ of $\boldsymbol{H}^{\infty}$. Then, the stopping time k^* is defined to be

$$k^* := \mathrm{argmin}\left\{k \in \mathbb{N} \mid \hat{\mathcal{R}}_{\boldsymbol{H}^{\infty}}\left(\frac{1}{\sqrt{\eta k}}\right) > \frac{1}{2\mathrm{e}\sigma\eta k}\right\} - 1. \tag{3.6}$$

In essence, the optimal stopping time decreases with the noise level σ and increases with the model complexity, measured by the eigenvalues of $\boldsymbol{H}^{\infty}$.

Remark 3.2.3 *(k^* for neural networks) To derive the order of k^* for overparametrized neural network, a sharp characterization of the eigen-distribution of $\boldsymbol{H}^{\infty}$ is needed. To the*

best of the authors' knowledge, no such results are available yet. Even though as $m \to \infty$, neural network resembles its linearization (NTK), it doesn't necessarily mean such a stopping rule can be easily derived for finite-width neural networks. In general, theoretical guarantees of an early stopping rule for training overparametrized neural networks is challenging and left for future work.

Besides early stopping, explicit regularizations are usually employed in deep learning models to balance the bias-variance trade-off and prevent overfitting, for example, weight decay [112], batch normalization [40], dropout [38], etc., to prevent overfitting. In the next section, we investigate the ℓ_2 regularization [113]–[115] and demonstrate its effectiveness in the nonparametric regression setting.

3.3 ℓ_2-Regularized Gradient Descent for Noisy Data

Without any regularization, GD overfits the training data and the estimation error is bounded away from zero. Instead, we propose using the ℓ_2-regularized gradient descent defined as

$$\begin{aligned}\mathrm{vec}(\boldsymbol{W}_D(k+1)) =& \mathrm{vec}(\boldsymbol{W}_D(k)) - \eta_1 \boldsymbol{Z}_D(k)(\boldsymbol{u}_D(k) - \boldsymbol{y}) \\ & - \eta_2 \mu \mathrm{vec}(\boldsymbol{W}_D(k)),\end{aligned} \tag{3.7}$$

where $\eta_1, \eta_2 > 0$ are step sizes, and $\mu > 0$ is a tuning parameter. It can be easily seen that (3.7) is the GD update rule on the following loss function

$$\Phi_1(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u}\|_2^2 + \frac{\mu}{2}\|\mathrm{vec}(\boldsymbol{W})\|_2^2. \tag{3.8}$$

The ℓ_2 regularization has long been used in practical training neural networks and is equivalent to "weight decay" [112] when using GD [116]. In the NTK literature, ℓ_2 regularization is also considered as a way to improve generalization [103], [104]. However, we are among the first to directly analyze the ℓ_2-regularized GD trajectories of overparametrized neural networks and show its connection to kernel ridge regression using NTK. In the rest of

this work, we use subscript D to denote the variables under the regularized GD (3.7), e.g., $\boldsymbol{u}_D(k)$ for the predictions at the k-th iteration.

Theorem 3.3.1 *Let λ_0 be the largest number such that with probability at least $1-\delta_n$, $\lambda_{\min}(\boldsymbol{H}^\infty) \geq \lambda_0$, and $\delta_n \to 0$ as n goes to infinity*[1]*. For sufficiently large n, suppose $\mu \asymp n^{\frac{d-1}{2d-1}}$, $\eta_1 \asymp \eta_2 = o(n^{-\frac{3d-1}{2d-1}})$, $\tau = O(1)$, $m \geq \tau^{-2}\mathrm{ploy}(n, \lambda_0^{-1})$, and the iteration number k satisfies $\log\left(\mathrm{ploy}_1(n, \tau, 1/\lambda_0)\right) \lesssim \eta_2\mu k \lesssim \log\left(\mathrm{ploy}_2(\tau, 1/n, \sqrt{m})\right)$. Then we have*

$$\|\boldsymbol{u}_D(k) - \boldsymbol{H}^\infty(C\mu I + \boldsymbol{H}^\infty)^{-1}\boldsymbol{y}\|_2 = O_{\mathbb{P}}\left(\sqrt{n}(1-\eta_2\mu)^k\right), \tag{3.9}$$

$$\|\mathrm{vec}(\boldsymbol{W}_D(k)) - (1-\eta_2\mu)^k\mathrm{vec}(\boldsymbol{W}_D(0))\|_2 = O_{\mathbb{P}}(1), \tag{3.10}$$

for some constant $C > 0$. Moreover, during the training process, the mean squared loss satisfies

$$\Phi(\boldsymbol{W}_D(k))/n \leq (1-\eta_2\mu)^k\Phi(\boldsymbol{W}_D(0))/n + O_{\mathbb{P}}(1). \tag{3.11}$$

In the above theorem, three upper bounds are provided. In (3.9), we provide an upper bound on the difference between the prediction using one-hidden-layer neural networks and that obtained by (3.4), which converges to zero as the sample size goes to infinity. This indicates that the ℓ_2 penalty on neural network weights has similar effects to penalizing the RKHS norm as in (3.3). Combining (3.9) and Theorem 3.1.2, we can conclude that the ℓ_2-regularized one-hidden-layer ReLU neural network recovers the true function on the training data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

In (3.10), we provide an upper bound on the distance between the weight matrix at the k-th iteration and the "decayed" initialization $\boldsymbol{W}_D(0)$. Under the conditions in Theorem 3.3.1, their distance measured in Frobenius norm is bounded by some constant depending on the underlying true function. Unlike the results in [17], the upper bound presented in (3.10) does not depend on data. Therefore, as long as the underlying function is within the RKHS generated by NTK, the total movement of all the weights is not large even if the data observed are corrupted by noises.

[1]Potential dependency of λ_0 on n is suppressed for notational simplicity.

In (3.11), we give a characterization of how the training objective decreases over iterations, which is reminiscent of Theorem 4.1 in [16]. Unlike the results without regularization, our ℓ_2-regularized objective is not expected to converge to zero, i.e., no data interpolation, which is essential to ensure the best trade-off between the bias and variance.

Remark 3.3.2 *(More Iterations) The required iteration number k in Theorem 3.3.1 is approximately $(\eta_2\mu)^{-1}$, up to a logarithmic term. We believe the upper bound on k is not necessary and may be relaxed. The stated results are expected to hold if $k \to \infty$ and we conjecture that the output will converge to the optimal solution of kernel ridge regression as in (3.4). Simulation results in Section 3.4 support our conjecture and we leave the technical proof for future work.*

Remark 3.3.3 *(Neural Network Width) In the previous result, the requirement for the width $m \geq \tau^{-2}\mathrm{ploy}(n, \lambda_0^{-1})$ indicates that m is in polynomial order of sample size. Such a overparametrization is not uncommon in the NTK literature. It should be noted that there is a huge gap between overparametrized, finite-width networks and infinite-width networks. The former is still a network while the latter reduces to the exact NTK methods. It remains an active field of research on characterizing the size and approximation error dependence between the two [106].*

Next, we extend the results in Theorem 3.3.1 and establish the L_2 convergence rate for neural networks trained with ℓ_2-regularized GD.

Theorem 3.3.4 *Suppose the assumptions of Theorem 3.3.1 hold. Then we have*

$$\|f_{\boldsymbol{W}_D(k),\boldsymbol{a}} - f^*\|_2^2 = O_{\mathbb{P}}(n^{-\frac{d}{2d-1}}).$$

The above theorem states that with probability tending to one, the neural network estimator can still recover the true function with the optimal convergence rate of $n^{-\frac{d}{2(2d-1)}}$, demonstrating the effectiveness of the ℓ_2 regularization for noisy data. Unlike other optimality results established for neural networks [2], [50], our convergence rate result applies to overparametrized networks and is obtainable using the ℓ_2-regularized GD.

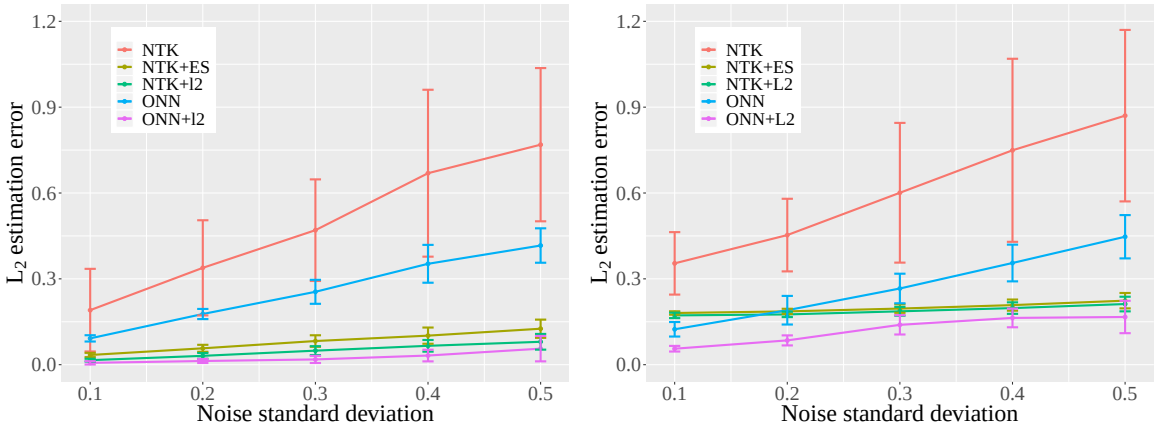


Figure 3.1. The results for f_1^* are shown on the left figure and the results for f_2^* are shown on the right figure. The L_2 estimation errors are shown for all methods vs. σ, with their standard deviations plotted as vertical bars. Similarly for both f_1^* and f_2^*, we observe that NTK and ONN do not recover the true function well. Early stopping and ℓ_2 regularization perform similarly for NTK, especially for f_2^*. ONN+ℓ_2 performs the best in both cases.

3.4 Numerical Studies

In practice, regularization techniques are widely used in training deep learning models. Among others, [33], [114], [117]–[119] have investigated the effectiveness of ℓ_2 regularization and early stopping in training DNNs, and comprehensive comparisons have been made empirically against other regularization techniques. Therefore, one major goal of this section is not to show state-of-the-art performance using ℓ_2 regularization, but to use it as an example to illustrate, from a nonparametric perspective, the necessity of regularization in training overparametrized neural networks with GD. Another goal is to demonstrate the robustness of our theory when some underlying assumptions are violated, e.g., one hidden layer, ReLU activation function and data on a sphere, etc.

Specifically, we consider NTK without regularization (NTK), NTK with early stopping[2] (NTK+ES), NTK with ℓ_2 regularization (NTK+ℓ_2), overparametrized neural network with and without ℓ_2 regularization, denoted as ONN and ONN+ℓ_2, respectively. For ONN, we use two-hidden-layer ReLU neural networks and $m = 500$ for each layer. To train the

[2]As specified in Theorem 3.2.2, the optimal stopping time k^* in (3.6) depends on σ, which is to be estimated from data. In our simulation, we directly use the true value.

neural networks, instead of GD, we consider the more popular RMSProp optimizer [120] with the default setting. For ONN+ℓ_2 and NTK+ℓ_2, the tuning parameter μ is selected by cross-validation.

Neural network setup The neural network used in all experiments is a 2-layer ReLU neural network with $m = 500$ nodes in each hidden layer. All the weighs are initialized with the Glorot uniform initializer, also called as Xavier uniform initializer [121], which is the default choice in the TensorFlow Keras Sequential module. All the weights are trained by RMSProp [120] optimizer with the default setting, e.g. learning rate of 0.001, etc. All ONN experiments are conducted using TensorFlow 2 with Python API.

3.4.1 Simulated Data

Consider the $d = 2$ case where the training data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are i.i.d. sampled from unif$([-1, 1]^2)$. We set $n = 100$ and let noises follow $N(0, \sigma^2)$. Two target functions are considered: $f_1^*(\boldsymbol{x}) = 0$ and $f_2^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x}$. The L_2 estimation error is approximated using a noiseless test dataset $\{(\bar{\boldsymbol{x}}_{\mathrm{i}}, f^*(\bar{\boldsymbol{x}}_{\mathrm{i}}))\}_{\mathrm{i}=1}^{1000}$ where $\bar{\boldsymbol{x}}_{\mathrm{i}}$'s are new samples i.i.d. from unif$([-1, 1]^2)$. We choose $\sigma = 0.1, 0.2, ..., 0.5$ and for each σ value, 100 replications are run to estimate the mean and standard deviation of the L_2 estimation error. Results are presented in Figure 3.1. The learning rate for NTK+ES is $\eta = 0.01$ and the GD update rule is as specified in (3.30). In the ℓ_2-regularized methods, the tuning parameter μ for each task is chosen by cross validation. The validation dataset is of size 100 that is also noiseless and follows the same generating mechanism as the test dataset. For NTK+ℓ_2, we use a grid search of interval $[0, 1]$ with $\mu = 0.01, 0.02, \ldots, 1$ and for ONN+ℓ_2, the μ candidates are $0.1, 0.2, \ldots, 10$. In both cases, we observe that the optimal μ increases with the noise level σ. For f_2^*, we plot the chosen μ and k^* for NTK+ℓ_2 and NTK+ES respectively vs. σ. For each σ value, the reported value is the average of 100 replications. The results are shown in Figure 3.2.

Figure 3.1 clearly demonstrates that ONN and NTK do not recover the true function well. As is explained in the section, without regularization, overfitting the training data is harmful for the L_2 estimation. To illustrate this point, we show the trained estimators of f_2^* for all the methods in Figure 3.3 when $\sigma = 0.1$.

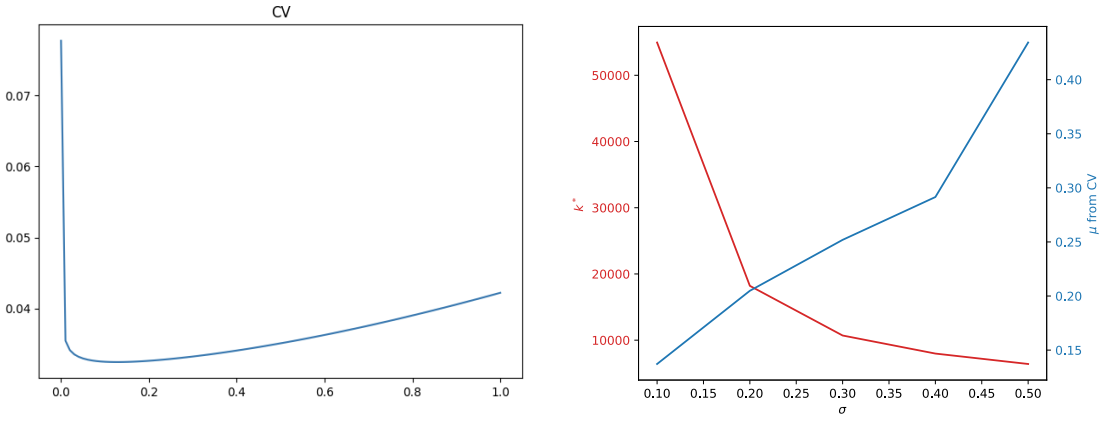


Figure 3.2. Left: Cross-validation of μ in NTK+ℓ_2 for fitting f_2^* when $\sigma = 0.1$. The horizontal axis is values of μ (100 points from 0.01 to 1) and the vertical axis is the validation mean squared error. The cross-validated μ in this case is 0.13. Right: Optimal stopping time k^* in NTK+ES and cross-validated μ in NTK+ℓ_2 for fitting f_2^* are shown vs. σ. The optimal GD stopping time decrease with noise level while the best μ increases with σ.

3.4.2 Real Data

To showcase our results on the L_2 estimation, an ideal dataset is one that can be well-fitted by neural networks so that we can treat it as noiseless and then manually inject random noises. Inspired by the numerical studies in [104], we consider the MNIST dataset (digits 5 vs. 8 relabeled as -1 and 1), where the test accuracy can reach over 99% by shallow fully connected neural networks [122]. For images 5 and 8, the training and test split are the default.[3] We change label 5 and 8 to -1 and 1 respectively. No further pre-processing is done to the dataset. For NTK+ES, the learning rate is $\eta = 0.0001$ and the GD update rule is as specified in (3.30). To account for the high data dimension, we divide the NTK matrix $\boldsymbol{H}^\infty$ by d. For the ONN+ℓ_2 and NTK+ℓ_2, we choose μ by cross-validation and the candidates are $\mu = 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000$ for ONN+ℓ_2 and $\mu = 1, 2, 3, \ldots, 100$ for NTK+ℓ_2. The training/validation split is 80%/20% for cross-validation so the actual training data size is 9107 for all methods (ONN, NTK and NTK+ES do not use the validation dataset). The cross-validated μ for ONN+ℓ_2 and optimal stopping time k^* for NTK+ES are shown in Figure 3.4, together with the cross-validation results specifically for $\sigma = 1$.

[3] http://yann.lecun.com/exdb/mnist/

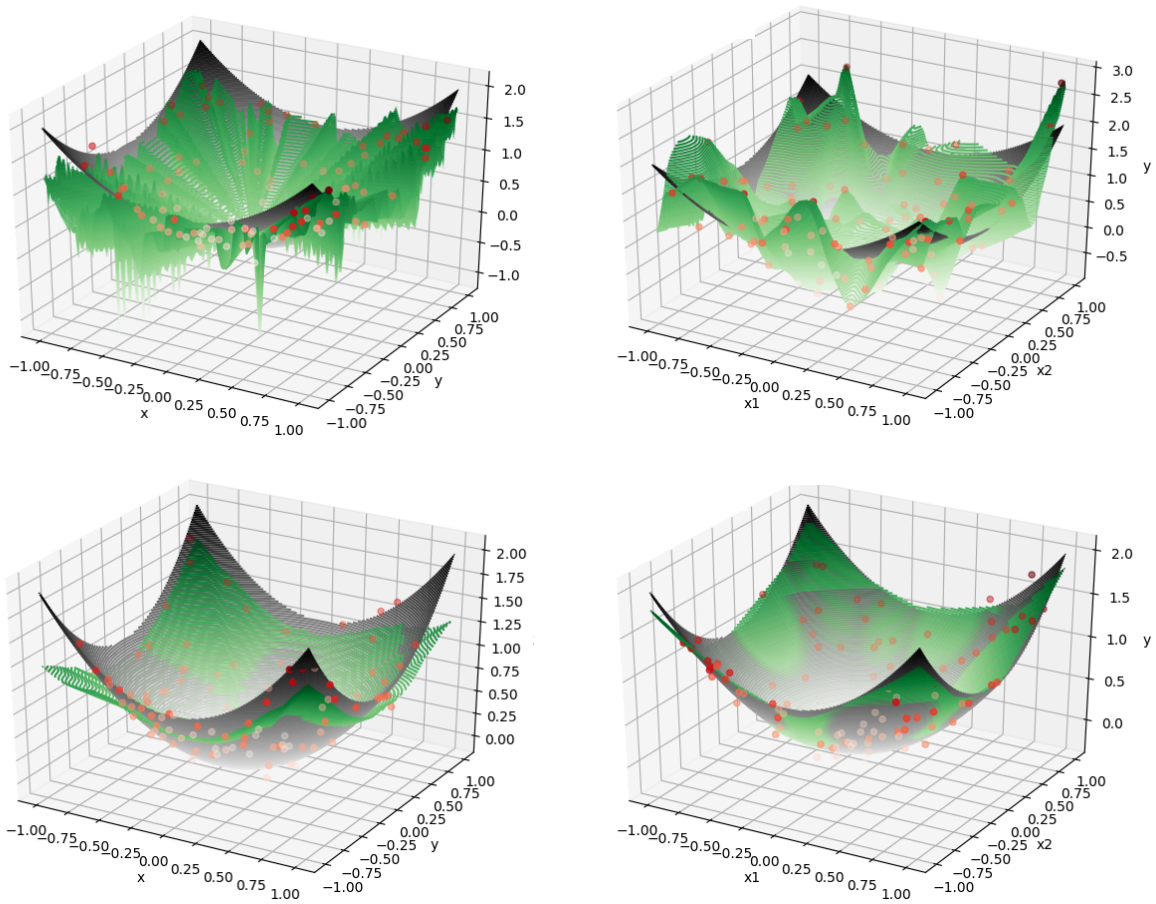


Figure 3.3. Visualizations for the trained estimators of NTK (top left), NTK+ℓ_2 (bottom left), ONN (top right) and ONN+ℓ_2 (bottom right). Training data are plotted as red dots. The green surface is the estimator and the grey surface is the true function f_2^*. Both surfaces are approximated by grid points $(\mathrm{i}/100, \mathrm{j}/100)$ for i, j from -100 to 100. As can be seen in the top row, without regularization, the estimators overfit training data. The fitted estimators are very rough and don't recover the true function well.

Even though the dataset is for classification, we can treat the labels as continuous and learn the true function under the proposed regression setting. We use $\boldsymbol{y}^*$ to denote the true labels and manually add noises $\boldsymbol{\epsilon}$ to the training data, where each element of $\boldsymbol{\epsilon}$ follows $N(0, \sigma^2)$ independently. The perturbed labels are denoted by $\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{\epsilon}$. By gradually increase σ, we investigate how ONN and ONN+ℓ_2 perform under the additive label noises setting.

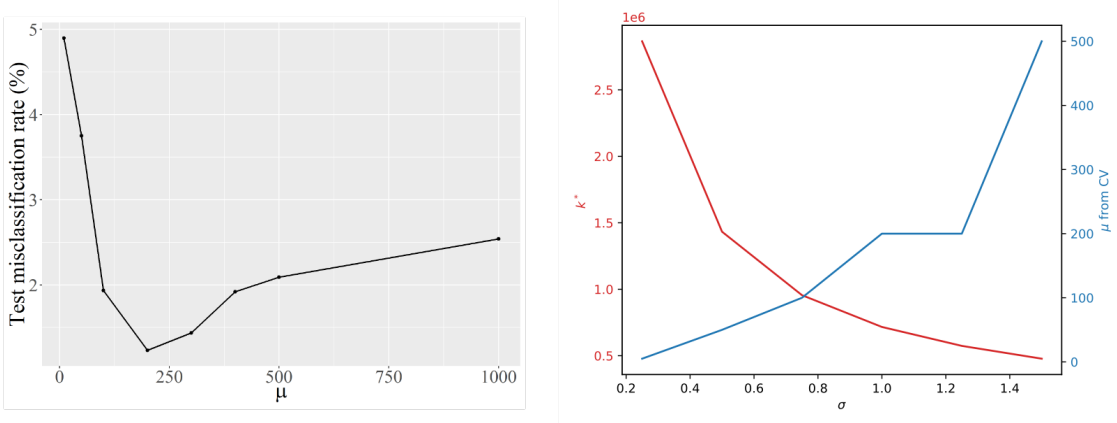


Figure 3.4. Left: Cross-validation result for μ in ONN+ℓ_2 when $\sigma = 1$ (with extra μ candidates of 300 and 400). In the range of $\mu = 5$ to $\mu = 1000$, we can clearly see a V-shape and the best μ in this case is 200. Right: Optimal stopping time k^* in NTK+ES and cross-validated μ in ONN+ℓ_2 for MNIST dataset are shown vs. σ. The optimal stopping time decreases with noise level while the best μ increases with σ.

Remark 3.4.1 *(Additive label noises) To manually inject noises to classification data, many works consider replacing part of the labels by random labels [17], [33]. However, such noises are not i.i.d. and cannot be applied to the regression setting. Similar additive label noises are also considered in [104].*

The training dataset contains $n = 11272$ vectorized images of dimension $d = 784$. The test dataset size is 1866. For ONN+ℓ_2, our training objective function is Φ_1 as in (3.8) and setting $\mu = 0$ corresponds to the objective function of training ONN. On test dataset, which is *not contaminated* by noises, we use the sign of the output for classification and calculate the misclassification rate as a measure of estimation performance. To be more specific, a test image $\bar{\boldsymbol{x}}$ is classified as label 8 if $\hat{f}(\bar{\boldsymbol{x}}) \geq 0$, and label 5 if $\hat{f}(\bar{\boldsymbol{x}}) < 0$, where $\hat{f}$ is the neural network estimator. The misclassification rate is the percentage of incorrect classifications on the test images. We choose $\sigma = 0, 0.25, ..., 1.5$ and for each σ value, 100 replications are run to estimate the mean and standard deviation of the test misclassification rate. How the training root mean square error (RMSE) and test misclassification rate evolve during training when $\sigma = 1$ for ONN and ONN+ℓ_2 is also investigated. The results are reported in Figure 3.5 and 3.6.

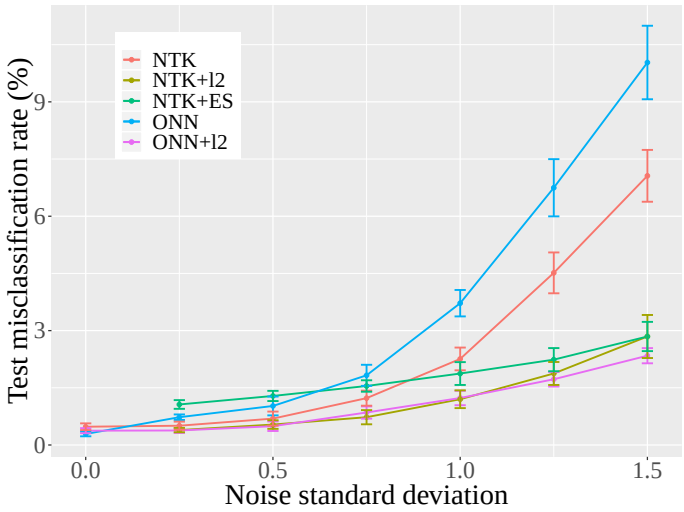


Figure 3.5. The test misclassification rates for all methods vs. σ with their standard deviations plotted as vertical bars is shown in the figure. NTK+ES for $\sigma = 0$ is omitted since k^* is not well-defined when $\sigma = 0$ and NTK+ES in this case should be the same as NTK, i.e. $k^* = \infty$. As σ increases, all misclassification rates increase but NTK+ℓ_2 and ONN+ℓ_2 perform significantly better than NTK and ONN with smaller misclassification rate and better stability, i.e., the standard deviation is smaller. The NTK+ES is the green line and it performs the worst when $\sigma \leq 0.5$ but better than NTK and ONN when $\sigma \geq 1$.

Remark 3.4.2 *(NTK+ES) The performance of NTK+ES is shown in Figure 3.5. Unlike in the simulated dataset where NTK+ES and NTK+ℓ_2 perform almost identically, NTK+ES performs noticeably worst for the MNIST dataset, especially when σ is small. One possible explanation lies in our additive label noise setting. Even though we treat the labels as continuous during training, the reported misclassification rate only depends on the sign of the label. If σ is small, the probability of changing signs is small. This may be one of the reasons that NTK, ONN perform relatively well for small σ's, since if the signs remain the same, it is not very harmful to overfit the labels. Note that NTK+ℓ_2 and ONN+ℓ_2 choose small μ's such that it is not very different from NTK and ONN. The stopping rule in NTK+ES, on the other hand, doesn't take the classification setting into consideration and tends to underestimate the stopping time when the additive label noises are small. Nonetheless, we don't recommend NTK+ES for handling large datasets. Firstly, the noise level σ needs to be estimated, which*

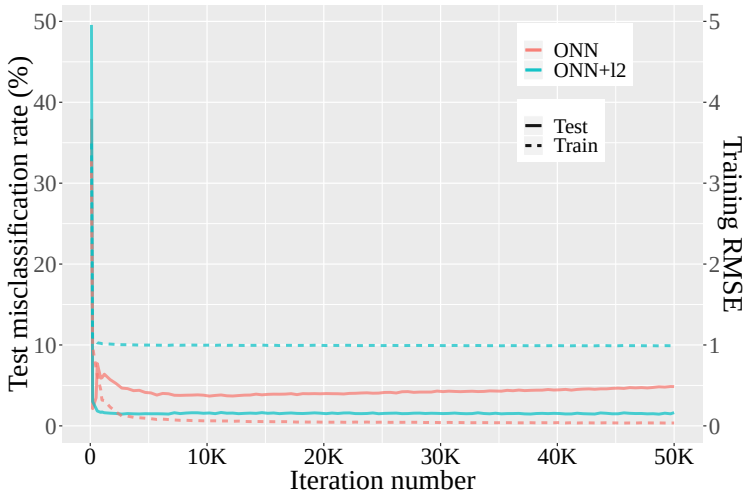


Figure 3.6. The figure shows how the training RMSE and test misclassification rate evolve across iterations for ONN and ONN+ℓ_2 when $\sigma = 1$. For both methods, the training RMSEs decrease fast in the first 1K iterations. However, as the ONN training RMSE flattens after 10K iterations, its test misclassification rate goes up while that for ONN+ℓ_2 remains flat even after 50K iterations, which supports our conjecture in Remark 3.3.2. The right figure also reveals the potential early stopping time for ONN around iteration 10K, which has test misclassification rate comparable to that of ONN+ℓ_2.

brings extra instability to the algorithm. Secondly, NTK+ES is very computationally intensive, especially for the eigenvalues of the NTK matrix.

3.5 Discussion

From a nonparametric perspective, this section studies overparametrized neural networks trained with GD and establishes optimal L_2 convergence rates for trained neural network estimators under the ℓ_2 regularization. On one hand, our result broadens the NTK literature by incorporating an explicit penalty term in the training objective. On the other hand, our convergence analysis extends the statistical theory of deep neural networks by bringing algorithmic guarantees into the network estimator and offsetting the extra complexity from overparametrization through delicate GD analysis. Our simulation results corroborate the theoretical analysis and imply that the assumptions of our theory may be relaxed. More investigations along this direction would advance our statistical understandings of deep learning. For example, our work can be further improved by relaxing the sphere

assumption on the input data and the iteration number k imposed in Theorems 3.3.1 and 3.3.4. Additionally, as empirically shown in numerical experiments, it is possible to extend our theory to multi-layer neural networks with other types of activation functions and training algorithms.

The nonparametric perspective is potentially helpful in understanding other popular regularization techniques, e.g., batch normalization [40], data augmentation [123], knowledge distillation [74], etc. On the other hand, novel and problem-specific regularization approaches may be motivated during the convergence analysis that inspires better performance in practice.

3.6 Technical Proofs

We introduce some additional notations. Denote $\boldsymbol{y}^* = (f^*(x_1), \cdots, f^*(x_n))^\top$ as the the vector of underlying function's functional values at sample points. Let $\mathbb{I}_r(\boldsymbol{x}) = \mathbb{I}\{\boldsymbol{w}_r^\top \boldsymbol{x} \geq 0\}$ and

$$\boldsymbol{z}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \begin{pmatrix} a_1 \mathbb{I}_1(\boldsymbol{x})\boldsymbol{x} \\ \vdots \\ a_m \mathbb{I}_m(\boldsymbol{x})\boldsymbol{x} \end{pmatrix} \in \mathbb{R}^{md \times 1}. \tag{3.12}$$

Thus, $\boldsymbol{Z}(k) = (\boldsymbol{z}(\boldsymbol{x}_1), ..., \boldsymbol{z}(\boldsymbol{x}_n))|_{\boldsymbol{W}=\boldsymbol{W}(k)}$. When the context is clear, we omit the dimension and write $\boldsymbol{I}_d$ as $\boldsymbol{I}$.

Proof of Lemma 3.1.1 We will use the following lemma, which states the Mercer decomposition of h as in (3.2).

Lemma 3.6.1 (Mercer decomposition of NTK h) *For any $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{S}^{d-1}$, we have the following decomposition of the NTK,*

$$h(\boldsymbol{s}, \boldsymbol{t}) = \sum_{k=0}^{\infty} \mu_k \sum_{\mathrm{j}=1}^{N(d,k)} Y_{k,\mathrm{j}}(\boldsymbol{s}) Y_{k,\mathrm{j}}(\boldsymbol{t}),$$

where $Y_{k,\mathrm{j}}$, $\mathrm{j} = 1, ..., N(d,k)$ are spherical harmonic polynomials of degree k, and the non-negative eigenvalues μ_k satisfy $\mu_k \asymp k^{-d}$, and $\mu_k = 0$ if $k = 2\mathrm{j}+1$ for $k \geq 2$.

The proof of Lemma 3.6.1 is similar to the proof of Proposition 5 in [108]. The difference is that the Proposition 5 in [108] considers the kernel function

$$h_1(\boldsymbol{s},\boldsymbol{t}) = 4h(\boldsymbol{s},\boldsymbol{t}) + \frac{\sqrt{1-(\boldsymbol{s}^\top\boldsymbol{t})^2}}{\boldsymbol{\pi}},$$

and we only need to consider the kernel function $h(\boldsymbol{s},\boldsymbol{t})$. A generalization of Proposition 5 in [108] can be found in Theorem 3.5 of [124].

Note that in the proof of Lemma 3.6.1,

$$N(d,\mathrm{j}) = \frac{2\mathrm{j}+d-2}{\mathrm{j}}\begin{pmatrix}\mathrm{j}+d-3\\ d-2\end{pmatrix} = \frac{\Gamma(\mathrm{j}+d-2)}{\Gamma(d-1)\Gamma(\mathrm{j})},$$

where Γ is the Gamma function. By the Stirling approximation, we have $\Gamma(x) \approx \sqrt{2\boldsymbol{\pi}}x^{x-1/2}\mathrm{e}^{-x}$. Therefore, we have the number $N(d,\mathrm{j})$ is equivalent to j^{d-2}. Thus, by Lemma 3.6.1, the j-th eigenvalue λ_j can be denoted by

$$\lambda_\mathrm{j} = \mu_l, \text{ for } \sum_{\mathrm{i}=1}^{l-1} N(d,2\mathrm{i}) \le \mathrm{j} < \sum_{\mathrm{i}=1}^{l} N(d,2\mathrm{i}),$$

which can be approximated by $\lambda_\mathrm{j} \asymp \mu_l$, for $(2l-2)^{d-1} \le \mathrm{j} < (2l)^{d-1}$. By Lemma 3.6.1, we have $\mu_l \asymp l^{-d}$, which implies $\lambda_\mathrm{j} \asymp \mathrm{j}^{-\frac{d}{d-1}}$.

Proof of Theorem 3.1.2

Proof Let $\mathcal{G}$ be a metric space equipped with a metric d_g. The δ-covering number of the metric space $(\mathcal{G},d_g)$, denoted by $N(\delta,\mathcal{G},d_g)$, is the minimum integer N so that there exist N distinct balls in $(\mathcal{G},d_g)$ with radius δ, and the union of these balls covers $\mathcal{G}$. Let $H(\delta,\mathcal{G},d_g) = \log N(\delta,\mathcal{G},d_g)$ be the entropy of the metric space $(\mathcal{G},d_g)$. We first present an upper bound on the entropy of the metric space $(\mathcal{N},\|\cdot\|_\infty)$, where the proof can be found in Section 3.6.3. ■

Lemma 3.6.2 *Let $\mathcal{N}$ be the reproducing kernel Hilbert space generated by the NTK h defined in (3.2), equipped with norm $\|\cdot\|_{\mathcal{N}}$. The entropy $H(\delta, \mathcal{N}(1), \|\cdot\|_\infty)$ can be bounded by*

$$H(\delta, \mathcal{N}(1), \|\cdot\|_\infty) \leq A_0 \delta^{-\frac{2(d-1)}{d}},$$

where $\mathcal{N}(1) = \{f : f \in \mathcal{N}, \|f\|_{\mathcal{N}} \leq 1\}$, and $A_0 > 0$ is a constant not depending on δ.

For the regression problem, consider a general penalized least-square estimator

$$\hat{f} := \underset{f \in \mathcal{N}}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{\mathrm{i}=1}^{n} (y_{\mathrm{i}} - f(\boldsymbol{x}_{\mathrm{i}}))^2 + \lambda_n^2 I^v(f) \right),$$

where $\lambda_n > 0$ is the smoothing parameter and $I : \mathcal{N} \to [0, \infty)$ is a pseudo-norm measuring the complexity. We use the RKHS norm $\|f\|_{\mathcal{N}(\Omega)}$ in our case. Let $\|\cdot\|_n$ denote the empirical norm. The following lemma establishes the rate of convergence for the estimator $\hat{f}$.

Lemma 3.6.3 (Lemma 10.2 in [92]) *Assume Gaussian noises and entropy bound $H(\delta, \mathcal{N}(1), \|\cdot\|_n) \leq A\delta^{-\alpha}$ for some constants $A > 0$ and $0 < \alpha < 2$. If $v \geq \frac{2\alpha}{2+\alpha}$, $I(f^*) > 0$ and*

$$\lambda_n^{-1} = O_{\mathbb{P}}\left(n^{1/(2+\alpha)}\right) I^{(2v-2\alpha+v\alpha)/2(2+\alpha)}(f^*).$$

Then we have

$$\|\hat{f} - f^*\|_n = O_{\mathbb{P}}(\lambda_n) I^{v/2}(f^*)$$

and $I(\hat{f}) = O_{\mathbb{P}}(1) I(f^*)$.

To bound the difference between empirical norm and L_2 norm, we utilize the following lemma. For a class of functions $\mathcal{F}$, define for $z > 0$

$$J_\infty(z, \mathcal{F}) := C_0 \inf_{\delta > 0} \left[z \int_{\delta/4}^{1} \sqrt{\mathcal{H}_\infty(uz/2, \mathcal{F})} du + \sqrt{n}\delta z \right].$$

Lemma 3.6.4 (Theorem 2.2 in [125]) *Let*

$$R := \sup_{f \in \mathcal{F}} \|f\|_2, \;\; K := \sup_{f \in \mathcal{F}} \|f\|_\infty$$

Then, for all $t > 0$, with probability at least $1 - \exp[-t]$,

$$\sup_{f\in\mathcal{F}}\left|\|f\|_n^2 - \|f\|_2^2\right|/C_1 \leq \frac{2RJ_\infty(K,\mathcal{F}) + RK\sqrt{t}}{\sqrt{n}} + \frac{4J_\infty^2(K,\mathcal{F}) + K^2 t}{n}$$

where $C_1 > 0$ is some constant not depending on n.

Proof of Theorem 3.1.2

Proof Consider our estimator $\hat{f}$ as in (3.4), in which case, $v = 2$ and $I(f)$ is the RKHS norm of f. Since $\|f\|_n \leq \|f\|_\infty$, Lemma 3.6.2 indicates that $\alpha = 2(d-1)/d < 2$. By choosing $\lambda_n \asymp n^{-d/(4d-2)}$, which corresponds to $\mu \asymp n^{(d-1)/(2d-1)}$ in (3.3), Lemma 3.6.3 yields that

$$\|\hat{f} - f^*\|_n^2 = O_{\mathbb{P}}(n^{-d/(2d-1)}) \quad \text{and} \quad \|\hat{f}\|_{\mathcal{N}}^2 = O_{\mathbb{P}}(1).$$

Now we use Lemma 3.6.4 to obtain a bound on $\|\hat{f} - f^*\|_2$. First consider $\{f - f^* : f \in \mathcal{N}(1)\}$. Since $\|f\|_{\mathcal{N}} \leq 1$ for every $f \in \mathcal{N}(1)$, we have $K, R = O(1)$. By the entropy bound in Lemma 3.6.2 we have $J_\infty(z, \mathcal{N}(1)) \leq 2C_0 z^{1/d}$. Therefore, Lemma 3.6.4 yields

$$\sup_{f\in\mathcal{N}(1)}\left|\|f - f^*\|_n^2 - \|f - f^*\|_2^2\right| = O_{\mathbb{P}}\left(\sqrt{\frac{1}{n}}\right).$$

Combined with $\|\hat{f} - f^*\|_n^2 = O_{\mathbb{P}}(n^{-d/(2d-1)})$, we can conclude that for any $t > 0$ large enough, $\|\hat{f} - f^*\|_2^2 = O(\sqrt{t/n})$ with probability at least $1 - \exp(-t)$. Utilizing Lemma 3.6.4 again with $R = O(\sqrt{t/n})$ we have for some $C > 0$,

$$\mathbb{P}\left(\sup_{f\in\mathcal{G}(R)}\left|\|f - f^*\|_n^2 - \|f - f^*\|_2^2\right| \leq \frac{Ct}{n}\right) \geq 1 - \mathrm{e}^{-t},$$

where $\mathcal{G}(R) := \{f \in \mathcal{N}(1) : \|f - f^*\|_2 \leq R\}$. Notice that $\hat{f} \in \mathcal{G}(R)$ with probability at least $1 - \exp(-t)$. Therefore, $\|\hat{f} - f^*\|_2^2 = O(n^{-d/(2d-1)} + t/n)$ with probability at least $1 - 2\exp(-t)$. ■

3.6.1 Proofs of main theorems in Section 3.2

For brevity, let $\hat{f}_k = f_{\boldsymbol{W}(k),\boldsymbol{a}}$. For two positive semidefinite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we write $\boldsymbol{A} \geq \boldsymbol{B}$ to denote that $\boldsymbol{A} - \boldsymbol{B}$ is positive semidefinite and $\boldsymbol{A} > \boldsymbol{B}$ to denote that $\boldsymbol{A} - \boldsymbol{B}$ is positive definite. This partial order of positive semidefinite matrices is also known as Loewner order. We focus on the L_2 loss of our estimator $\hat{f}_k$ after k GD updates. Let $\tilde{f}$ denote the kernel regression solution with kernel $h(\cdot,\cdot)$ that interpolates all $\{(\boldsymbol{x}_{\mathrm{i}}, f^*(\boldsymbol{x}_{\mathrm{i}}))\}_{\mathrm{i}=1}^n$, i.e.,

$$g(\boldsymbol{x}) = h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*. \tag{3.13}$$

We first provide some lemmas used in this section. The proofs of lemmas are presented in Section 3.6.3. Lemma 3.6.5 states some basic inequalities that are also used in the proof of Theorem 3.3.1. Lemma 3.6.6 provides the convergence rate of interpolant using NTK. Lemmas 3.6.7 can be found in [17]. Lemma 3.6.8 is implied by the proof in [17]. Lemma 3.6.9 provides some bounds on the related quantities used in the proofs of Theorems 3.2.1 and 3.3.4. Lemma 3.6.10 provide some properties of Loewner order.

Lemma 3.6.5 *Let μ be as in Theorem 3.1.2. Then we have*

$$\begin{aligned}
h(\boldsymbol{s},\boldsymbol{s}) - h(\boldsymbol{s},\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}h(\boldsymbol{X},\boldsymbol{s}) &\geq 0,\\
\int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^\infty + \mu\boldsymbol{I})^{-2}h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x} &= O_{\mathbb{P}}(n^{-\frac{d}{2d-1}}),\\
\int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{x}) - h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x} &= O_{\mathbb{P}}(n^{-\frac{1}{2d-1}}),
\end{aligned}$$

where $h(\boldsymbol{x},\boldsymbol{X}) = (h(\boldsymbol{x},\boldsymbol{x}_1),...,h(\boldsymbol{x},\boldsymbol{x}_n))$ and $h(\boldsymbol{X},\boldsymbol{x}) = h(\boldsymbol{x},\boldsymbol{X})^\top$.

Lemma 3.6.6 *Assume the true function $f^* \in \mathcal{N}$ with finite RKHS norm, then $g(\boldsymbol{x})$ defined (3.13) satisfies*

$$\|g - f^*\|_2 = O_{\mathbb{P}}\left(n^{-1/2}\right).$$

Lemma 3.6.7 (Lemma C.1 in [17]) *If* $\lambda_0 = \lambda_{min}(\boldsymbol{H}^{\infty}) > 0$, $m = \Omega\left(\frac{n^6}{\lambda_0^4 \tau^2 \delta^3}\right)$ *and* $\eta = O\left(\frac{\lambda_0}{n^2}\right)$, *with probability at least* $1-\delta$ *over the random initialization, we have*

$$\|\boldsymbol{w}_r(k) - \boldsymbol{w}_r(0)\|_2 \leq R_0, \quad \forall\ r \in [m], \forall\ k \geq 0,$$

where $R_0 = \frac{4\sqrt{n}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2}{\sqrt{m}\lambda_0}$.

Lemma 3.6.8 ([17]) *Denote* $u_{\mathrm{i}}(k) = f_{\boldsymbol{W}(k),\boldsymbol{a}}(\boldsymbol{x}_{\mathrm{i}})$ *to be the network's prediction on the* i*-th input and let* $\boldsymbol{u}(k) = (u_1(k), ..., u_n(k))^{\top} \in \mathbb{R}^n$ *denote all* n *predictions on the points* $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ *at iteration* k. *We have*

$$\boldsymbol{u}(k) - \boldsymbol{y} = (\boldsymbol{I} - \eta \boldsymbol{H}^{\infty})^k (\boldsymbol{u}(0) - \boldsymbol{y}) + \mathrm{e}(k)$$

where

$$\|\mathrm{e}(k)\|_2 = O\left(k\left(1 - \frac{\eta\lambda_0}{4}\right)^{k-1} \frac{\eta n^{5/2}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^2}{\sqrt{m}\lambda_0\tau\delta}\right).$$

Lemma 3.6.9 *With probability at least* $1-\delta$, *we have*

(a) $\|\boldsymbol{Z}(k) - \boldsymbol{Z}(0)\|_F = O\left(\frac{n^{3/4}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^{1/2}}{\sqrt{m^{1/2}\lambda_0\tau\delta}}\right)$;

(b) $\|\boldsymbol{H}(0) - \boldsymbol{H}^{\infty}\|_F = O\left(\frac{n\sqrt{\log(n/\delta)}}{\sqrt{m}}\right)$;

(c) $\|\boldsymbol{z}_0(\cdot)^{\top}\boldsymbol{Z}(0) - h(\cdot, \boldsymbol{X})\|_2 = O\left(\frac{\sqrt{n}\sqrt{\log(n/\delta)}}{\sqrt{m}}\right)$;

(d) $\|z_0(\cdot)^{\top}\mathrm{vec}(\boldsymbol{W}(0))\|_2 = O\left(\tau\sqrt{\log(1/\delta)}\right)$.

Lemma 3.6.10 (Properties of Loewner order) *For two positive semi-definite matrices* $\boldsymbol{A}$ *and* $\boldsymbol{B}$,

(a). Suppose $\boldsymbol{A}$ *is non-singular, then* $\boldsymbol{A} \geq \boldsymbol{B} \iff \lambda_{max}(\boldsymbol{B}\boldsymbol{A}^{-1}) \leq 1$ *and* $\boldsymbol{A} > \boldsymbol{B} \iff \lambda_{\max}(\boldsymbol{B}\boldsymbol{A}^{-1}) > 1$, *where* $\lambda_{\max}(\cdot)$ *denotes the maximum eigenvalue of the input matrix.*

(b). Suppose $\boldsymbol{A}$, $\boldsymbol{B}$ *and* $\boldsymbol{Q}$ *are positive definite,* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are exchangeable, then* $\boldsymbol{A} \geq \boldsymbol{B} \Longrightarrow \boldsymbol{A}\boldsymbol{Q}\boldsymbol{A} \geq \boldsymbol{B}\boldsymbol{Q}\boldsymbol{B}$.

Proof of Theorem 3.2.1

Proof For notational simplification, we use $\hat{f}_k = f_{\boldsymbol{W}(k),\boldsymbol{a}}$. Define

$$\tilde{f}_k(\boldsymbol{x}) = \mathrm{vec}(\boldsymbol{W}(k))^\top \boldsymbol{z}_0(\boldsymbol{x}),$$

where $\boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})|_{\boldsymbol{W}=\boldsymbol{W}(0)}$. Then we can write the following decomposition

$$\hat{f}_k - f^* = (\hat{f}_k - \tilde{f}_k) + (\tilde{f}_k - g) + (g - f^*) = \Delta_1 + \Delta_2 + \Delta_3, \tag{3.14}$$

where g is as in (3.13). It follows from Lemma 3.6.6 that

$$\|\Delta_3\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{1}{n}}\right). \tag{3.15}$$

For Δ_1, under the assumptions of Lemma 3.6.7, with high probability, we have $\|\boldsymbol{w}_r(k) - \boldsymbol{w}_r(0)\|_2 \leq R_0$. Thus, for fixed $\boldsymbol{x}$, we have

$$|\boldsymbol{w}_r(k)^\top \boldsymbol{x} - \boldsymbol{w}_r(0)^\top \boldsymbol{x}| \leq \|\boldsymbol{w}_r(k) - \boldsymbol{w}_r(0)\|_2 \|\boldsymbol{x}\|_2 \leq R_0.$$

Define event

$$B_r(\boldsymbol{x}) = \{|\boldsymbol{w}_r(0)^\top \boldsymbol{x}| \leq R_0\}, \forall r \in [m].$$

If $\mathbb{I}\{B_r(\boldsymbol{x})\} = 0$, then we have $\mathbb{I}_{r,k}(\boldsymbol{x}) = \mathbb{I}_{r,0}(\boldsymbol{x})$, where $\mathbb{I}_{r,k}(\boldsymbol{x}) = \mathbb{I}\{\boldsymbol{w}_r(k)^\top \boldsymbol{x} \geq 0\}$. Therefore, for any fixed $\boldsymbol{x}$, we have

$$\begin{aligned}
|\hat{f}_k(\boldsymbol{x}) - \tilde{f}_k(\boldsymbol{x})| &= \left| \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r (\mathbb{I}_{r,k}(\boldsymbol{x}) - \mathbb{I}_{r,0}(\boldsymbol{x})) \boldsymbol{w}_r(k)^\top \boldsymbol{x} \right| \\
&= \left| \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \mathbb{I}\{B_r(\boldsymbol{x})\} (\mathbb{I}_{r,k}(\boldsymbol{x}) - \mathbb{I}_{r,0}(\boldsymbol{x})) \boldsymbol{w}_r(k)^\top \boldsymbol{x} \right| \\
&\leq \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_r(\boldsymbol{x})\} |\boldsymbol{w}_r(k)^\top \boldsymbol{x}| \\
&\leq \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_r(\boldsymbol{x})\} \left(|\boldsymbol{w}_r(0)^\top \boldsymbol{x}| + |\boldsymbol{w}_r(k)^\top \boldsymbol{x} - \boldsymbol{w}_r(0)^\top \boldsymbol{x}| \right) \\
&\leq \frac{2R_0}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_r(x)\}
\end{aligned}$$

Recall that $\|\boldsymbol{x}\|_2 = 1$, which implies that $\boldsymbol{w}_r(0)^\top \boldsymbol{x}$ is distributed as $N(0, \tau^2)$. Therefore, we have

$$\mathbb{E}[\mathbb{I}\{B_r(x)\}] = \mathbb{P}\left(|\boldsymbol{w}_r(0)^\top \boldsymbol{x}| \leq R_0 \right) = \int_{-R_0}^{R_0} \frac{1}{\sqrt{2\pi}\tau} \exp\left\{ -\frac{u^2}{2\tau^2} \right\} du \leq \frac{2R_0}{\sqrt{2\pi}\tau}.$$

By Markov's inequality, with probability at least $1 - \delta$, we have

$$\sum_{r=1}^{m} \mathbb{I}\{B_r(x)\} \leq \frac{2mR_0}{\sqrt{2\pi}\tau\delta}.$$

Thus, we have

$$\|\Delta_1\|_2 \leq \frac{2R_0}{\sqrt{m}} \| \sum_{r=1}^{m} \mathbb{I}\{B_r(\cdot)\} \|_2 \leq \frac{4\sqrt{m}R_0^2}{\sqrt{2\pi}\tau\delta} = O\left(\frac{n\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^2}{\sqrt{m}\tau\lambda_0^2\delta} \right). \tag{3.16}$$

Next, we evaluate Δ_2. Recall that the GD update rule is

$$\mathrm{vec}(\boldsymbol{W}(\mathrm{j}+1)) = \mathrm{vec}(\boldsymbol{W}(\mathrm{j})) - \eta \boldsymbol{Z}(\mathrm{j})(\boldsymbol{u}(\mathrm{j}) - \boldsymbol{y}), \mathrm{j} \geq 0.$$

Applying Lemma 3.6.8, we can get

$$
\begin{aligned}
&\operatorname{vec}(\boldsymbol{W}(k)) - \operatorname{vec}(\boldsymbol{W}(0)) \\
=&\sum_{\mathrm{j}=0}^{k-1}(\operatorname{vec}(\boldsymbol{W}(\mathrm{j}+1)) - \operatorname{vec}(\boldsymbol{W}(\mathrm{j}))) \\
=&-\sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(\mathrm{j})(\boldsymbol{u}(\mathrm{j}) - \boldsymbol{y}) \\
=&\sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(\mathrm{j})(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^{\mathrm{j}}(\boldsymbol{y} - \boldsymbol{u}(0)) - \sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(\mathrm{j})\mathrm{e}(\mathrm{j}) \\
=&\sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(0)(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^{\mathrm{j}}(\boldsymbol{y} - \boldsymbol{u}(0)) + \sum_{\mathrm{j}=0}^{k-1}\eta(\boldsymbol{Z}(\mathrm{j}) - \boldsymbol{Z}(0))(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^{\mathrm{j}}(\boldsymbol{y} - \boldsymbol{u}(0)) - \sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(\mathrm{j})\mathrm{e}(\mathrm{j}) \\
=&\sum_{\mathrm{j}=0}^{k-1}\eta\boldsymbol{Z}(0)(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^{\mathrm{j}}(\boldsymbol{y} - \boldsymbol{u}(0)) + \zeta(k).
\end{aligned}
$$

For the first term of $\zeta(k)$, applying Lemma 3.6.9 (a), with probability at least $1-\delta$, we get

$$
\begin{aligned}
&\|\sum_{\mathrm{j}=0}^{k-1}\eta(\boldsymbol{Z}(\mathrm{j}) - \boldsymbol{Z}(0))(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^{\mathrm{j}}(\boldsymbol{y} - \boldsymbol{u}(0))\|_2 \\
\leq&\sum_{\mathrm{j}=0}^{k-1} O\left(\frac{n^{3/4}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^{1/2}}{\sqrt{m^{1/2}\lambda_0\tau\delta}}\right)\eta\|\boldsymbol{I} - \eta\boldsymbol{H}^{\infty}\|_2^{\mathrm{j}}\|(\boldsymbol{y} - \boldsymbol{u}(0))\|_2 \\
\leq&O\left(\frac{n^{3/4}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^{3/2}}{\sqrt{m^{1/2}\lambda_0\tau\delta}}\right)\sum_{\mathrm{j}=0}^{k-1}\eta(1-\eta\lambda_0)^{\mathrm{j}} \\
=&O\left(\frac{n^{3/4}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^{3/2}}{m^{1/4}\tau^{1/2}\lambda_0^{3/2}\delta^{1/2}}\right).
\end{aligned}
$$

Denote that $z_{\mathrm{i}}(\mathrm{j}) = z(\boldsymbol{x}_{\mathrm{i}})|_{\boldsymbol{W}=\boldsymbol{W}(\mathrm{j})}$. By (3.12), we have $\|\boldsymbol{z}_{\mathrm{i}}(\mathrm{j})\|_2 \leq 1$. Thus,

$$
\|\boldsymbol{Z}(\mathrm{j})\|_F = \left(\sum_{\mathrm{i}=1}^{n}\|\boldsymbol{z}_{\mathrm{i}}(\mathrm{j})\|_2^2\right)^{\frac{1}{2}} \leq \sqrt{n} \quad ,\forall\ \mathrm{j} \geq 0. \tag{3.17}
$$

For the second term of $\zeta(k)$, we have

$$\begin{aligned}
&\|\sum_{\mathrm{j}=0}^{k-1}\eta \boldsymbol{Z}(\mathrm{j})\mathrm{e}(\mathrm{j})\|_2\\
\leq&\sum_{\mathrm{j}=0}^{k-1}\eta\|\boldsymbol{Z}(\mathrm{j})\|_F\|\mathrm{e}(\mathrm{j})\|_2\\
\leq&\sum_{\mathrm{j}=0}^{k-1}\eta\sqrt{n}O\left(\mathrm{j}\left(1-\frac{\eta\lambda_0}{4}\right)^{\mathrm{j}-1}\frac{\eta n^{5/2}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^2}{\sqrt{m}\tau\lambda_0\delta}\right)\\
=&O\left(\frac{n^3\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^2}{\sqrt{m}\lambda_0^3\tau\delta}\right).
\end{aligned}$$

Therefore,

$$\|\zeta(k)\|_2=O\left(\frac{n^{3/4}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^{3/2}}{m^{1/4}\tau^{1/2}\lambda_0^{3/2}\delta^{1/2}}\right)+O\left(\frac{n^3\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^2}{\sqrt{m}\lambda_0^3\tau\delta}\right). \tag{3.18}$$

Define $\boldsymbol{G}_k=\sum_{\mathrm{j}=0}^{k-1}\eta(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^{\mathrm{j}}$. Recalling that $\boldsymbol{y}=\boldsymbol{y}^*+\boldsymbol{\epsilon}$, for fixed $\boldsymbol{x}$, we have

$$\begin{aligned}
\tilde{f}_k(\boldsymbol{x})-g(\boldsymbol{x})=&\boldsymbol{z}_0(\boldsymbol{x})^\top\mathrm{vec}(\boldsymbol{W}(k))-h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\\
=&\boldsymbol{z}_0(\boldsymbol{x})^\top\Big[\boldsymbol{Z}(0)\boldsymbol{G}_k(\boldsymbol{y}-\boldsymbol{u}(0))+\zeta(k)+\mathrm{vec}(\boldsymbol{W}(0))\Big]\\
=&\Big[h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{G}_k-(\boldsymbol{H}^\infty)^{-1})\boldsymbol{y}^*+h(\boldsymbol{x},\boldsymbol{X})\boldsymbol{G}_k\boldsymbol{\epsilon}\Big]+\Big[\boldsymbol{z}_0(\boldsymbol{x})^\top\boldsymbol{Z}(0)-h(\boldsymbol{x},\boldsymbol{X})\Big]\boldsymbol{G}_k\boldsymbol{y}\\
&+\Big[\boldsymbol{z}_0(\boldsymbol{x})^\top\mathrm{vec}(\boldsymbol{W}(0))+\boldsymbol{z}_0(\boldsymbol{x})^\top\zeta(k)-\boldsymbol{z}_0(\boldsymbol{x})^\top\boldsymbol{Z}(0)\boldsymbol{G}_k\boldsymbol{u}(0)\Big]\\
=&\Delta_{21}(\boldsymbol{x})+\Delta_{22}(\boldsymbol{x})+\Delta_{23}(\boldsymbol{x}).
\end{aligned} \tag{3.19}$$

Using Lemma 3.6.9 (c), we can bound Δ_{22} as

$$\begin{aligned}
\|\Delta_{22}\|_2\leq&\|\boldsymbol{z}_0(\boldsymbol{x})^\top\boldsymbol{Z}(0)-h(\boldsymbol{x},\boldsymbol{X})\|_2\|\boldsymbol{G}_k\boldsymbol{y}\|_2\\
\leq&O\left(\frac{\sqrt{n}\sqrt{\log(n/\delta)}}{\sqrt{m}}\right)\|(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}\|_2\\
=&O\left(\frac{\sqrt{n}\sqrt{\log(n/\delta)}\|\boldsymbol{y}\|_2}{\sqrt{m}\lambda_0}\right).
\end{aligned} \tag{3.20}$$

Since the i-th coordinate of $\boldsymbol{u}(0)$ is

$$u_{\mathrm{i}}(0) = \boldsymbol{z}_0(\boldsymbol{x}_{\mathrm{i}})^\top \mathrm{vec}(\boldsymbol{W}(0)) = \sum_{r=1}^{m} a_r \boldsymbol{w}(0)^\top \boldsymbol{x}_{\mathrm{i}} \mathbb{I}\{\boldsymbol{w}(0)^\top \boldsymbol{x}_{\mathrm{i}}\},$$

where $a_r \sim \mathrm{unif}\{1, -1\}$ and $\boldsymbol{w}(0)^\top \boldsymbol{x}_{\mathrm{i}} \sim N(0, \tau^2)$, it is easy to prove that $u_{\mathrm{i}}(0)$ has zero mean and variance τ^2. This implies $\mathbb{E}[\|\boldsymbol{u}(0)\|_2^2] = O(n\tau^2)$. By Markov's inequality, with probability at least $1 - \delta$, we have $\|\boldsymbol{u}(0)\|_2 = O\left(\frac{\sqrt{n}\tau}{\delta}\right)$. Similar to (3.17), we can obtain $\|\boldsymbol{Z}(0)\|_F = O(\sqrt{n})$. Thus,

$$|\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0) \boldsymbol{G}_k \boldsymbol{u}(0)| \leq \|\boldsymbol{z}_0(\boldsymbol{x})\|_2 \|\boldsymbol{Z}(0)\|_F \|\boldsymbol{G}_k \boldsymbol{u}(0)\|_2 \leq \sqrt{n} \|(\boldsymbol{H}^\infty)^{-1} \boldsymbol{u}(0)\|_2 = O\left(\frac{n\tau}{\lambda_0 \delta}\right). \tag{3.21}$$

Combining Lemma 3.6.9 (d), (3.18) and (3.21), we obtain

$$\begin{aligned}
\|\Delta_{23}\|_2 \leq & \|\boldsymbol{z}_0(\cdot)^\top \mathrm{vec}(\boldsymbol{W}(0))\|_2 + \|\boldsymbol{z}_0(\cdot)\|_2 \|\zeta(k)\|_2 + \|\boldsymbol{z}_0(\cdot)^\top \boldsymbol{Z}(0) \boldsymbol{G}_k \boldsymbol{u}(0)\|_2 \\
= & O\left(\tau \sqrt{\log(1/\delta)}\right) + O\left(\frac{n^{3/4} \|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^{3/2}}{m^{1/4} \tau^{1/2} \lambda_0^{3/2} \delta^{1/2}}\right) + O\left(\frac{n^3 \|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^2}{\sqrt{m} \lambda_0^3 \tau \delta}\right) + O\left(\frac{n\tau}{\lambda_0 \delta}\right) \\
= & O\left(\frac{n^{3/4} \|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^{3/2}}{m^{1/4} \tau^{1/2} \lambda_0^{3/2} \delta^{1/2}}\right) + O\left(\frac{n^3 \|\boldsymbol{y} - \boldsymbol{u}(0)\|_2^2}{\sqrt{m} \lambda_0^3 \tau \delta}\right) + O\left(\frac{n\tau}{\lambda_0 \delta}\right).
\end{aligned} \tag{3.22}$$

By (3.14) and (3.19), we can rewrite $\hat{f}_k - f^*$ as

$$\hat{f}_k - f^* = \Delta_{21} + (\Delta_1 + \Delta_3 + \Delta_{22} + \Delta_{23}) := \Delta_{21} + \Xi,$$

Next we bound the expected value of $\|\Xi\|_2^2$ over noise, $\mathbb{E}_{\boldsymbol{\epsilon}} \|\Xi\|_2^2$. Note that we have

$$\mathbb{E}_{\boldsymbol{\epsilon}} \|\boldsymbol{y}\|_2^2 = \mathbb{E}_{\boldsymbol{\epsilon}} \|\boldsymbol{y}^* + \boldsymbol{\epsilon}\|_2^2 \leq 2\boldsymbol{y}^{*\top} \boldsymbol{y}^* + 2\mathbb{E}_{\boldsymbol{\epsilon}} \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon} = O(n). \tag{3.23}$$

By Markov's inequality, with probability $1-\delta$ over random initialization, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2 \leq& \left(\mathbb{E}_{\boldsymbol{\epsilon}}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^2\right)^{\frac{1}{2}}\\ \leq& \left(\frac{3\mathbb{E}_{\boldsymbol{W}(0),\boldsymbol{a}}\left[\boldsymbol{u}(0)^\top\boldsymbol{u}(0)+\boldsymbol{y}^{*\top}\boldsymbol{y}^*+\mathbb{E}_{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}\right]}{\delta}\right)^{\frac{1}{2}}\\ =& O\left(\sqrt{\frac{n(1+\tau^2)}{\delta}}\right)=O\left(\sqrt{\frac{n}{\delta}}\right),\end{aligned} \tag{3.24}$$

where the last equality of 3.24 is because $\tau^2 \lesssim 1$. By (3.15), (3.16), (3.20), (3.22), (3.23) and (3.24), $\mathbb{E}_{\boldsymbol{\epsilon}}\|\Xi\|_2^2$ can be upper bounded as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\Xi\|_2^2 \leq& 4\mathbb{E}_{\boldsymbol{\epsilon}}(\|\Delta_1\|_2^2+\|\Delta_3\|_2^2+\|\Delta_{22}\|_2^2+\|\Delta_{23}\|_2^2)\\ =&\mathbb{E}_{\boldsymbol{\epsilon}}\left[O\left(\frac{n^2\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^4}{m\tau^2\lambda_0^4\delta^2}\right)+O\left(\frac{1}{n}\right)+O\left(\frac{n\log(n/\delta)\|\boldsymbol{y}\|_2^2}{m\lambda_0^2}\right)\right]+4\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{23}\|_2^2\\ \leq& O\left(\frac{n^4}{m\tau^2\lambda_0^4\delta^4}\right)+O\left(\frac{1}{n}\right)+O\left(\frac{n^2\log(n/\delta)}{m\lambda_0^2\delta}\right)+O\left(\frac{n^2\tau^2}{\lambda_0^2\delta^2}\right)+\\ &+\mathbb{E}_{\boldsymbol{\epsilon}}\left[O\left(\frac{n^{3/2}\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^3}{m^{1/2}\tau\lambda_0^3\delta}\right)+O\left(\frac{n^6\|\boldsymbol{y}-\boldsymbol{u}(0)\|_2^4}{m\tau^2\lambda_0^6\delta^2}\right)\right]\\ =& O\left(\frac{n^4}{m\tau^2\lambda_0^4\delta^4}\right)+O\left(\frac{1}{n}\right)+O\left(\frac{n^2\log(n/\delta)}{m\lambda_0^2\delta}\right)+O\left(\frac{n^2\tau^2}{\lambda_0^2\delta^2}\right)\\ &+O\left(\frac{n^3}{\sqrt{m}\tau\lambda_0^3\delta^{5/2}}\right)+O\left(\frac{n^8}{m\tau^2\lambda_0^6\delta^4}\right)\\ =& O\left(\frac{1}{n}\right)+O\left(\frac{n^2\tau^2}{\lambda_0^2\delta^2}\right)+\frac{\text{poly}\left(n,\frac{1}{\lambda_0},\frac{1}{\delta}\right)}{m^{\frac{1}{2}}\tau}.\end{aligned}$$

In the following, we will evaluate Δ_{21} and discuss how the iteration number k would affect the L_2 estimation error $\|\hat{f}_k - f^*\|_2^2$.

Case 1: The iteration number k cannot be too small By taking expectation of $\|\Delta_{21}\|_2^2$ over the noise, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}\|_2^2 =& \int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{X})\left[(\boldsymbol{H}^\infty)^{-1}-\boldsymbol{G}_k)\boldsymbol{y}^*\boldsymbol{y}^{*\top}((\boldsymbol{H}^\infty)^{-1}-\boldsymbol{G}_k)+\boldsymbol{G}_k^2\right]h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x}\\ =& \int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{M}_k(\boldsymbol{H}^\infty)^{-1}h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{M}_k =&(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k\boldsymbol{S}(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k+(\boldsymbol{I}-(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k)^2\\ =&[(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k-(\boldsymbol{S}+\boldsymbol{I})^{-1}](\boldsymbol{S}+\boldsymbol{I})[(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k-(\boldsymbol{S}+\boldsymbol{I})^{-1}]+\boldsymbol{I}-(\boldsymbol{S}+\boldsymbol{I})^{-1}\end{aligned} \tag{3.25}$$

and $\boldsymbol{S}=\boldsymbol{y}^*\boldsymbol{y}^{*\top}$. If $k\geq C_0\left(\frac{\log n}{\eta\lambda_0}\right)$ for some constant $C_0>1$, we have

$$(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k\leq(1-\eta\lambda_0)^k\boldsymbol{I}\leq\exp\{-\eta\lambda_0 k\}\boldsymbol{I}\leq\exp\{-C_0\log n\}\boldsymbol{I}=\frac{1}{n^{C_0}}\boldsymbol{I},$$

Since $1+\|\boldsymbol{y}^*\|_2^2\leq C_1 n$ for some constant C_1, we have

$$\lambda_{\max}\left(\frac{1}{n^{C_0}}(\boldsymbol{S}+\boldsymbol{I})\right)=\frac{1+\|\boldsymbol{y}^*\|_2^2}{n^{C_0}}\leq\frac{C_1}{n^{C_0-1}}<1.$$

By Lemma 3.6.10 (a), we have

$$(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k\leq\frac{1}{n^{C_0}}\boldsymbol{I}<(\boldsymbol{S}+\boldsymbol{I})^{-1}.$$

Therefore, we have

$$(\boldsymbol{S}+\boldsymbol{I})^{-1}-(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k\geq(\boldsymbol{S}+\boldsymbol{I})^{-1}-\frac{1}{n^{C_0}}\boldsymbol{I},$$

where $(\boldsymbol{S}+\boldsymbol{I})^{-1}-(\boldsymbol{I}-\eta\boldsymbol{H}^\infty)^k$ and $(\boldsymbol{S}+\boldsymbol{I})^{-1}-n^{-C_0}\boldsymbol{I}$ are positive definite matrices. It is also obvious that the two matrices are exchangeable. By Lemma 3.6.10 (b) and (3.25), we have

$$\boldsymbol{M}_k\geq\left(1-\frac{1}{n^{C_0}}\right)^2\boldsymbol{I}+\frac{1}{n^{2C_0}}\boldsymbol{S}.$$

Then we have

$$\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}\|_2^2\geq\left(1-\frac{1}{n^{C_0}}\right)^2 I_1+\frac{1}{n^{2C_0}}I_2\geq c_0 I_1$$

where $c_0 \in (0,1)$ is a constant,

$$I_1 = \int h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty)^{-2} h(\boldsymbol{X}, \boldsymbol{x}) d\boldsymbol{x}, \quad \text{and} \quad I_2 = \int [h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty)^{-1} \boldsymbol{y}^*]^2 d\boldsymbol{x}.$$

By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}} \|\hat{f}_k - f^*\|_2^2 =& \mathbb{E}_{\boldsymbol{\epsilon}} \|\Delta_{21} + \Xi\|_2^2 \\
\geq& \frac{1}{2} \mathbb{E}_{\boldsymbol{\epsilon}} \|\Delta_{21}\|_2^2 - \mathbb{E}_{\boldsymbol{\epsilon}} \|\Xi\|_2^2 \\
\geq& \frac{c_0}{2} I_1 - O\left(\frac{1}{n}\right) - O\left(\frac{n^2 \tau^2}{\lambda_0^2 \delta^2}\right) - \frac{\text{poly}\left(n, \frac{1}{\lambda_0}, \frac{1}{\delta}\right)}{m^{\frac{1}{2}} \tau}.
\end{aligned} \tag{3.26}$$

Let $\tau \leq C_3 \frac{\lambda_0 \delta}{n} \|(\boldsymbol{H}^\infty)^{-1} h(\boldsymbol{X}, \cdot)\|_2$ for some constant $C_3 > 0$ such that the third term of (3.26) is bounded by $\frac{c_0}{4} \|(\boldsymbol{H}^\infty)^{-1} h(\boldsymbol{X}, \cdot)\|_2^2$. Therefore, $\mathbb{E}_{\boldsymbol{\epsilon}} \|\hat{f}_k - f^*\|_2^2$ can be lower bounded as

$$\mathbb{E}_{\boldsymbol{\epsilon}} \|\hat{f}_k - f^*\|_2^2 \geq C_1^* \|(\boldsymbol{H}^\infty)^{-1} h(\boldsymbol{X}, \cdot)\|_2^2 - O\left(\frac{1}{n}\right), \tag{3.27}$$

where $C_1^* > 0$ is a constant. Note that I_1 is $\mathbb{E}_\epsilon \|\hat{f}_\infty - g^*\|_2^2$, where $g^* \equiv 0$ and $\hat{f}_\infty$ is the interpolated estimator of g^*, as in Theorem 3.2.2. Therefore, by Theorem 3.2.2, there exists a constant c_1 such that $\mathbb{E}_\epsilon \|\hat{f}_\infty - g^*\|_2^2 \geq c_1$, which implies $I_1 \geq c_1$. Taking n large enough such that the second term in (3.27) is smaller than $C_1^* c_1$, we finish the proof of the case that k is large.

Case 2: The iteration number k cannot be too large We can rewrite Δ_{21} as

$$\begin{aligned}
\Delta_{21} =& h(\boldsymbol{x}, \boldsymbol{X}) \boldsymbol{G}_k (\boldsymbol{y}^* + \boldsymbol{\epsilon}) - h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty)^{-1} \boldsymbol{y}^* \\
=& \Delta_{21}^* - h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty)^{-1} \boldsymbol{y}^*.
\end{aligned}$$

Since

$$\boldsymbol{G}_k = \sum_{\mathrm{j}=0}^{k-1} \eta (\boldsymbol{I} - \eta \boldsymbol{H}^\infty)^{\mathrm{j}} = \sum_{\mathrm{j}=0}^{k-1} \eta \sum_{\mathrm{i}=1}^{n} (1 - \eta \lambda_{\mathrm{i}})^{\mathrm{j}} \boldsymbol{v}_{\mathrm{i}} \boldsymbol{v}_{\mathrm{i}}^\top \leq \eta k \boldsymbol{I},$$

we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}^*\|_2^2 =& \int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{X})\boldsymbol{G}_k(\boldsymbol{S}+\boldsymbol{I})\boldsymbol{G}_k h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x}\\ \leq& \eta^2k^2\int_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{S}+\boldsymbol{I})h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x}\\ =& \eta^2k^2\left(\int_{\boldsymbol{x}\in\Omega}\left[h(\boldsymbol{x},\boldsymbol{X})\boldsymbol{y}^*\right]^2 d\boldsymbol{x}+\|h(\cdot,\boldsymbol{X})\|_2^2\right)\\ =& O\left(\eta^2k^2n^2\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\hat{f}_k-f^*\|_2^2 =& \mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}^*+\Xi-h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2\\ \geq& \frac{1}{2}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2-\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}^*+\Xi\|_2^2\\ \geq& \frac{1}{2}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2-2\mathbb{E}_{\boldsymbol{\epsilon}}\|\Delta_{21}^*\|_2^2-2\mathbb{E}_{\boldsymbol{\epsilon}}\|\Xi\|_2^2\\ \geq& \frac{1}{2}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2-O\left(\eta^2k^2n^2\right)\\ &-O\left(\frac{1}{n}\right)-O\left(\frac{n^2\tau^2}{\lambda_0^2\delta^2}\right)-\frac{\text{poly}\left(n,\frac{1}{\lambda_0},\frac{1}{\delta}\right)}{m^{\frac{1}{2}}\tau}.\end{aligned}\tag{3.28}$$

Let $k\leq C_1\left(\frac{1}{\eta n}\right)$ for some constant $C_1>0$ such that the the second term of (3.28) can be bounded by $\frac{1}{8}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2$. Let $\tau\leq C_2\left(\frac{\delta\lambda_0}{n}\right)$ for some constant $C_2>0$ such that the fourth term in (3.28) can be bounded by $\frac{1}{8}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2$. Note that we can also choose m such that the fifth term in (3.28) is bounded by $\frac{1}{8}\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2$. Therefore, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\|\hat{f}_k-f^*\|_2^2 \geq& C_2^*\|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}^*\|_2^2-O\left(\frac{1}{n}\right)\\ \geq& C_3^*\|f^*\|_2^2-O\left(\frac{1}{n}\right),\end{aligned}\tag{3.29}$$

where the last inequality is because of Lemma 3.6.6, and $C_2^*>0$ is a constant. By taking n large enough such that the second term in (3.29) is smaller than $C_3^*\|f^*\|_2^2/2$, we finish the proof. ■

Proof of Theorem 3.2.2

Proof Let's first introduce the GD update for the kernel ridge regression. By the representer theorem [126], the kernel estimator can be written as

$$\hat{f}(\boldsymbol{x}) = \sum_{\mathrm{i}=1}^{n} \omega_{\mathrm{i}} h(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{i}}) := h(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{\omega},$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ is the coefficient vector. Consider using the squared loss

$$\Phi(\boldsymbol{\omega}) = \frac{1}{2} \sum_{\mathrm{i}=1}^{n} (\hat{f}(\boldsymbol{x}_{\mathrm{i}}) - y_{\mathrm{i}})^2.$$

Let $\boldsymbol{\omega}_k$ be the $\boldsymbol{\omega}$ at the k-th GD iteration and choose $\boldsymbol{\omega}_0 = \mathbf{0}$. Then, the GD update rule for estimating $\boldsymbol{\omega}$ can be expressed as

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k - \eta \left((\boldsymbol{H}^{\infty})^2 \boldsymbol{\omega} - \boldsymbol{H}^{\infty} \boldsymbol{y} \right) \tag{3.30}$$

In the formulation of the stopping rule, two quantities play an important role: first, the running sum of the step sizes $\alpha_{\mathrm{j}} := \sum_{\mathrm{i}=0}^{\mathrm{j}} \eta_{\mathrm{i}}$, and secondly, the eigenvalues $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_n \geq 0$ of the empirical kernel matrix H^{∞}, which are computable from the data. Recall the definition of the optimal stopping time k^* as in (3.6). The following lemma establishes the L_2 estimation results for $\hat{f}_{k^*}$ for kernels with polynomial eigendecay. ■

Lemma 3.6.11 (Corollary 1 in [110]) *Suppose that variables $\{\boldsymbol{x}_{\mathrm{i}}\}_{\mathrm{i}=1}^{n}$ are sampled i.i.d. and the kernel class $\mathcal{N}$ satisfies the polynomial eigenvalue decay $\lambda_{\mathrm{j}} \lesssim \mathrm{j}^{-2\nu}$ for some $\nu > 1/2$. Then there is a universal constant C such that*

$$\mathbb{E}\|\hat{f}_{k^*} - f^*\|_2^2 \leq C \left(\frac{\sigma^2}{n} \right)^{\frac{2\nu}{2\nu+1}}.$$

Moreover, if $\lambda_{\mathrm{j}} \asymp \mathrm{j}^{-2\nu}$ for all $\mathrm{j} = 1, 2, \ldots$, then for all iterations $k = 1, 2, \ldots$,

$$\mathbb{E}\|\hat{f}_{k^*} - f^*\|_2^2 \geq \frac{\sigma^2}{4} \min \left\{ 1, \ \frac{(\alpha_k)^{\frac{1}{2\nu}}}{n} \right\}.$$

By Lemma 3.1.1, apply Lemma 3.6.11 with $2\nu = d/(d-1)$ and the running sum of the step sizes $\alpha_k = k\eta$ gives the convergence rate.

Moreover, if $k \to \infty$, i.e., interpolation of training data, the lower bound result in Lemma 3.6.11 implies $\mathbb{E}\|f_{\hat{T}} - f^*\|_2^2 \gtrsim \sigma^2$ that doesn't converge to 0.

3.6.2 Proofs of main theorems in Section 3.3

Proof of Theorem 3.3.1

Proof Consider event

$$A_{ir} = \{\exists \boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w} - (1-\eta_2\mu)^k \boldsymbol{w}_r(0)\|_2 \le R, \mathbb{I}\{\boldsymbol{x}_i^\top \boldsymbol{w}_r(0) \ge 0\} \ne \mathbb{I}\{\boldsymbol{x}_i^\top \boldsymbol{w} \ge 0\}\},$$

where R will be determined later. Set $S_i = \{r \in [m] : \mathbb{I}\{A_{ir}\} = 0\}$ and $S_i^\perp = [m] \backslash S_i$. Then A_{ir} happens if and only if $|\boldsymbol{w}_r(0)^\top \boldsymbol{x}_i| < R/(1-\eta_2\mu)^k$. By concentration inequality of Gaussian, we have $\mathbb{P}(A_{ir}) = \mathbb{P}(|\boldsymbol{w}_r(0)^\top \boldsymbol{x}_i| < R/(1-\eta_2\mu)^k \le \frac{2R}{\sqrt{2\pi}\tau(1-\eta_2\mu)^k}$. Thus, it follows the union bound inequality that with probability at least $1-\delta$ we have

$$\sum_{i=1}^n |S_i^\perp| \le \frac{CmnR}{\delta(1-\eta_2\mu)^k}, \tag{3.31}$$

where C is a positive constant.

Let $\boldsymbol{u}_D(l) = (u_{D,1}(l), ..., u_{D,n}(l))^\top \in \mathbb{R}^n$ be the predictions on the points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ using the modified GD at the k-th iteration. We first study the difference between two predictions $\boldsymbol{u}_D(l+1)$ and $\boldsymbol{u}_D(l)$. For any $i \in [n]$, we have

$$\begin{aligned}
u_{D,i}(l+1) - (1-\eta_2\mu)u_{D,i}(l) =& \frac{1}{\sqrt{m}} \sum_{r=1}^m a_r(\sigma(\boldsymbol{w}_{D,r}(l+1)^\top \boldsymbol{x}_i) - (1-\eta_2\mu)\sigma(\boldsymbol{w}_{D,r}(l)^\top \boldsymbol{x}_i)) \\
=& \frac{1}{\sqrt{m}} \sum_{r \in S_i^\perp} a_r(\sigma(\boldsymbol{w}_{D,r}(l+1)^\top \boldsymbol{x}_i) - (1-\eta_2\mu)\sigma(\boldsymbol{w}_{D,r}(l)^\top \boldsymbol{x}_i)) \\
&+ \frac{1}{\sqrt{m}} \sum_{r \in S_i} a_r(\sigma(\boldsymbol{w}_{D,r}(l+1)^\top \boldsymbol{x}_i) - (1-\eta_2\mu)\sigma(\boldsymbol{w}_{D,r}(l)^\top \boldsymbol{x}_i)) \\
=& I_{1,i}(l) + I_{2,i}(l).
\end{aligned} \tag{3.32}$$

The first term $I_{1,\mathrm{i}}(l)$ can be bounded by

$$\begin{aligned}
I_{1,\mathrm{i}}(l) =& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}^{\perp}} a_r(\sigma(\boldsymbol{w}_{D,r}(l+1)^{\top}\boldsymbol{x}_{\mathrm{i}}) - (1-\eta_2\mu)\sigma(\boldsymbol{w}_{D,r}(l)^{\top}\boldsymbol{x}_{\mathrm{i}})) \\
\leq& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}^{\perp}} \left|(\boldsymbol{w}_{D,r}(l+1) - (1-\eta_2\mu)\boldsymbol{w}_{D,r}(l))^{\top}\boldsymbol{x}_{\mathrm{i}}\right| \\
\leq& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}^{\perp}} \|\boldsymbol{w}_{D,r}(l+1) - (1-\eta_2\mu)\boldsymbol{w}_{D,r}(l)\|_2 \\
=& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}^{\perp}} \|\frac{\eta_1}{\sqrt{m}} a_r \sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l) - y_{\mathrm{j}})\mathbb{I}_{r,\mathrm{j}}(l)\boldsymbol{x}_{\mathrm{j}}\|_2 \\
\leq& \frac{\eta_1}{m} \sum_{r\in S_{\mathrm{i}}^{\perp}} \sum_{\mathrm{j}=1}^{n} |u_{D,\mathrm{j}}(l) - y_{\mathrm{j}}| \\
\leq& \frac{\eta_1\sqrt{n}|S_{\mathrm{i}}^{\perp}|}{m}\|\boldsymbol{u}_D(l) - \boldsymbol{y}\|_2.
\end{aligned} \tag{3.33}$$

In (3.33), the second and the last inequalities are by the Cauchy-Schwarz inequality. The second term $I_{2,\mathrm{i}}(l)$ can be bounded by

$$\begin{aligned}
I_{2,\mathrm{i}}(l) =& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}} a_r(\sigma(\boldsymbol{w}_{D,r}(l+1)^{\top}\boldsymbol{x}_{\mathrm{i}}) - (1-\eta_2\mu)\sigma(\boldsymbol{w}_{D,r}(l)^{\top}\boldsymbol{x}_{\mathrm{i}})) \\
=& \frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}} a_r\mathbb{I}_{r,\mathrm{i}}(l)(\boldsymbol{w}_{D,r}(l+1) - (1-\eta_2\mu)\boldsymbol{w}_{D,r}(l))^{\top}\boldsymbol{x}_{\mathrm{i}} \\
=& -\frac{1}{\sqrt{m}} \sum_{r\in S_{\mathrm{i}}} a_r\mathbb{I}_{r,\mathrm{i}}(l) \left(\frac{\eta_1}{\sqrt{m}} a_r \sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l) - y_{\mathrm{j}})\mathbb{I}_{r,\mathrm{j}}(l)\boldsymbol{x}_{\mathrm{j}}\right)^{\top} \boldsymbol{x}_{\mathrm{i}} \\
=& -\frac{\eta_1}{m} \sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l) - y_{\mathrm{j}})\boldsymbol{x}_{\mathrm{j}}^{\top}\boldsymbol{x}_{\mathrm{i}} \sum_{r\in S_{\mathrm{i}}} \mathbb{I}_{r,\mathrm{i}}(l)\mathbb{I}_{r,\mathrm{j}}(l) \\
=& -\eta_1 \sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l) - y_{\mathrm{j}})\boldsymbol{H}_{\mathrm{ij}}(l) + I_{3,\mathrm{i}}(l),
\end{aligned} \tag{3.34}$$

where

$$I_{3,\mathrm{i}}(l) = \frac{\eta_1}{m} \sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l) - y_{\mathrm{j}})\boldsymbol{x}_{\mathrm{j}}^{\top}\boldsymbol{x}_{\mathrm{i}} \sum_{r\in S_{\mathrm{i}}^{\perp}} \mathbb{I}_{r,\mathrm{i}}(l)\mathbb{I}_{r,\mathrm{j}}(l).$$

The term $I_{3,\mathrm{i}}(l)$ in (3.34) can be bounded by

$$\begin{aligned}|I_{3,\mathrm{i}}(l)| \leq & \left|\frac{\eta_1}{m}\sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l)-y_{\mathrm{j}})\boldsymbol{x}_{\mathrm{j}}^{\top}\boldsymbol{x}_{\mathrm{i}}\sum_{r\in S_{\mathrm{i}}^{\perp}}\mathbb{I}_{r,\mathrm{i}}(l)\mathbb{I}_{r,\mathrm{j}}(l)\right| \\ \leq & \frac{\eta_1}{m}|S_{\mathrm{i}}^{\perp}|\sum_{\mathrm{j}=1}^{n}|u_{D,\mathrm{j}}(l)-y_{\mathrm{j}}| \\ \leq & \frac{\eta_1\sqrt{n}|S_{\mathrm{i}}^{\perp}|}{m}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2. \end{aligned} \tag{3.35}$$

Plugging (3.33) and (3.34) into (3.32), we have

$$u_{D,\mathrm{i}}(l+1)-(1-\eta_2\mu)u_{D,\mathrm{i}}(l)=-\eta_1\sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l)-y_{\mathrm{j}})\boldsymbol{H}_{\mathrm{ij}}(l)+I_{1,\mathrm{i}}(l)+I_{3,\mathrm{i}}(l),$$

which leads to

$$\boldsymbol{u}_D(l+1)-(1-\eta_2\mu)\boldsymbol{u}_D(l)=-\eta_1\boldsymbol{H}(l)(\boldsymbol{u}_D(l)-\boldsymbol{y})+\boldsymbol{I}(l), \tag{3.36}$$

where $\boldsymbol{I}(l)=(I_{1,1}(l)+I_{3,1}(l),...,I_{1,n}(l)+I_{3,n}(l))^{\top}$. By the triangle inequality, we have

$$\|\boldsymbol{u}_D(l+1)-(1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2 \leq \|\eta_1\boldsymbol{H}(l)(\boldsymbol{u}_D(l)-\boldsymbol{y})\|_2+\|\boldsymbol{I}(l)\|_2. \tag{3.37}$$

By (3.31), (3.33), and (3.35), the term $\|\boldsymbol{I}(l)\|_2$ in (3.37) can be bounded by

$$\begin{aligned}\|\boldsymbol{I}(l)\|_2 \leq & \sum_{\mathrm{i}=1}^{n}|I_{3,\mathrm{i}}(l)|+|I_{1,\mathrm{i}}(l)| \leq \sum_{\mathrm{i}=1}^{n}\frac{2\eta_1\sqrt{n}|S_{\mathrm{i}}^{\perp}|}{m}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2 \\ \leq & \frac{2\eta_1\sqrt{n}}{m}\frac{CmnR}{\delta(1-\eta_2\mu)^k}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2=\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2. \end{aligned} \tag{3.38}$$

Gershgorin's theorem [127] implies

$$\lambda_{\max}(H(l)) \leq \max_{\mathrm{j}}\sum_{\mathrm{i}=1}^{n}H_{\mathrm{ij}}(l) \leq n.$$

Therefore, the term $\|\eta_1 \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y})\|_2$ in (3.37) can be bounded by

$$\|\eta_1 \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y})\|_2 \leq \eta_1 \lambda_{\max}(H(l)) \|\boldsymbol{u}_D(l) - \boldsymbol{y}\|_2 \leq \eta_1 n \|\boldsymbol{u}_D(l) - \boldsymbol{y}\|_2.$$

By (3.37) and (3.38), $\|\boldsymbol{y} - \boldsymbol{u}_D(l+1)\|_2$ can be bounded by

$$\begin{aligned}
\|\boldsymbol{y} - \boldsymbol{u}_D(l+1)\|_2^2 =& \|\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2 - 2(\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l))^\top (\boldsymbol{u}_D(l+1) - (1-\eta_2\mu)\boldsymbol{u}_D(l)) \\
&+ \|\boldsymbol{u}_D(l+1) - (1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2 \\
=& \|\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2 + 2\eta_1 (\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l))^\top \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y}) \\
&- 2\eta_1 (\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l))^\top \boldsymbol{I}(l) + \|\boldsymbol{u}_D(l+1) - (1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2 \\
=& T_1 + T_2 + T_3 + T_4. \qquad (3.39)
\end{aligned}$$

The first term T_1 can be bounded by

$$\begin{aligned}
T_1 =& \|\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2 \\
=& \eta_2^2\mu^2 \|\boldsymbol{y}\|_2^2 + (1-\eta_2\mu)^2 \|\boldsymbol{y} - \boldsymbol{u}_D(l)\|_2^2 + 2\eta_2\mu(1-\eta_2\mu)\boldsymbol{y}^\top (\boldsymbol{y} - \boldsymbol{u}_D(l)) \\
\leq& (\eta_2^2\mu^2 + \eta_2\mu)\|\boldsymbol{y}\|_2^2 + (1+\eta_2\mu)(1-\eta_2\mu)^2 \|\boldsymbol{y} - \boldsymbol{u}_D(l)\|_2^2. \qquad (3.40)
\end{aligned}$$

The second term T_2 can be bounded by

$$\begin{aligned}
T_2 =& 2\eta_1 (\boldsymbol{y} - (1-\eta_2\mu)\boldsymbol{u}_D(l))^\top \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y}) \\
=& 2\eta_1 (1-\eta_2\mu)(\boldsymbol{y} - \boldsymbol{u}_D(l))^\top \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y}) + 2\eta_1\eta_2\mu \boldsymbol{y}^\top \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y}) \\
=& -2\eta_1 (1-\eta_2\mu)(\boldsymbol{y} - \boldsymbol{u}_D(l))^\top \boldsymbol{H}(l)(\boldsymbol{y} - \boldsymbol{u}_D(l)) + 2\eta_1\eta_2\mu \boldsymbol{y}^\top \boldsymbol{H}(l)(\boldsymbol{u}_D(l) - \boldsymbol{y}) \\
\leq& 4\eta_1\eta_2\mu n \|\boldsymbol{y}\|_2^2 + 4\eta_1\eta_2\mu n \|\boldsymbol{u}_D(l) - \boldsymbol{y}\|_2^2.
\end{aligned}$$

Using (3.38), the third term T_3 can be bounded by

$$\begin{aligned}
T_3 =& -2\eta_1(\boldsymbol{y}-(1-\eta_2\mu)\boldsymbol{u}_D(l))^\top \boldsymbol{I}(l)\\
=& -2\eta_1(1-\eta_2\mu)(\boldsymbol{y}-\boldsymbol{u}_D(l))^\top \boldsymbol{I}(l)+2\eta_1\eta_2\mu\boldsymbol{y}^\top \boldsymbol{I}(l)\\
\le& 2\eta_1(1-\eta_2\mu)\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2+4\eta_1\eta_2\mu\|\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu\|\boldsymbol{I}(l)\|_2^2\\
\le& 2\eta_1(1-\eta_2\mu)\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu\|\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2.
\end{aligned}$$

The fourth term T_4 can be bounded by

$$\begin{aligned}
T_4 =& \|\boldsymbol{u}_D(l+1)-(1-\eta_2\mu)\boldsymbol{u}_D(l)\|_2^2\\
\le& 2\|\eta_1\boldsymbol{H}(l)(\boldsymbol{u}_D(l)-\boldsymbol{y})\|_2^2+2\|\boldsymbol{I}(l)\|_2^2\\
\le& 2\eta_1^2n^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2+2\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2. \qquad (3.41)
\end{aligned}$$

Plugging (3.40) - (3.41) into (3.39), we have

$$\begin{aligned}
&\|\boldsymbol{y}-\boldsymbol{u}_D(l+1)\|_2^2\\
\le&(\eta_2^2\mu^2+\eta_2\mu)\|\boldsymbol{y}\|_2^2+(1+\eta_2\mu)(1-\eta_2\mu)^2\|\boldsymbol{y}-\boldsymbol{u}_D(l)\|_2^2+4\eta_1\eta_2\mu n\|\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu n\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2\\
&+2\eta_1(1-\eta_2\mu)\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu\|\boldsymbol{y}\|_2^2+4\eta_1\eta_2\mu\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2\\
&+2\eta_1^2n^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2+2\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2\\
=&a_1\|\boldsymbol{y}\|_2^2+a_2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2, \qquad (3.42)
\end{aligned}$$

where

$$\begin{aligned}
a_1 =&(\eta_2^2\mu^2+\eta_2\mu)+4\eta_1\eta_2\mu n+4\eta_1\eta_2\mu \le 2\eta_2\mu+8\eta_1\eta_2\mu n,\\
a_2 =&(1+\eta_2\mu)(1-\eta_2\mu)^2+4\eta_1\eta_2\mu n+2\eta_1(1-\eta_2\mu)\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\\
&+4\eta_1\eta_2\mu\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2+2\eta_1^2n^2+2\left(\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}\right)^2\\
\le&1-\left(\eta_2\mu-4\eta_1\eta_2\mu n-2\eta_1\frac{2C\eta_1 n^{3/2}R}{\delta(1-\eta_2\mu)^k}-2\eta_1^2n^2\right)\\
=&1-\nu_0.
\end{aligned}$$

By the conditions imposed on η_1,η_2,μ,m, the dominating terms in a_1 and ν_0 are both $\eta_2\mu$. Thus $a_1=o(1/n)$, $\nu_0=o(1/n)$ and $a_1/\nu_0=O(1)$. Using (3.42) iteratively, we have

$$\begin{aligned}
\|\boldsymbol{y}-\boldsymbol{u}_D(l+1)\|_2^2 \le& a_1\|\boldsymbol{y}\|_2^2+a_2\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2^2\\
\le&\ldots\le\sum_{\mathrm{i}=0}^{l}(1-\nu_0)^{\mathrm{i}}(a_1\|\boldsymbol{y}\|_2^2)+(1-\nu_0)^{l+1}\|\boldsymbol{y}-\boldsymbol{u}_D(0)\|_2^2\\
\le&\frac{a_1\|\boldsymbol{y}\|_2^2}{\nu_0}+(1-\nu_0)^{l+1}\|\boldsymbol{y}-\boldsymbol{u}_D(0)\|_2^2.
\end{aligned}$$

By the modified GD rule, we have

$$\boldsymbol{w}_{D,r}(l+1)-(1-\eta_2\mu)\boldsymbol{w}_{D,r}(l)=-\frac{\eta_1}{\sqrt{m}}a_r\sum_{\mathrm{j}=1}^{n}(u_{D,\mathrm{j}}(l)-y_{\mathrm{j}})\mathbb{I}_{r,\mathrm{j}}(l)\boldsymbol{x}_{\mathrm{j}},$$

which implies

$$\|\boldsymbol{w}_{D,r}(l+1)-(1-\eta_2\mu)\boldsymbol{w}_{D,r}(l)\|_2\le\frac{\eta_1\sqrt{n}}{\sqrt{m}}\|\boldsymbol{u}_D(l)-\boldsymbol{y}\|_2\le\frac{C\eta_1 n}{\sqrt{m}} \tag{3.43}$$

for some constant C. Using (3.43) iteratively yields

$$\begin{aligned}
&\|\boldsymbol{w}_{D,r}(l+1)-(1-\eta_2\mu)^{l+1}\boldsymbol{w}_{D,r}(0)\|_2\\
\leq&\|\boldsymbol{w}_{D,r}(l+1)-(1-\eta_2\mu)\boldsymbol{w}_{D,r}(l)\|_2+\|(1-\eta_2\mu)\boldsymbol{w}_{D,r}(0)-(1-\eta_2\mu)^{l+1}\boldsymbol{w}_{D,r}(l)\|_2\\
\leq&\frac{C\eta_1 n}{\sqrt{m}}+(1-\eta_2\mu)\|\boldsymbol{w}_{D,r}(l)-(1-\eta_2\mu)^{l}\boldsymbol{w}_{D,r}(0)\|_2\\
\leq&\ldots\leq\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\frac{C\eta_1 n}{\sqrt{m}}\leq\frac{C\eta_1 n}{\eta_2\mu\sqrt{m}}.
\end{aligned}\tag{3.44}$$

By similar approach as in the proof of Lemma C.2 of [16], we can show that with probability at least $1-\delta$ with respect to random initialization,

$$\|\boldsymbol{Z}(l)-\boldsymbol{Z}(0)\|_F^2\leq\frac{2nR}{\sqrt{2\pi}\tau\delta(1-\eta_2\mu)^k}+\frac{n}{m}=O\left(\frac{\eta_1 n^2}{(1-\eta_2\mu)^k\eta_2\mu\sqrt{m}\delta^{3/2}\tau}\right),\forall l\in[k],$$

and

$$\|\boldsymbol{H}(l)-\boldsymbol{H}(0)\|_F\leq\frac{4n^2R}{\sqrt{2\pi}\tau}+\frac{2n^2\delta}{m}=O\left(\frac{\eta_1 n^3}{(1-\eta_2\mu)^k\eta_2\mu\sqrt{m}\delta^{3/2}\tau}\right),\forall l\in[k].$$

By Lemma C.3 of [16], we have with probability at least $1-\delta$ with respect to random initialization,

$$\|\boldsymbol{H}(0)-\boldsymbol{H}^{\infty}\|_F=O\left(\frac{n\sqrt{\log(n/\delta)}}{\sqrt{m}}\right).$$

By (3.36), we have

$$\begin{aligned}
\boldsymbol{u}_D(l+1)-(1-\eta_2\mu)\boldsymbol{u}_D(l)=&-\eta_1\boldsymbol{H}(l)(\boldsymbol{u}_D(l)-\boldsymbol{y})+\boldsymbol{I}(l)\\
=&-\eta_1\boldsymbol{H}^{\infty}(\boldsymbol{u}_D(l)-\boldsymbol{y})+\boldsymbol{I}(l)-\eta_1(\boldsymbol{H}(l)-\boldsymbol{H}^{\infty})(\boldsymbol{u}_D(l)-\boldsymbol{y}),
\end{aligned}$$

which yields

$$\boldsymbol{u}_D(l+1)-B=\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)(\boldsymbol{u}_D(l)-B)+\boldsymbol{I}(l)-\eta_1(\boldsymbol{H}(l)-\boldsymbol{H}^{\infty})(\boldsymbol{u}_D(l)-\boldsymbol{y}),\tag{3.45}$$

where

$$B = (\eta_2\mu I + \eta_1 \boldsymbol{H}^\infty)^{-1}\eta_1 \boldsymbol{H}^\infty \boldsymbol{y} = \eta_1 \boldsymbol{H}^\infty(\eta_2\mu I + \eta_1 \boldsymbol{H}^\infty)^{-1}\boldsymbol{y}.$$

Iteratively using (3.45), we have

$$\begin{aligned}
\boldsymbol{u}_D(l+1) - B =& ((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^{l+1}(\boldsymbol{u}_D(0) - B) \\
&+ \sum_{\mathrm{i}=0}^{l} ((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^{\mathrm{i}} (\boldsymbol{I}(l-\mathrm{i}) - \eta_1(\boldsymbol{H}(l-\mathrm{i}) - \boldsymbol{H}^\infty)(\boldsymbol{u}_D(l-\mathrm{i}) - \boldsymbol{y})) \\
=& ((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^{l+1}(\boldsymbol{u}_D(0) - B) + \mathrm{e}_l, \qquad (3.46)
\end{aligned}$$

where

$$\mathrm{e}_l = \sum_{\mathrm{i}=0}^{l} ((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^{\mathrm{i}} (\boldsymbol{I}(l-\mathrm{i}) - \eta_1(\boldsymbol{H}(l-\mathrm{i}) - \boldsymbol{H}^\infty)(\boldsymbol{u}_D(l-\mathrm{i}) - \boldsymbol{y})).$$

The term e_l can be bounded by

$$\begin{aligned}
\|\mathrm{e}_l\|_2 =& \|\sum_{\mathrm{i}=0}^{l} ((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^{\mathrm{i}} (\boldsymbol{I}(l-\mathrm{i}) - \eta_1(\boldsymbol{H}(l-\mathrm{i}) - \boldsymbol{H}^\infty)(\boldsymbol{u}_D(l-\mathrm{i}) - \boldsymbol{y}))\|_2 \\
\le& \sum_{\mathrm{i}=0}^{l} \|(1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty\|_2^{\mathrm{i}} (\|\boldsymbol{I}(l-\mathrm{i})\|_2 + \eta_1\|\boldsymbol{H}(l-\mathrm{i}) - \boldsymbol{H}^\infty\|_2\|\boldsymbol{u}_D(l-\mathrm{i}) - \boldsymbol{y}\|_2) \\
\le& \sum_{\mathrm{i}=0}^{l} (1-\eta_2\mu)^{\mathrm{i}} O\left(\frac{2C\eta_1^2 n^{5/2}}{\eta_2\mu\sqrt{m}\delta^{3/2}(1-\eta_2\mu)^k} + \frac{\eta_1^2 n^{7/2}}{(1-\eta_2\mu)^k\eta_2\mu\sqrt{m}\delta^2\tau}\right) \\
=& O\left(\frac{\eta_1^2 n^{7/2}}{\eta_2^2\mu^2\sqrt{m}\delta^2(1-\eta_2\mu)^k\tau}\right). \qquad (3.47)
\end{aligned}$$

By (3.46) and taking $l = k-1$, with probability at least $1-\delta$ with respect to the random initialization, the difference $\boldsymbol{u}_D(k) - B$ can be bounded by

$$\begin{aligned}
\|\boldsymbol{u}_D(k) - B\|_2 \le& \|((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^\infty)^k(\boldsymbol{u}_D(0) - B)\|_2 + \|\mathrm{e}_k\|_2 \\
=& O\left(\sqrt{n}(1-\eta_2\mu-\eta_1\lambda_0)^k + \frac{n^{7/2}}{\mu^2\sqrt{m}\delta^2(1-\eta_2\mu)^k\tau}\right) \\
=& O\left(\sqrt{n}(1-\eta_2\mu)^k + \frac{n^{7/2}}{\mu^2\sqrt{m}\delta^2(1-\eta_2\mu)^k\tau}\right).
\end{aligned}$$

This implies that

$$\|\boldsymbol{u}_D(k) - B\|_2 = O_{\mathbb{P}}\left(\sqrt{n}(1-\eta_2\mu)^k + \frac{n^{7/2}}{\mu^2\sqrt{m}(1-\eta_2\mu)^k\tau}\right).$$

By choosing $m = \text{poly}(n, 1/\tau, 1/\lambda_0)$ such that $\frac{n^{7/2}}{\mu^2\sqrt{m}(1-\eta_2\mu)^k\tau} \leq \sqrt{n}(1-\eta_2\mu)^k$, we finish the proof of (3.9).

Now consider $\text{vec}(\boldsymbol{W}_D(l+1))$. Direct calculation shows that

$$\begin{aligned}
\text{vec}(\boldsymbol{W}_D(l+1)) =&(1-\eta_2\mu)\text{vec}(\boldsymbol{W}_D(l)) - \eta_1\boldsymbol{Z}(l)(\boldsymbol{u}_D(l)-\boldsymbol{y}) \\
=&(1-\eta_2\mu)\text{vec}(\boldsymbol{W}_D(l)) - \eta_1\boldsymbol{Z}(0)(\boldsymbol{u}_D(l)-\boldsymbol{y}) - \eta_1(\boldsymbol{Z}(l)-\boldsymbol{Z}(0))(\boldsymbol{u}_D(l)-\boldsymbol{y}) \\
=&(1-\eta_2\mu)^{l+1}\text{vec}(\boldsymbol{W}_D(0)) - \eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}(\boldsymbol{u}_D(l-\mathrm{i})-\boldsymbol{y}) \\
&-\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\eta_1(\boldsymbol{Z}(l)-\boldsymbol{Z}(0))(\boldsymbol{u}_D(l)-\boldsymbol{y}). \qquad (3.48)
\end{aligned}$$

Plugging

$$\boldsymbol{u}_D(l+1) = \left((1-\eta_2\mu)I - \eta_1\boldsymbol{H}^{\infty}\right)^{l+1}(\boldsymbol{u}_D(0)-B) + \mathrm{e}_l + B$$

into (3.48), we have

$$
\begin{aligned}
&\mathrm{vec}(\boldsymbol{W}_D(l+1)) - (1-\eta_2\mu)^{l+1}\mathrm{vec}(\boldsymbol{W}_D(0)) \\
=&-\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)^{l-\mathrm{i}}(\boldsymbol{u}_D(0)-B) \\
&-\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}(\mathrm{e}_{l-\mathrm{i}-1}+B-\boldsymbol{y})-\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\eta_1(\boldsymbol{Z}(l)-\boldsymbol{Z}(0))(\boldsymbol{u}_D(l)-\boldsymbol{y}) \\
=&\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)^{l-\mathrm{i}}\eta_1\boldsymbol{H}^{\infty}(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y} \\
&-\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)^{l-\mathrm{i}}\boldsymbol{u}_D(0) \\
&-\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\mathrm{e}_{l-\mathrm{i}-1}-\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}(B-\boldsymbol{y}) \\
&-\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\eta_1(\boldsymbol{Z}(l)-\boldsymbol{Z}(0))(\boldsymbol{u}_D(l)-\boldsymbol{y}) \\
=&E_1-E_2+E_3-T_5-E_4. \qquad (3.49)
\end{aligned}
$$

Let

$$
\begin{aligned}
\boldsymbol{T}_l =&\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)^{l-\mathrm{i}} \\
=&(1-\eta_2\mu)^{l}\sum_{\mathrm{i}=0}^{l}\left(I-\frac{\eta_1}{(1-\eta_2\mu)}\boldsymbol{H}^{\infty}\right)^{\mathrm{i}} \qquad (3.50)
\end{aligned}
$$

and

$$
\boldsymbol{a}_1 =\eta_1\boldsymbol{H}^{\infty}(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}.
$$

The first term E_1 can be bounded by

$$\begin{aligned}\|E_1\|_2^2 =&\|\eta_1 \boldsymbol{Z}(0)\boldsymbol{T}_l \boldsymbol{a}_1\|_2^2\\ =&\eta_1^2 \boldsymbol{a}_1^\top \boldsymbol{T}_l \boldsymbol{Z}(0)^\top \boldsymbol{Z}(0)\boldsymbol{T}_l \boldsymbol{a}_1\\ =&\eta_1^2 \boldsymbol{a}_1^\top \boldsymbol{T}_l \boldsymbol{H}^\infty \boldsymbol{T}_l \boldsymbol{a}_1 + \eta_1^2 \boldsymbol{a}_1^\top \boldsymbol{T}_l (\boldsymbol{H}(0)-\boldsymbol{H}^\infty)\boldsymbol{T}_l \boldsymbol{a}_1\\ =&\eta_1^2 \boldsymbol{a}_1^\top \boldsymbol{T}_l \boldsymbol{H}^\infty \boldsymbol{T}_l \boldsymbol{a}_1 + \eta_1^2 O\left(\frac{n\sqrt{\log(n/\delta)}}{\sqrt{m}}\right)\boldsymbol{a}_1^\top \boldsymbol{T}_l^2 \boldsymbol{a}_1. \end{aligned} \tag{3.51}$$

By (3.50), we have

$$\boldsymbol{T}_l = (1-\eta_2\mu)^l \sum_{\mathrm{j}=1}^n \frac{1-(1-\frac{\eta_1}{(1-\eta_2\mu)}\lambda_\mathrm{j})^{l+1}}{\frac{\eta_1}{(1-\eta_2\mu)}\lambda_\mathrm{j}} \boldsymbol{v}_\mathrm{j}\boldsymbol{v}_\mathrm{j}^\top \preceq \frac{(1-\eta_2\mu)^l}{\eta_1\lambda_0}\boldsymbol{I},$$

and

$$\boldsymbol{T}_l \boldsymbol{H}^\infty \boldsymbol{T}_l = (1-\eta_2\mu)^{2l} \sum_{\mathrm{j}=1}^n \left(\frac{1-(1-\frac{\eta_1}{(1-\eta_2\mu)}\lambda_\mathrm{j})^{2l+2}}{\frac{\eta_1}{(1-\eta_2\mu)}\lambda_\mathrm{j}}\right)^2 \lambda_\mathrm{j}\boldsymbol{v}_\mathrm{j}\boldsymbol{v}_\mathrm{j}^\top \preceq \frac{(1-\eta_2\mu)^{l+1}}{\eta_1^2}(\boldsymbol{H}^\infty)^{-1}.$$

Therefore,

$$\begin{aligned}\eta_1^2 \boldsymbol{a}_1^\top \boldsymbol{T}_l \boldsymbol{H}^\infty \boldsymbol{T}_l \boldsymbol{a}_1 \leq& (1-\eta_2\mu)^{2l+2}\boldsymbol{a}_1^\top (\boldsymbol{H}^\infty)^{-1}\boldsymbol{a}_1,\\ \eta_1^2 O\left(\frac{n\sqrt{\log(n/\delta)}}{\sqrt{m}}\right)\boldsymbol{a}_1^\top \boldsymbol{T}_l^2 \boldsymbol{a}_1 \leq& O\left(\frac{n^2(1-\eta_2\mu)^{2l}\sqrt{\log(n/\delta)}}{\sqrt{m}\lambda_0^2}\right).\end{aligned}$$

Together with (3.51), we have

$$\|E_1\|_2^2 = (1-\eta_2\mu)^{2l+2}\boldsymbol{a}_1^\top (\boldsymbol{H}^\infty)^{-1}\boldsymbol{a}_1 + O\left(\frac{n^2(1-\eta_2\mu)^{2l}\sqrt{\log(n/\delta)}}{\sqrt{m}\lambda_0^2}\right). \tag{3.52}$$

By similar approach, the second term E_2 can be bounded by

$$\begin{aligned}\|E_2\|_2^2 =&\|\eta_1 \boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\left((1-\eta_2\mu)I-\eta_1\boldsymbol{H}^{\infty}\right)^{l-\mathrm{i}}\boldsymbol{u}_D(0)\|_2^2\\ =&\eta_1^2\boldsymbol{u}_D(0)^{\top}\boldsymbol{T}_1(l)\boldsymbol{Z}(0)^{\top}\boldsymbol{Z}(0)\boldsymbol{T}_1(l)\boldsymbol{u}_D(0)\\ =&\eta_1^2\boldsymbol{u}_D(0)^{\top}\boldsymbol{T}_1(l)\boldsymbol{H}^{\infty}\boldsymbol{T}_1(l)\boldsymbol{u}_D(0)+\eta_1^2\boldsymbol{u}_D(0)^{\top}\boldsymbol{T}_1(l)(\boldsymbol{H}(0)-\boldsymbol{H}^{\infty})\boldsymbol{T}_1(l)\boldsymbol{u}_D(0)\\ =&(1-\eta_2\mu)^{2l+2}\boldsymbol{u}_D(0)^{\top}(\boldsymbol{H}^{\infty})^{-1}\boldsymbol{u}_D(0)+O\left(\frac{n^2(1-\eta_2\mu)^{2l}\sqrt{\log(n/\delta)}}{\sqrt{m}\lambda_0^2}\right).\end{aligned}\tag{3.53}$$

By (3.47), the third term E_3 can be bounded by

$$\begin{aligned}\|E_3\|_2^2 =&\|\eta_1\boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\mathrm{e}_{l-\mathrm{i}-1}\|_2^2\\ =&\eta_1^2\left(\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\mathrm{e}_{l-\mathrm{i}-1}\right)^{\top}\boldsymbol{H}(0)\left(\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\mathrm{e}_{l-\mathrm{i}-1}\right)\\ =&O\left(\frac{\eta_1^6 n^8}{\eta_2^6\mu^6 m\delta^4(1-\eta_2\mu)^{2k}\tau^2}\right).\end{aligned}\tag{3.54}$$

The fourth term E_4 can be bounded by

$$\begin{aligned}\|E_4\|_2^2 =&\|\sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}\eta_1(\boldsymbol{Z}(l)-\boldsymbol{Z}(0))(\boldsymbol{u}_D(l)-\boldsymbol{y})\|_2^2\\ =&O\left(\frac{\eta_1^3 n^3}{(1-\eta_2\mu)^k\eta_2^3\mu^3\sqrt{m}\delta^{3/2}\tau}\right).\end{aligned}\tag{3.55}$$

Note that

$$\begin{aligned}B-\boldsymbol{y} =&\eta_1\boldsymbol{H}^{\infty}(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}-\boldsymbol{y}\\ =&(\eta_1\boldsymbol{H}^{\infty}-\eta_2\mu I-\eta_1\boldsymbol{H}^{\infty})(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}\\ =&-\eta_2\mu(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}.\end{aligned}$$

Therefore, the remaining term T_5 can be bounded by

$$\begin{aligned}\|T_5\|_2^2 =&\|\eta_1 \boldsymbol{Z}(0) \sum_{\mathrm{i}=0}^{l}(1-\eta_2\mu)^{\mathrm{i}}(B-\boldsymbol{y})\|_2^2 \\ \leq&\eta_1^2 \boldsymbol{y}^\top(\eta_2\mu I+\eta_1\boldsymbol{H}^\infty)^{-1}\boldsymbol{H}^\infty(\eta_2\mu I+\eta_1\boldsymbol{H}^\infty)^{-1}\boldsymbol{y} \\ \leq&\boldsymbol{y}^\top(\eta_2\mu/\eta_1 I+\boldsymbol{H}^\infty)^{-1}\boldsymbol{H}^\infty(\eta_2\mu/\eta_1 I+\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}.\end{aligned}$$

By the assumption that $\eta_2 \asymp \eta_1$, the term T_5 can be further bounded by

$$\|T_5\|_2^2 \leq \boldsymbol{y}^\top(C\mu I+\boldsymbol{H}^\infty)^{-1}\boldsymbol{H}^\infty(C\mu I+\boldsymbol{H}^\infty)^{-1}\boldsymbol{y}. \tag{3.56}$$

The right-hand side of (3.56) is $\|\hat{f}\|_{\mathcal{N}}^2$, where $\hat{f}$ is defined in (3.4). The term $\|\hat{f}\|_{\mathcal{N}}^2$ can be bounded by some constant as in Theorem 3.1.2. This also implies

$$\boldsymbol{a}_1^\top(\boldsymbol{H}^\infty)^{-1}\boldsymbol{a}_1 = \eta_1^2\boldsymbol{y}^\top(\eta_2\mu I+\eta_1\boldsymbol{H}^\infty)^{-1}\boldsymbol{H}^\infty(\eta_2\mu I+\eta_1\boldsymbol{H}^\infty)^{-1}\boldsymbol{y} = O(1). \tag{3.57}$$

Note also that

$$\boldsymbol{u}_D(0)^\top(\boldsymbol{H}^\infty)^{-1}\boldsymbol{u}_D(0) = O\left(\frac{n\tau^2}{\lambda_0}\right). \tag{3.58}$$

By the assumptions of Theorem 3.3.1, plugging (3.51)-(3.58) into (3.49), and taking the iteration number at k, we can conclude that

$$\begin{aligned}&\|\mathrm{vec}(\boldsymbol{W}_D(k)) - (1-\eta_2\mu)^k\mathrm{vec}(\boldsymbol{W}_D(0))\|_2^2 \\ =&O((1-\eta_2\mu)^{2k}) + O\left(\frac{n^2(1-\eta_2\mu)^{2k-2}\sqrt{\log(n/\delta)}}{\sqrt{m}\lambda_0^2}\right) \\ &+ O\left(\frac{n\tau^2}{\lambda_0}(1-\eta_2\mu)^{2k}\right) + O\left(\frac{n^2(1-\eta_2\mu)^{2k-2}\sqrt{\log(n/\delta)}}{\sqrt{m}\lambda_0^2}\right) \\ &+ O\left(\frac{n^8}{\mu^6 m\delta^4(1-\eta_2\mu)^{2k}\tau^2}\right) + O\left(\frac{n^3}{(1-\eta_2\mu)^k\mu^3\sqrt{m}\delta^{3/2}\tau}\right) + O(1) \\ =&O(1),\end{aligned} \tag{3.59}$$

where the last equality is because we can select some polynomials such that all the terms in (3.59) except the $O(1)$ term converge to zero, and $\exp(-2\eta_2\mu k) \leq (1-\eta_2\mu)^k \leq \exp(-\eta_2\mu k)$ for sufficiently large n. This finishes the proof of (3.10) in Theorem 3.3.1. ■

Proof of Theorem 3.3.4

Proof For notational simplification, we use $\hat{f}_k = f_{\boldsymbol{W}(k),\boldsymbol{a}}$. Similar to the proof of Theorem 3.2.1, we define

$$\tilde{f}_k(\boldsymbol{x}) = \mathrm{vec}(\boldsymbol{W}_D(k))^\top \boldsymbol{z}_0(\boldsymbol{x}),$$

where $\boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})|_{\boldsymbol{W}_D=\boldsymbol{W}_D(0)}$. Then we can write the following decomposition

$$\begin{aligned}\hat{f}_k(\boldsymbol{x}) - f^*(\boldsymbol{x}) =& (\hat{f}_k(\boldsymbol{x}) - \tilde{f}_k(\boldsymbol{x})) + (\tilde{f}_k(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})) + (\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x})) \\ =& \Delta_1(\boldsymbol{x}) + \Delta_2(\boldsymbol{x}) + \Delta_3(\boldsymbol{x}),\end{aligned}$$

where $\hat{f}$ is as in (3.4). It follows from Theorem 3.1.2 that

$$\|\Delta_3\|_2^2 = O_{\mathbb{P}}\left(n^{-\frac{d}{2d-1}}\right). \tag{3.60}$$

Next, we consider Δ_1. From (3.44), it can be seen that

$$\|\boldsymbol{w}_{D,r}(k) - (1-\eta_2\mu)^k \boldsymbol{w}_{D,r}(0)\|_2 \leq \frac{C\eta_1 n}{\eta_2\mu\sqrt{m}}.$$

Define event

$$B_{D,r}(\boldsymbol{x}) = \{|(1-\eta_2\mu)^k \boldsymbol{w}_{D,r}(0)^\top \boldsymbol{x}| \leq R_1\}, \forall r \in [m],$$

where $R_1 = \frac{C\eta_1 n}{\eta_2 \mu \sqrt{m}}$. If $\mathbb{I}\{B_{D,r}(\boldsymbol{x})\} = 0$, then we have $\mathbb{I}_{r,k}(\boldsymbol{x}) = \mathbb{I}_{r,0}(\boldsymbol{x})$, where $\mathbb{I}_{r,k}(\boldsymbol{x}) = \mathbb{I}\{\boldsymbol{w}_{D,r}(k)^\top \boldsymbol{x} \geq 0\}$. Therefore, for any fixed $\boldsymbol{x}$,

$$
\begin{aligned}
|\Delta_1(\boldsymbol{x})| &= |\hat{f}_k(\boldsymbol{x}) - \tilde{f}_k(\boldsymbol{x})| \\
&= \left| \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r (\mathbb{I}_{r,k}(\boldsymbol{x}) - \mathbb{I}_{r,0}(\boldsymbol{x})) \boldsymbol{w}_{D,r}(k)^\top \boldsymbol{x} \right| \\
&= \left| \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \mathbb{I}\{B_{D,r}(\boldsymbol{x})\} (\mathbb{I}_{r,k}(\boldsymbol{x}) - \mathbb{I}_{r,0}(\boldsymbol{x})) \boldsymbol{w}_{D,r}(k)^\top \boldsymbol{x} \right| \\
&\leq \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_{D,r}(\boldsymbol{x})\} |\boldsymbol{w}_{D,r}(k)^\top \boldsymbol{x}| \\
&\leq \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_{D,r}(\boldsymbol{x})\} \left(|(1-\eta_2\mu)^k \boldsymbol{w}_{D,r}(0)^\top \boldsymbol{x}| + |\boldsymbol{w}_{D,r}(k)^\top \boldsymbol{x} - (1-\eta_2\mu)^k \boldsymbol{w}_r(0)^\top \boldsymbol{x}| \right) \\
&\leq \frac{2R_1}{\sqrt{m}} \sum_{r=1}^{m} \mathbb{I}\{B_{D,r}(x)\}.
\end{aligned}
$$

Note that $\|\boldsymbol{x}\|_2 = 1$, which implies that $\boldsymbol{w}_{D,r}(0)^\top \boldsymbol{x}$ is distributed as $N(0, \tau^2)$. Therefore, we have

$$
\begin{aligned}
\mathbb{E}[\mathbb{I}\{B_{D,r}(x)\}] &= \mathbb{P}\left(|(1-\eta_2\mu)^k \boldsymbol{w}_{D,r}(0)^\top \boldsymbol{x}| \leq R_1 \right) \\
&= \int_{-R_1/(1-\eta_2\mu)^k}^{R_1/(1-\eta_2\mu)^k} \frac{1}{\sqrt{2\pi}\tau} \exp\left\{ -\frac{u^2}{2\tau^2} \right\} du \leq \frac{2R_1}{\sqrt{2\pi}(1-\eta_2\mu)^k \tau}.
\end{aligned}
$$

By Markov's inequality, with probability at least $1-\delta$, we have

$$
\sum_{r=1}^{m} \mathbb{I}\{B_{D,r}(x)\} \leq \frac{2mR_1}{\sqrt{2\pi}(1-\eta_2\mu)^k \tau \delta}.
$$

Thus, we have with probability at least $1-\delta$,

$$
\|\Delta_1\|_2 \leq \frac{2R_1}{\sqrt{m}} \| \sum_{r=1}^{m} \mathbb{I}\{B_{D,r}(\cdot)\} \|_2 \leq \frac{4\sqrt{m} R_1^2}{\sqrt{2\pi}(1-\eta_2\mu)^k \tau \delta} = O\left(\frac{n^2}{\sqrt{m}\lambda_0^2 \delta^2 (1-\eta_2\mu)^k \tau} \right),
$$

which implies

$$
\|\Delta_1\|_2 = O_{\mathbb{P}}\left(\frac{n^2}{\sqrt{m}\lambda_0^2 (1-\eta_2\mu)^k \tau} \right). \tag{3.61}
$$

Now we bound Δ_2. Note that Define $\boldsymbol{G}_k = \sum_{\mathrm{j}=0}^{k-1} \eta(\boldsymbol{I} - \eta \boldsymbol{H}^\infty)^{\mathrm{j}}$. Recalling that $\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{\epsilon}$, for fixed $\boldsymbol{x}$, we have

$$\begin{aligned}
\Delta_2(\boldsymbol{x}) =& \tilde{f}_k(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) \\
=& \boldsymbol{z}_0(\boldsymbol{x})^\top \mathrm{vec}(\boldsymbol{W}_D(k)) - h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y} \\
=& \boldsymbol{z}_0(\boldsymbol{x})^\top E_1 - \boldsymbol{z}_0(\boldsymbol{x})^\top E_2 + \boldsymbol{z}_0(\boldsymbol{x})^\top E_3 - \boldsymbol{z}_0(\boldsymbol{x})^\top T_5 - \boldsymbol{z}_0(\boldsymbol{x})^\top E_4 \\
&+ (1-\eta_2\mu)^k \boldsymbol{z}_0(\boldsymbol{x})^\top \mathrm{vec}(\boldsymbol{W}_D(0)) - h(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}, \qquad (3.62)
\end{aligned}$$

where E_1, E_2, E_3, T_5, E_4 are as in (3.49). Noting that $\|z_0(\boldsymbol{x})\|_2 = O_{\mathbb{P}}(1)$, we have that

$$|\boldsymbol{z}_0(\boldsymbol{x})^\top E_1|^2 \le \|\boldsymbol{z}_0(\boldsymbol{x})\|_2^2 \|E_1\|_2^2 = O_{\mathbb{P}}((1-\eta_2\mu)^{2k}) + O_{\mathbb{P}}\left(\frac{n^2(1-\eta_2\mu)^{2k-2}\sqrt{\log(n)}}{\sqrt{m}\lambda_0^2}\right), \qquad (3.63)$$

$$|\boldsymbol{z}_0(\boldsymbol{x})^\top E_2|^2 \le \|\boldsymbol{z}_0(\boldsymbol{x})\|_2^2 \|E_2\|_2^2 = O_{\mathbb{P}}\left(\frac{n\tau^2}{\lambda_0}(1-\eta_2\mu)^{2k}\right) + O_{\mathbb{P}}\left(\frac{n^2(1-\eta_2\mu)^{2k-2}\sqrt{\log(n)}}{\sqrt{m}\lambda_0^2}\right), \qquad (3.64)$$

$$|\boldsymbol{z}_0(\boldsymbol{x})^\top E_3|^2 \le \|\boldsymbol{z}_0(\boldsymbol{x})\|_2^2 \|E_3\|_2^2 = O_{\mathbb{P}}\left(\frac{\eta_1^6 n^8}{\eta_2^6\mu^6 m(1-\eta_2\mu)^{2k}\tau^2}\right), \qquad (3.65)$$

$$|\boldsymbol{z}_0(\boldsymbol{x})^\top E_4|^2 \le \|\boldsymbol{z}_0(\boldsymbol{x})\|_2^2 \|E_4\|_2^2 = O_{\mathbb{P}}\left(\frac{n^3}{(1-\eta_2\mu)^k\mu^3\sqrt{m}\delta^{3/2}\tau}\right), \qquad (3.66)$$

where (3.63) is because of (3.52) and (3.57), (3.64) is because of (3.53) and (3.58), (3.65) is because of (3.54), and (3.66) is because of (3.55). By Lemma 3.6.9 (d), the term $(1-\eta_2\mu)^k \boldsymbol{z}_0(\boldsymbol{x})^\top \mathrm{vec}(\boldsymbol{W}_D(0))$ in (3.62) can be bounded by

$$\|(1-\eta_2\mu)^k \boldsymbol{z}_0(\cdot)^\top \mathrm{vec}(\boldsymbol{W}_D(0))\|_2 = O_{\mathbb{P}}((1-\eta_2\mu)^k\tau).$$

Define

$$B = \eta_1 \boldsymbol{H}^\infty(\eta_2\mu I + \eta_1 \boldsymbol{H}^\infty)^{-1}\boldsymbol{y}.$$

Note that

$$\begin{aligned} B-\boldsymbol{y} =&\eta_1\boldsymbol{H}^{\infty}(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}-\boldsymbol{y}\\ =&(\eta_1\boldsymbol{H}^{\infty}-\eta_2\mu I-\eta_1\boldsymbol{H}^{\infty})(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}\\ =&-\eta_2\mu(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}. \end{aligned}$$

Therefore, the remaining term in (3.62) $-\boldsymbol{z}_0(\boldsymbol{x})^\top T_5 - h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}$ can be bounded by

$$\begin{aligned} &-\boldsymbol{z}_0(\boldsymbol{x})^\top T_5 - h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\\ =&-\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)\sum_{\mathrm{i}=0}^{k-1}\eta_1(1-\eta_2\mu)^{\mathrm{i}}(B-\boldsymbol{y})-h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\\ =&-\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)\eta_1\frac{1-(1-\eta_2\mu)^k}{\eta_2\mu}(B-\boldsymbol{y})-h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\\ =&\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)\eta_1(1-(1-\eta_2\mu)^k)(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}-h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\\ =&(\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)-h(\boldsymbol{x},\boldsymbol{X}))(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}-\eta_1(1-\eta_2\mu)^k\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)(\eta_2\mu I+\eta_1\boldsymbol{H}^{\infty})^{-1}\boldsymbol{y}. \end{aligned} \tag{3.67}$$

The first term in (3.67) can be bounded by

$$\begin{aligned} &\|(\boldsymbol{z}_0(\cdot)^\top \boldsymbol{Z}(0)-h(\cdot,\boldsymbol{X}))(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2\\ \leq&\|(\boldsymbol{z}_0(\cdot)^\top \boldsymbol{Z}(0)-h(\cdot,\boldsymbol{X}))\|_2\|(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2\\ =&O_{\mathbb{P}}\left(\frac{n\sqrt{\log(n)}\eta_1}{\sqrt{m}\eta_2\mu}\right), \end{aligned} \tag{3.68}$$

where we utilize

$$\|(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2^2=\boldsymbol{y}^\top(\boldsymbol{H}^{\infty}+\eta_2\mu/\eta_1 I)^{-2}\boldsymbol{y}\leq\frac{\eta_1^2}{\eta_2^2\mu^2}\|\boldsymbol{y}\|_2^2=O_{\mathbb{P}}\left(\frac{\eta_1^2}{\eta_2^2\mu^2}n\right),$$

and Lemma 3.6.9 (c).

The second term in (3.67) can be bounded by

$$\begin{aligned}
&\|(1-\eta_2\mu)^k \boldsymbol{z}_0(\cdot)^\top \boldsymbol{Z}(0)(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2 \\
\leq &(1-\eta_2\mu)^k \|(\boldsymbol{z}_0(\cdot)^\top \boldsymbol{Z}(0) - h(\cdot,\boldsymbol{X}))(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2 \\
&+ (1-\eta_2\mu)^k \|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_2 \\
\leq &O_{\mathbb{P}}\left(\frac{n\sqrt{\log(n)}\eta_1}{\sqrt{m}\eta_2\mu}\right) + (1-\eta_2\mu)^k \|h(\cdot,\boldsymbol{X})(\boldsymbol{H}^\infty + \eta_2\mu/\eta_1 I)^{-1}\boldsymbol{y}\|_{\mathcal{N}} \\
= &O_{\mathbb{P}}((1-\eta_2\mu)^k), \qquad (3.69)
\end{aligned}$$

where the second inequality is because of (3.68) and the last equality is because of Theorem 3.1.2 and the assumption $\eta_1 \asymp \eta_2$. Plugging (3.63)-(3.69) to (3.62), we can conclude that

$$\|\Delta_2\|_2 = o_{\mathbb{P}}(n^{-\frac{d}{2d-1}}), \qquad (3.70)$$

by choosing k and m as in Theorem 3.3.4. Combining (3.61), (3.70), and (3.60) finishes the proof. ■

3.6.3 Proof of lemmas

Proof of Lemma 3.6.1

Proof The proof of Lemma 3.6.1 mainly from Appendix C of [108] and Appendix D of [128], with some modifications.

We first review some background of spherical harmonic analysis [129], [130]. Let $Y_{k,\mathrm{j}}$ be the spherical harmonics of degree k on $\mathcal{S}^{d-1}$, where $N(p.k) = \frac{2k+d-2}{k}\begin{pmatrix} k+d-3 \\ d-2 \end{pmatrix}$. Then $Y_{k,\mathrm{j}}$ is an orthonormal basis of $L_2(\mathcal{S}^{p-1}, d\xi)$, where $d\xi$ is the uniform measure on the sphere. Then we have

$$\sum_{\mathrm{j}=1}^{N(d,k)} Y_{k,\mathrm{j}}(\boldsymbol{s})Y_{k,\mathrm{j}}(\boldsymbol{t}) = N(d,k)P_k(\boldsymbol{s}^\top \boldsymbol{t}),$$

where P_k is the k-th Legendre polynomial in dimension d, given by

$$P_k(t) =(-1/2)^k \frac{\Gamma(\frac{d-1}{2})}{\Gamma(k+\frac{d-1}{2})}(1-t^2)^{(3-d)/2}\left(\frac{d}{dt}\right)^k (1-t^2)^{k+(d-3)/2}.$$

The polynomials P_k are orthogonal in $L_2([-1,1])d\nu$, where the measure $d\nu = (1-t^2)^{(d-3)/2}dt$ with Lebesgue measure dt, and

$$\int_{[-1,1]} P_k^2(t)(1-t^2)^{(d-3)/2}dt = \frac{w_{d-1}}{w_{d-2}}\frac{1}{N(d,k)},$$

where $w_{d-1} = \frac{2\pi^{d/2}}{\Gamma(d/2)}$. Furthermore, it can be shown that [129]

$$tP_k(t) = \frac{k}{2k+d-2}P_{k-1}(t) + \frac{k+d-2}{2k+d-2}P_{k+1}(t),$$

for $k \geq 1$, and for j $= 0$ we have $tP_0(t) = P_1(t)$. This implies that for large k enough, we have

$$\mu_k = \frac{k}{2k+d-2}\mu_{0,k-1} + \frac{k+d-2}{2k+d-2}\mu_{0,k+1},$$

where $\mu_{0,k-1}$ and $\mu_{0,k+1}$ are as in Lemma 17 of [108]. By Lemma 17 of [108], we have $\mu_{0,k} \asymp k^{-d}$ for large k, if $k = 1$ mod 2. This finish the proof of Lemma 3.6.1. ■

Proof of Lemma 3.6.2

Proof By Theorem 1 of [131] and Lemma 3.6.1, we can see that the function space $\mathcal{N}$ is a subspace of the Sobolev space $H^s(\mathcal{S}^{d-1})$. Therefore, the entropy of $\mathcal{N}(1)$ can be bounded if the entropy of $H^{d/2}(\mathcal{S}^{d-1})(1)$ can be bounded. By Theorem 1.2 of [132], we have that the k-th entropy number $\mathrm{e}_k(T)$ can be bounded by $k^{-d/(2(d-1))}$. This implies that

$$H(\delta, \mathcal{N}(1), \|\cdot\|_{L_\infty}) \leq A\delta^{-\frac{2(d-1)}{d}}.$$

■

Proof of Lemma 3.6.5

Proof The first inequality follows the fact that h is positive definite, which implies the inverse of

$$\begin{pmatrix} h(\boldsymbol{s},\boldsymbol{s}) & h(\boldsymbol{X},\boldsymbol{s}) \\ h(\boldsymbol{s},\boldsymbol{X}) & \boldsymbol{h}^{\infty} \end{pmatrix}$$

is positive definite. By block matrix inverse, we have the first inequality in Lemma 3.6.5 holds.

The second inequality and third inequality are direct results of Theorem 3.1.2 implies

$$\begin{aligned}
&\mathbb{E}_{\epsilon,\boldsymbol{X}}(\|\hat{g}_n - g^*\|_2^2) \\
=&\int_{\mathbb{S}^{d-1}} (g^*(\boldsymbol{x}) - h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}\boldsymbol{y}^*)^2 + h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-2}h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x} = O_{\mathbb{P}}(n^{-\frac{d}{2d-1}})
\end{aligned}$$

for any function g^* with $\|g^*\|_{\mathcal{N}} \leq 1$. Then we have

$$\int_{\mathbb{S}^{d-1}} h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-2}h(\boldsymbol{X},\boldsymbol{x})d\boldsymbol{x} = O_{\mathbb{P}}(n^{-\frac{d}{2d-1}}),$$

which finishes the proof of the second equality. Let $g^*(\boldsymbol{x}) = h(\boldsymbol{s},\boldsymbol{x})$, then we have

$$\int_{\mathbb{S}^{d-1}} (h(\boldsymbol{s},\boldsymbol{x}) - h(\boldsymbol{x},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s}))^2 d\boldsymbol{x} = O_{\mathbb{P}}(n^{-\frac{d}{2d-1}}).$$

By the interpolation inequality, we have

$$\begin{aligned}
&h(\boldsymbol{s},\boldsymbol{s}) - h(\boldsymbol{s},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s})) \\
\leq&\|h(\boldsymbol{s},\cdot) - h(\cdot,\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s}))\|_{\infty} \\
\leq&C\|h(\boldsymbol{s},\cdot) - h(\cdot,\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s}))\|_2^{1-\frac{d-1}{d}}\|h(\boldsymbol{s},\cdot) - h(\cdot,\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s})\|_{\mathcal{N}}^{\frac{d-1}{d}} \\
=&O_{\mathbb{P}}(n^{-\frac{1}{2d-1}})(h(\boldsymbol{s},\boldsymbol{s}) + h(\boldsymbol{s},\boldsymbol{X})(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}\boldsymbol{H}^{\infty}(\boldsymbol{H}^{\infty}+\mu\boldsymbol{I})^{-1}h(\boldsymbol{X},\boldsymbol{s}))^{\frac{d-1}{d}} \\
\leq&O_{\mathbb{P}}(n^{-\frac{1}{2d-1}})(h(\boldsymbol{s},\boldsymbol{s}) + h(\boldsymbol{s},\boldsymbol{X})(\boldsymbol{H}^{\infty})^{-1}h(\boldsymbol{X},\boldsymbol{s}))^{\frac{d-1}{d}} = O_{\mathbb{P}}(n^{-\frac{1}{2d-1}}),
\end{aligned}$$

where the last inequality follows the first inequality of Lemma 3.6.5. ■

Proof of Lemma 3.6.6

Proof Given that g and f^* have the same value at all $\boldsymbol{x}_{\mathrm{i}}$'s, the empirical norm $\|g - f^*\|_n = 0$. Notice that both g and f^* are in the RKHS generated by the NTK h, denoted by $\mathcal{N}$. Utilizing Lemma 3.6.2 and 3.6.4 similarly as in the proof of Theorem 3.1.2, we have $R, K = O(1)$ and $J_\infty(z, \mathcal{N}) \lesssim z^{1/d}$, which leads to

$$\sup_{h \in \mathcal{G}(R)} \left| \|h\|_n^2 - \|h\|_2^2 \right| = O_{\mathbb{P}}\left(\sqrt{\frac{1}{n}}\right),$$

where $\mathcal{G}(R) := \{g \in \mathcal{N}(1) : \|g - g^*\|_2 \leq R\}$. Therefore, we can conclude that $\|g - f^*\|_2 = O_{\mathbb{P}}(n^{-1/2})$. ■

Proof of Lemma 3.6.9

Proof The proof of (a) and (b) can be found in [17].

For (c), the i-th coordinates of $\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0)$ and $h(\boldsymbol{x}, \boldsymbol{X})$ are

$$\frac{1}{m}\sum_{r=1}^{m} \boldsymbol{x}^\top \boldsymbol{x}_{\mathrm{i}} \mathbb{I}\{\boldsymbol{w}_r^\top(0)\boldsymbol{x} \geq 0\}\mathbb{I}\{\boldsymbol{w}_r^\top(0)\boldsymbol{x}_{\mathrm{i}} \geq 0\}, \quad \text{and} \quad \mathbb{E}_{\boldsymbol{w}\sim N(0,\boldsymbol{I})}[\boldsymbol{x}^\top \boldsymbol{x}_{\mathrm{i}} \mathbb{I}\{\boldsymbol{w}^\top \boldsymbol{x} \geq 0\}\mathbb{I}\{\boldsymbol{w}^\top \boldsymbol{x}_{\mathrm{i}} \geq 0\}],$$

respectively. $\forall \mathrm{i} \in [n]$, $(\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0))_{\mathrm{i}}$ is the average of m i.i.d. random variables, which have expectation $h_{\mathrm{i}}(\boldsymbol{x}, \boldsymbol{X})$ and bounded in $[0, 1]$. For any fixed $\boldsymbol{x}$, by Hoeffding's inequality, with probability at least $1 - \delta^*$,

$$|(\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0))_{\mathrm{i}} - h_{\mathrm{i}}(\boldsymbol{x}, \boldsymbol{X})| \leq \sqrt{\frac{\log(2/\delta^*)}{2m}}$$

holds. By defining $\delta = n\delta^*$ and applying a union bound over all $\mathrm{i} \in [n]$, with probability at least $1 - \delta$, we have

$$\|\boldsymbol{z}_0(\boldsymbol{x})^\top \boldsymbol{Z}(0) - h(\boldsymbol{x}, \boldsymbol{X})\|_2^2 = O\left(n\frac{\log(2n/\delta)}{2m}\right)$$

For (d), since

$$\boldsymbol{z}_0(\boldsymbol{x})^\top \mathrm{vec}(\boldsymbol{W}(0)) = \frac{1}{\sqrt{m}}\sum_{r=1}^{m} a_r \mathbb{I}\{\boldsymbol{w}_r(0)^\top \boldsymbol{x} \geq 0\}\boldsymbol{w}_r(0)^\top \boldsymbol{x}$$

Define random variables V_r, $r \in [m]$ as

$$V_r = a_r \mathbb{I}\{\boldsymbol{w}_r(0)^\top \boldsymbol{x} \geq 0\} \boldsymbol{w}_r(0)^\top \boldsymbol{x}$$

Since

$$\boldsymbol{w}_r(0)^\top \boldsymbol{x} \ \sim \ N(0, \tau^2) \quad \text{and} \quad a_r \ \sim \ \text{unif}\{1, -1\}.$$

It's easy to prove that V_r, $r \in [m]$ are i.i.d. with mean 0 and sub-Gaussian parameter τ. By Hoeffding's inequality, at fixed bx, with probability at least $1 - \delta$, we have

$$\left| \frac{1}{\sqrt{m}} \sum_{r=1}^{m} V_r \right| \leq \sqrt{2}\tau \sqrt{\log(2/\delta)}.$$

Thus $\|\boldsymbol{z}_0(\cdot)^\top \text{vec}(\boldsymbol{W}(0))\|_2 = O\left(\tau \sqrt{\log(1/\delta)}\right)$. ■

4. SUMMARY

Deep learning has achieved breakthroughs in many machine learning tasks. In contrast to the great empirical success, theoretical understanding of deep learning is still lacking. It is my firm believe that statistics has a lot more to offer for deep learning theories. This thesis aims to investigate the nonparametric perspective of DNNs. Through the lens of nonparametric estimation, statistical optimality is established for DNNs in popular tasks such as regression and classification. We have shown that, without much modification, DNN estimators can adapt to different kinds of underlying low-dimensional structures of the data and alleviate the curse of dimensionality. Even though the optimization of DNNs is highly non-convex, training algorithm can be brought into the nonparametric framework and act as a way of regularization. Statistical optimality can also be proven with algorithmic guarantees.

Our results contribute to the current literature of statistical deep learning. The combination of classical statistical results and recent advances in approximation, optimization, generalization of DNNs brings out great potentials into understanding why deep learning works so well in practice. Along this line, more work could be done for more complicated network structures, e.g., CNN, ResNet, etc. and on more estimation problems such as density estimation. On one hand, this type of analysis can potentially explain in theory, the advantages of popular deep learning models and training techniques. On the other hand, from such theoretical analysis, new techniques for training better and more robust deep learning models could be motivated.

REFERENCES

[1] E. Mammen, A. B. Tsybakov, *et al.*, "Smooth discrimination analysis," *The Annals of Statistics*, vol. 27, no. 6, pp. 1808–1829, 1999.

[2] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with relu activation function," *arXiv preprint arXiv:1708.06633*, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[8] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 183–192, 1989.

[9] B. Hanin, "Universal function approximation by deep neural nets with bounded width and relu activations," *arXiv preprint arXiv:1708.02691*, 2017.

[10] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.

[11] J. Lu, Z. Shen, H. Yang, and S. Zhang, "Deep network approximation for smooth functions," *arXiv preprint arXiv:2001.03040*, 2020.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] P. Zhou and J. Feng, "The landscape of deep learning algorithms," *arXiv preprint arXiv:1705.07038*, 2017.

[14] H. Fu, Y. Chi, and Y. Liang, *Local geometry of one-hidden-layer neural networks for logistic regression*, 1802.

[15] S. Liang, R. Sun, Y. Li, and R. Srikant, "Understanding the loss surface of neural networks for binary classification," *arXiv preprint arXiv:1803.00909*, 2018.

[16] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[17] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," *arXiv preprint arXiv:1901.08584*, 2019.

[18] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," *arXiv preprint arXiv:1811.03962*, 2018.

[19] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," *arXiv preprint arXiv:1811.04918*, 2018.

[20] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in neural information processing systems*, 2016, pp. 586–594.

[21] S. Liang, R. Sun, J. D. Lee, and R. Srikant, "Adding one neuron can eliminate all bad local minima," in *Advances in Neural Information Processing Systems*, 2018, pp. 4350–4360.

[22] K. Kawaguchi and L. P. Kaelbling, "Elimination of all bad local minima in deep learning," *arXiv preprint arXiv:1901.00279*, 2019.

[23] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in neural information processing systems*, 2018, pp. 8571–8580.

[24] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," *arXiv preprint arXiv:1904.11955*, 2019.

[25] L. Chizat and F. Bach, "A note on lazy training in supervised differentiable programming," *arXiv preprint arXiv:1812.07956*, 2018.

[26] G. Yang, "Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation," *arXiv preprint arXiv:1902.04760*, 2019.

[27] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[28] D. McAllester, "Simplified pac-bayesian margin bounds," in *Learning theory and Kernel machines*, Springer, 2003, pp. 203–215.

[29] B. Neyshabur, S. Bhojanapalli, and N. Srebro, "A pac-bayesian approach to spectrally-normalized margin bounds for neural networks," *arXiv preprint arXiv:1707.09564*, 2017.

[30] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Conference on Learning Theory*, 2015, pp. 1376–1401.

[31] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 6240–6249.

[32] A. Ledent, Y. Lei, and M. Kloft, "Improved generalisation bounds for deep learning through $l\infty$ *coveringnumbers*," *arXiv preprint arXiv:1905.12430*, 2019.

[33] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[34] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 1731–1741.

[35] M. S. Advani and A. M. Saxe, "High-dimensional dynamics of generalization error in neural networks," *arXiv preprint arXiv:1710.03667*, 2017.

[36] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation," *arXiv preprint arXiv:1903.08560*, 2019.

[37] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Linearized two-layers neural networks in high dimension," *arXiv preprint arXiv:1904.12191*, 2019.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] C. H. Martin and M. W. Mahoney, "Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning," *arXiv preprint arXiv:1810.01075*, 2018.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[41] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, "Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks.," *Journal of Machine Learning Research*, vol. 20, no. 63, pp. 1–17, 2019.

[42] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 5947–5956.

[43] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," *arXiv preprint arXiv:1703.11008*, 2017.

[44] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, "Fantastic generalization measures and where to find them," *arXiv preprint arXiv:1912.02178*, 2019.

[45] T. Hu, Z. Shang, and G. Cheng, "Sharp rate of convergence for deep neural network classifiers under the teacher-student setting," *arXiv preprint arXiv:2001.06892*, 2020.

[46] W. Wang, T. Hu, C. Lin, and G. Cheng, "Regularization matters: A nonparametric perspective on overparametrized neural network," *arXiv preprint arXiv:2007.02486*, 2020.

[47] S. Siegel, "Nonparametric statistics," *The American Statistician*, vol. 11, no. 3, pp. 13–19, 1957.

[48] C. J. Stone, "Optimal global rates of convergence for nonparametric regression," *The annals of statistics*, pp. 1040–1053, 1982.

[49] R. Liu, B. Boukai, and Z. Shang, "Optimal nonparametric inference via deep neural network," *arXiv preprint arXiv:1902.01687*, 2019.

[50] B. Bauer, M. Kohler, *et al.*, "On deep learning as a remedy for the curse of dimensionality in nonparametric regression," *The Annals of Statistics*, vol. 47, no. 4, pp. 2261–2285, 2019.

[51] M. Kohler and S. Langer, "Deep versus deeper learning in nonparametric regression," *Submitted for publication*, 2018.

[52] M. Imaizumi and K. Fukumizu, "Deep neural networks learn non-smooth functions effectively," *arXiv preprint arXiv:1802.04474*, 2018.

[53] J. Schmidt-Hieber, "Deep relu network approximation of functions on a manifold," *arXiv preprint arXiv:1908.00695*, 2019.

[54] M. Kohler, A. Krzyzak, and S. Langer, "Estimation of a function of low local dimensionality by deep neural networks," *arXiv preprint arXiv:1908.11140*, 2019.

[55] T. Suzuki and A. Nitanda, "Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic besov space," *arXiv preprint arXiv:1910.12799*, 2019.

[56] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[57] A. P. Korostelev and A. B. Tsybakov, *Minimax theory of image reconstruction*. Springer Science & Business Media, 2012, vol. 82.

[58] Y. Kim, I. Ohn, and D. Kim, "Fast convergence rates of deep neural networks for classification," *arXiv preprint arXiv:1812.03599*, 2018.

[59] M. Kohler, A. Krzyzak, and B. Walter, "On the rate of convergence of image classifiers based on convolutional neural networks," *arXiv preprint arXiv:2003.01526*, 2020.

[60] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, "Iris recognition with off-the-shelf cnn features: A deep learning perspective," *IEEE Access*, vol. 6, pp. 18 848–18 855, 2017.

[61] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Springer, 2014, pp. 818–833.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[64] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[65] M. H. Farrell, T. Liang, and S. Misra, "Deep neural networks for estimation and inference: Application to causal effects and other semiparametric estimands," *arXiv preprint arXiv:1809.09953*, 2018.

[66] T. Suzuki, "Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: Optimal rate and curse of dimensionality," *arXiv preprint arXiv:1810.08033*, 2018.

[67] R. Nakada and M. Imaizumi, "Adaptive approximation and estimation of deep neural network to intrinsic dimensionality," *arXiv preprint arXiv:1907.02177*, 2019.

[68] K. Oono and T. Suzuki, "Approximation and non-parametric estimation of resnet-type convolutional neural networks," *arXiv preprint arXiv:1903.10047*, 2019.

[69] M. Chen, H. Jiang, W. Liao, and T. Zhao, "Efficient approximation of deep relu networks for functions on low dimensional manifolds," in *Advances in Neural Information Processing Systems*, 2019, pp. 8172–8182.

[70] M. Kohler and S. Langer, "On the rate of convergence of fully connected very deep neural network regression estimates," *arXiv preprint arXiv:1908.11133*, 2019.

[71] D. Saad and S. A. Solla, *Dynamics of on-line gradient descent learning for multilayer neural networks*, 1996.

[72] C. W. H. Mace and A. C. C. Coolen, "Statistical mechanical analysis of the dynamics of learning in perceptrons," *Statistics and Computing*, vol. 8, no. 1, pp. 55–88, 1998. DOI: 10.1023/A:1008896910704. [Online]. Available: https://doi.org/10.1023/A:1008896910704.

[73] A. Engel and C. Broeck, *Statistical Mechanics of Learning*. Cambridge University Press, 2002.

[74] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[75] J. Ba and R. Caruana, "Do deep nets really need to be deep?" In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2654–2662.

[76] S. Goldt, M. Advani, A. M. Saxe, F. Krzakala, and L. Zdeborová, "Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 6979–6989.

[77] B. Aubin, A. Maillard, j. barbier jean, F. Krzakala, N. Macris, and L. Zdeborová, "The committee machine: Computational to statistical gaps in learning a two-layers neural network," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H.

Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Curran Associates, Inc., 2018, pp. 3223–3234.

[78] Y. Tian, "A theoretical framework for deep locally connected relu network," *arXiv preprint arXiv:1809.10829*, 2018.

[79] Y. Tian, "Over-parameterization as a catalyst for better generalization of deep relu network," *arXiv preprint arXiv:1909.13458*, 2019.

[80] X. Zhang, Y. Yu, L. Wang, and Q. Gu, "Learning one-hidden-layer relu networks via gradient descent," *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[81] Y. Cao and Q. Gu, "Tight sample complexity of learning one-hidden-layer convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 611–10 621.

[82] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine learning research*, vol. 7, no. Nov, pp. 2541–2563, 2006.

[83] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in neural information processing systems*, 2017, pp. 6231–6239.

[84] M. Telgarsky, "Representation benefits of deep feedforward networks," *arXiv preprint arXiv:1509.08101*, 2015.

[85] N. Srebro, K. Sridharan, and A. Tewari, "Optimistic rates for learning with a smooth loss," *arXiv preprint arXiv:1009.3896*, 2010.

[86] A. B. Tsybakov *et al.*, "Optimal aggregation of classifiers in statistical learning," *The Annals of Statistics*, vol. 32, no. 1, pp. 135–166, 2004.

[87] J. Bai, Q. Song, and G. Cheng, "Rate optimal variational bayesian inference for sparse dnn," *arXiv preprint arXiv:1910.04355*, 2019.

[88] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, 2014, pp. 2924–2932.

[89] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[90] Y. Lin, "Support vector machines and the bayes rule in classification," *Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 259–275, 2002.

[91] P. Petersen and F. Voigtlaender, "Optimal approximation of piecewise smooth functions using deep relu neural networks," *Neural Networks*, vol. 108, pp. 296–330, 2018.

[92] S. van de Geer, *Empirical Processes in M-estimation*. Cambridge University Press, 2000.

[93] T. Serra, C. Tjandraatmadja, and S. Ramalingam, "Bounding and counting linear regions of deep neural networks," *arXiv preprint arXiv:1711.02114*, 2017.

[94] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. S. Dickstein, "On the expressive power of deep neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 2847–2854.

[95] D. Yarotsky and A. Zhevnerchuk, "The phase diagram of approximation rates for deep neural networks," *arXiv preprint arXiv:1906.09477*, 2019.

[96] I. Steinwart, C. Scovel, *et al.*, "Fast rates for support vector machines using gaussian kernels," *The Annals of Statistics*, vol. 35, no. 2, pp. 575–607, 2007.

[97] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Advances in Neural Information Processing Systems*, 2018, pp. 8157–8166.

[98] D. Zou and Q. Gu, "An improved analysis of training over-parameterized deep neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 2053–2062.

[99] Z. Ji and M. Telgarsky, "The implicit bias of gradient descent on nonseparable data," in *Conference on Learning Theory*, 2019, pp. 1772–1798.

[100] Z. Ji and M. Telgarsky, "Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks," *arXiv preprint arXiv:1909.12292*, 2019.

[101] K. Lyu and J. Li, "Gradient descent maximizes the margin of homogeneous neural networks," *arXiv preprint arXiv:1906.05890*, 2019.

[102] Y. Cao and Q. Gu, "Generalization error bounds of gradient descent for learning overparameterized deep ReLU networks," *arXiv preprint arXiv:1902.01384*, 2019.

[103] C. Wei, J. D. Lee, Q. Liu, and T. Ma, "Regularization matters: Generalization and optimization of neural nets vs their induced kernel," in *Advances in Neural Information Processing Systems*, 2019, pp. 9709–9721.

[104] W. Hu, Z. Li, and D. Yu, "Simple and effective regularization methods for training on noisily labeled data with generalization guarantee," in *International Conference on Learning Representations*, 2020.

[105] A. Geifman, A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and R. Basri, "On the similarity between the laplace and neural tangent kernels," *NeurIPS 2020*, 2020.

[106] A. Nitanda, G. Chinot, and T. Suzuki, "Gradient descent can learn less over-parameterized two-layer neural networks on classification problems," *arXiv preprint arXiv:1905.09870*, 2019.

[107] A. Nitanda and T. Suzuki, "Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime," *arXiv preprint arXiv:2006.12297*, 2020.

[108] A. Bietti and J. Mairal, "On the inductive bias of neural tangent kernels," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 873–12 884.

[109] M. Yuan, D.-X. Zhou, *et al.*, "Minimax optimal rates of estimation in high dimensional additive models," *The Annals of Statistics*, vol. 44, no. 6, pp. 2564–2593, 2016.

[110] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping and non-parametric regression: An optimal data-dependent stopping rule," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 335–366, 2014.

[111] M. Kohler and A. Krzyzak, "Over-parametrized deep neural networks do not generalize well," *arXiv preprint arXiv:1912.03925*, 2019.

[112] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992, pp. 950–957.

[113] B. Bilgic, I. Chatnuntawech, A. P. Fan, K. Setsompop, S. F. Cauley, L. L. Wald, and E. Adalsteinsson, "Fast image reconstruction with l2-regularization," *Journal of magnetic resonance imaging*, vol. 40, no. 1, pp. 181–191, 2014.

[114] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv preprint arXiv:1706.05350*, 2017.

[115] E. Phaisangittisagul, "An analysis of the regularization between l2 and dropout in single hidden layer neural network," in *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, IEEE, 2016, pp. 174–179.

[116] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[117] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in Neural Information Processing Systems*, 2001, pp. 402–408.

[118] L. Prechelt, "Early stopping-but when?" In *Neural Networks: Tricks of the Trade*, Springer, 1998, pp. 55–69.

[119] A. Lewkowycz and G. Gur-Ari, "On the training dynamics of deep networks with l_2 regularization," *arXiv preprint arXiv:2006.08643*, 2020.

[120] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,"

[121] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[122] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[123] T. Dao, A. Gu, A. J. Ratner, V. Smith, C. De Sa, and C. Ré, "A kernel theory of modern data augmentation," *Proceedings of machine learning research*, vol. 97, p. 1528, 2019.

[124] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu, "Towards understanding the spectral bias of deep learning," *arXiv preprint arXiv:1912.01198*, 2019.

[125] S. van de Geer, "On the uniform convergence of empirical norms and inner products, with application to causal inference," *Electronic Journal of Statistics*, vol. 8, no. 1, pp. 543–574, 2014.

[126] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.

[127] R. S. Varga, *Gershgorin and His Circles*. Springer Science & Business Media, 2010, vol. 36.

[128] F. Bach, "Breaking the curse of dimensionality with convex neural networks," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 629–681, 2017.

[129] K. Atkinson and W. Han, *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*. Springer Science & Business Media, 2012, vol. 2044.

[130] E. Costas and F. Christopher, *Spherical Harmonics in p Dimensions*. World Scientific, 2014.

[131] J. S. Brauchart and J. Dick, "A characterization of Sobolev spaces on the sphere and an extension of Stolarsky's invariance principle to arbitrary smoothness," *Constructive Approximation*, vol. 38, no. 3, pp. 397–445, 2013.

[132] H. P. Wang, K. Wang, and J. Wang, "Entropy numbers of Besov classes of generalized smoothness on the sphere," *Acta Mathematica Sinica, English Series*, vol. 30, no. 1, pp. 51–60, 2014.

VITA

Tianyang Hu was born in Chaoyang, Liaoning, China in 1991. He received a bachelor's degree in Mathematics from Tsinghua University, China in 2014 and a master's degree in Statistics from The University of Chicago in 2016. He then joined the Statistics Ph.D. program at Purdue University, where he earned a doctoral degree in Statistics in 2020. His research interests include theoretical deep learning, machine learning, nonparametric estimation, and high-dimensional statistics.