

Classification of Data Generated by Gaussian Mixture Models Using Deep ReLU Networks

Tian-Yi Zhou

*H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA*

TZHOU306@GATECH.EDU

Xiaoming Huo

*H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA*

HUO@ISYE.GATECH.EDU

Editor: Bharath Sriperumbudur

Abstract

This paper studies the binary classification of unbounded data from \mathbb{R}^d generated under Gaussian Mixture Models (GMMs) using deep ReLU neural networks. We obtain — for the first time — non-asymptotic upper bounds and convergence rates of the excess risk (excess misclassification error) for the classification without restrictions on model parameters. While the majority of existing generalization analysis of classification algorithms relies on a bounded domain, we consider an unbounded domain by leveraging the analyticity and fast decay of Gaussian distributions. To facilitate our analysis, we give a novel approximation error bound for general analytic functions using ReLU networks, which may be of independent interest. Gaussian distributions can be adopted nicely to model data arising in applications, e.g., speeches, images, and texts; our results provide a theoretical verification of the observed efficiency of deep neural networks in practical classification problems.

Keywords: binary classification, Gaussian Mixture Model, excess risk, ReLU neural networks, statistical learning theory

1. Introduction

This paper studies the binary classification of unbounded data generated by a mixture of Gaussian distributions using neural networks. We assume our data in \mathbb{R}^d follow a class of distribution largely used to model real-world data, namely the Gaussian Mixture Model (GMM). Many studies have shown that GMM is an effective model for audio, speech, image, and text processing, e.g., see (Reynolds et al., 2000; Portilla et al., 2003; Blekas et al., 2005). The universality of GMM (Goodfellow et al., 2016) motivates us to study the classification problem under such distributional assumptions on data.

In this paper, we consider data $X \in \mathbb{R}^d$ drawn from a GMM with two classes, denoted as $\{-1, 1\}$, and members of each class are drawn from a mixture of Gaussian distributions. Denote the domain by $\mathcal{X} = \mathbb{R}^d$ and the output set by $\mathcal{Y} = \{-1, 1\}$. We also denote by ρ a joint distribution on $Z := \mathcal{X} \times \mathcal{Y}$ for a GMM to be specified later. We are interested in learning a binary classifier $f : \mathbb{R}^d \rightarrow \{-1, 1\}$ using deep neural networks (DNNs). To

evaluate the effectiveness of a classifier f , we conduct a misclassification error analysis. Specifically, we examine its excess risk (excess misclassification error). For any classifier $\text{sgn}(f)$ induced by a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, its misclassification error is defined as

$$R(f) := \mathbb{E}[\mathbb{1}_{\{Y \cdot \text{sgn}(f(X)) = -1\}}] = \mathbb{P}(Y \cdot \text{sgn}(f(X)) = -1).$$

A Bayes classifier f_c minimizes the misclassification error and gives the best prediction of Y for a given X :

$$f_c(X) = \begin{cases} 1, & \text{if } \mathbb{P}(Y = 1|X) \geq \mathbb{P}(Y = -1|X), \\ -1, & \text{if } \mathbb{P}(Y = 1|X) < \mathbb{P}(Y = -1|X). \end{cases} \quad (1)$$

We aim to learn a classifier f as close as possible to f_c using trainable DNNs. The accuracy of a classifier can be characterized by the excess risk given by: $R(f) - R(f_c)$. In this work, we establish fast convergence rates of excess risk of classifiers under the GMM model generated by DNNs (given in Theorem 6).

The mathematical analysis of classification algorithms was initiated upon the introduction of support vector machines (Cortes and Vapnik, 1995; Vapnik, 1999) with a focus on margin-based analysis. Shortly after, the universality of classification induced by kernel-based regularization schemes was established in (Steinwart, 2001, 2002). Tsybakov’s noise condition (Tsybakov, 2004), together with a comparison theorem (Zhang, 2004), have facilitated the analysis of the excess risk of classification algorithms. Since then, a significant body of literature has emerged to study the theoretical guarantees of kernel methods in classification (Chen et al., 2004; Steinwart, 2005; Wu et al., 2007; Kim and Scott, 2009). All the existing work is carried out on a bounded domain (Steinwart and Christmann, 2008; Campbell and Ying, 2011). The study of classification algorithms continues to be an active area of research in both theory and practice.

Today, neural networks are widely considered a popular choice for classification tasks in the machine learning community, often preferred over kernel methods. Since the last decade, the development of powerful GPUs and large data sets has enabled the training of deep and complex neural networks. These led to breakthroughs in many fields, including computer vision, speech recognition, and natural language processing. A rapidly growing line of literature demonstrates the accuracy and effectiveness of DNNs in tackling classification tasks arising in practice, e.g., text and image classifications (Krizhevsky et al., 2012; He et al., 2016).

Given the unboundedness of Gaussian distributions, we study the classification of GMM on an unbounded domain. We would like to highlight that all existing results of classification, whether by ReLU neural networks or kernel-based classifiers, rely on a bounded input domain, e.g., the unit cube $[0, 1]^d$ (Kim et al., 2021; Bos and Schmidt-Hieber, 2022; Shen et al., 2022), the unit sphere \mathbb{S}^{d-1} (Feng et al., 2021). However, since Gaussian distributions are unbounded, existing results cannot be applied. In contrast to the prior works, our paper considers the unbounded domain \mathbb{R}^d . Many existing approaches in the mathematical analysis of classification problems, such as covering numbers and integral operators, do not apply to unbounded input spaces. We extend the analysis from a bounded to an unbounded domain by leveraging the fast decay and analyticity of Gaussian functions. By not restricting data in a bounded set, our work speaks directly to many modern classification tasks in practice.

To conduct a generalization analysis of neural network classifiers, we adopt the Hinge loss function. Given a random sample $z := \{(x_i, y_i)\}_{i=1}^n$ drawn from $\mathbb{R}^d \times \{-1, 1\}$, it is natural to find a classifier that minimizes the empirical risk $\frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{y_i \cdot \text{sgn}(f(x_i)) = -1\}}$. However, minimizing the empirical risk with the 0-1 loss considered by (Tsybakov, 2004; Audibert and Tsybakov, 2007) is NP-hard and thus computationally infeasible (Bartlett et al., 2006). In this paper, we adopt the well-known Hinge loss function $\phi(t) := \max\{0, 1 - t\}$ to make computations feasible. Learning a neural network classifier with Hinge loss is relatively straightforward owing to the gradient descent algorithm (Molitor et al., 2021; George et al., 2024). Also, there exists a well-established and neat comparison theorem from (Zhang, 2004) with respect to Hinge loss, which facilitates the generalization analysis of neural network classifiers.

The effectiveness of a classifier can be evaluated by its excess risk. Excess risk bounds are typically given regarding the underlying distribution ρ , the loss function, and the classification algorithm. Before we get into the main results of this paper, we would like to review some findings on the excess risk of different classifiers in the literature. We would like to pay special attention to their assumptions on the distribution ρ and their uses of loss functions.

1.1 Related Work

Here, we review some related work. Previously, (Jalali et al., 2019) studied the classification of GMM data in \mathbb{R}^d using 1-layer and 2-layer neural networks with C^∞ sigmoid-type activation functions. It considered the set $\mathcal{S}_{D,t} = \{x \in \mathbb{R}^d : D(x) \geq t\}$ with $t > 0$ and D being the GMM discriminant function. This is a bounded set on \mathbb{R}^d depending on the threshold t . It established a bound of the relative error $\left| \frac{\hat{D}(X) - D(X)}{D(X)} \right|$ for approximation on this set by 2-layer sigmoid networks. Neither estimates of approximation error nor excess risk is given. They imposed several regularity assumptions on the activation function, which ReLU does not satisfy.

Due to the availability of scalable computing and stochastic optimization techniques, sigmoid neural networks have taken a back seat to ReLU networks in the last decade. Deep ReLU networks are extensively used nowadays in practice because they have overcome optimization hurdles of vanishing gradients and exhibit superior empirical performances.

In the past three years, a handful of stimulating papers, e.g., (Feng et al., 2021; Kim et al., 2021; Shen et al., 2022), have studied the theoretical guarantees of ReLU neural networks for binary classification on bounded domains under structural assumptions of the regression function, noise or decision boundary. Note that the regression function is defined as the conditional mean $f_\rho(X) = \mathbb{E}[Y|X]$. Tsybakov’s noise condition (Tsybakov, 2004) with noise exponent $q \geq 0$ assumes that $\mathbb{P}(\{X \in \mathcal{X} : |f_\rho(X)| \leq t\}) = \mathcal{O}(t^q)$. A noteworthy work is (Kim et al., 2021). It showed that, with Hinge loss and noise exponent q , the empirical risk minimizer generated from ReLU fully-connected neural networks (ReLU FNNs) achieves rates of $\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+(d-1)(q+1)}}\right)$ and $\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+d}}\right)$ for the excess risk when the decision boundary belongs to $C^{0,\alpha}([0, 1]^d)$ or when the regression function $f_\rho(X) \in C^{0,\alpha}([0, 1]^d)$. Here $C^{0,\alpha}([0, 1]^d)$ denotes the space of Hölder continuous functions with order

$\alpha > 0$ in $[0, 1]^d$. Note that the results in (Kim et al., 2021) are derived under some additional conditions on the density of X .

Moreover, two recent works studied the binary classification by ReLU convolutional neural networks (ReLU CNNs) (Feng et al., 2021; Shen et al., 2022). Feng et al. (Feng et al., 2021) considered the p -norm loss $\phi(t) := \max\{0, 1-t\}^p, p \geq 1$ (1-norm loss is the hinge loss) and input data supported on the sphere \mathbb{S}^{d-1} in \mathbb{R}^d . The approximation error bound and the excess risk for a target function in the Sobolev space $W_p^r(\mathbb{S}^{d-1})$ (with $r > 0, p \geq 1$) are derived under a varying power condition. Two quantities including $\beta = \max\{1, (d + 3 + r)/(2(d - 1))\}$ and $\tau \in [0, 1]$ are involved in the convergence rates. More recently, (Shen et al., 2022) established convergence rates of the excess risk for classification with a class of convex loss functions. They considered data drawn from d -dimensional cube $[0, 1]^d$, which is compact. We would like to point out that the excess risk estimates of the above-mentioned works all contain a constant term depending on r or α . For example, Feng et al. gave excess risk bounds containing a constant 2^β that increases exponentially with the smoothness index $r > 0$ (Feng et al., 2021, Theorem 2).

Table 1 summarizes convergence rates of the excess risks in the existing literature and this paper, where the logarithm factors are dropped for brevity. The table is based on Table 3 in (Shen et al., 2022).

The regression function under a GMM is entire, implying that it is infinitely differentiable. **All the above-mentioned excess risk estimates in the literature increase to infinity when we take the smoothness index goes to be infinity, due to the constant terms involved.** In our work, we use the analyticity of the GMM regression function and obtain a novel result. We establish an excess risk estimate of order $\mathcal{O}\left(n^{-\frac{q+1}{q+2}}\right)$, where $q \geq 0$. Our result does not depend on any smoothness index or the dimension d . From Table 1, we can see that our convergence rate is faster than all existing results in the literature.

In the field of statistics, there are two classes of classification approaches — namely generative classification and discriminative classification. For GMM, the generative approach is to train a classifier by estimating the parameters (means and covariances) of the Gaussian components and then derive the Bayes classifier using the parameter estimates. On the other hand, the discriminative approach is to estimate the Bayes classifier from samples directly. Detailed and full definitions of generative and discriminative classifications can be found in (Ng and Jordan, 2001; Christmann, 2002; Li et al., 2015). We would like to point out that the classification of GMM by neural network is a discriminative approach since the procedure does not involve the estimation of GMM parameters.

We now turn our attention to the related works in classifications of GMM in the statistics literature. There is a long and continuing history of research on GMM. A complete review of GMM-related literature is not feasible. Here, we would like to focus on papers most related to our work, which are papers studying the classification of GMM.

A handful of statistics papers have studied the binary classification of GMM by generative approaches (Li et al., 2015, 2017). They considered a special kind of GMM — GMM consisting of only two Gaussian distributions in \mathbb{R}^d (each Gaussian distribution corresponds to one class) with identical covariance Σ . Let μ_0 and μ_1 denote the Gaussian means. Their analysis relies on the identical covariance assumption that the Bayes classifier is a linear function of a given sample $x \in \mathbb{R}^d$. Under a sparsity condition $\|\Sigma^{-1}(\mu_1 - \mu_0)\|_0 = s$ and

Reference	Function space	Loss	Condition	Rate
Tsybakov (2004)	measurable functions	0-1 loss	q -noise condition;	$\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+(d-1)q}}\right)$
Kim et al. (2021)	ReLU FNNs	Hinge	decision boundary $\in C^{0,\alpha}([0, 1]^d)$	$\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+(d-1)(q+1)}}\right)$
			q -noise condition; $f_\rho \in C^{0,\alpha}([0, 1]^d)$	$\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+d}}\right)$
Feng et al. (2021)	ReLU CNNs	Hinge	$f_\rho \in W_p^r(\mathbb{S}^{d-1})$	$\mathcal{O}\left(n^{-\frac{r}{2\beta(d-1)+r(2-\tau)}}\right)$
		p -norm		$\mathcal{O}\left(n^{-\frac{pr}{2(\beta+1)(d-1)+2pr(2-\tau)}}\right)$
		2-norm	$f_\rho \in W_p^r(\mathbb{S}^{d-1});$ q -noise condition	$\mathcal{O}\left(n^{-\frac{2rq}{(2+q)((\beta+1)(d-1)+2r}}\right)$
Shen et al. (2022)		Hinge	$f_\rho \in W_p^r([0, 1]^d);$ q -noise condition	$\mathcal{O}\left(n^{-\frac{r(q+1)}{d+2r(q+1)}}\right)$
		Logistic		$\mathcal{O}\left(n^{-\frac{r}{2d+4r}}\right)$
		Least square		$\mathcal{O}\left(n^{-\frac{4r}{3d+16r}}\right)$
Theorem 6 in this work	ReLU FNNs	Hinge	q -noise condition; GMM	$\mathcal{O}\left(n^{-\frac{q+1}{q+2}}\right)$

Table 1: This table compares the excess risk in the existing literature and this paper. The logarithm factors are dropped for brevity. Recall that $C^{0,\alpha}$ is the Hölder space of order $\alpha > 0$, W_p^r is the Sobolev space of order $r > 0$ and $p \geq 1$, the quantity $\beta = \max\{1, (d+3+r)/(2(d-1))\}$, and $\tau \in [0, 1]$.

some additional constraints on μ_0 and μ_1 , Li et al. (Li et al., 2015) derived an excess risk estimate depending on s, d , and n , achieved by some regularized logistic regression classifiers (under the 0-1 loss).

Subsequent to the above-mentioned work, Li et al. (Li et al., 2017) derived the excess risk estimates (under the 0-1 loss) achieved by Fisher’s linear discriminant under the same GMM setting, but with an additional assumption that the Gaussian distributions are isotropic (that is, $\Sigma = \sigma^2 I$ for some known σ). Since the analysis in the papers (Li et al., 2015) and (Li et al., 2017) require the Bayes classifier to be linear, their approaches cannot be applied to a general GMM setting.

In this work, we will first prove that deep ReLU neural networks can generate functions that approximate entire functions well on the whole unbounded domain. Using this approximation result and the fact that a GMM regression function is entire, we are able to establish an excess risk estimate of the classification of a general GMM. Our results apply to the binary classification of a general GMM without any assumptions on the number of Gaussian components or Gaussian parameters, especially the covariance or sparsity.

1.2 Our Contributions

To our best knowledge, this paper presents the best rate for the excess risk of classification with a GMM without restrictions on the domain, model parameters, or the number of Gaussian components. Our main contributions can be summarized as follows.

1. Our first result (Theorem 1) proves that there exists a ReLU DNN that can approximate the GMM discriminant function, which will be defined shortly, to any arbitrary accuracy as the depth of the network grows. This novel approximation error bound is given explicitly in terms of the network parameters and model parameters. By approximating the discriminant function, such a ReLU DNN can, in turn, approximate the Bayes classifier f_c well w.r.t. the misclassification error.
2. Next, our second result (Theorem 2) shows that there exists a ReLU DNN that can approximate a general analytic function well. This approximation result is of independent interest and may be useful in other problems.
3. We propose a special ReLU fully-connected neural network architecture for learning the Bayes classifier from GMM data (Section 4). The design of the special network is based on the results in Theorem 1, which guarantees a small approximation error when the network is sufficiently deep. With Hinge loss and a Tsybakov-type noise condition, our third result (Theorem 6) establishes a fast convergence rate of the excess risk of order $\mathcal{O}\left(n^{-\frac{q+1}{q+2}}(\log n)^4\right)$ using the proposed ReLU network architecture, where $q \geq 0$ is the noise exponent. We do not require the domain to be bounded. The convergence rate we obtained is faster than the existing results, and it does not depend on the dimension d , which demonstrates that ReLU networks can overcome the curse of dimensionality in classification.

The rest of the paper is organized as follows. In Section 2, we describe the setup of the binary classification problem and the class of ReLU-activated neural networks used in classification. In Section 3, we establish convergence rates on approximating GMM discriminant functions (Theorem 1) and general analytic functions (Theorem 2) via deep ReLU networks. In Section 4, we outline the construction of a special ReLU network architecture for learning the Bayes classifier from GMM data. In Section 5, we show that this network architecture achieves a fast convergence rate of excess risk under Tsybakov-type noise condition (Theorem 6). The proofs of Theorem 1 and Theorem 6 are provided in Sections 6 and 7, respectively. Concluding remarks are given in Section 8. Whenever possible, we relegate proofs of results, technical lemmas, and propositions to the Appendix.

2. Problem Formulations

In this section, we present the problem formulations. In Subsection 2.1, we describe our binary classification problem and the Gaussian Mixture Model from which our data is drawn. We then establish connections between the defined classification problem with the approximation of GMM discriminant function. In Subsection 2.2, we present a Tsybakov-type noise condition that is crucial for conducting generalization analysis of neural network classifiers. Lastly, in Subsection 2.3, we formulate the ReLU fully-connected neural networks we consider for classification.

2.1 Gaussian Mixture Models (GMM) and their classifiers

Consider the binary classification problem with data from the domain $\mathcal{X} = \mathbb{R}^d$ and output label $\mathcal{Y} = \{-1, 1\}$. A joint distribution ρ on $Z := \mathcal{X} \times \mathcal{Y}$ can be decomposed into the marginal distribution ρ_X on \mathcal{X} and the conditional distributions $\rho(\cdot|X)$ at $X \in \mathcal{X}$.

We are interested in learning a binary classifier $f : \mathbb{R}^d \rightarrow \{-1, 1\}$. As mentioned earlier, the Bayes classifier f_c (also known as the Bayes decision function) takes the form (1). Since the regression function f_ρ is given by the conditional mean $f_\rho(X) = \mathbb{E}[Y|X]$ at $X \in \mathcal{X}$, we observe that $f_c = \text{sgn}(f_\rho)$ (i.e., f_c is the sign of f_ρ).

In this paper, we study the binary classification problem with data generated from a Gaussian Mixture Model (GMM). Let $P^+ = \mathbb{P}(y = 1)$ and $P^- = \mathbb{P}(y = -1) = 1 - P^+$ denote the prior probabilities that a data point is drawn from the positive and negative class, respectively. Assume that members of each class are drawn from a mixture of Gaussian distributions. Also, assume that there are overall K different Gaussian distributions to draw from, where K is a positive integer at least 2. Each Gaussian distribution is assigned uniquely to one of the two classes. The assignment of the Gaussian distributions to the two classes is represented by sets \mathcal{T}^+ and \mathcal{T}^- (i.e., $\mathcal{T}^+ \cap \mathcal{T}^- = \emptyset$ and $\mathcal{T}^+ \cup \mathcal{T}^- = \{1, \dots, K\}$).

Conditioned on being in the positive class, let p_i , $i \in \mathcal{T}^+$, denote the probability that the data comes from Gaussian distribution i . Let $\mathbf{card}(S)$ be the cardinality of the set S . Then, for the positive class, the data are drawn from a mixture of $\mathbf{card}(\mathcal{T}^+)$ Gaussian distributions with densities

$$\mathcal{N}(x, \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(x - \mu_i)^T (\Sigma_i)^{-1} (x - \mu_i)\right), \quad i \in \mathcal{T}^+,$$

where $\mu_i \in \mathbb{R}^d$ and $\Sigma_i \in \mathbb{R}^{d \times d}$ denote the mean and the covariance matrix of Gaussian distribution i . We do not impose any constraints on the means or the covariances, except that we require the covariances to be non-degenerate. Similarly, data belonging to the negative class is drawn from $\mathbf{card}(\mathcal{T}^-)$ Gaussian distributions with densities $\mathcal{N}(x, \mu_i, \Sigma_i)$, for $i \in \mathcal{T}^-$.

For the positive class, we define its discriminant function $D^+ : \mathbb{R}^d \rightarrow \mathbb{R}$ in terms of the prior probabilities $\{p_i\}_{i \in \mathcal{T}^+}$:

$$D^+(x) := \mathbb{P}(x, y = 1) = P^+ \sum_{i \in \mathcal{T}^+} p_i \mathcal{N}(x, \mu_i, \Sigma_i),$$

while the discriminant function for the negative class $D^- : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined in the same way:

$$D^-(x) := \mathbb{P}(x, y = -1) = P^- \sum_{j \in \mathcal{T}^-} p_j \mathcal{N}(x, \mu_j, \Sigma_j).$$

Observe that the density function of the marginal distribution ρ_X equals $D^+(X) + D^-(X)$. The regression function f_ρ can thus be expressed as

$$f_\rho(X) = \mathbb{E}[Y|X] = \mathbb{P}(Y = 1|X) - \mathbb{P}(Y = -1|X) = \frac{D^+(X) - D^-(X)}{D^+(X) + D^-(X)}, \quad (2)$$

which implies

$$f_c(X) = \text{sgn}(f_\rho(X)) = \text{sgn}(D^+(X) - D^-(X)).$$

In other words, we can learn the Bayes classifier $f_c(X)$ by learning the sign of $D^+(X) - D^-(X)$. Denote by $D(X) = D^+(X) - D^-(X)$ our function of interest. Notice that D is of the form

$$D(x) = D^+(x) - D^-(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x)), \quad x \in \mathbb{R}^d, \quad (3)$$

where β_j are the constant coefficients, and $u_j(x)$ is a quadratic term given by

$$u_j(x) = (x - \mu_j)^T (\Sigma_j)^{-1} (x - \mu_j) / 2, \quad \forall j = 1, \dots, K. \quad (4)$$

Essentially, D is a linear combination of K Gaussian functions. Since all exponential functions are analytic everywhere, we know that D is entire and thus infinitely differentiable. This nice property of D enables us to learn a binary classifier that learns the Bayes classifier f_c with fast learning rates.

2.2 Tsybakov-type Noise Condition

The Tsybakov noise condition is widely used to study the quantitative behaviors of classification algorithms, e.g., (Tsybakov, 2004; Audibert and Tsybakov, 2007; Kim et al., 2021; Feng et al., 2021). The original Tsybakov noise condition (Mammen and Tsybakov, 1999; Tsybakov, 2004) is stated in terms of the regression function f_ρ . It assumes that for some $c_0 > 0$ and $q \in [0, \infty)$, $P(\{X \in \mathcal{X} : |f_\rho(X)| \leq t\}) \leq c_0 t^q, \forall t > 0$. Intuitively, it describes (with parameter q) how the regression function behaves around the boundary $\{x : f_\rho(x) = 0\}$. Bigger q means there is a jump of f_ρ near the boundary $\{x : f_\rho(x) = 0\}$, which is favorable for classification; smaller q (i.e., q close to 0) means there is a plateau behavior near the boundary, which is considered difficult for classification. For the extreme case of hard margin when q is infinity, it implies that f_ρ is bounded away from 0.

Throughout this paper, we assume a Tsybakov-type noise condition as follows.

Assumption 1. *Assume a Tsybakov-type noise condition that for some $c_0 > 0$ and $q \in [0, \infty)$, there holds*

$$P(\{X \in \mathcal{X} : |D(X)| \leq t\}) \leq c_0 t^q, \quad \forall t > 0, \quad (5)$$

where q is often referred to as the **noise exponent**.

Since $D(X) = (D^+(X) + D^-(X))f_\rho(X)$ in our GMM setting, and we know the factor $D^+(X) + D^-(X)$ is bounded above on the whole space \mathbb{R}^d and bounded below on any bounded domain, (5) is of the same type of Tsybakov noise condition.

2.3 Formulation of ReLU Fully-connected Neural Network (ReLU FNN)

Throughout this paper, we study deep fully-connected neural networks equipped with ReLU activation functions (ReLU FNNs), where the ReLU function is defined by $\sigma(a) = \max\{a, 0\}$. We consider deep ReLU FNNs that take d -dimensional inputs and produce one-dimensional outputs.

To mathematically define such a class of deep ReLU FNNs, we adopt the notation used by (Schmidt-Hieber, 2020) with slight modification. The network architecture (L, \mathbf{p}) consists of a positive integer L , which indicates the number of hidden layers (also known as the depth), and a width vector $\mathbf{p} = (p_1, \dots, p_L) \in \mathbb{N}^L$ which indicates the width in each hidden layer. A deep ReLU FNN with architecture (L, \mathbf{p}) can be written in the following compositional form

$$f(x) := f_\theta(x) = a \cdot \sigma \left(W^{(L)} \cdot \sigma \left(W^{(L-1)} \dots \sigma \left(W^{(1)} x + b^{(1)} \right) \dots + b^{(L-1)} \right) + b^{(L)} \right), \quad (6)$$

where $x \in \mathbb{R}^d$ is the input, $a \in \mathbb{R}^{p_L}$ is the outer weight, $W^{(i)}$ is a $p_i \times p_{i-1}$ weight matrix with $p_0 = d$, and $b^{(i)} \in \mathbb{R}^{p_i}$ is the bias vector, for $i = 1, \dots, L$. Denote by $\mathbf{W} = \{W^{(i)}\}_{i=1}^L$ the set of all weight matrices, $\mathbf{b} = \{b^{(i)}\}_{i=1}^L$ the set of all bias vectors, and $\theta = \{\mathbf{W}, \mathbf{b}, a\}$ the collection of all trainable parameters in the network.

From now on, we use $\mathcal{F}(L, \mathbf{p})$ to represent the set of functions of the form (6) produced by a class of ReLU FNNs with architecture (L, \mathbf{p}) .

3. Main Results I: Universal Approximation Theorem for GMM discriminant functions and general analytic functions

For $x \in \mathbb{R}^d$, we denote by $\|x\|$ the standard Euclidean norm in \mathbb{R}^d , unless otherwise specified. Recall that we assume our GMM model consists of K Gaussian distributions in total, each with mean and covariance denoted by μ_j and Σ_j , for $j = 1, \dots, K$. We define μ^* to be

$$\mu^* = \max_{1 \leq j \leq K} \|\mu_j\|. \quad (7)$$

Let $\tilde{\sigma}$ be the smallest eigenvalue of all the K covariance matrices $\{\Sigma_j\}_{j=1}^K$.

Our first result proves that there exists a ReLU FNN that approximates our function of interest $D(x) = D^+(x) - D^-(x)$ very well for x on a cube $[-b, b]^d$, while keeping bounded outside. The proof of Theorem 1 is given in Section 6.

Theorem 1. *Let $b \geq 1, m, \ell \in \mathbb{N}, C_K^0 \geq \frac{\sqrt{d}}{\sqrt{\tilde{\sigma}}}, C_K^1 \geq \frac{\mu^*}{\sqrt{\tilde{\sigma}}}$ and $R_2 \geq \sqrt{d}(b+1)C_K^0 + C_K^1$. Consider the GMM discriminant function D defined in (3). If $\ell \geq 2 \log((R_2)^2 d e) / \log(2)$, there exists a function $\tilde{D} : \mathbb{R}^d \rightarrow \mathbb{R}$ implementable by a ReLU FNN $\in \mathcal{F}(L, \mathbf{p})$ with $L = m + 3 + (m + 1)(\ell + 1)$ and the width vector $\mathbf{p} \in \mathbb{N}^L$ given by*

$$\begin{cases} p_1 = 4d + 2dK, p_2 = 1 + dK, \\ p_3 = p_{m+4} = 4dK, \\ p_{i+s(m+1)} = 10dK, & \text{if } s = 0, 1, i = 4, \dots, m+3, \\ p_{(j+2)(m+1)+4} = 5K(2^j), & \text{if } j = 0, \dots, \ell-1, \\ p_{(j+2)(m+1)+i} = 11K(2^j), & \text{if } j = 0, \dots, \ell-1, i = 5, \dots, m+4, \end{cases}$$

and with all weights and biases taking values in $[-4, 4]$ except for the L -th layer such that

$$\left| \tilde{D}(x) - D(x) \right| \leq C_{R_2} \left(\frac{2^{\ell+1}}{4^{m+1}} + 2^{-\frac{\ell(2^\ell)}{2}} \right), \quad \forall x \in [-b, b]^d \quad (8)$$

and

$$\left| \tilde{D}(x) \right| \leq C_{R_2}, \quad \forall x \in \mathbb{R}^d, \quad (9)$$

where $C_{R_2} = \left(\sum_{j=1}^K |\beta_j| \right) e^{(R_2)^{2d}}$.

Here are some interpretations of Theorem 1. On one hand, Inequality (8) shows that when the data is bounded in a d -dimensional cube, the function \tilde{D} closely approximates D to any arbitrary accuracy as the depth of the ReLU network grows (i.e., as m, ℓ goes to infinity). On the other hand, even when the data is unbounded in \mathbb{R}^d , Inequality (9) shows that the function value $\tilde{D}(x)$ is bounded by some constant. The error bounds given in (8) and (9) are important tools for deriving the excess risk bound later in Theorem 6.

Observe that $D(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x))$ is an entire function. In light of this, we can extend the above universal approximation theorem to a more general class of analytic functions. We first recall the definition of an analytic function. If a function t is analytic throughout a disk $|z - u_0| < R$, centered at u_0 and with radius $R > 0$, then t has the power series representation

$$t(z) = \sum_{i=0}^{\infty} \frac{t^{(i)}(u_0)}{i!} (z - u_0)^i, \quad |z - u_0| < R. \quad (10)$$

In other words, series (10) converges to $t(z)$ when z lies in the aforementioned open disk.

Now, we present our result on approximating a univariate analytic function via a deep ReLU FNN.

Theorem 2. *Let $m, \ell \in \mathbb{N}$, $1 < R_1 < R_0 \leq \infty$. Consider a univariate function $t(u)$ on $(-R_0, R_0)$, which can be extended to an analytic function on the disk $|z| < R_0$. For input $x \in [-1, 1]$, there exists a function F implementable by a deep ReLU FNN $\in \mathcal{F}(\ell(m+1), \mathbf{p})$ with width vector $\mathbf{p} \in \mathbb{N}^{\ell(m+1)}$ given by, for $j = 0 \dots, \ell - 1$,*

$$\begin{cases} p_{j(m+1)+1} = 5 \left(2^j \right), \\ p_{j(m+1)+i} = 11 \left(2^j \right), \quad \text{for } i = 2, \dots, m+1, \end{cases}$$

and with all weights and biases taking values in $[-4, 4]$ except the last layer such that

$$\sup_{x \in [-1, 1]} |F(x) - t(x)| \leq C_{R_1} \left(\frac{1}{4^{m+1}} + \frac{1}{R_1^{2\ell}} \right), \quad (11)$$

where $C_{R_1} = \frac{2^{7 + \lceil \log 4 / \log R_1 \rceil}}{(R_1 - 1)} \sup_{i \in \mathbb{Z}_+} \left| \frac{t^{(i)}(0)}{i!} (R_1)^i \right|$.

Theorem 2 shows that there exists a deep ReLU FNN that can approximate a general analytic function defined on $[-1, 1]$ to any arbitrary accuracy as the ReLU network grows. This approximation result is of independent interest and may be useful in other problems. The proof of Theorem 2 can be found in Appendix B.

Theorem 2 considers univariate analytic functions on a compact interval. A question arises at this point: Is it possible to generalize this result to multivariate analytic functions (on a compact domain)? While we believe such a generalization may be possible, it

requires some significant work, and thus we leave this for future study. The challenge of generalization to multivariate analytic functions is that our main tool to derive Theorem 2 is a monomial gate approximating univariate monomials with neat constant terms. It is not obvious how this monomial gate can approximate multivariate monomials with controlled constant terms.

4. Methodology

To solve the GMM classification problem effectively, our primary goal is to learn the optimal Bayes classifier $f_c = \text{sgn}(D)$ well, where D is the GMM discriminant function defined earlier in (3). To do so, we propose a special ReLU FNN architecture for learning the Bayes classifier from GMM data. This special ReLU FNN has an expansive binary-tree structure. The design of this network architecture is based on the results in Theorem 1, which guarantees a small approximation error for a sufficiently deep ReLU network. For brevity, we refer to this special network as the Expansive Binary-Tree ReLU network (EBTnet).

In this section, we first outline the design of the EBTnet (Subsection 4.1). Then, we describe a preprocessing subnetwork (Subsection 4.2). Lastly, in Subsection 4.3, we define our final network architecture and the hypothesis space for classification.

We will show, later in Section 5, that this network architecture (i.e., EBTnets followed after the preprocessing subnetwork) achieves a good excess risk bound.

4.1 Expansive Binary-tree network

Recall that Theorem 1 shows that there exists a function \tilde{D} implementable by a ReLU FNN that approximates D well. From the Taylor’s expansion of exponential functions, we know that $D(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x))$ can be expressed as a linear combination of monomials.

The results in Theorem 1 guide us to construct a monomial gate, which is a ReLU FNN designed to approximate linear combinations of monomial functions. This network has an expansive binary-tree structure. We start by introducing two important building blocks of the monomial gate, namely the squaring gate \hat{f}_m and the product gate $\hat{\Phi}$.

The squaring gate $\hat{f}_m =: \hat{f}_{m,\theta}$ is a ReLU FNN $\in \mathcal{F}(m, (5, 5, \dots, 5))$, where $m \in \mathbb{N}$. Each \hat{f}_m has parameters – weights \mathbf{W} , bias vectors \mathbf{b} and outer weights a – all taking values on $[-4, 4]$. This network architecture, introduced by (Yarotsky, 2017), is used to approximate the quadratic function $f(u) = u^2$ for any input $u \geq 0$. It is demonstrated in (Yarotsky, 2017, Proposition 2) that there exists a specific function $f_m(u) \in \mathcal{F}(m, (5, 5, \dots, 5))$ with all parameters bounded by 4 such that

$$f_m(u) \in [0, 1] \quad \text{and} \quad |u^2 - f_m(u)| \leq 4^{-(m+1)} \quad \forall u \in [0, 1]. \quad (12)$$

Here and later, we use the hat sign in \hat{f}_m to denote a network output function with flexible parameter choices while f_m without the hat sign denotes a specific network output function with specific parameter choices.

The main idea for constructing this specific squaring gate is to approximate u^2 by the network output $f_m(u) := u - \sum_{s=1}^m \frac{g_s(u)}{2^{2s}}$. Here, $g_s(u) := \underbrace{g \circ g \circ \dots \circ g}_s(u)$ is a s -compositions s folds

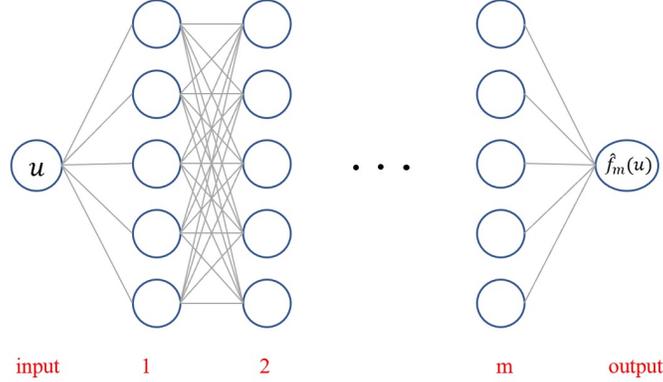


Figure 1: An illustration of the network architecture \widehat{f}_m for approximating u^2 for input $u \geq 0$. \widehat{f}_m is a ReLU FNN with m hidden layers, and each layer has 5 neurons. The red numbers below the architecture indicate the order of hidden layers.

of hat functions $g : [0, \infty) \rightarrow [0, 1]$ defined as

$$g(u) = 2\sigma(u) - 4\sigma\left(u - \frac{1}{2}\right) + 2\sigma(u - 1) = \begin{cases} 2u, & \text{if } 0 \leq u \leq \frac{1}{2}, \\ 2(1 - u), & \text{if } \frac{1}{2} < u \leq 1, \\ 0, & \text{if } u > 1. \end{cases} \quad (13)$$

We see that the function g can be implemented by a single-layer ReLU network with input $u \geq 0$. It follows that we can construct the squaring gate \widehat{f}_m , which belongs to $\mathcal{F}(m, (5, 5, \dots, 5))$. A more detailed discussion on \widehat{f}_m is given later in Subsection 6.1.

Next, invoking the identity

$$u \cdot v = \left| \frac{u+v}{2} \right|^2 - \left| \frac{u-v}{2} \right|^2$$

and \widehat{f}_m , we are able to construct a product gate $\widehat{\Phi} =: \widehat{\Phi}_\theta$, which is a ReLU FNN belonging to $\mathcal{F}(m+1, (4, 10, 10, \dots, 10))$ for $m \in \mathbb{N}$. A similar network construction can be found in (Suh et al., 2023, Lemma D.2.2). The first hidden layer of the product gate $\widehat{\Phi}$ takes $u, v \in \mathbb{R}$ as inputs and outputs $|\frac{u+v}{2}|$ and $|\frac{u-v}{2}|$ via $|u| = \sigma(u) + \sigma(-u)$. Then, $|\frac{u+v}{2}|$ and $|\frac{u-v}{2}|$ become inputs for two identical \widehat{f}_m respectively. We know that \widehat{f}_m takes m hidden layers (each with a width of 5) to output $\widehat{f}_m(|\frac{u+v}{2}|) \approx |\frac{u+v}{2}|^2$ and $\widehat{f}_m(|\frac{u-v}{2}|) \approx |\frac{u-v}{2}|^2$, respectively. These outputs are merged together via

$$\widehat{\Phi}(u, v) = \widehat{f}_m\left(\left|\frac{u+v}{2}\right|\right) - \widehat{f}_m\left(\left|\frac{u-v}{2}\right|\right).$$

We prove, later in Proposition 7 (in Subsection 6.1), that a specific function $\Phi \in \mathcal{F}(m+1, (4, 10, 10, \dots, 10))$ with a specific f_m can approximate the multiplication $u \cdot v$ to any arbitrary accuracy for m sufficiently large.

Now we are in a position to introduce the monomial gate – a ReLU network architecture that employs an expansive binary-tree structure. The aforementioned product gate $\widehat{\Phi}$ is an

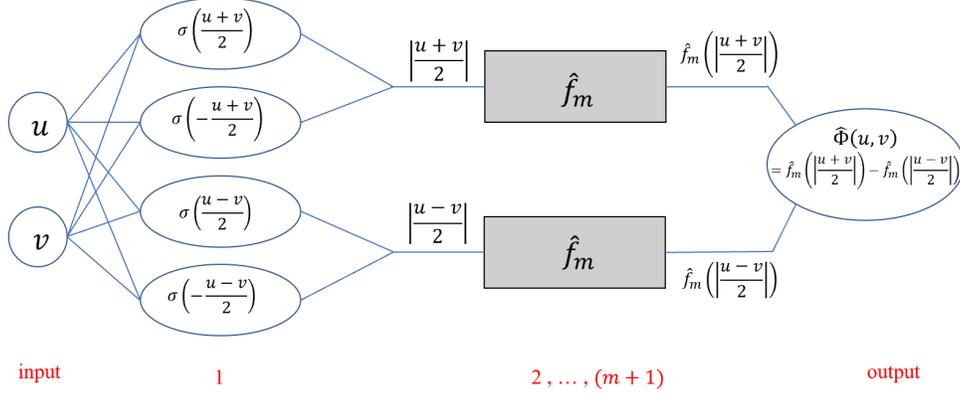


Figure 2: An illustration of $\widehat{\Phi}$ for approximating $u \cdot v$ with inputs $(u, v) \in \mathbb{R}^2$. $\widehat{\Phi}$ is a ReLU FNN with $m+1$ hidden layers. The first hidden layer generates $\frac{|u+v|}{2}$ and $\frac{|u-v|}{2}$, which become the inputs of two squaring gates \widehat{f}_m respectively. The red numbers below the architecture indicate the order of hidden layers.

important building block of this network. The monomial gate is constructed to approximate monomial functions of degree $k \in \mathbb{N}$, that is u^k with input $u \in [-1, 1]$. This ReLU FNN, as illustrated in Figure 3, belongs to $\mathcal{F}(\ell(m+1), \mathbf{p})$, where the width vector $\mathbf{p} \in \mathbb{N}^{\ell(m+1)}$ is given by, for $j = 0 \dots, \ell - 1$,

$$\begin{cases} p_{j(m+1)+1} = 5 (2^j), \\ p_{j(m+1)+i} = 11 (2^j), \quad \text{for } i = 2, \dots, m+1. \end{cases}$$

Notice that $\widehat{\Phi}(u, u) = \widehat{f}_m(|u|) - \widehat{f}_m(0)$. Define the functions $\{\widehat{h}_k(u)\}_{k=1}^{2^\ell}$ on \mathbb{R} for $\ell \in \mathbb{N}$ by

$$\{\widehat{h}_k(u)\}_{k=1}^2 = \left\{ \widehat{h}_1(u) = u, \widehat{h}_2(u) = \widehat{\Phi}(u, u) = \widehat{f}_m(|u|) - \widehat{f}_m(0) \right\}, \quad (14)$$

and iteratively for $j = 1, \dots, \ell - 1$,

$$\widehat{h}_{2^j+i}(u) = \widehat{\Phi}(\widehat{h}_{2^j}(u), \widehat{h}_i(u)), \quad i = 1, \dots, 2^j. \quad (15)$$

The key idea for constructing the monomial gate is to employ an expansive binary-tree structure. The network comprises ℓ subnetworks, each equipped with product gates $\widehat{\Phi}$. The 1st subnetwork takes the input u and outputs $\widehat{h}_2(u) = \widehat{\Phi}(u, u)$ and u . The outputs of the 1st subnetwork become the inputs of the 2nd subnetwork, which outputs

$$\{u, \widehat{h}_2(u)\} \rightarrow \{u, \widehat{h}_2(u), \widehat{h}_3(u) = \widehat{\Phi}(\widehat{h}_2(u), u), \widehat{h}_4(u) = \widehat{\Phi}(\widehat{h}_2(u), \widehat{h}_2(u))\}.$$

The k -th subnetwork has 2^{k-1} product gates $\widehat{\Phi}$. This subnetwork takes in the outputs from the $(k-1)$ -th subnetwork and computes

$$\{u, \widehat{h}_2(u), \dots, \widehat{h}_{2^{k-1}}(u)\} \rightarrow \{u, \widehat{h}_2(u), \dots, \widehat{h}_{2^{k-1}}(u), \dots, \widehat{h}_{2^k}(u)\}.$$

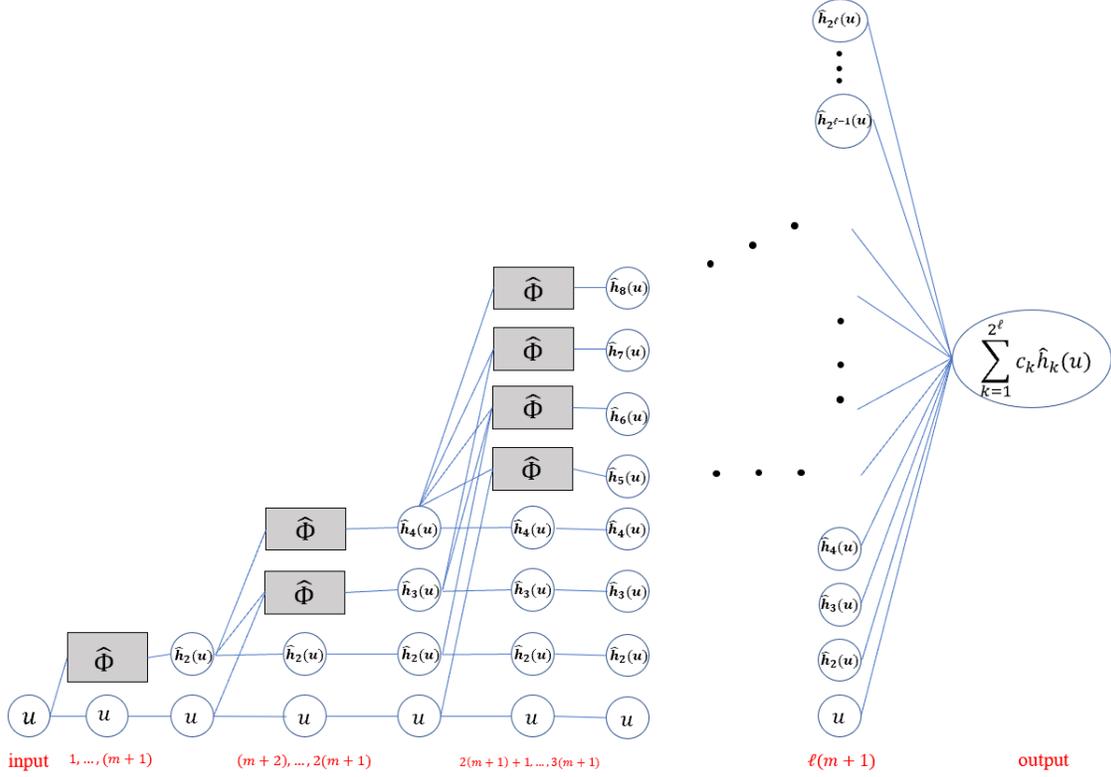


Figure 3: An illustration of the monomial gate for approximating the set of monomials $\{u^k\}_{k=1}^{2^\ell}$ with input $u \in [-1, 1]$. The network has an expansive binary-tree structure. It comprises ℓ subnetworks equipped with the product gate $\hat{\Phi}$. The width of the k -th subnetwork is doubled from that of the $(k - 1)$ -th subnetwork for $k = 1, \dots, \ell$.

In this way, we can see that the width of the k -th subnetwork is doubled from that of the $(k - 1)$ -th subnetwork for $k = 1, \dots, \ell$, thereby forming the expansive binary-tree structure. The final output of the monomial gate (i.e., an EBTnet) is $\sum_{k=1}^{2^\ell} c_k \hat{h}_k(u)$, which is a linear combination of $\hat{h}_k(u)$ with some constant coefficients c_k for $k = 1, \dots, 2^\ell$.

Later in Proposition 8 (in Subsection 6.2), we prove that there exists a specific set of functions $\{h_k(u)\}_{k=1}^{2^\ell}$, implemented by this EBTnet with specific parameter choices, approximating the set of monomials $\{u^k\}_{k=1}^{2^\ell}$ up to any arbitrary accuracy, given that the depth of the network is sufficiently large. The approximation error bound is obtained by induction.

4.2 Preprocessing subnetwork: High Dimensional Truncation of Unbounded Data

Notice that input data generated from GMM is unbounded. Later when we conduct a generalization analysis of ReLU network classifiers, we need to estimate the covering num-

bers of a set of output functions generated from the EBTnet. However, the unboundedness of input data makes estimating covering numbers infeasible. We do not choose to make the assumption that data is bounded because Gaussian distributions are unbounded, and real-world features, such as images and speeches, are typically unbounded.

To circumvent the challenges that come with unbounded data, we adopt a specific preprocessing subnetwork which is placed prior to the EBTnets. This preprocessing subnetwork contains a high-dimensional truncation unit. This truncation unit, which will be defined shortly, manually projects d -dimensional unbounded input to a bounded domain for $d \in \mathbb{N}$.

We first consider the case $d = 1$. Let $b > 0$, define a univariate trapezoid-shaped function $T_b : \mathbb{R} \rightarrow \mathbb{R}$ by

$$T_b(u) := \sigma(u + b + 1) - \sigma(u + b) - \sigma(u - b) + \sigma(u - b - 1) \quad (16)$$

$$= \begin{cases} u + b + 1, & \text{if } -b - 1 \leq u < -b, \\ 1, & \text{if } -b \leq u \leq b, \\ -u + b + 1, & \text{if } b < u \leq b + 1, \\ 0, & \text{if } u < -b - 1 \text{ or } u > b + 1. \end{cases}$$

With $u \in \mathbb{R}$, T_b can be implemented by a single-layer ReLU network without any free parameter.

Now consider $d > 1$. We extend the univariate trapezoid-shaped function to a higher dimensional space. With input $u = (u_1, u_2, \dots, u_d) \in \mathbb{R}^d$, define a d -dimensional truncation function $\Psi_b : \mathbb{R}^d \rightarrow \mathbb{R}$ by

$$\Psi_b(u) := \sigma \left\{ \sum_{i=1}^d T_b(u_i) - (d - 1) \right\}. \quad (17)$$

The truncation function Ψ_b alone can be implemented by a ReLU FNN with 1 hidden layer (of width $4d$) without any free parameter, as illustrated below:

$$\begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} \rightarrow \begin{bmatrix} \sigma(u_1 + b + 1) \\ \sigma(u_1 + b) \\ \sigma(u_1 - b) \\ \sigma(u_1 - b - 1) \\ \vdots \\ \vdots \\ \vdots \\ \sigma(u_d + b + 1) \\ \sigma(u_d + b) \\ \sigma(u_d - b) \\ \sigma(u_d - b - 1) \end{bmatrix} \rightarrow \Psi_b(u) = \sigma \left\{ \sum_{i=1}^d T_b(u_i) - (d - 1) \right\}.$$

A similar truncation network is given in (Shaham et al., 2018). The following Lemma presents the truncation property of Ψ_b . Its proof is given in Appendix A.1.

Lemma 3. *Let $b > 0$, and Ψ_b be defined by (17). With input $u \in \mathbb{R}^d$, we have*

$$0 \leq \Psi_b(u) \leq 1$$

and

$$\Psi_b(u) = \begin{cases} 0, & \text{if } u \notin [-b-1, b+1]^d, \\ 1, & \text{if } u \in [-b, b]^d. \end{cases}$$

Lemma 3 tells us that applying Ψ_b to any data $u \in \mathbb{R}^d$ will project the data onto the interval $[0, 1]$. Particularly, if u lies outside the cube $[-b-1, b+1]^d$, Ψ_b will truncate u to 0 (i.e., makes u vanishes).

Now, we apply the truncation function Ψ_b to our data, which is $x \in \mathbb{R}^d$ generated from some GMM. Recall that we assume our model has a total of K Gaussian distributions, each with mean and covariance denoted by μ_j and Σ_j , for $j = 1, \dots, K$. Since x is unbounded, it follows that

$$u_j(x) = (x - \mu_j)^T (\Sigma_j)^{-1} (x - \mu_j) / 2, \quad \forall j = 1, \dots, K,$$

are also unbounded. To learn the GMM discriminant function $D(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x))$ (previously defined in (3)) by ReLU FNNs, we need to input $u_j(x)$ into the EBTnet. But since $u_j(x)$ is unbounded, we first apply Ψ_b to project $u_j(x)$ onto a bounded domain.

Let $r_{i,j}(x)$ be the i -th component of the vector $(\Sigma_j)^{-1/2} \frac{(x - \mu_j)}{\sqrt{2}}$ for $i = 1, \dots, d$ and $j = 1, \dots, K$. They are affine functions of x and thus can be implemented by a ReLU FNN via $r_{i,j}(x) = \sigma(r_{i,j}(x)) - \sigma(-r_{i,j}(x))$. We use a ReLU FNN belonging to $\mathcal{F}(2, (2, 1))$ with the hypothesis space

$$\mathring{\mathcal{H}} = \left\{ \mathring{W} \cdot x + \mathring{b} : \mathring{W} \in \mathbb{R}^d, \|\mathring{W}\| \leq C_K^0, |\mathring{b}| \leq C_K^1 \right\} \quad (18)$$

to learn $r_{i,j}(x)$ by $\hat{r}_{i,j}(x) = \mathring{W}_{i,j} \cdot x + \mathring{b}_{i,j}$, where C_K^0 and C_K^1 are tunable parameters.

Instead of directly applying Ψ_b truncate to $\hat{r}(x)$, we make use of Ψ_b and the product gate $\hat{\Phi}$ simultaneously. **More specifically, we construct a preprocessing subnetwork to compute $\hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x))$ with input $x \in \mathbb{R}^d$.**

Let us explain the purpose of computing $\hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x))$ here. From Lemma 3, we know that

$$\begin{cases} \hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x)) = \hat{\Phi}(0, \hat{r}_{i,j}(x)) = 0, & \text{if } x \notin [-b-1, b+1]^d, \\ \hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x)) = \hat{\Phi}(1, \hat{r}_{i,j}(x)), & \text{if } x \in [-b, b]^d. \end{cases} \quad (19)$$

for all $i = 1, \dots, d$ and $j = 1, \dots, K$.

Recall that $\Phi(u, v)$ is a specific network architecture that is used to approximate the multiplication $u \cdot v$. Equations (19) tells us that if x lies in $[-b, b]^d$, the preprocessing subnetwork generates $\hat{\Phi}(1, \hat{r}_{i,j}(x))$, which is used to approximate $1 \cdot r_{i,j}(x) = r_{i,j}(x)$. On the other hand, if x lies outside the d -dimensional cube $[-b-1, b+1]^d$, the preprocessing subnetwork outputs 0. This is how the preprocessing network makes $\hat{r}_{i,j}(x)$ vanishes when $x \in \mathbb{R}^d$ is too large.

The following flowchart (20) illustrates the preprocessing subnetwork. It is equipped with one truncation unit Ψ_b and $d \cdot K$ product gates $\hat{\Phi}$. This subnetwork alone belongs to $\mathcal{F}(m+3, (4d+2dK, 1+dK, 4dK, 10dK, 10dK, \dots, 10dK))$. The final outputs is the

collection of $\widehat{\Phi}(\Psi_b(x), \widehat{r}_{i,j}(x))$ for $i = 1, \dots, d$, and $j = 1, \dots, K$.

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix} \Psi_b(x) \\ \widehat{r}_{1,1}(x) \\ \vdots \\ \widehat{r}_{d,1}(x) \\ \vdots \\ \widehat{r}_{1,K}(x) \\ \vdots \\ \widehat{r}_{d,K}(x) \end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix} \widehat{\Phi}(\Psi_b(x), \widehat{r}_{1,1}(x)) \\ \vdots \\ \widehat{\Phi}(\Psi_b(x), \widehat{r}_{d,K}(x)) \end{bmatrix} \quad (20)$$

Next, we will define our final ReLU network architecture. In the final network architecture, the preprocessing subnetwork is placed at the beginning.

4.3 Defining the Hypothesis Space

Here, we define our final neural network architecture and the hypothesis space \mathcal{H} for classification. Recall the Bayes classifier f_c given by

$$f_c(x) = \text{sgn}(D(x)) = \text{sgn} \left(\sum_{j=1}^K \beta_j \exp(-u_j(x)) \right),$$

where β_j are constant coefficients given by the model, and $u_j(x) = (x - \mu_j)^T (\Sigma_j)^{-1} (x - \mu_j) / 2$.

The function space \mathcal{H} consists of functions implementable by a ReLU FNN and closely approximate f_c for a given $x \in \mathcal{X}$. Before we introduce \mathcal{H} , we first define the function $\sigma_\lambda : \mathbb{R} \rightarrow [-1, 1]$ for some $0 < \lambda \leq 1$ to be the linear combination of four scaled ReLU units given by

$$\sigma_\lambda(u) := \sigma\left(\frac{u}{\lambda}\right) - \sigma\left(\frac{u}{\lambda} - 1\right) - \sigma\left(-\frac{u}{\lambda}\right) + \sigma\left(-\frac{u}{\lambda} + 1\right) = \begin{cases} 1, & \text{if } u \geq \lambda, \\ \frac{u}{\lambda}, & \text{if } u \in [-\lambda, \lambda), \\ -1, & \text{if } u < -\lambda. \end{cases} \quad (21)$$

We can see that if λ is close to 0, $\sigma_\lambda(u)$ is close to $\text{sgn}(u)$. In other words, we use the function σ_λ to approximate the sign function.

Recall that $\mu^* = \max_{1 \leq j \leq K} \|\mu_j\|$. Here, we give the definition of our hypothesis space \mathcal{H} . We will show, later in Theorem 6, that functions in \mathcal{H} can indeed learn f_c with a fast learning rate (i.e., the excess risk converges to 0 fast).

Definition 4 (Hypothesis Space \mathcal{H}). *Let $0 < \lambda \leq 1$, $b \geq 1$ and $m, \ell \in \mathbb{N}$. Also let $R_2 \geq 1 + \frac{1}{\sqrt{\sigma}}(\sqrt{d}(b+1) + \mu^*)$. Define $h_{k,j}^* : \mathbb{R}^d \rightarrow \mathbb{R}$ by*

$$h_{k,j}^*(x) = \widehat{h}_k \left(\frac{1}{d} \sum_{i=1}^d \widehat{\Phi} \left(\widehat{\Phi}(\Psi_b(x), \widehat{r}_{i,j}(x)), \widehat{\Phi}(\Psi_b(x), \widehat{r}_{i,j}(x)) \right) \right). \quad (22)$$

With input $x \in \mathcal{X} = \mathbb{R}^d$, the hypothesis space \mathcal{H} is given by

$$\mathcal{H} = \left\{ \sigma_\lambda \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j} h_{k,j}^*(x) + c_0 \right) : |c_{k,j}|, |c_0| \leq C_K, \mathbf{W} = \{W_i\}_{i=1}^m \in [-4, 4]^{5 \times 5}, \right. \\ \left. \mathbf{b} = \{b_i\}_{i=1}^m \in [-4, 4]^5, a \in [-4, 4]^5, \hat{r}_{i,j} \in \hat{\mathcal{H}} \right\}.$$

Here, C_K is a positive tunable parameter equal to or greater than $C_{R_2} = \sum_{j=1}^K |\beta_j| \exp((R_2)^2 d)$.

Remark 5. The hypothesis space \mathcal{H} consists of functions implementable by ReLU FNNs $\in \mathcal{F}(L, \mathbf{p})$ where $L = m + 3 + (\ell + 1)(m + 1) + 1$ and the width vector $\mathbf{p} \in \mathbb{N}^L$ given by

$$\begin{cases} p_1 = 4d + 2dK, p_2 = 1 + dK, \\ p_3 = p_{m+4} = 4dK, \\ p_{i+s(m+1)} = 10dK, & \text{if } s = 0, 1, i = 4, \dots, m + 3, \\ p_{(j+2)(m+1)+4} = 5K (2^j), & \text{if } j = 0, \dots, \ell - 1, \\ p_{(j+2)(m+1)+i} = 11K (2^j), & \text{if } j = 0, \dots, \ell - 1, i = 5, \dots, m + 4, \\ p_L = 4. \end{cases}$$

The beginning of the network is a preprocessing subnetwork that outputs $\hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x))$ for $i = 1, \dots, d$, and $j = 1, \dots, K$. Each of these outputs is used to approximate $r_{i,j}(x)$. Next, each output from the preprocessing subnetwork enters one product gate $\hat{\Phi}$ (i.e., we have K product gates $\hat{\Phi}$ followed by the preprocessing subnetwork). This group of product gates together outputs $\frac{1}{d} \sum_{i=1}^d \hat{\Phi}(\hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x)), \hat{\Phi}(\Psi_b(x), \hat{r}_{i,j}(x)))$ for $j = 1, \dots, K$, which is used to approximate $\frac{1}{d} \sum_{i=1}^d (r_{i,j}(x))^2 = \frac{1}{d} u_j(x)$. Then, each of these products enters a monomial gate (EBTnet), and all outputs of the K monomial gates are merged together via $\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j} h_{k,j}^*(x) + c_0$. If ℓ is large, this output can be used to approximate a linear combination of $\{(u_j(x))^k\}_{k=0}^\infty$, and thereby approximate $D(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x))$ with suitable choices of coefficients $c_{k,j}$ and c_0 . The last layer of the network is a scaling unit σ_λ , which is close to the sign function if λ is close to 0. We use the output function $f(x) \in \mathcal{H}$ to learn our target Bayes classifier $f_c(x) = \text{sgn}(D(x))$.

5. Main Results II: Generalization Analysis of ReLU Network Classifier

Consider the hypothesis space \mathcal{H} defined above in Definition 4. For any function f in \mathcal{H} , the misclassification error w.r.t. the probability measure ρ is defined as

$$R(f) := \mathbb{E}[\mathbb{1}_{\{Y \cdot \text{sgn}(f(X)) = -1\}}] = \mathbb{P}(Y \cdot \text{sgn}(f(X)) = -1), \quad \forall f \in \mathcal{H} \quad (23)$$

To show functions in \mathcal{H} can learn f_c sufficiently well, we aim to find a $f \in \mathcal{H}$ that minimizes the excess risk $R(f) - R(f_c) \geq 0$. In practice, the probability measures ρ are usually unknown. The classifier f will be learned based on a random sample $z := \{(x_i, y_i)\}_{i=1}^n$

drawn independently and identically distributed from ρ , where n is the sample size. We find a classifier that minimizes the empirical risk:

$$\hat{f}_n := \arg \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{y_i \cdot \text{sgn}(f(x_i)) = -1\}}.$$

However, the empirical risk w.r.t. 0-1 loss – the natural loss function for binary classification – is non-continuous and non-convex (Bartlett et al., 2006). Instead of minimizing the 0-1 loss, we can adopt some convex loss function $V : \mathcal{X} \times \mathcal{Y} \rightarrow [0, \infty)$ to make computation feasible. The Hinge loss, defined as $\phi(t) := \max\{0, 1 - t\}$, is one of the most commonly used loss functions in maximum-margin classifications, most notably the support vector machine (Rosasco et al., 2004). The generalization error associated with the Hinge loss ϕ for f is defined by

$$\begin{aligned} \varepsilon(f) &= \int_{\mathcal{Z}} \phi(yf(x)) d\rho = \int_{\mathcal{X}} \int_{\mathcal{Y}} \phi(yf(x)) d\rho(y|x) d\rho_X \\ &= \int_{\mathcal{X}} (\phi(f(x))P(y = 1|x) + \phi(-f(x))P(y = -1|x)) d\rho_X. \end{aligned}$$

Given a sample $z = \{(x_i, y_i)\}_{i=1}^n$, define the empirical risk of f w.r.t. ϕ over z as

$$\varepsilon_z(f) := \frac{1}{n} \sum_{i=1}^n \phi(y_i f(x_i)). \quad (24)$$

Our goal is to find a classifier $f \in \mathcal{H}$ that minimizes the empirical risk w.r.t. ϕ , that is, the empirical risk minimizer (ERM) defined as

$$f_z := \arg \min_{f \in \mathcal{H}} \varepsilon_z(f) \quad (25)$$

The well-known Comparison Theorem in classification in (Zhang, 2004) suggests that, for the Hinge loss ϕ and any measurable function $f : \mathcal{X} \rightarrow \mathbb{R}$,

$$R(\text{sgn}(f)) - R(f_c) \leq \varepsilon(f) - \varepsilon(f_c). \quad (26)$$

In other words, we can minimize the excess generalization error (also known as the excess ϕ -error) $\varepsilon(f_z) - \varepsilon(f_c)$ to, in turn, bound the excess risk $R(\text{sgn}(f_z)) - R(f_c)$. To derive the convergence rate of the excess generalization error, we assume the Tsybakov-type noise condition given earlier in Assumption 1. As a recap, Assumption 1 asserts that for some $c_0 > 0$ and $q \in [0, \infty)$, there holds

$$P(\{X \in \mathcal{X} : |D(X)| \leq t\}) \leq c_0 t^q, \quad \forall t > 0,$$

where q is often referred to as the noise exponent.

Our third result derives the convergence rate of the excess generalization error $\varepsilon(f_z) - \varepsilon(f_c)$, which in turn gives the convergence rate of the excess risk. In other words, the following theorem establishes the learning rate of ReLU networks on learning the Bayes classifier f_c of GMM data. To our best knowledge, this is the first generalization error bound for classifications under a general GMM setting without constraints on model parameters or the number of Gaussian components. In particular, we do not impose any sparsity condition on the covariance matrices. The proof of Theorem 6 is given in Section 7.4.

Theorem 6. Let $n \geq 3, q > 0, b \geq 1, C_K^0 \geq \frac{\sqrt{d}}{\sqrt{\delta}}, C_K^1 \geq \frac{\mu^*}{\sqrt{\delta}}, R_2 = 2\sqrt{d}(C_K^0 + C_K^1)b, C_K = c_2 \exp(c_3 b^2)$ with $c_2 \geq \sum_{j=1}^K |\beta_j|, c_3 \geq 4d^2(C_K^0 + C_K^1)^2$. Let f_z be the empirical risk minimizer of the hypothesis space \mathcal{H} with the chosen C_K . Suppose the noise condition (5) holds for some noise exponent q and constant $c_0 > 0$. Take $\lambda = n^{-\frac{1}{q}}, m = \ell(2^{\ell-1}), b = \sqrt{c'_q \ell(2^{\ell-1})}$ with $c'_q = \frac{q(\log 2)}{\frac{1}{16\sigma^*} + c_3 q}$ and ℓ to be the smallest integer satisfying $\ell 2^{\ell-1} \geq \frac{16\sigma^*}{c'_q} \log n$. For any $0 < \delta < 1$, with probability $1 - \delta$, there holds,

$$R(\text{sgn}(f_z)) - R(f_c) \leq \varepsilon(f_z) - \varepsilon(f_c) \leq C_{q,d} \log\left(\frac{2}{\delta}\right) (\log n)^4 \left(\frac{1}{n}\right)^{\frac{q+1}{q+2}}, \quad (27)$$

where $C_{q,d}$ is a positive constant independent of n or δ .

We can see that the excess risk bound depends on q , the noise exponent. Theorem 6 tells us that when $q = 0$ (no noise assumption), the convergence rate of the excess risk is of $\mathcal{O}\left((1/n)^{\frac{1}{2}}(\log n)^4\right)$. When q increases (stronger noise assumption), the convergence rate of the excess risk approaches $\mathcal{O}\left((1/n)(\log n)^4\right)$.

In the classic literature, convergence rates of the excess risk of order $\mathcal{O}\left((1/n)^{\frac{1}{2}}\right)$ with 0-1 loss were established (Mammen and Tsybakov, 1999). Moreover, it is proven in (Tsybakov, 2004) that, when the ERM is taken over all measurable classifiers, the minimax lower bound of the excess risk is $\mathcal{O}\left(n^{-\frac{\alpha(q+1)}{\alpha(q+2)+(d-1)q}}\right)$, under Tsybakov noise condition and when the decision boundary is α -Hölder smooth.

Recall Table 1 given earlier in Section 1. This table compares our finding in Theorem 6 with the existing results on excess risk in the literature. Comparatively, our result does not depend on any smoothness or regularity measure, whereas the existing results from (Feng et al., 2021; Kim et al., 2021; Shen et al., 2022) contain constant terms depending on smoothness index. For example, the excess risk estimate given in (Feng et al., 2021, Theorem 2) contains a constant term that increases exponentially with the smoothness index $r > 0$. Since the regression function under GMM is infinitely differentiable, existing results increase to infinity when we take the smoothness index to be infinity.

Moreover, notice that our convergence rate does not depend on the dimension d . Such a result matches the well-established phenomenon that when the smoothness of the functions is proportional to the dimensionality, then the rates of excess risk are dimension-independent, e.g., the Barron class (E et al., 2022).

All existing results on function approximations by ReLU neural networks consider only functions on bounded domains. Specifically, approximation of polynomials (and thereby functions with smoothness) on a bounded domain by ReLU networks has been extensively studied, starting from the seminal work (Yarotsky, 2017). However, there is no result on function approximation on an unbounded domain by ReLU networks, even for polynomials. Our novelty in this aspect is that we utilize the special composition form of the GMM discriminant function: It is a composition of quadratic polynomials on the whole space \mathbb{R}^d with a univariate exponential function. Our network is designed specifically to learn functions of this form. Our approach uses the positivity of these quadratic polynomials and the decay of the exponential function e^{-u} , which enables us to reduce the approximation

from the whole space to a bounded domain. It is possible to extend our results to the setting where the exponential function e^{-u} is replaced with some other entire function decaying at infinity. However, this is not the main focus of this work on GMM classification.

6. Proof of Theorem 1: Approximation of GMM Discriminant Function by ReLU Network

In this section, we present the proof of Theorem 1. The proof of Theorem 1 can be divided into three steps. First, we prove that the new product gate can approximate multiplication well (Subsection 6.1). Then, using the new product gate, we study how ReLU networks can approximate monomial functions (Subsection 6.2). After that, we prove Theorem 1 by showing how the GMM discriminant function D can be well approximated by ReLU networks (Subsection 6.3).

6.1 A Novel Product gate

From (Yarotsky, 2017), it is shown that a m -layer ReLU FNN can approximate $f(u) = u^2$ to an accuracy $\frac{1}{4^{m+1}}$ with input $u \in [0, 1]$. Inspired by Yarotsky's results, we propose a new product gate that achieves the same approximation accuracy on $[0, 1]^2$ and has a linear increment on \mathbb{R}^2 while adopting a structure that enables efficient approximation of monomials later.

Recall the hat function $g : [0, \infty) \rightarrow [0, 1]$ we defined earlier in (13) by extending the construction by (Yarotsky, 2017) on the interval $[0, 1]$ to $[0, \infty)$. The function g can be regarded as the output of a ReLU network with 1 hidden layer and 3 neurons. The s -composition g_s of g with itself on $[0, \infty)$ takes the form

$$g_s(u) := \underbrace{g \circ g \circ \dots \circ g(u)}_{s \text{ folds}} = \begin{cases} 2^s \left(u - \frac{2k}{2^s}\right), & \text{if } u \in \left[\frac{2k}{2^s}, \frac{2k+1}{2^s}\right], k = 0, 1, \dots, 2^{s-1} - 1, \\ 2^s \left(\frac{2k}{2^s} - u\right), & \text{if } u \in \left[\frac{2k-1}{2^s}, \frac{2k}{2^s}\right], k = 1, 2, \dots, 2^{s-1}, \\ 0, & \text{if } u > 1. \end{cases}$$

With input $u \geq 0$, we can generate g_s for $s = 1, 2, \dots, m$ by a ReLU network of m layers. Denote a function vector $\boldsymbol{\sigma}$ by

$$\boldsymbol{\sigma}(u) = \begin{bmatrix} \sigma(u-1) \\ \sigma(u-1/2) \\ \sigma(u) \end{bmatrix}. \quad (28)$$

By g_s and $\sigma(u) = u$ for $u \geq 0$, the following flow chart illustrates how the functions $\{u, g_1(u), g_2(u), \dots, g_m(u)\}$ are produced by a ReLU FNN with m hidden layers:

$$u \rightarrow [\boldsymbol{\sigma}(u)] \rightarrow \begin{bmatrix} \boldsymbol{\sigma}(g_1(u)) \\ u \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{\sigma}(g_2(u)) \\ g_1(u) \\ u \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} \boldsymbol{\sigma}(g_{m-1}(u)) \\ \sum_{s=1}^{m-2} \frac{g_s(u)}{2^{2s}} \\ u \end{bmatrix} =: G_m(u) \rightarrow f_m(u), \quad (29)$$

where $f_m : [0, \infty) \rightarrow [0, \infty)$ is defined as a linear combination of $\{g_s\}_{s=1}^m$ given by

$$f_m(u) = u - \sum_{s=1}^m (f_{s-1}(u) - f_s(u)) = \begin{cases} u - \sum_{s=1}^m \frac{g_s(u)}{2^{2s}}, & \text{if } 0 \leq u \leq 1, \\ u, & \text{if } u > 1. \end{cases}$$

As observed in (Yarotsky, 2017), on the interval $[0, 1]$, f_m is the piecewise linear interpolation of u^2 on 2^m subintervals on $[0, 1]$ with breakpoints $\{0, \frac{1}{2^m}, \dots, \frac{2^m}{2^m} = 1\}$. From (29), we see that f_m can be implemented by a ReLU FNN with m hidden layers, each of which consists of 5 nodes or units. All the parameters take values on $[-4, 4]$.

Now, motivated by the identity $u \cdot v = \left|\frac{u+v}{2}\right|^2 - \left|\frac{u-v}{2}\right|^2$, we introduce a novel product gate $\Phi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ as

$$\Phi(u, v) = f_m \left(\left| \frac{u+v}{2} \right| \right) - f_m \left(\left| \frac{u-v}{2} \right| \right), \quad \forall u, v \in \mathbb{R}. \quad (30)$$

It follows that the multiplication function $u \cdot v$ can be well-approximated by a ReLU network, as stated in the proposition below.

Proposition 7. *Let $m \in \mathbb{N}$. With the input $(u, v) \in \mathbb{R}^2$, the function $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ can be implemented by a ReLU FNN $\in \mathcal{F}(m+1, (4, 10, 10, \dots, 10))$ with all the parameters take values on $[-4, 4]$ such that*

1. $\Phi(u, v) = 0$ if $u = 0$ or $v = 0$;
2. $|\Phi(u, v)| \leq |u| + |v|$;
3. If $u, v \in [-1, 1]$, $\Phi(u, v) \in [-1, 1]$ and it achieves an approximation accuracy

$$|\Phi(u, v) - u \cdot v| \leq 4^{-(m+1)}.$$

Proof Note that $|t| = \sigma(t) + \sigma(-t)$ for all $t \in \mathbb{R}$. Hence $\sigma\left(\frac{u+v}{2}\right) + \sigma\left(-\frac{u+v}{2}\right)$ gives $\left|\frac{u+v}{2}\right|$, which will be the input of a subnetwork f_m . The following flow chart demonstrates how the product gate Φ is generated:

$$\begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} \sigma\left(\frac{u+v}{2}\right) \\ \sigma\left(-\frac{u+v}{2}\right) \\ \sigma\left(\frac{u-v}{2}\right) \\ \sigma\left(-\frac{u-v}{2}\right) \end{bmatrix} \rightarrow \begin{bmatrix} \sigma\left(\left|\frac{u+v}{2}\right|\right) \\ \sigma\left(\left|\frac{u-v}{2}\right|\right) \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} G_m\left(\left|\frac{u+v}{2}\right|\right) \\ G_m\left(\left|\frac{u-v}{2}\right|\right) \end{bmatrix} \rightarrow \Phi(u, v).$$

The complexity of the network Φ follows from that of f_m .

If $u = 0$ or $v = 0$, we have $\left|\frac{u+v}{2}\right| = \left|\frac{u-v}{2}\right|$ and thereby $\Phi(u, v) = 0$. Observe that $0 \leq f_m(u) \leq u$ for $u \in [0, \infty)$, it follows that for $u, v \in \mathbb{R}$,

$$|\Phi(u, v)| = \left| f_m \left(\left| \frac{u+v}{2} \right| \right) - f_m \left(\left| \frac{u-v}{2} \right| \right) \right| \leq \left| \frac{u+v}{2} \right| + \left| \frac{u-v}{2} \right| \leq |u| + |v|.$$

It is shown in (Yarotsky, 2017, Proposition 2) that $0 \leq f_m(u) - u^2 \leq 4^{-(m+1)}$ and $0 \leq f_m(u) \leq 1$ for $u \in [0, 1]$. We have, for $u, v \in [-1, 1]$, $-1 \leq \Phi(u, v) \leq 1$ and

$$\begin{aligned} \Phi(u, v) - u \cdot v &= \left\{ f_m \left(\left| \frac{u+v}{2} \right| \right) - \left| \frac{u+v}{2} \right|^2 \right\} - \left\{ f_m \left(\left| \frac{u-v}{2} \right| \right) - \left| \frac{u-v}{2} \right|^2 \right\} \\ &\in [-4^{-(m+1)}, 4^{-(m+1)}]. \end{aligned}$$

This proves the proposition. ■

One of the advantages of our product gate is that for input on the domain $[0, 1]$, the output is also on $[0, 1]$. Such consistency of the domain and range $[0, 1]$ helps us to define the monomial gate in an elegant way in the next subsection.

6.2 Monomial Gate

Following the idea of the product gate introduced in the previous subsection, we construct a network to approximate monomial functions u^k on $[-1, 1]$, with $k \in \mathbb{N}$. This network is a monomial gate (i.e., EBTnet) defined in subsection 4.1 with specific parameter choices. Define the functions $\{h_k(u)\}_{k=1}^{2^\ell}$ on \mathbb{R} for $\ell \in \mathbb{Z}_+$ by

$$\{h_k(u)\}_{k=1}^2 = \{h_1(u) = u, h_2(u) = \Phi(u, u)\}, \quad (31)$$

and iteratively for $j = 1, \dots, \ell - 1$,

$$h_{2^j+k}(u) = \Phi(h_{2^j}(u), h_k(u)), \quad k = 1, \dots, 2^j. \quad (32)$$

Here, we focus on the input domain $[-1, 1]$.

Proposition 8. *Let $\ell \in \mathbb{Z}_+$. Consider the functions $\{h_k(u)\}_{k=1}^{2^\ell}$ defined in (31) and (32). For input $u \in [-1, 1]$, there exists a ReLU FNN that outputs the set of functions $\{h_k(u)\}_{k=1}^{2^\ell}$ such that for $j = 0, 1, \dots, \ell - 1$ and $k = 1, \dots, 2^j$,*

$$h_{2^j+k}(u) \in [-1, 1] \quad (33)$$

and

$$\left| h_{2^j+k}(u) - u^{2^j+k} \right| \leq \frac{2^{j+1} - 1}{4^{m+1}}. \quad (34)$$

This ReLU FNN belongs to $\mathcal{F}(\ell(m+1), \mathbf{p})$, where the width vector \mathbf{p} is given by, for $j = 0, \dots, \ell - 1$,

$$\begin{cases} p_{j(m+1)+1} = 5(2^j), \\ p_{j(m+1)+i} = 11(2^j), \quad \text{for } i = 2, \dots, m+1. \end{cases}$$

All parameters take value in $[-4, 4]$.

Proof The network described here is an EBTnet defined in subsection 4.1. It consists of ℓ subnetworks, each with depth $m+1$, forming an expansive binary tree structure. We prove our statements by induction on j .

The case $j = 0$ is obvious: the 1st subnetwork takes u a sinput and outputs $\{h_2(u), h_1(u)\}$. For $u \in [-1, 1]$, we have

$$h_2(u) = \Phi(u, u) = f_m(|u|) \in [0, 1]$$

and

$$|h_2(u) - u^2| = |f_m(|u|) - u^2| = f_m(|u|) - u^2 \leq 4^{-(m+1)} = \frac{2-1}{4^{(m+1)}}.$$

Now, assume the statements are true for the j -th subnetwork, where $0 \leq j \leq \ell - 1$. The $(j+1)$ -th subnetwork takes $[h_i(u)]_{i=1}^{2^j}$ as inputs, and outputs $h_{2^j+1}(u), \dots, h_{2^{j+1}}(u)$.

By the induction hypothesis, $h_{2^j-1+k}(u) \in [-1, 1]$ and $|h_{2^j-1+k}(u) - u^{2^j-1+k}| \leq \frac{2^j-1}{4^{m+1}}$ for $u \in [-1, 1]$ and $k = 1, \dots, 2^{j-1}$. We obtain, for $u \in [-1, 1]$ and $k = 1, \dots, 2^j$,

$$h_{2^j+k}(u) = \Phi(h_{2^j}(u), h_k(u)) \in [-1, 1],$$

and

$$\begin{aligned}
 & |h_{2^{j+k}}(u) - u^{2^j+k}| \\
 = & |h_{2^{j+k}}(u) - h_{2^j}(u) \cdot h_k(u) + h_{2^j}(u) \cdot h_k(u) - u^{2^j+k}| \\
 \leq & |\Phi(h_{2^j}(u), h_k(u)) - h_{2^j}(u) \cdot h_k(u)| + \left| \left(h_{2^j}(u) - u^{2^j} \right) h_k(u) + u^{2^j} (h_k(u) - u^k) \right| \\
 \leq & \frac{1}{4^{m+1}} + 2 \cdot \frac{2^j - 1}{4^{m+1}} = \frac{2^{j+1} - 1}{4^{m+1}}.
 \end{aligned}$$

This completes the induction procedure and the proof of Proposition 8. \blacksquare

6.3 Proof of Theorem 1

Recall the GMM discriminant function D defined in (3) by

$$D(x) = D^+(x) - D^-(x) = \sum_{j=1}^K \beta_j \exp(-u_j(x)),$$

where β_j are constant coefficients, and $u_j(x) = (x - \mu_j)^T (\Sigma_j)^{-1} (x - \mu_j) / 2$.

Recall that $r_{i,j}(x)$ is the i -th component of the vector $(\Sigma_j)^{-1/2} \frac{(x - \mu_j)}{\sqrt{2}}$ for $i = 1, \dots, d$ and $j = 1, \dots, K$. Also, recall that $\mu^* = \max_{1 \leq j \leq K} \|\mu_j\|$ and $\tilde{\sigma}$ is the smallest eigenvalue of all the K covariance matrices. We have for $j = 1, \dots, K$,

$$|r_{i,j}(x)| \leq \|(\Sigma_j)^{-1/2} (x - \mu_j)\| \leq \frac{1}{\sqrt{\tilde{\sigma}}} \|x - \mu_j\| \leq \frac{1}{\sqrt{\tilde{\sigma}}} (\|x\| + \mu^*). \quad (35)$$

Here, we present the proof of Theorem 1.

Proof of Theorem 1. Since the input $x \in \mathbb{R}^d$ is unbounded, $u_j(x)$ is unbounded for $j = 1, \dots, K$. We first apply the preprocessing subnetwork to truncate x . This part of the neural network is a fixed network structure (i.e., all network parameters are not free). The flowchart below showcases the preprocessing unit of our neural network:

$$\begin{array}{c}
 \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \rightarrow \begin{bmatrix} \Psi_b(x) \\ r_{1,1}(x) \\ \vdots \\ r_{d,1}(x) \\ \vdots \\ r_{1,K}(x) \\ \vdots \\ r_{d,K}(x) \end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix} \Phi(\Psi_b(x), r_{1,1}(x)) \\ \vdots \\ \Phi(\Psi_b(x), r_{d,K}(x)) \end{bmatrix}
 \end{array}$$

It follows from Proposition 7 that $|\Phi(\Psi_b(x), r_{i,j}(x))| \leq |\Psi_b(x)| + |r_{i,j}(x)|$.

Then we have, from Lemma 3, for $j = 1, \dots, K$,

$$\begin{cases} \Phi(\Psi_b(x), r_{i,j}(x)) = \Phi(0, r_{i,j}(x)) = 0, & \text{if } x \notin [-b-1, b+1]^d, \\ \Phi(\Psi_b(x), r_{i,j}(x)) = \Phi(1, r_{i,j}(x)), & \text{if } x \in [-b, b]^d. \end{cases} \quad (36)$$

Recall the function space \mathcal{H} defined earlier in (18). Every $\hat{r}_{i,j} \in \mathcal{H}$ has the form $\hat{r}_{i,j}(x) = \hat{W}_{i,j} \cdot x + \hat{b}_{i,j}$ with $\|\hat{W}_{i,j}\| \leq C_K^0, |\hat{b}_{i,j}| \leq C_K^1$. In Theorem 1, we choose $C_K^0 \geq \frac{\sqrt{d}}{\sqrt{\sigma}}$ and $C_K^1 \geq \frac{\mu^*}{\sqrt{\sigma}}$. We observe from (35) that $r_{i,j} \in \mathcal{H}$ due to the choices of C_K^0, C_K^1 here. Also, $|r_{i,j}(x)| \leq \frac{1}{\sqrt{\sigma}}(\sqrt{d}(b+1) + \mu^*)$ for $x \in [-b-1, b+1]^d$. But $R_2 \geq \sqrt{d}(b+1)C_K^0 + C_K^1$. Then for $j = 1, \dots, K$ and $i = 1, \dots, d$, $|r_{i,j}(x)/R_2| \leq 1$ and

$$\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \leq 1, \quad \forall x \in [-b-1, b+1]^d \quad (37)$$

and thus for all $x \in \mathbb{R}^d$ by (36). Observe that

$$\begin{aligned} D(x) &= \sum_{j=1}^K \beta_j \exp(-u_j(x)) = \sum_{j=1}^K \beta_j \exp \left(- \sum_{i=1}^d (r_{i,j}(x))^2 \right) \\ &= \sum_{j=1}^K \beta_j \exp \left(-(R_2)^2 \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right). \end{aligned}$$

By the Taylor expansion of the exponential function $\exp(-(R_2)^2 u)$ for $u \in \mathbb{R}$, we can further write D as

$$D(x) = \sum_{j=1}^K \beta_j \sum_{k=0}^{\infty} \frac{(-1)^k (R_2)^{2k} d^k}{k!} \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k.$$

Note that $\Phi(u, u) = f_m(|u|)$ by the definition of Φ in (30). Now define $\tilde{D} : \mathbb{R}^d \rightarrow \mathbb{R}$ by

$$\tilde{D}(x) = \sum_{j=1}^K \beta_j \left\{ 1 + \sum_{k=1}^{2^\ell} \frac{(-1)^k (R_2)^{2k} d^k}{k!} h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) \right\}. \quad (38)$$

For brevity, we wrote $f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right)$ instead of $\Phi \left(\Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right), \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right)$.

We would like to highlight that $\sigma_\lambda(\tilde{D}) \in \mathcal{H}$. In other words, \tilde{D} can be implemented by a ReLU FNN described in Remark 5 excluding the last scaling layer σ_λ .

For $k = 1, \dots, 2^\ell$ and $x \in \mathbb{R}^d$, we have

$$\begin{aligned} & h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) \\ &= h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) - \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right)^k \\ & \quad + \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right)^k. \end{aligned}$$

Applying Proposition 8, by (37) and $0 \leq f_m(u) \leq 1$ for $u \in [0, 1]$, we get for $k = 1, \dots, 2^\ell$ and $x \in \mathbb{R}^d$,

$$\left| h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) \right| \leq 1. \quad (39)$$

By (39), when $x \in [-b-1, b+1]^d$,

$$|\tilde{D}(x)| \leq \sum_{j=1}^K |\beta_j| \left\{ 1 + \sum_{k=1}^{2^\ell} \frac{((R_2)^2 d)^k}{k!} \right\} \leq \sum_{j=1}^K |\beta_j| \exp((R_2)^2 d). \quad (40)$$

When $x \notin [-b-1, b+1]^d$, by (36), we have $f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) = 0$ and thereby $|\tilde{D}(x)| \leq \left| \sum_{j=1}^K \beta_j \right| \leq \sum_{j=1}^K |\beta_j| (\exp((R_2)^2 d))$. This proves (9) in Theorem 1.

Now if $x \in [-b, b]^d$, we know $\Psi_b(x) = 1$ from Lemma 3. Applying the approximation error bound (12) for f_m , we get

$$\left| f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) - \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right| = \left| f_m \left(\left| \Phi \left(1, \frac{r_{i,j}(x)}{R_2} \right) \right| \right) - \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right| \leq 4^{-(m+1)}.$$

It follows from Proposition 8 that for $x \in [-b, b]^d$, $k = 1, \dots, 2^\ell$,

$$\begin{aligned} & \left| h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) - \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \\ & \leq \frac{2^\ell - 1}{4^{m+1}} + \left| \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right)^k - \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \\ & \leq \frac{2^\ell - 1}{4^{m+1}} + \frac{k}{d} \sum_{i=1}^d \left| f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) - \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right| \\ & \leq \frac{2^\ell - 1}{4^{m+1}} + k \left(4^{-(m+1)} \right) = \frac{2^\ell + k - 1}{4^{m+1}}. \end{aligned}$$

Here, we have used the Mean Value Theorem to bound $|u^k - v^k| \leq k|u - v|$ for $u, v \in [0, 1]$.

Then, for $x \in [-b, b]^d$, we have

$$\begin{aligned} & \left| \tilde{D}(x) - D(x) \right| \\ & \leq \left| \tilde{D}(x) - \sum_{j=1}^K \beta_j \sum_{k=0}^{2^\ell} \frac{(-1)^k (R_2)^{2k} d^k}{k!} \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \\ & \quad + \left| \sum_{j=1}^K \beta_j \sum_{k=2^\ell+1}^{\infty} \frac{(-1)^k (R_2)^{2k} d^k}{k!} \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \\ & \leq \sum_{j=1}^K |\beta_j| \sum_{k=0}^{2^\ell} \frac{(R_2)^{2k} d^k}{k!} \left| h_k \left(\frac{1}{d} \sum_{i=1}^d f_m \left(\left| \Phi \left(\Psi_b(x), \frac{r_{i,j}(x)}{R_2} \right) \right| \right) \right) - \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \end{aligned}$$

$$\begin{aligned}
 & + \left| \sum_{j=1}^K \beta_j \sum_{k=2^\ell+1}^{\infty} \frac{(-1)^k (R_2)^{2k} d^k}{k!} \left(\frac{1}{d} \sum_{i=1}^d \left(\frac{r_{i,j}(x)}{R_2} \right)^2 \right)^k \right| \\
 & \leq \sum_{j=1}^K |\beta_j| \left\{ \sum_{k=1}^{2^\ell} \frac{((R_2)^2 d)^k}{k!} \left(\frac{2^\ell + k - 1}{4^{m+1}} \right) \right\} + \sum_{j=1}^K |\beta_j| \sum_{k=2^\ell+1}^{\infty} \frac{((R_2)^2 d)^k}{k!} \\
 & \leq \sum_{j=1}^K |\beta_j| \left\{ \sum_{k=1}^{2^\ell} \frac{((R_2)^2 d)^k}{k!} \left(\frac{2^{\ell+1}}{4^{m+1}} \right) + \sum_{k=2^\ell+1}^{\infty} \frac{((R_2)^2 d)^k}{k!} \right\}.
 \end{aligned}$$

By Stirling's formula,

$$\sqrt{2\pi k} \left(\frac{k}{e} \right)^k \exp\left(\frac{1}{12k+1} \right) < k! < \sqrt{2\pi k} \left(\frac{k}{e} \right)^k \exp\left(\frac{1}{12k} \right),$$

we know that

$$\sum_{k=2^\ell+1}^{\infty} \frac{((R_2)^2 d)^k}{k!} < \sum_{k=2^\ell+1}^{\infty} \frac{((R_2)^2 d)^k}{\sqrt{2\pi k}} \left(\frac{e}{k} \right)^k \exp\left(-\frac{1}{12k+1} \right) \leq \sum_{k=2^\ell+1}^{\infty} \left(\frac{(R_2)^2 de}{k} \right)^k.$$

Since $\ell \geq \frac{2 \log((R_2)^2 de)}{\log 2}$, we have $((R_2)^2 de)^2 \leq 2^\ell + 1$. We then apply the bound $2^\ell + 1 \leq k$ for $k \geq 2^\ell + 1$ and find

$$\sum_{k=2^\ell+1}^{\infty} \left(\frac{(R_2)^2 de}{k} \right)^k \leq \sum_{k=2^\ell+1}^{\infty} \left(\frac{1}{\sqrt{k}} \right)^k \leq \left(\frac{1}{\sqrt{2^\ell}} \right)^{2^\ell} \sum_{k=2^\ell+1}^{\infty} \left(\frac{1}{\sqrt{2^\ell}} \right)^{k-2^\ell} \leq 2^{-\frac{\ell(2^\ell)}{2}},$$

where we have bounded $k \geq 2^\ell + 1$ by 2^ℓ from below and then $\sqrt{2^\ell}$ by 2 from below.

Therefore, for $x \in [-b, b]^d$, there holds

$$|\tilde{D}(x) - D(x)| \leq \left(\sum_{j=1}^K |\beta_j| \right) \exp((R_2)^2 d) \left(\frac{2^{\ell+1}}{4^{m+1}} + 2^{-\frac{\ell(2^\ell)}{2}} \right),$$

which verifies (8) in Theorem 1. The proof of Theorem 1 is complete. ■

7. Proof of Theorem 6: Generalization Analysis

In this section, we derive the high probability upper bound of excess generalization error $\varepsilon(f_z) - \varepsilon(f_c)$ for proving Theorem 6.

To start, we decompose $\varepsilon(f_z) - \varepsilon(f_c)$ into estimation error terms and an approximation error term (Subsection 7.1). Then, we bound the estimation error terms (in Subsection 7.2) that involve estimating the covering number of our hypothesis space \mathcal{H} . After that, we bound the approximation error term (Subsection 7.3). Lastly, by combining all the error estimates together, we are able to derive the proof of Theorem 6 (Subsection 7.3).

7.1 Error decomposition

We consider the following error decomposition. Similar error decompositions can be found in (Zhou and Huo, 2024; Huang et al., 2022).

Lemma 9 (Decomposition of $\varepsilon(f_z) - \varepsilon(f_c)$). *Let $f_{\mathcal{H}}$ be any functions in \mathcal{H} defined in Definition 4. There holds*

$$\varepsilon(f_z) - \varepsilon(f_c) \leq \{\varepsilon(f_z) - \varepsilon_z(f_z)\} + \{\varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}})\} + \{\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)\}. \quad (41)$$

Proof We express $\varepsilon(f_z) - \varepsilon(f_c)$ by inserting empirical risks as follows

$$\varepsilon(f_z) - \varepsilon(f_c) = \{\varepsilon(f_z) - \varepsilon_z(f_z)\} + \{\varepsilon_z(f_z) - \varepsilon_z(f_{\mathcal{H}})\} + \{\varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}})\} + \{\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)\}.$$

Both f_z and $f_{\mathcal{H}}$ lies on the hypothesis space \mathcal{H} . From the definition of f_z at (25), f_z minimizes the empirical risk $\varepsilon_z(f)$ over \mathcal{H} . Thus we have $\varepsilon_z(f_z) - \varepsilon_z(f_{\mathcal{H}}) \leq 0$. This yields the expression (41). \blacksquare

$\{\varepsilon(f_z) - \varepsilon_z(f_z)\}$ is the first estimation error (also known as the sample error) term, $\{\varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}})\}$ is the second estimation error term, whereas $\{\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)\}$ — which does not depend on the data — is the approximation error term induced by $f_{\mathcal{H}}$. To give an upper bound to the excess generalization error, we will proceed to bound these three error terms respectively.

7.2 Upper Bound of Estimation Errors

In this subsection, we derive an upper bound of the estimation errors $\varepsilon(f_z) - \varepsilon_z(f_z) + \varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}})$.

We first rewrite it by inserting $\varepsilon(f_c)$ and $\varepsilon_z(f_c)$:

$$\varepsilon(f_z) - \varepsilon_z(f_z) + \varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}}) = \varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c)) \quad (42)$$

$$+ \varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)). \quad (43)$$

In other words, to bound $\varepsilon(f_z) - \varepsilon_z(f_z) + (\varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}}))$, we should bound the R.H.S. of (42) and the R.H.S. of (43) respectively.

The following devotes to an upper bound of R.H.S. of (42).

7.2.1 UPPER BOUND OF $\varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c))$

The expression (42)

$$\varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c)) \leq \sup_{f \in \mathcal{H}} \{\varepsilon(f) - \varepsilon(f_c) - (\varepsilon_z(f) - \varepsilon_z(f_c))\}$$

can be estimated by the theory of uniform convergence. Since our domain $\mathcal{X} = \mathbb{R}^d$ is unbounded, deriving covering number estimates for our hypothesis space \mathcal{H} is difficult.

Let

$$B := \sum_{i \in \mathcal{T}^+} \frac{P^+ p_i}{\sqrt{(2\pi)^d |\Sigma_i|}} + \sum_{j \in \mathcal{T}^-} \frac{P^- p_j}{\sqrt{(2\pi)^d |\Sigma_j|}}. \quad (44)$$

Recall the Tsybakov-type noise condition we stated in Assumption 1. The following Lemma presents an upper bound of the second moment and thereby the variance of $\phi(yf(x)) - \phi(yf_c(x))$ for any function $f : \mathcal{X} \rightarrow [-1, 1]$ under the noise condition (5).

Lemma 10. *Let $0 \leq q \leq \infty$. Also let $\phi(t) = \max\{0, 1 - t\}$ to be the Hinge loss function. Consider the constant B given by (44). If noise condition (5) holds for some noise exponent q and constant $c_0 > 0$, then for every function $f : \mathcal{X} \rightarrow [-1, 1]$, there holds*

$$\mathbb{E} [\{\phi(yf(x)) - \phi(yf_c(x))\}^2] \leq 8 (c_0)^{\frac{1}{q+1}} (B(\varepsilon(f) - \varepsilon(f_c)))^{\frac{q}{q+1}}. \quad (45)$$

The proof of Lemma 10 is given in Appendix A.2.

For $\epsilon > 0$, denote by $\mathcal{N}(\epsilon, \mathcal{K}) = \mathcal{N}(\epsilon, \mathcal{K}, \|\cdot\|_\infty)$ the ϵ -covering number of a set of functions \mathcal{K} with respect to $\|\cdot\|_\infty := \sup_z |f(z)|$. More specifically, $\mathcal{N}(\epsilon, \mathcal{K})$ is the minimal $N \in \mathbb{N}$ such that there exists functions $\{f_1, \dots, f_N\} \in \mathcal{K}$ satisfying

$$\min_{1 \leq i \leq N} \|f - f_i\|_\infty \leq \epsilon, \quad \forall f \in \mathcal{K}. \quad (46)$$

Observe that the Hinge loss function $\phi(t) = \max\{0, 1 - t\}$ is Lipschitz continuous on \mathbb{R} with Lipschitz constant $M = 1$ because

$$|\phi(t_1) - \phi(t_2)| \leq |t_1 - t_2| \quad \forall t_1, t_2 \in \mathbb{R}. \quad (47)$$

Next, we construct a function set \mathcal{G} induced by functions in \mathcal{H} . The following Lemma tells us that the covering number of \mathcal{G} , denoted by $\mathcal{N}(\epsilon, \mathcal{G})$, is no greater than $\mathcal{N}(\epsilon, \mathcal{H})$. After that, we will proceed to estimate $\mathcal{N}(\epsilon, \mathcal{H})$ which will help us derive an upper bound of $\varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c))$.

Lemma 11. *For $\phi(t) = \max\{0, 1 - t\}$, define the set of functions on $Z = \mathcal{X} \times \mathcal{Y}$ given by*

$$\mathcal{G} := \{\phi(yf(x)) - \phi(yf_c(x)) : f \in \mathcal{H}\}, \quad (48)$$

where \mathcal{H} is the hypothesis space defined at (4). For $\epsilon > 0$, there holds

$$\mathcal{N}(\epsilon, \mathcal{G}) \leq \mathcal{N}(\epsilon, \mathcal{H}). \quad (49)$$

Proof For any $f_1, f_2 \in \mathcal{H}$ and $(x, y) \in Z$, it follows from (47)

$$\begin{aligned} |\{\phi(yf_1(x)) - \phi(yf_c(x))\} - \{\phi(yf_2(x)) - \phi(yf_c(x))\}| &= |\phi(yf_1(x)) - \phi(yf_2(x))| \\ &\leq |yf_1(x) - yf_2(x)| \\ &\leq \|f_1 - f_2\|_\infty, \end{aligned}$$

which implies $\mathcal{N}(\epsilon, \mathcal{G}) \leq \mathcal{N}(\epsilon, \mathcal{H})$. ■

For $\ell \in \mathbb{N}$, let $c^* = [-C_K, C_K]^{2^\ell}$ with C_K to be a positive constant given in Definition 4. The following Proposition devotes to an upper bound of the covering number of our hypothesis space \mathcal{H} . Its proof is relatively long and is given in Appendix C.

Proposition 12 (Covering number of the hypothesis space \mathcal{H}). *Let \mathcal{H} be defined by Definition 4 with $C_K^0 \geq \frac{\sqrt{d}}{\sqrt{\sigma}}$, $C_K^1 \geq \frac{\mu^*}{\sqrt{\sigma}}$, and $R_2 \geq \sqrt{d}(b+1)C_K^0 + C_K^1$. For $0 < \epsilon \leq 1$, $0 < \lambda \leq 1$, $b \geq 1$, $m, \ell \in \mathbb{N}$, there holds*

$$\log \mathcal{N}(\epsilon, \mathcal{H}) \leq C' m 2^\ell \log \left(\frac{bC_K}{\lambda \epsilon} \right) + 4\ell(2^\ell) \log(C_K) + C'' m^2 \ell(2^\ell), \quad (50)$$

where C', C'' are positive constants independent of m, ℓ, b, λ, C_K or ϵ .

As a simple corollary combining Proposition 12 and Lemma 11, for every $0 < \epsilon \leq 1$, there holds

$$\log \mathcal{N}(\epsilon, \mathcal{G}) \leq C' m 2^\ell \log \left(\frac{bC_K}{\lambda \epsilon} \right) + 4\ell(2^\ell) \log(C_K) + C'' m^2 \ell(2^\ell),$$

where C', C'' are positive constants independent of ℓ, m, b, λ, C_K or ϵ . We will next apply this covering number estimate to derive a high probability upper bound of the estimation error term $\varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c))$, which is the R.H.S. of (42). The proof of the following Lemma (Lemma 13) is given in Appendix A.3.

Lemma 13. *Let $q \geq 0, m, \ell \in \mathbb{N}, 0 < \lambda \leq 1$. Suppose noise condition (5) holds for some noise exponent q and constant $c_0 > 0$. For any $0 < \delta < 1, n \geq 3$, with probability $1 - \delta/2$, there holds*

$$\begin{aligned} & \varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c)) \\ & \leq C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log \left(\frac{2}{\delta} \right) + m 2^\ell \log \left(\frac{bC_K}{\lambda} \right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n} \right)^{\frac{q+1}{q+2}} + \frac{\varepsilon(f_z) - \varepsilon(f_c)}{2}, \end{aligned}$$

where $C_{q,B}$ is a constant depending on q, c_0, B only.

7.2.2 UPPER BOUND OF $\varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))$

Now we move on to estimate $\varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))$, which is the R.H.S. of (43).

Define a random variable $\xi(z) := \xi(x, y) = \phi(yf_{\mathcal{H}}(x)) - \phi(yf_c(x))$. We have

$$\begin{aligned} & \varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)) \\ & = \frac{1}{n} \sum_{i=1}^n \left\{ \phi(y_i f_{\mathcal{H}}(x_i)) - \phi(y_i f_c(x_i)) \right\} - \int_Z \phi(yf_{\mathcal{H}}(x)) - \phi(yf_c(x)) d\rho \\ & = \frac{1}{n} \sum_{i=1}^n \xi(z_i) - \mathbb{E}[\xi(z)] \end{aligned}$$

is a function of a single random variable ξ and thus can be estimated by Bernstein's inequality (see, e.g., (Györfi et al., 2002, Lemma A.2)). To apply Bernstein's inequality, we need first to establish an upper bound of the variance of ξ , denoted by $\sigma^2 = \text{Var}[\xi]$. To achieve so, we apply Lemma 10 to $f_{\mathcal{H}}$. With a bound of σ^2 in hand, we can obtain a high probability upper bound of $\varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))$.

Lemma 14. *Let $0 \leq q \leq \infty$. Suppose noise condition (5) holds for some q and constant $c_0 > 0$. For any $0 < \delta < 1$, with probability $1 - \delta/2$, there holds*

$$\varepsilon_z(f_{\mathcal{H}}) - \varepsilon_z(f_c) - (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)) \leq \frac{4}{n} \log \left(\frac{2}{\delta} \right) + 2(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))^{\frac{q}{q+1}}. \quad (51)$$

The proof of the above Lemma is given in Appendix A.4.

7.3 Upper Bound of the Approximation Error

Note that $\sigma_\lambda(\tilde{D}) \in \mathcal{H}$ where \tilde{D} (defined in (38)) is the approximation of the function D in Theorem 1. Recall that we use $f_{\mathcal{H}}$ denote any functions in \mathcal{H} . In this subsection, we derive a tight upper bound for the approximation error $\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)$ by taking $f_{\mathcal{H}} = \sigma_\lambda(\tilde{D})$.

Recall σ^* , the largest eigenvalue of all the covariance matrices Σ_j for $j = 1, \dots, K$. Also recall μ^* defined earlier in (7) as $\mu^* = \max_{1 \leq i \leq K} \|\mu_i\|$.

Lemma 15. *Let $b \geq \max\{2\mu^*, 1\}$, $0 < \lambda \leq 1$. Let $\tau > 0$ such that $\|\tilde{D} - D\|_{L^\infty[-b, b]^d} \leq \tau$. Assume noise condition (5) holds for some noise exponent q and constant $c_0 > 0$. There holds*

$$\left| \varepsilon\left(\sigma_\lambda\left(\tilde{D}\right)\right) - \varepsilon(f_c) \right| \leq 2 \left(\sum_{i=1}^K \frac{p_i}{\sqrt{|\Sigma_i|}} (4\sigma^*)^{\frac{d}{2}} \right) \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2c_0(\tau^q + (\tau + \lambda)^q). \quad (52)$$

Proof We know that

$$\begin{aligned} |\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)| &= \int_{\mathcal{X}} \int_{\mathcal{Y}} |yf_{\mathcal{H}}(x) - yf_c(x)| d\rho(y|x) d\rho_X \\ &= \int_{\mathcal{X}} |f_{\mathcal{H}}(x) - f_c(x)| |f_\rho(x)| d\rho_X \leq \int_{\mathcal{X}} |f_{\mathcal{H}}(x) - f_c(x)| d\rho_X. \end{aligned}$$

Since ρ_X has density function $D^+(x) + D^-(x)$, we know

$$|\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)| \leq \int_{\mathcal{X}} |f_{\mathcal{H}}(x) - f_c(x)| (D^+(x) + D^-(x)) dx. \quad (53)$$

To further bound the R.H.S. of (53), we consider the cases $\|x\| \geq b$ and $\|x\| < b$ separately, with $b \geq \max\{2\mu^*, 1\}$. For the case $\|x\| \geq b$, we have $\|x - \mu_j\| \geq \|x\| - \|\mu_j\| \geq \|x\| - \mu^* \geq b/2$. Hence, we have

$$(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \geq \frac{\|x - \mu_j\|^2}{\sigma^*} \geq \frac{b^2}{4\sigma^*}.$$

Observe that $|f_{\mathcal{H}}(x)| \leq 1$ and $|f_c(x)| \leq 1$. Making use of the decay of Gaussian density function \mathcal{N} , we get

$$\begin{aligned} & \int_{\{x \in \mathcal{X} : \|x\| \geq b\}} |f_{\mathcal{H}}(x) - f_c(x)| (D^+(x) + D^-(x)) dx \\ & \leq 2 \int_{\{x \in \mathcal{X} : \|x\| \geq b\}} (D^+(x) + D^-(x)) dx \\ & \leq 2 \int_{\{x \in \mathcal{X} : \|x\| \geq b\}} \left(\sum_{i=1}^K \frac{p_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma^*}\right) \right) dx. \end{aligned}$$

But

$$\exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma^*}\right) \leq \exp\left(-\frac{\|x - \mu_i\|^2}{4\sigma^*}\right) \exp\left(-\frac{b^2}{16\sigma^*}\right).$$

We use $\int_{\{x \in \mathcal{X} : \|x\| \geq b\}} e^{-\frac{\|x - \mu_j\|^2}{4\sigma^*}} dx < \sqrt{(2\pi)^d} (2\sqrt{\sigma^*})^d$ and obtain

$$\begin{aligned}
 & \int_{\{x \in \mathcal{X} : \|x\| \geq b\}} |f_{\mathcal{H}}(x) - f_c(x)| (D^+(x) + D^-(x)) dx \\
 \leq & 2 \exp\left(-\frac{b^2}{16\sigma^*}\right) \sum_{i=1}^K \frac{p_i}{\sqrt{|\Sigma_i|}} \int_{\{x \in \mathcal{X} : \|x\| \geq b\}} \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{\|x - \mu_i\|^2}{4\sigma^*}\right) dx \\
 \leq & 2 \exp\left(-\frac{b^2}{16\sigma^*}\right) \sum_{i=1}^K \frac{p_i}{\sqrt{|\Sigma_i|}} (4\sigma^*)^{d/2}.
 \end{aligned}$$

Take $f_{\mathcal{H}} = \sigma_{\lambda}(\tilde{D}) \in \mathcal{H}$. For $\|x\| < b$, we consider the cases $|D(x)| \leq \tau$ and $|D(x)| > \tau$ separately for some $\tau > 0$ satisfying $\|\tilde{D} - D\|_{L^\infty[-b, b]^d} \leq \tau$. For the case $\|x\| < b$ and $|D(x)| \leq \tau$, it follows from the noise condition (5) that

$$\begin{aligned}
 \int_{\{x \in \mathcal{X} : \|x\| < b, |D(x)| \leq \tau\}} |f_{\mathcal{H}}(x) - f_c(x)| d\rho_X & \leq 2 \int_{\{x \in \mathcal{X} : \|x\| < b, |D(x)| \leq \tau\}} d\rho_X \\
 & = 2 \cdot \mathbb{P}(\{x \in \mathcal{X} : \|x\| < b, |D(x)| \leq \tau\}) \\
 & \leq 2 \cdot \mathbb{P}(\{x \in \mathcal{X} : |D(x)| \leq \tau\}) \\
 & \leq 2c_0\tau^q.
 \end{aligned}$$

For $\|x\| < b$ and $|D(x)| > \tau$, since $\|\tilde{D} - D\|_{L^\infty[-b, b]^d} \leq \tau$, we have $\text{sgn}(\tilde{D}) = \text{sgn}(D) = f_c$. If $\sigma_{\lambda}(\tilde{D}(x)) = \{1, -1\}$, then $\sigma_{\lambda}(\tilde{D}(x))$ is exactly equal to $f_c(x)$ which implies $|\sigma_{\lambda}(\tilde{D}(x)) - f_c(x)| = 0$. Thus, we have

$$\begin{aligned}
 & \int_{\{x \in \mathcal{X} : \|x\| < b, |D(x)| > \tau\}} |\sigma_{\lambda}(\tilde{D})(x) - f_c(x)| d\rho_X \\
 = & \int_{\{x \in \mathcal{X} : \|x\| < b, |D(x)| > \tau, |\sigma_{\lambda}(\tilde{D}(x))| < 1\}} |\sigma_{\lambda}(\tilde{D})(x) - f_c(x)| d\rho_X \\
 \leq & 2 \cdot \mathbb{P}(x \in \mathcal{X} : \|x\| < b, |D(x)| > \tau, |\sigma_{\lambda}(\tilde{D})(x)| < 1) \\
 \leq & 2 \cdot \mathbb{P}(x \in \mathcal{X} : \|x\| < b, |\sigma_{\lambda}(\tilde{D})(x)| < 1) \\
 = & 2 \cdot \mathbb{P}(x \in \mathcal{X} : \|x\| < b, |\tilde{D}(x)| < \lambda) \\
 \leq & 2 \cdot \mathbb{P}(x \in \mathcal{X} : |D(x)| < \tau + \lambda) \\
 \leq & 2c_0(\tau + \lambda)^q.
 \end{aligned}$$

Here, we have used the equivalence between $|\sigma_{\lambda}(\tilde{D})(x)| < 1$ and $|\tilde{D}(x)| < \lambda$ and the condition $\|\tilde{D} - D\|_{L^\infty[-b, b]^d} \leq \tau$ for getting $|D(x)| < \tau + \lambda$ when $\|x\| < b$ and $|\tilde{D}(x)| < \lambda$. Combining the above estimates, we get the desired bound and prove the lemma. \blacksquare

7.4 Combining error bounds together

Now that we derived the upper bounds of the estimation errors and the approximation error, we can combine them together to prove Theorem 6.

Proof of Theorem 6. With probability at least $1 - \delta$, we have

$$\begin{aligned}
 & \varepsilon(f_z) - \varepsilon(f_c) \\
 & \leq \{\varepsilon(f_z) - \varepsilon_z(f_z)\} + \{\varepsilon_z(f_{\mathcal{H}}) - \varepsilon(f_{\mathcal{H}})\} + \{\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)\} \\
 & \leq C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log\left(\frac{2}{\delta}\right) + m 2^\ell \log\left(\frac{bC_K}{\lambda}\right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} \\
 & \quad + \frac{\varepsilon(f_z) - \varepsilon(f_c)}{2} + \frac{4}{n} \log\left(\frac{2}{\delta}\right) + 2(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))^{\frac{q}{q+1}} + \varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c).
 \end{aligned}$$

Take τ as in Lemma 15. This implies, with probability at least $1 - \delta$,

$$\begin{aligned}
 & \varepsilon(f_z) - \varepsilon(f_c) \\
 & \leq 2C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log\left(\frac{2}{\delta}\right) + m 2^\ell \log\left(\frac{bC_K}{\lambda}\right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} \\
 & \quad + \frac{8}{3n} \log\left(\frac{2}{\delta}\right) + \frac{8}{n} \log\left(\frac{2}{\delta}\right) + 4(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))^{\frac{q}{q+1}} + 2(\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)) \\
 & \leq 2C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log\left(\frac{2}{\delta}\right) + m 2^\ell \log\left(\frac{bC_K}{\lambda}\right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} \\
 & \quad + \frac{11}{n} \log\left(\frac{2}{\delta}\right) + \left(2 + 4(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}}\right) (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)) \\
 & = 2C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log\left(\frac{2}{\delta}\right) + m 2^\ell \log\left(\frac{bC_K}{\lambda}\right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} + \frac{11}{n} \log\left(\frac{2}{\delta}\right) \\
 & \quad + \left(4 + 8(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}}\right) \left\{ \left(\sum_{i=1}^K \frac{p_i}{\sqrt{\Sigma_i}} (4\sigma^*)^{\frac{d}{2}} \right) \exp\left(-\frac{b^2}{16\sigma^*}\right) + c_0 (\tau^q + (\tau + \lambda)^q) \right\}.
 \end{aligned}$$

Now take $c'_{q,B} = 4 + 8(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}}$, $c_1 = \sum_{i=1}^K \frac{p_i}{\sqrt{\Sigma_i}} (4\sigma^*)^{\frac{d}{2}}$. We have, with probability at least $1 - \delta$,

$$\begin{aligned}
 \varepsilon(f_z) - \varepsilon(f_c) & \leq 2C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell) \log(C_K) + \log\left(\frac{2}{\delta}\right) + m 2^\ell \log\left(\frac{bC_K}{\lambda}\right) \right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} \\
 & \quad + \frac{11}{n} \log\left(\frac{2}{\delta}\right) + c'_{q,B} \left(c_1 \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2c_0 (\tau + \lambda)^q \right).
 \end{aligned}$$

Recall $R_2 \geq \sqrt{d}(b+1)C_K^0 + C_K^1$ and $C_K \geq \sum_{j=1}^K |\beta_j| e^{(R_2)^2 d}$ are positive tunable parameters. From (8) of Theorem 1, we know that

$$\|\tilde{D} - D\|_{L^\infty[-b,b]^d} \leq C_K \left(\frac{2^{\ell+1}}{4^{m+1}} + 2^{-\frac{\ell(2^\ell)}{2}} \right) = C_K \left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right).$$

We take $R_2 = 2\sqrt{d}(C_K^0 + C_K^1)b$, $C_K = c_2 \exp(c_3 b^2)$ and $\tau = c_2 \exp(c_3 b^2) \left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right)$, where $c_2 \geq \sum_{j=1}^K |\beta_j|$, $c_3 \geq (2\sqrt{d}(C_K^0 + C_K^1))^2 d = 4d^2(C_K^0 + C_K^1)^2 \geq 4d^2 \frac{(\sqrt{d} + \mu^*)^2}{\bar{\sigma}}$. Then $\log(C_K) = \log(c_2) + c_3 b^2$. It follows that with probability at least $1 - \delta$,

$$\begin{aligned} & \varepsilon(f_z) - \varepsilon(f_c) \\ & \leq 2C_{q,B} \left(m^2 \ell 2^\ell + \ell(2^\ell)(\log(c_2) + c_3)b^2 + \log\left(\frac{2}{\bar{\delta}}\right) + m2^\ell \left(\log\left(\frac{b}{\lambda}\right) + (\log(c_2) + c_3)b^2 \right) \right)^{\frac{q+1}{q+2}} \\ & \quad \left(\frac{\log n}{n} \right)^{\frac{q+1}{q+2}} + \frac{11}{n} \log\left(\frac{2}{\bar{\delta}}\right) \\ & \quad + c'_{q,B} \left(c_1 \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2c_0 \left(c_2 \exp(c_3 b^2) \left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right) + \lambda \right)^q \right). \end{aligned}$$

We take $m = \ell(2^{\ell-1})$. We can easily verify that $\left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right) \leq 2 \cdot 2^{-\frac{\ell(2^\ell)}{2}} = 2^{1-\ell(2^{\ell-1})}$. We then choose b such that $\exp\left(-\frac{b^2}{16\sigma^*}\right) = \left(\exp(c_3 b^2) 2^{-\ell(2^{\ell-1})} \right)^q$, that is

$$-\frac{b^2}{16\sigma^*} = c_3 q b^2 - \ell q(2^{\ell-1})(\log 2) \iff b^2 = \frac{q(\log 2)\ell(2^{\ell-1})}{\frac{1}{16\sigma^*} + c_3 q}.$$

We thus take $b = \sqrt{c'_q \ell(2^{\ell-1})}$ with $c'_q = \frac{q(\log 2)}{\frac{1}{16\sigma^*} + c_3 q}$. Next, we take $\lambda = n^{-\frac{1}{q}}$. By our choices of b, m, λ and by applying $(a + b)^q \leq 2^q \max\{a^q, b^q\}$, we can see that

$$\begin{aligned} & c'_{q,B} \left(c_1 \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2c_0 \left(c_2 \exp(c_3 b^2) \left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right) + \lambda \right)^q \right) \\ & \leq c'_{q,B} \left(c_1 \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2^{q+1} c_0 \max \left\{ c_2^q \exp(c_3 q b^2) \left(\frac{2^\ell}{2(4^m)} + 2^{-\frac{\ell(2^\ell)}{2}} \right)^q, \lambda^q \right\} \right) \\ & \leq c'_{q,B} \left(c_1 \exp\left(-\frac{b^2}{16\sigma^*}\right) + 2^{q+1} c_0 c_2^q \exp(c_3 q b^2) \left(2^{1-\ell(2^{\ell-1})} \right)^q + 2^{q+1} c_0 \lambda^q \right) \\ & \leq c'_{q,B} \left((c_1 + 2^{q+1} c_0 c_2^q 2^q) \exp\left(-\frac{b^2}{16\sigma^*}\right) + \frac{2^{q+1} c_0}{n} \right). \end{aligned}$$

Finally, we take ℓ to be the smallest positive integer such that

$$\exp\left(-\frac{b^2}{16\sigma^*}\right) = \exp\left(-\frac{c'_q \ell(2^{\ell-1})}{16\sigma^*}\right) \leq \frac{1}{n}$$

which means $\ell 2^{\ell-1} \geq \frac{16\sigma^*}{c'_q} \log n$. With this choice, the above quantity is bounded by $\frac{c''_{q,B}}{n}$, where $c''_{q,B} = c'_{q,B}((c_1 + 2^{q+1} c_0 c_2^q 2^q) + 2^{q+1} c_0)$.

Observe from $\sigma^* \geq \bar{\sigma}$ that

$$\frac{16\sigma^*}{c'_q} = \frac{16\sigma^*}{q(\log 2)} \left(\frac{1}{16\sigma^*} + c_3 q \right) \geq \frac{16\sigma^* 4d^2 (\sqrt{d} + \mu^*)^2 q}{q(\log 2) \bar{\sigma}} \geq \frac{16\sigma^* 4d^3}{(\log 2) \bar{\sigma}} \geq 16(4d^3) \geq 64.$$

Thus, the restriction on ℓ implies $\ell \geq 4$. But ℓ is the smallest integer satisfying $(\ell - 1)2^{\ell-2} < \frac{16\sigma^*}{c'_q} \log n$ and thereby $\ell 2^\ell < 16(\ell - 1)2^{\ell-1} < \frac{16^2\sigma^*}{c'_q} \log n$. Thus, with our choices of ℓ, b, m, λ , we see that $m^2\ell 2^\ell = \ell^3 2^{3\ell-2} \leq \left(\frac{16^2\sigma^* \log n}{c'_q}\right)^3$ and

$$\ell(2^\ell)(\log(c_2) + c_3)b^2 \leq (\log(c_2) + c_3) \frac{(\log 2)q}{\frac{1}{16\sigma^*} + c_3q} (\ell(2^\ell))^2 \leq (\log(c_2) + c_3) \frac{(16^2\sigma^* \log n)^2}{c'_q}$$

and

$$\begin{aligned} & m2^\ell \left(\log\left(\frac{b}{\lambda}\right) + (\log(c_2) + c_3)b^2 \right) \\ & \leq m2^\ell ((\log(c_2) + c_3 + 1)b^2 - \log \lambda) \\ & \leq \ell 2^{2\ell-1} \left((\log(c_2) + c_3 + 1)c'_q \ell(2^{\ell-1}) + \frac{\log n}{q} \right) \\ & \leq (\log(c_2) + c_3 + 1) \frac{(16^2\sigma^* \log n)^3}{(c'_q)^2} + \frac{1}{q} \left(\frac{16^2\sigma^*}{c'_q} \right)^2 (\log n)^3. \end{aligned}$$

The proof is complete. ■

8. Conclusions

In this paper, we establish universal approximation theorems for GMM discriminant functions and general analytic functions using ReLU neural networks. Moreover, with Hinge loss and a Tsybakov-type noise condition, we obtain a fast convergence rate of the excess risk of order $O\left(n^{-\frac{q+1}{q+2}}(\log n)^4\right)$ for binary classification of GMM data by deep ReLU networks. Our convergence rate is better than the existing ones in the literature by leveraging the analyticity of the Gaussian function. Also, our convergence rate does not depend on the dimension d , demonstrating that neural networks can overcome the curse of dimensionality in classification.

To our best knowledge, our work is the first to study the generalization of classification with a GMM without restrictions on model parameters or the number of Gaussian components. This is also the first paper studying the statistical guarantees of neural network classifiers on an unbounded domain. Our findings shed light on the practical effectiveness of deep neural networks in classification problems, considering the universality of the Gaussian distribution across various data feature spaces such as speeches, images, and texts.

There has been an active line of research studying the theoretical aspects of multi-class classifications, see e.g., (Lei et al., 2015; Bos and Schmidt-Hieber, 2022). A future direction will be to extend our work to a multi-class classification problem. It would also be interesting to study the classification of GMM data with respect to a more general class of convex surrogate losses, e.g., the cross-entropy loss and logistic loss. These problems deserve further study in the future.

Acknowledgments

The authors are partially sponsored by NSF grants DMS 2015363 and CCF-1740776. Zhou is also partially supported by Georgia Tech Algorithm and Randomness Center (ARC-ACO) Fellowship.

Appendices

Appendix A. Proof of Supporting Lemmas

A.1 Proof of Lemma 3

Proof The proof is straightforward. Since $0 \leq T_b(u_i) \leq 1$, we have $\sum_{i=1}^d T_b(u_i) - (d-1) \leq 1$ then $0 \leq \Psi_b(u) \leq 1$ for $u \in \mathbb{R}^d$. If $u_j \in [-b, b]$ for all $j \in \{1, \dots, d\}$, then $\sum_{i=1}^d T_b(u_i) = d$ which gives $\Psi_b(u) = 1$. If $u_j \notin [-b-1, b+1]$ for at least one $j \in \{1, \dots, d\}$, then we have $\sum_{i=1}^d T_b(u_i) \leq d-1$ and thus $\Psi_b(u) = 0$. \blacksquare

A.2 Proof of Lemma 10

Proof Since $f(x) \in [-1, 1]$, $\phi(yf(x)) - \phi(yf_c(x)) = 1 - yf(x) - (1 - yf_c(x)) = y(f_c(x) - f(x))$. It follows that

$$\mathbb{E} [\{\phi(yf(x)) - \phi(yf_c(x))\}^2] = \mathbb{E}[y^2(f_c(x) - f(x))^2] = \int_{\mathcal{X}} (f_c(x) - f(x))^2 d\rho_X \quad (54)$$

and

$$\begin{aligned} \varepsilon(f) - \varepsilon(f_c) &= \int_{\mathcal{X}} \int_{\mathcal{Y}} y(f_c(x) - f(x)) d\rho(y|x) d\rho_X = \int_{\mathcal{X}} (f_c(x) - f(x)) f_\rho(x) d\rho_X \\ &= \int_{\mathcal{X}} |f_c(x) - f(x)| |f_\rho(x)| d\rho_X \end{aligned}$$

because, once again, $f_c = \text{sgn}(f_\rho) \in \{-1, 1\}$.

Let $t > 0$. Consider these two subsets of domain \mathcal{X} : $\mathcal{X}_t^+ = \{x \in \mathcal{X} : |D(x)| > t\}$ and $\mathcal{X}_t^- = \{x \in \mathcal{X} : |D(x)| \leq t\}$. On the set \mathcal{X}_t^+ , we apply $|f_c(x) - f(x)| \leq 2$ and get

$$|f_c(x) - f(x)|^2 \leq 2|f_c(x) - f(x)| \frac{|D(x)|}{t}.$$

On the set \mathcal{X}_t^- , we have $|f_c(x) - f(x)|^2 \leq 4$. Recall $B = \sum_{i \in \mathcal{T}^+} \frac{P^+ p_i}{\sqrt{(2\pi)^d |\Sigma_i|}} + \sum_{j \in \mathcal{T}^-} \frac{P^- p_j}{\sqrt{(2\pi)^d |\Sigma_j|}}$. From (2) and (3), we have $|D(x)| = |D^+(x) - D^-(x)| = |(D^+(x) + D^-(x)) f_\rho(x)| \leq B |f_\rho(x)|$. It follows from the noise condition (5) that

$$\begin{aligned} \mathbb{E} [\{\phi(yf(x)) - \phi(yf_c(x))\}^2] &= \int_{\mathcal{X}_t^-} (f_c(x) - f(x))^2 d\rho_X + \int_{\mathcal{X}_t^+} (f_c(x) - f(x))^2 d\rho_X \\ &\leq 4\mathbb{P}(\{x \in \mathcal{X} : |D(x)| \leq t\}) + \frac{2}{t} \int_{\mathcal{X}_t^+} |f_c(x) - f(x)| |D(x)| d\rho_X \\ &\leq 4\mathbb{P}(\{x \in \mathcal{X} : |D(x)| \leq t\}) + \frac{2B}{t} \int_{\mathcal{X}} |f_c(x) - f(x)| |f_\rho(x)| d\rho_X \end{aligned}$$

$$\leq 4c_0 t^q + \frac{2B}{t} (\varepsilon(f) - \varepsilon(f_c)).$$

Now set $t = \left(\frac{B(\varepsilon(f) - \varepsilon(f_c))}{2c_0} \right)^{1/(q+1)}$, we obtain our desired upper bound. \blacksquare

A.3 Proof of Lemma 13

Lemma 13 present a high probability upper bound of the estimation error term $\varepsilon(f_z) - \varepsilon(f_c) - (\varepsilon_z(f_z) - \varepsilon_z(f_c))$. To prove Lemma 13, we make use of the following concentration inequality found in (Cucker and Zhou, 2007, Lemma 10.20):

Lemma 16. *Let $0 \leq \gamma \leq 1$, $C_1, C_2 \geq 0$ and \mathcal{K} be a set of functions on Z such that for every $f \in \mathcal{K}$, $\mathbb{E}[f] \geq 0$, $\|f - \mathbb{E}[f]\|_\infty \leq C_2$ and $\mathbb{E}[f^2] \leq C_1(\mathbb{E}[f])^\gamma$. Then for all $\epsilon > 0$, with probability at least*

$$1 - \mathcal{N}(\epsilon, \mathcal{K}) \exp\left(-\frac{n\epsilon^{2-\gamma}}{2(C_1 + \frac{C_2}{3}\epsilon^{1-\gamma})}\right),$$

there holds

$$\sup_{f \in \mathcal{F}} \left\{ \frac{\mathbb{E}[f] - \frac{1}{n} \sum_{i=1}^n f(z_i)}{((\mathbb{E}[f])^\gamma + \epsilon^\gamma)^{1/2}} \right\} \leq 4\epsilon^{1-\frac{\gamma}{2}}. \quad (55)$$

Recall the function set $\mathcal{G} = \{\phi(yf(x)) - \phi(yf_c(x)) : f \in \mathcal{H}\}$ defined earlier in Lemma 11. To achieve the upper bound given in Lemma 13, we will apply Lemma 16 on the function set \mathcal{G} .

Proof of Lemma 13. To apply Lemma 16 to the function set $\mathcal{G} = \{\phi(yf(x)) - \phi(yf_c(x)) : f \in \mathcal{H}\}$, we need first to check that the three assumptions stated in Lemma 16 are satisfied. That is, for every function $g \in \mathcal{G}$, there holds

1. $\mathbb{E}[g] \geq 0$,
2. $\|f - \mathbb{E}[g]\|_\infty \leq C_2$,
3. $\mathbb{E}[g^2] \leq C_1(\mathbb{E}[g])^\gamma$,

for some constants $C_1, C_2 \geq 0$, and $0 \leq \gamma \leq 1$.

We have $\mathbb{E}[g] = \varepsilon(f) - \varepsilon(f_c) \geq 0$ because the Bayes classifier f_c minimizes the generalization error $\varepsilon(f)$ over all functions. Since $f \in \mathcal{H}$ is bounded in $[-1, 1]$, $|g(x)| = |\phi(yf(x)) - \phi(yf_c(x))| = |1 - yf(x) - (1 - yf_c(x))| = |y(f_c(x) - f(x))| \leq 2$. We know $|E[g]| \leq 2$. It follows that $\|g - \mathbb{E}[g]\|_\infty \leq 4$. So the second assumption is satisfied with $C_2 = 4$. Recall from Lemma 10 that $\mathbb{E}[g^2] \leq C_1(\mathbb{E}[g])^\gamma$ with $C_1 = 8(c_0)^{\frac{1}{q+1}}(B)^{\frac{q}{q+1}}$ and $\gamma = \frac{q}{q+1}$ with $c_0 > 0, q \geq 0$. So we proved that all three assumptions are satisfied for every $g \in \mathcal{G}$.

We can now apply Lemma 16 to \mathcal{G} . Lemma 16 tells us that for every $0 < \epsilon \leq 1$, with probability at least

$$1 - \mathcal{N}(\epsilon, \mathcal{G}) \exp\left(-\frac{n\epsilon^{2-\gamma}}{2(C_1 + \frac{C_2}{3}\epsilon^{1-\gamma})}\right)$$

$$\begin{aligned}
 &= 1 - \mathcal{N}(\epsilon, \mathcal{G}) \exp\left(-\frac{n\epsilon^{2-\gamma}}{16(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} + \frac{4}{3}\epsilon^{1-\gamma}}\right) \\
 &\geq 1 - \exp\left(C'm2^\ell \log\left(\frac{bC_K}{\lambda\epsilon}\right) + 4\ell(2^\ell) \log(C_K) + C''m^2\ell(2^\ell) - \frac{n\epsilon^{2-\gamma}}{16(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} + \frac{4}{3}}\right),
 \end{aligned}$$

there holds

$$\sup_{g \in \mathcal{G}} \left\{ \frac{\mathbb{E}[g] - \frac{1}{n} \sum_{i=1}^n g(z_i)}{((\mathbb{E}[g])^\gamma + \epsilon^\gamma)^{1/2}} \right\} \leq 4\epsilon^{1-\frac{\gamma}{2}},$$

which implies

$$\mathbb{E}[g] - \frac{1}{n} \sum_{i=1}^n g(z_i) \leq 4\epsilon^{1-\frac{\gamma}{2}} ((\mathbb{E}[g])^\gamma + \epsilon^\gamma)^{1/2}, \quad \forall g \in \mathcal{G}$$

and thereby

$$\varepsilon(f) - \varepsilon(f_c) - (\varepsilon_z(f) - \varepsilon_z(f_c)) \leq 4\epsilon^{1-\frac{\gamma}{2}} ((\varepsilon(f) - \varepsilon(f_c))^\gamma + \epsilon^\gamma)^{1/2}, \quad \forall f \in \mathcal{H}.$$

For brevity, we choose not to plug in $\gamma = \frac{q}{q+1} \in [0, 1]$ for now.

Setting the above confidence bound to be $1 - \delta/2$, then the solution ϵ satisfies

$$C'm2^\ell \log\left(\frac{1}{\epsilon}\right) - \frac{n\epsilon^{2-\gamma}}{16(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} + \frac{4}{3}} = \log\left(\frac{\delta}{2}\right) - C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) - 4\ell(2^\ell) \log(C_K) - C''m^2\ell(2^\ell).$$

Let $\hat{\epsilon} = \epsilon^{2-\gamma}$ and $c_{q,B} = 16(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} + \frac{4}{3} > 0$. We can see that $c_{q,B}$ is a constant depending on q, c_0, B only. We then have

$$\frac{C'm2^\ell}{2-\gamma} \log\left(\frac{1}{\hat{\epsilon}}\right) - \frac{n\hat{\epsilon}}{c_{q,B}} = \log\left(\frac{\delta}{2}\right) - C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) - 4\ell(2^\ell) \log(C_K) - C''m^2\ell(2^\ell).$$

We solve for $\hat{\epsilon}$ with the above equation. Note that the function $T : (0, 1] \rightarrow \mathbb{R}$ defined by $T(u) = \frac{C'm2^\ell}{2-\gamma} \log\left(\frac{1}{u}\right) - \frac{nu}{c_{q,B}}$ is decreasing. Take

$$A = \left(\frac{C'm2^\ell}{2-\gamma} + \log\left(\frac{2}{\delta}\right) + C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) + 4\ell(2^\ell) \log(C_K) + C''m^2\ell(2^\ell) \right) c_{q,B}.$$

For $n \geq 3$ (which implies $\log n > 1$), there holds $\frac{A(\log n)}{n} \geq \frac{1}{n}$. It follows that

$$\begin{aligned}
 &T\left(\frac{A(\log n)}{n}\right) \\
 &\leq \frac{C'm2^\ell}{2-\gamma} \log n - (\log n) \left(\frac{C'm2^\ell}{2-\gamma} + \log\left(\frac{2}{\delta}\right) + C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) + 4\ell(2^\ell) \log(C_K) + C''m^2\ell(2^\ell) \right) \\
 &\leq -(\log n) \left(\log\left(\frac{2}{\delta}\right) + C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) + 4\ell(2^\ell) \log(C_K) + C''m^2\ell(2^\ell) \right) \\
 &\leq \log\left(\frac{\delta}{2}\right) - C'm2^\ell \log\left(\frac{bC_K}{\lambda}\right) - 4\ell(2^\ell) \log(C_K) - C''m^2\ell(2^\ell).
 \end{aligned}$$

Since T is decreasing, we have $\hat{\epsilon} \leq \frac{A(\log n)}{n}$, which implies $\epsilon \leq \left(\frac{A(\log n)}{n}\right)^{1/(2-\gamma)}$.

Now, take $f = f_z$ and the above estimate of ϵ , we obtain

$$\epsilon(f_z) - \epsilon(f_c) - (\epsilon_z(f_z) - \epsilon_z(f_c)) \leq 4 \left(\frac{A(\log n)}{n} (\epsilon(f_z) - \epsilon(f_c))^\gamma\right)^{1/2} + 4 \left(\frac{A(\log n)}{n}\right)^{1/(2-\gamma)}$$

with probability at least $1 - \delta/2$. Plug in $\gamma = \frac{q}{q+1}$, we get with probability at least $1 - \delta/2$,

$$\epsilon(f_z) - \epsilon(f_c) - (\epsilon_z(f_z) - \epsilon_z(f_c)) \leq 4 \left(\frac{A(\log n)}{n}\right)^{1/2} (\epsilon(f_z) - \epsilon(f_c))^{\frac{q}{2(q+1)}} + 4 \left(\frac{A(\log n)}{n}\right)^{\frac{q+1}{q+2}}.$$

We then apply Young's Inequality for products (Young, 1912):

$$a \cdot b \leq \frac{a^p}{p} + \frac{b^{p^*}}{p^*} \quad \text{with } a \geq 0, b \geq 0, p > 1, p^* > 1 \text{ and } \frac{1}{p} + \frac{1}{p^*} = 1$$

to further upper bound $4 \left(\frac{A(\log n)}{n}\right)^{1/2} (\epsilon(f_z) - \epsilon(f_c))^{\frac{q}{2(q+1)}}$. We get

$$\begin{aligned} 4 \left(\frac{A(\log n)}{n}\right)^{1/2} (\epsilon(f_z) - \epsilon(f_c))^{\frac{q}{2(q+1)}} &\leq \frac{\left(4 \left(\frac{A(\log n)}{n}\right)^{1/2}\right)^{\frac{2(q+1)}{q+2}}}{\frac{2(q+1)}{q+2}} + \frac{\epsilon(f_z) - \epsilon(f_c)}{\frac{2(q+1)}{q}} \\ &\leq 2^{\frac{3q+2}{q+2}} \frac{q+2}{q+1} \left(\frac{A(\log n)}{n}\right)^{\frac{q+1}{q+2}} + \frac{\epsilon(f_z) - \epsilon(f_c)}{2}. \end{aligned} \quad (56)$$

Notice that $A \leq c_{q,B} \left(\frac{C'}{2-(q/(q+1))} + C''\right) \left(m^2 \ell(2^\ell) + \log\left(\frac{2}{\delta}\right) + 4\ell(2^\ell) \log(C_K) + m2^\ell \log\left(\frac{bC_K}{\lambda}\right)\right)$. Plug in this upper bound of A and (56), we finally get, with probability at least $1 - \delta/2$,

$$\begin{aligned} &\epsilon(f_z) - \epsilon(f_c) - (\epsilon_z(f_z) - \epsilon_z(f_c)) \\ &\leq \left(2^{\frac{3q+2}{q+2}} \frac{q+2}{q+1} + 4\right) \left(\frac{A(\log n)}{n}\right)^{\frac{q+1}{q+2}} + \frac{\epsilon(f_z) - \epsilon(f_c)}{2} \\ &\leq C_{q,B} \left(m^2 \ell 2^\ell + \log\left(\frac{2}{\delta}\right) + \ell(2^\ell) \log(C_K) + m2^\ell \log\left(\frac{bC_K}{\lambda}\right)\right)^{\frac{q+1}{q+2}} \left(\frac{\log n}{n}\right)^{\frac{q+1}{q+2}} + \frac{\epsilon(f_z) - \epsilon(f_c)}{2}, \end{aligned}$$

where $C_{q,B}$ is a positive constant depending only on q, c_0, B . The proof is complete. \blacksquare

A.4 Proof of Lemma 14

Proof Notice that $|\xi(z)| = |\phi(yf_{\mathcal{H}}(x)) - \phi(yf_c(x))| \leq 2$. It follows from Lemma 10 that

$$\sigma^2 = \text{Var}[\xi(z)] \leq \text{E}[\xi(z)^2] \leq 8(c_0)^{\frac{1}{q+1}} (B(\epsilon(f_{\mathcal{H}}) - \epsilon(f_c)))^{\frac{q}{q+1}}.$$

By the one-sided Bernstein's inequality, for any $\eta > 0$, there holds, with probability at least $1 - \exp\left(-\frac{n\eta^2}{2(\sigma^2 + 2\eta/3)}\right)$,

$$\epsilon_z(f_{\mathcal{H}}) - \epsilon_z(f_c) - (\epsilon(f_{\mathcal{H}}) - \epsilon(f_c)) \leq \eta. \quad (57)$$

Setting this confidence bound to be $1 - \delta/2$, we get a quadratic equation $\frac{n\eta^2}{2(\sigma^2+2\eta/3)} = \log(2/\delta)$ for η . We solve this equation and get a positive solution η^* given by

$$\begin{aligned} \eta^* &= \frac{\frac{4}{3} \log\left(\frac{2}{\delta}\right) + \sqrt{\frac{16}{9} \left(\log\left(\frac{2}{\delta}\right)\right)^2 + 8n\sigma^2 \log\left(\frac{2}{\delta}\right)}}{2n} \\ &\leq \frac{2}{3n} \log\left(\frac{2}{\delta}\right) + \frac{2}{3n} \log\left(\frac{2}{\delta}\right) + \frac{\sqrt{2\sigma^2 \log\left(\frac{2}{\delta}\right)}}{\sqrt{n}} \\ &\leq \frac{4}{3n} \log\left(\frac{2}{\delta}\right) + 4\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{n}} (c_0)^{\frac{1}{2(q+1)}} (B(\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)))^{\frac{q}{2(q+1)}} \\ &\leq \frac{4}{3n} \log\left(\frac{2}{\delta}\right) + 2 \left(\frac{\log\left(\frac{2}{\delta}\right)}{n} + (c_0)^{\frac{1}{q+1}} (B(\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c)))^{\frac{q}{q+1}} \right) \\ &\leq \frac{4}{n} \log\left(\frac{2}{\delta}\right) + 2(c_0)^{\frac{1}{q+1}} B^{\frac{q}{q+1}} (\varepsilon(f_{\mathcal{H}}) - \varepsilon(f_c))^{\frac{q}{q+1}}. \end{aligned}$$

Here, we have used $2\sqrt{ab} \leq a + b$ in the third inequality. ■

Appendix B. Proof of Theorem 2: Approximation of Analytic Functions

In this part, we present the proof of Theorem 2. We apply our monomial gate to approximate univariate analytic functions.

Proof of Theorem 2. The convergence of the Taylor series of t at R_1 implies

$$\widetilde{M} := \sup_{i \in \mathbb{Z}_+} \left| \frac{t^{(i)}(0)}{i!} (R_1)^i \right| < \infty.$$

It follows that, for $\ell \in \mathbb{N}$ and $u \in [-1, 1]$,

$$\begin{aligned} \left| t(u) - t(0) - \sum_{i=1}^{2^\ell} \frac{t^{(i)}(0)}{i!} (R_1)^i \left(\frac{u}{R_1}\right)^i \right| &= \left| \sum_{i=2^{\ell+1}}^{\infty} \frac{t^{(i)}(0)}{i!} (R_1)^i \left(\frac{u}{R_1}\right)^i \right| \leq \widetilde{M} \sum_{i=2^{\ell+1}}^{\infty} \left| \frac{u}{R_1} \right|^i \\ &\leq \frac{\widetilde{M}}{R_1 - 1} R_1^{-2^\ell}. \end{aligned}$$

To further approximate $t(u)$, consider the monomial gate defined in Section 2 with input $u \in [-1, 1]$. We can construct a deep ReLU network of depth $(m+1) \cdot \ell$ which outputs the function

$$F(u) = t(0) + \sum_{i=1}^{2^\ell} \frac{t^{(i)}(0)}{i!} (R_1)^i \frac{h_i(u)}{(R_1)^i},$$

where $\{h_i\}_{i=1}^{2^\ell}$ are outputs of the monomial gate defined in Proposition 8. Recall from Proposition 8 that $\{h_i(u)\}_{i=1}^{2^\ell}$ approximate $\{u^i\}_{i=1}^{2^\ell}$ to an accuracy

$$\left| h_{2^j+k}(u) - u^{2^j+k} \right| \leq \frac{2^{j+1} - 1}{4^{m+1}}, \quad \text{for } u \in [-1, 1], \quad j = 0, \dots, \ell - 1, \quad k = 1, \dots, 2^j.$$

We have

$$\begin{aligned}
 |F(u) - t(u)| &\leq \left| \sum_{i=1}^{2^\ell} \frac{t^{(i)}(0)}{i!} (R_1)^i \frac{h_i(u) - u^i}{(R_1)^i} \right| + \left| \sum_{i=2^{\ell+1}}^{\infty} \frac{t^{(i)}(0)}{i!} (R_1)^i \left(\frac{u}{R_1} \right)^i \right| \\
 &\leq \widetilde{M} \sum_{i=1}^{2^\ell} \frac{|h_i(u) - u^i|}{(R_1)^i} + \frac{\widetilde{M}}{R_1 - 1} R_1^{-2^\ell} \\
 &\leq \widetilde{M} \sum_{j=0}^{\ell-1} \sum_{k=1}^{2^j} \frac{|h_{2^j+k}(u) - u^{2^j+k}|}{(R_1)^{2^j+k}} + \frac{\widetilde{M}}{R_1 - 1} R_1^{-2^\ell} \\
 &\leq \widetilde{M} \left(\sum_{j=0}^{\ell-1} \sum_{k=1}^{2^j} \frac{2^{j+1}-1}{4^{m+1}} + \frac{1}{R_1^{2^\ell} (R_1 - 1)} \right) \\
 &\leq \widetilde{M} \left(\frac{1}{4^{m+1}} \sum_{j=0}^{\ell-1} \frac{2^{j+1}}{(R_1)^{2^j}} \sum_{k=1}^{2^j} \frac{1}{(R_1)^k} + \frac{1}{R_1^{2^\ell} (R_1 - 1)} \right) \\
 &\leq \frac{\widetilde{M}}{(R_1 - 1)} \left(\frac{1}{4^{m+1}} \sum_{j=0}^{\ell-1} \frac{2^{j+1}}{(R_1)^{2^j}} + \frac{1}{R_1^{2^\ell}} \right).
 \end{aligned}$$

Observe that if $j \geq 5$, we have $2^j \geq j^2$ which implies $(R_1)^{2^j} \geq (R_1)^{j^2}$. Hence, for $j \geq \max\{5, \frac{\log 4}{\log R_1}\}$, $(R_1)^{2^j} \geq (R_1^j)^j \geq 4^j$. Then,

$$\begin{aligned}
 |F(u) - t(u)| &\leq \frac{\widetilde{M}}{(R_1 - 1)} \left(\frac{1}{4^{m+1}} \left(\sum_{j=0}^{\max\{4, \lfloor \log 4 / \log R_1 \rfloor\}} 2^{j+1} + 2 \sum_{j=\max\{4, \lfloor \log 4 / \log R_1 \rfloor\}+1}^{\ell-1} \left(\frac{1}{2} \right)^j \right) + \frac{1}{R_1^{2^\ell}} \right) \\
 &\leq \frac{\widetilde{M}}{(R_1 - 1)} \left(\frac{1}{4^{m+1}} \left(2^{4+\lfloor \log 4 / \log R_1 \rfloor+2} + 2 \right) + \frac{1}{R_1^{2^\ell}} \right) \\
 &\leq \frac{\widetilde{M}}{(R_1 - 1)} \left(\frac{2^{7+\lfloor \log 4 / \log R_1 \rfloor}}{4^{m+1}} + \frac{1}{R_1^{2^\ell}} \right).
 \end{aligned}$$

$$\text{Take } C = \frac{\widetilde{M}(2^{7+\lfloor \log 4 / \log R_1 \rfloor})}{(R_1 - 1)} = \frac{2^{7+\lfloor \log 4 / \log R_1 \rfloor}}{(R_1 - 1)} \sup_{i \in \mathbb{Z}_+} \left| \frac{t^{(i)}(0)}{i!} (R_1)^i \right|. \quad \blacksquare$$

Appendix C. Proof of Proposition 12: Covering Number of the Hypothesis Space \mathcal{H}

In this part, we derive the upper bound of the covering number of the hypothesis space \mathcal{H} to prove Proposition 12.

We will first give uniform bounds of squaring gate (Subsection C.1), product gate (Subsection C.2), and monomial gate (Subsection C.3). Finally, we apply these uniform bounds to prove Proposition 12 (Subsection C.4).

We begin by giving some notations. For any vector $\nu \in \mathbb{R}^n$, define $\|\nu\|_\infty := \max_{1 \leq i \leq n} |\nu_i|$ and $\|\nu\|_1 := \sum_{i=1}^n |\nu_i|$. For any matrix $A \in \mathbb{R}^{m \times n}$, define $\|A\|_\infty := \max_{1 \leq i \leq m} \sum_{j=1}^n |A_{i,j}|$, which is the maximum absolute row sum of the matrix and equals the operator norm of $A : (\mathbb{R}^n, \|\cdot\|_\infty) \rightarrow (\mathbb{R}^m, \|\cdot\|_\infty)$. For a function $f : [-T, T] \rightarrow \mathbb{R}$, define $\|f\|_{L^\infty[-T, T]} := \sup_{x \in [-T, T]} |f(x)|$. Denote further

$$|f|_{Lip1} := \sup_{x, y \in [-T, T], x \neq y} \frac{|f(x) - f(y)|}{|x - y|}$$

as the Lipschitz-1 seminorm of a function on $[-T, T]$.

C.1 Uniform Bound of Squaring Gate \widehat{f}_m

For $T > 0$ and $x \in [-T, T]$, let $H^{(0)} = x$ and define iteratively for $j = 1, \dots, m$,

$$H^{(j)}(x) := H_{\mathbf{W}, \mathbf{b}}^{(j)}(x) = \sigma(W^{(j)}H^{(j-1)}(x) + b^{(j)}) \quad (58)$$

with $W^{(1)} \in [-4, 4]^{5 \times 1}$, $W^{(j)} \in [-4, 4]^{5 \times 5}$ for $j \geq 2$ and $b^{(j)} \in [-4, 4]^5$. For each $H^{(j)} = \{(H^{(j)})_1, \dots, (H^{(j)})_5\}$, define

$$\|H^{(j)}\|_{L^\infty[-T, T]} := \max_{1 \leq i \leq 5} \|(H^{(j)})_i\|_{L^\infty[-T, T]}$$

and

$$|H^{(j)}|_{Lip1} := \max_{1 \leq i \leq 5} \left| (H^{(j)})_i \right|_{Lip1}.$$

Lemma 17. *For each $j = 1, \dots, m$ and $H^{(j)}$ defined by (58) with $T > 0$, there holds*

$$\|H^{(j)}\|_{L^\infty[-T, T]} \leq 20^j T + 4 \left(\frac{20^j - 1}{20 - 1} \right), \quad (59)$$

and

$$|H^{(j)}|_{Lip1} \leq 20^j. \quad (60)$$

Proof From $|\sigma(u)| \leq |u|$, we have, for $i = 1, \dots, 5$, $j = 1, \dots, m$,

$$\begin{aligned} \left| \left(H^{(j)}(x) \right)_i \right| &= \left| \sigma(W^{(j)}H^{(j-1)}(x) + b^{(j)})_i \right| \leq \left| (W^{(j)}H^{(j-1)}(x))_i + (b^{(j)})_i \right| \\ &\leq \|W^{(j)}\|_\infty \|H^{(j-1)}\|_{L^\infty[-T, T]} + \|b^{(j)}\|_\infty \\ &\leq 20 \|H^{(j-1)}\|_{L^\infty[-T, T]} + 4. \end{aligned}$$

This leads us to a recurrence relationship:

$$\begin{aligned} \|H^{(j)}\|_{L^\infty[-T, T]} &\leq 20 \|H^{(j-1)}\|_{L^\infty[-T, T]} + 4 \leq \dots \leq 20^j \|H^{(0)}\|_{L^\infty[-T, T]} + 4 \left(\frac{20^j - 1}{20 - 1} \right) \\ &= 20^j T + 4 \left(\frac{20^j - 1}{20 - 1} \right), \quad \forall j = 1, \dots, m. \end{aligned}$$

This completes the proof of (59). Applying $|\sigma(u) - \sigma(v)| \leq |u - v|$, we have for all $u \neq v \in [-1, 1]$,

$$\begin{aligned} \|H^{(j)}(u) - H^{(j)}(v)\|_\infty &\leq \|W^{(j)}H^{(j-1)}(u) + b^{(j)} - W^{(j)}H^{(j-1)}(v) - b^{(j)}\|_\infty \\ &\leq \|W^{(j)}\|_\infty \|H^{(j-1)}(u) - H^{(j-1)}(v)\|_\infty \\ &\leq 20 \|H^{(j-1)}(u) - H^{(j-1)}(v)\|_\infty \\ &\leq \dots \leq 20^j \|H^{(0)}(u) - H^{(0)}(v)\|_\infty = 20^j |u - v|. \end{aligned}$$

Hence, we obtain

$$|H^{(j)}|_{Lip1} = \sup_{u, v \in [-T, T], u \neq v} \frac{\|H^{(j)}(u) - H^{(j)}(v)\|_\infty}{|u - v|} \leq 20^j.$$

This completes the proof of (60). ■

Recall that the squaring gate $\widehat{f}_m =: \widehat{f}_{m, \theta}$ is a ReLU FNN $\in \mathcal{F}(m, (5, 5, \dots, 5))$ defined in Subsection 4.1. It has all the trainable parameters $\theta = \{\mathbf{W}, \mathbf{b}, a\}$ taking values on $[-4, 4]$. For $T > 0$, the next lemma devotes to a uniform bound of $\|\widehat{f}_{m, \theta} - \widehat{f}_{m, \tilde{\theta}}\|_{L^\infty[-T, T]}$ where $\theta = \{\mathbf{W}, \mathbf{b}, a\}$ and $\tilde{\theta} = \{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}, \widetilde{a}\}$ represent two different collections of network parameters. Denote by

$$\mathbf{W}^* := \left\{ \mathbf{W} = W^{(1)} \in [-4, 4]^{5 \times 1}, (W^{(j)})_{j=2}^m \in \mathbb{R}^{5 \times 5} : |W_{i,k}^{(j)}| \leq 4 \right\}$$

and

$$\mathbf{b}^* := \left\{ \mathbf{b} = (b^{(j)})_{j=1}^m \in \mathbb{R}^5 : |b_i^{(j)}| \leq 4 \right\}$$

and

$$a^* := \{a \in \mathbb{R}^5 : |a_i| \leq 4\}.$$

Definition 18 (η -net). *For arbitrary $\eta > 0$, let \mathbf{W}_η^* , \mathbf{b}_η^* , a_η^* be η -nets of \mathbf{W}^* , \mathbf{b}^* , a^* , respectively, meaning that, for each $\mathbf{W} \in \mathbf{W}^*$, $\mathbf{b} \in \mathbf{b}^*$ and $a \in a^*$, there exist $\widetilde{\mathbf{W}} \in \mathbf{W}_\eta^*$, $\widetilde{\mathbf{b}} \in \mathbf{b}_\eta^*$, $\widetilde{a} \in a_\eta^*$ such that*

$$\|\mathbf{W} - \widetilde{\mathbf{W}}\|_{\infty, \infty} \leq \eta, \quad \|\mathbf{b} - \widetilde{\mathbf{b}}\|_\infty \leq \eta, \quad \|a - \widetilde{a}\|_1 \leq \eta. \quad (61)$$

Here, $\|\mathbf{W}\|_{\infty, \infty} := \max_{1 \leq j \leq m} \|W^{(j)}\|_\infty$ and $\|\mathbf{b}\|_\infty := \max_{1 \leq j \leq m} \|b^{(j)}\|_\infty$.

Lemma 19. *Let $\eta > 0$, $T > 0$, and $m \in \mathbb{N}$. Let \mathbf{W}_η^* , \mathbf{b}_η^* , a_η^* be η -nets of \mathbf{W}^* , \mathbf{b}^* , a^* defined above in Definition 18. With the network input $x \in [-T, T]$, there hold*

$$|\widehat{f}_{m, \theta}|_{Lip1} \leq 20^{m+1} \quad (62)$$

and

$$\|\widehat{f}_{m, \theta} - \widehat{f}_{m, \tilde{\theta}}\|_{L^\infty[-T, T]} \leq (T + 1)(m + 1)20^m \eta. \quad (63)$$

Proof From Inequality (60) in Lemma 17, we know

$$\|H^{(j)}(u) - H^{(j)}(v)\|_\infty \leq 20^j |u - v|, \quad \forall j = 1, \dots, m.$$

It follows that

$$\|H^{(m)}(u) - H^{(m)}(v)\|_\infty \leq 20^m |u - v|, \quad \forall u, v \in [-T, T].$$

Then, we obtain

$$\begin{aligned} |\widehat{f}_m(u) - \widehat{f}_m(v)| &= \left| \sum_{i=1}^5 a_i \left(H^{(m)}(u) \right)_i - \sum_{i=1}^5 a_i \left(H^{(m)}(v) \right)_i \right| \\ &\leq 20 \left\| H^{(m)}(u) - H^{(m)}(v) \right\|_\infty \\ &\leq 20(20^m) |u - v| = 20^{m+1} |u - v|, \quad \forall u, v \in [-T, T]. \end{aligned}$$

In other words, \widehat{f}_m is Lipschitz continuous with Lipschitz constant 20^{m+1} , thus (62) holds.

Now we move on to prove (63). From (59) in Lemma 17, we know

$$\|H_{\mathbf{W}, \mathbf{b}}^{(m)}\|_{L^\infty[-T, T]} \leq 20^m T + 4 \left(\frac{20^m - 1}{20 - 1} \right).$$

As a result, there holds

$$\begin{aligned} |\widehat{f}_{m, \theta}(x) - \widehat{f}_{m, \tilde{\theta}}(x)| &= \left| \sum_{i=1}^5 a_i \left(H_{\mathbf{W}, \mathbf{b}}^{(m)}(x) \right)_i - \sum_{i=1}^5 \tilde{a}_i \left(H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}(x) \right)_i \right| \\ &= \left| \sum_{i=1}^5 (a_i - \tilde{a}_i) \left(H_{\mathbf{W}, \mathbf{b}}^{(m)}(x) \right)_i + \sum_{i=1}^5 \tilde{a}_i \left(H_{\mathbf{W}, \mathbf{b}}^{(m)}(x) - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}(x) \right)_i \right| \\ &\leq \sum_{i=1}^5 |a_i - \tilde{a}_i| \max_{1 \leq j \leq 5} \left| \left(H_{\mathbf{W}, \mathbf{b}}^{(m)}(x) \right)_j \right| + \sum_{i=1}^5 |\tilde{a}_i| \max_{1 \leq j \leq 5} \left| \left(H_{\mathbf{W}, \mathbf{b}}^{(m)}(x) - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}(x) \right)_j \right| \\ &\leq \eta \|H_{\mathbf{W}, \mathbf{b}}^{(m)}\|_{L^\infty[-T, T]} + 20 \|H_{\mathbf{W}, \mathbf{b}}^{(m)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}\|_{L^\infty[-T, T]} \\ &\leq \eta \left(20^m T + 4 \left(\frac{20^m - 1}{20 - 1} \right) \right) + 20 \|H_{\mathbf{W}, \mathbf{b}}^{(m)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}\|_{L^\infty[-1, 1]}. \end{aligned}$$

To proceed, we need to compute $\|H_{\mathbf{W}, \mathbf{b}}^{(m)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)}\|_{L^\infty[-T, T]}$. Applying $|\sigma(u) - \sigma(v)| \leq |u - v|$, for $i = 1, \dots, 5$, $j = 1, \dots, m$, we have

$$\begin{aligned} &\left| \left(H_{\mathbf{W}, \mathbf{b}}^{(j)}(x) - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j)}(x) \right)_i \right| \\ &\leq \left| \left((W^{(j)} - \widetilde{W}^{(j)}) H_{\mathbf{W}, \mathbf{b}}^{(j-1)}(x) \right)_i + \left(\widetilde{W}^{(j)} (H_{\mathbf{W}, \mathbf{b}}^{(j-1)}(x) - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j-1)}(x)) \right)_i + (b^{(j)} - \widetilde{b}^{(j)})_i \right| \\ &\leq \eta \|H_{\mathbf{W}, \mathbf{b}}^{(j-1)}\|_{L^\infty[-T, T]} + 20 \|H_{\mathbf{W}, \mathbf{b}}^{(j-1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j-1)}\|_{L^\infty[-T, T]} + \eta \\ &\leq \eta \left(20^{j-1} T + 4 \left(\frac{20^{j-1} - 1}{20 - 1} \right) \right) + 20 \|H_{\mathbf{W}, \mathbf{b}}^{(j-1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j-1)}\|_{L^\infty[-T, T]} + \eta. \end{aligned}$$

This brings us to a recurrence relationship:

$$\begin{aligned} \left\| H_{\mathbf{W}, \mathbf{b}}^{(j)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j)} \right\|_{L^\infty[-T, T]} &\leq 20 \| H_{\mathbf{W}, \mathbf{b}}^{(j-1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j-1)} \|_{L^\infty[-T, T]} + \eta \left(20^{j-1} T + 4 \left(\frac{20^{j-1} - 1}{20 - 1} \right) + 1 \right) \\ &\leq 20 \| H_{\mathbf{W}, \mathbf{b}}^{(j-1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(j-1)} \|_{L^\infty[-T, T]} + (T + 1) 20^{j-1} \eta. \end{aligned}$$

We thereby obtain

$$\begin{aligned} \left\| H_{\mathbf{W}, \mathbf{b}}^{(m)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(m)} \right\|_{L^\infty[-T, T]} &\leq 20^{m-1} \| H_{\mathbf{W}, \mathbf{b}}^{(1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(1)} \|_{L^\infty[-T, T]} + (T + 1)(m - 1) 20^{m-1} \eta \\ &\leq (T + 1)(20^{m-1}) \eta + (T + 1)(m - 1) 20^{m-1} \eta \\ &= (T + 1) m 20^{m-1} \eta, \end{aligned}$$

where we have used $\| H_{\mathbf{W}, \mathbf{b}}^{(1)} - H_{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}}^{(1)} \|_{L^\infty[-T, T]} \leq \| W^{(1)} - \widetilde{W}^{(1)} \|_\infty \| H^{(0)} \|_\infty + \| b^{(1)} - \widetilde{b}^{(1)} \|_\infty \leq (T + 1) \eta$. Finally, we get

$$\begin{aligned} \|\widehat{f}_{m, \theta} - \widehat{f}_{m, \widetilde{\theta}}\|_{L^\infty[-T, T]} &\leq \eta \left(20^m T + 4 \left(\frac{20^m - 1}{20 - 1} \right) \right) + 20(T + 1) m 20^{m-1} \eta \\ &\leq (T + 1)(m + 1) 20^m \eta. \end{aligned}$$

The proof of (63) is complete. ■

C.2 Uniform Bound of Product Gate $\widehat{\Phi}$

We proceed by looking at the product gate $\widehat{\Phi}(u, v) := \widehat{\Phi}_\theta(u, v) = \widehat{f}_{m, \theta}(|\frac{u+v}{2}|) - \widehat{f}_{m, \theta}(|\frac{u-v}{2}|)$ defined earlier in Subsection 4.1. Here, $\theta = \{\mathbf{W}, \mathbf{b}, a\}$ represents a set of trainable parameters taking values on $[-4, 4]$. $\widehat{\Phi}$ is a ReLU FNN $\in \mathcal{F}(m + 1, (4, 10, 10, \dots, 10))$ with all parameter values in $[-4, 4]$. Using the results in Lemma 19, we are able to derive the Lipschitz-1 seminorm of $\widehat{\Phi}$ and the uniform bound $\|\widehat{\Phi}_\theta - \widehat{\Phi}_{\widetilde{\theta}}\|_\infty$. Denote

$$|\widehat{\Phi}_\theta|_{Lip1} = \sup_{x, y \in [-T, T]^2, x \neq y} \frac{|\widehat{\Phi}_\theta(x) - \widehat{\Phi}_\theta(y)|}{\|x - y\|_1}.$$

Lemma 20. *Let $\eta > 0, T > 0$, and $m \in \mathbb{N}$. Let $\mathbf{W}_\eta^*, \mathbf{b}_\eta^*, a_\eta^*$ be η -nets of $\mathbf{W}^*, \mathbf{b}^*, a^*$ defined above in Definition 18. With the input $(u, v) \in [-T, T]^2$, there hold*

$$\|\widehat{\Phi}_\theta\|_{L^\infty[-T, T]^2} \leq 20^{m+1} T \tag{64}$$

and

$$|\widehat{\Phi}_\theta|_{Lip1} \leq 20^{m+1} \tag{65}$$

and

$$\|\widehat{\Phi}_\theta - \widehat{\Phi}_{\widetilde{\theta}}\|_{L^\infty[-T, T]^2} \leq 2(T + 1)(m + 1) 20^m \eta. \tag{66}$$

Proof Let us first prove (64). Applying the Lipschitz-1 seminorm of \widehat{f}_m from (62), we have, for every $u, v \in [-T, T]$,

$$\begin{aligned} |\widehat{\Phi}_\theta(u, v)| &= \left| \widehat{f}_{m, \theta} \left(\left| \frac{u+v}{2} \right| \right) - \widehat{f}_{m, \theta} \left(\left| \frac{u-v}{2} \right| \right) \right| \leq 20^{m+1} \left| \left| \frac{u+v}{2} \right| - \left| \frac{u-v}{2} \right| \right| \\ &\leq 20^{m+1} |v| \\ &\leq 20^{m+1} T. \end{aligned}$$

This proves (64). Next, for every $u_1, u_2, v_1, v_2 \in [-T, T]$, we have

$$\begin{aligned} &|\widehat{\Phi}(u_1, v_1) - \widehat{\Phi}(u_2, v_2)| \\ &\leq \left| \widehat{f}_m \left(\left| \frac{u_1 + v_1}{2} \right| \right) - \widehat{f}_m \left(\left| \frac{u_2 + v_2}{2} \right| \right) \right| + \left| \widehat{f}_m \left(\left| \frac{u_1 - v_1}{2} \right| \right) - \widehat{f}_m \left(\left| \frac{u_2 - v_2}{2} \right| \right) \right| \\ &\leq 20^{m+1} \left| \left| \frac{u_1 + v_1}{2} \right| - \left| \frac{u_2 + v_2}{2} \right| \right| + 20^{m+1} \left| \left| \frac{u_1 - v_1}{2} \right| - \left| \frac{u_2 - v_2}{2} \right| \right| \\ &\leq 20^{m+1} (|u_1 - u_2| + |v_1 - v_2|). \end{aligned}$$

This proves (65).

According to (63) in Lemma 19, $\|\widehat{f}_{m, \theta} - \widehat{f}_{m, \tilde{\theta}}\|_{L^\infty[-T, T]} \leq (T+1)(m+1)20^m \eta$. Then for $(u, v) \in [-T, T]^2$,

$$\begin{aligned} &\|\widehat{\Phi}_\theta(u, v) - \widehat{\Phi}_{\tilde{\theta}}(u, v)\|_\infty \\ &\leq \left\| \widehat{f}_{m, \theta} \left(\left| \frac{u+v}{2} \right| \right) - \widehat{f}_{m, \tilde{\theta}} \left(\left| \frac{u+v}{2} \right| \right) \right\|_\infty + \left\| \widehat{f}_{m, \tilde{\theta}} \left(\left| \frac{u-v}{2} \right| \right) - \widehat{f}_{m, \theta} \left(\left| \frac{u-v}{2} \right| \right) \right\|_\infty \\ &\leq \|\widehat{f}_{m, \theta} - \widehat{f}_{m, \tilde{\theta}}\|_{L^\infty[-T, T]} + \|\widehat{f}_{m, \theta} - \widehat{f}_{m, \tilde{\theta}}\|_{L^\infty[-T, T]} \\ &\leq 2(T+1)(m+1)20^m \eta. \end{aligned}$$

This proves (66). The proof of Lemma 20 is complete. \blacksquare

C.3 Uniform Bound of Monomial Gate \widehat{h}_k

We proceed by looking at the monomial gate $\{\widehat{h}_k := \widehat{h}_{k, \theta}\}_{k=1}^{2^\ell}$ for some $\ell \in \mathbb{N}$. Recall the definitions we made at (14) and (15):

$$\left\{ \widehat{h}_k(u) \right\}_{k=1}^2 = \left\{ \widehat{h}_1(u) = u, \widehat{h}_2(u) = \widehat{\Phi}(u, u) = \widehat{f}_m(|u|) - \widehat{f}_m(0) \right\},$$

and iteratively for $j = 1, \dots, \ell - 1$, and $i = 1, \dots, 2^j$,

$$\widehat{h}_{2^{j+i}}(u) = \widehat{\Phi} \left(\widehat{h}_{2^j}(u), \widehat{h}_i(u) \right) = \widehat{f}_m \left(\left| \frac{\widehat{h}_{2^j}(u) + \widehat{h}_i(u)}{2} \right| \right) - \widehat{f}_m \left(\left| \frac{\widehat{h}_{2^j}(u) - \widehat{h}_i(u)}{2} \right| \right).$$

We define

$$B_j := B_{j, \theta, \tilde{\theta}} = \max_{1 \leq k \leq 2^j} \|\widehat{h}_{k, \theta} - \widehat{h}_{k, \tilde{\theta}}\|_{L^\infty[-T, T]}, \quad \forall j = 1, \dots, \ell, \quad (67)$$

where $\theta = \{\mathbf{W}, \mathbf{b}, a\}$ and $\tilde{\theta} = \{\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}, \widetilde{a}\}$ represent two different collections of network parameters. The following Lemma presents the uniform bound of $\|\widehat{h}_{k,\theta}\|_{L^\infty[-T,T]}$ and B_j , respectively.

Lemma 21. *Let $T > 0, m, \ell \in \mathbb{N}$. There holds, for $j = 1, \dots, \ell$,*

$$\max_{1 \leq k \leq 2^j} \|\widehat{h}_{k,\theta}\|_{L^\infty[-T,T]} \leq 20^{j(m+1)} \quad (68)$$

and

$$\max_{1 \leq k \leq 2^j} |\widehat{h}_{k,\theta}|_{Lip1} \leq 2 \cdot 20^{j(m+1)}. \quad (69)$$

Let $\eta > 0$. Also let $\mathbf{W}_\eta^*, \mathbf{b}_\eta^*, a_\eta^*$ be η -nets of $\mathbf{W}^*, \mathbf{b}^*, a^*$ defined above in Definition 18. There holds, for $j = 1, \dots, \ell$,

$$B_j \leq 2^j (T+1)(m+1)(20^{m+1})^j \eta. \quad (70)$$

Proof Let us first prove (68). Recall that from (62) Lemma 19, we derived $|\widehat{f}_{m,\theta}|_{Lip1} \leq 20^{m+1}$ which implies

$$|\widehat{f}_m(u) - \widehat{f}_m(v)| \leq 20^{m+1}|u - v|, \quad \forall u, v \in [-T, T].$$

For $j = 1, \dots, \ell - 1, i = 1, \dots, 2^j$,

$$\begin{aligned} \|\widehat{h}_{2^j+i}\|_{L^\infty[-T,T]} &= \sup_{u \in [-T,T]} \left| \widehat{f}_m \left(\left| \frac{\widehat{h}_{2^j}(u) + \widehat{h}_i(u)}{2} \right| \right) - \widehat{f}_m \left(\left| \frac{\widehat{h}_{2^j}(u) - \widehat{h}_i(u)}{2} \right| \right) \right| \\ &\leq \sup_{u \in [-T,T]} 20^{m+1} |\widehat{h}_i(u)| = 20^{m+1} \|\widehat{h}_i\|_{L^\infty[-T,T]}. \end{aligned}$$

We thereby obtain the relation

$$\max_{1 \leq k \leq 2^j} \|\widehat{h}_k\|_{L^\infty[-T,T]} \leq 20^{m+1} \max_{1 \leq k \leq 2^{j-1}} \|\widehat{h}_k\|_{L^\infty[-T,T]}, \quad \forall j = 1, \dots, \ell.$$

By induction, we have $\max_{1 \leq k \leq 2^j} \|\widehat{h}_k\|_{L^\infty[-T,T]} \leq 20^{(j-1)(m+1)} \max_{1 \leq k \leq 2} \|\widehat{h}_k\|_{L^\infty[-T,T]} \leq 20^{j(m+1)}$. This proves (68). In the same way,

$$\begin{aligned} |\widehat{h}_{2^j+i}(u) - \widehat{h}_{2^j+i}(v)| &= \left| \widehat{\Phi} \left(\widehat{h}_{2^j}(u), \widehat{h}_i(u) \right) - \widehat{\Phi} \left(\widehat{h}_{2^j}(v), \widehat{h}_i(v) \right) \right| \\ &\leq |\widehat{\Phi}|_{Lip1} \left(|\widehat{h}_{2^j}(u) - \widehat{h}_{2^j}(v)| + |\widehat{h}_i(u) - \widehat{h}_i(v)| \right) \\ &\leq 20^{m+1} \left(|\widehat{h}_{2^j}|_{Lip1} + |\widehat{h}_i|_{Lip1} \right) |u - v|, \end{aligned}$$

which implies by induction

$$\max_{1 \leq k \leq 2^j} |\widehat{h}_k|_{Lip1} \leq 20^{m+1} \max_{1 \leq k \leq 2^{j-1}} 2|\widehat{h}_k|_{Lip1} \leq 2 \cdot 20^{j(m+1)}.$$

This proves (69).

Next, we move on to prove (70). From (66) of Lemma 20, we have

$$B_1 = \max_{1 \leq k \leq 2} \|\widehat{h}_{k,\theta} - \widehat{h}_{k,\tilde{\theta}}\|_{L^\infty[-T,T]} \leq \|\widehat{\Phi}_\theta - \widehat{\Phi}_{\tilde{\theta}}\|_{L^\infty[-T,T]^2} \leq 2(T+1)(m+1)20^m \eta.$$

Then, by the definition of B_j in (67), for $j = 1, \dots, \ell - 1$,

$$B_{j+1} = \max_{1 \leq k \leq 2^{j+1}} \|\widehat{h}_{k,\theta} - \widehat{h}_{k,\tilde{\theta}}\|_{L^\infty[-T,T]} = \max \left\{ B_j, \max_{1 \leq k \leq 2^j} \|\widehat{h}_{2^j+k,\theta} - \widehat{h}_{2^j+k,\tilde{\theta}}\|_{L^\infty[-T,T]} \right\}.$$

For $1 \leq k \leq 2^j$,

$$\begin{aligned} & \|\widehat{h}_{2^j+k,\theta} - \widehat{h}_{2^j+k,\tilde{\theta}}\|_{L^\infty[-T,T]} \\ & \leq \|\widehat{\Phi}_\theta(\widehat{h}_{2^j,\theta}, \widehat{h}_{k,\theta}) - \widehat{\Phi}_{\tilde{\theta}}(\widehat{h}_{2^j,\theta}, \widehat{h}_{k,\theta})\|_{L^\infty[-T,T]} + \|\widehat{\Phi}_{\tilde{\theta}}(\widehat{h}_{2^j,\theta}, \widehat{h}_{k,\theta}) - \widehat{\Phi}_{\tilde{\theta}}(\widehat{h}_{2^j,\tilde{\theta}}, \widehat{h}_{k,\tilde{\theta}})\|_{L^\infty[-T,T]} \\ & \leq 2(T+1)(m+1)20^m\eta + \|\widehat{\Phi}_{\tilde{\theta}}(\widehat{h}_{2^j,\theta}, \widehat{h}_{k,\theta}) - \widehat{\Phi}_{\tilde{\theta}}(\widehat{h}_{2^j,\tilde{\theta}}, \widehat{h}_{k,\tilde{\theta}})\|_{L^\infty[-T,T]} \\ & \leq 2(T+1)(m+1)20^m\eta + 2(20^{m+1})B_j. \end{aligned}$$

Now, plugging this into the above iteration relation, we get

$$B_{j+1} \leq 2(T+1)(m+1)20^m\eta + 2(20^{m+1})B_j, \quad \forall j = 1, \dots, \ell - 1,$$

which is a recurrence relationship. We finally get

$$\begin{aligned} B_j & \leq (2 \cdot 20^{m+1})^{j-1} B_1 + (2(T+1)(m+1)20^m\eta) (1 + 2 \cdot 20^{m+1} + \dots + (2 \cdot 20^{m+1})^{j-2}) \\ & \leq \frac{2^{j+1}(20^{m+1})^j}{2(20^{m+1}) - 1} (T+1)(m+1)20^m\eta \\ & \leq 2^j (T+1)(m+1)(20^{m+1})^j \eta. \end{aligned}$$

The proof is complete. ■

C.4 Proof of proposition 12

Recall $c^* = [-C_K, C_K]^{2^\ell+1}$ with C_K to be a positive constant given in Definition 4. For $\eta > 0$, let c_η^* be an η -net of c^* such that for each $c \in c^*$, there exists $\tilde{c} \in c_\eta^*$ such that

$$\|c - \tilde{c}\|_1 = \sum_{k=0}^{2^\ell} |c_k - \tilde{c}_k| \leq \eta. \quad (71)$$

Proof of Proposition 12. Applying (69) in Lemma 21 and the fact that $\widehat{h}_{1,\theta}(u) = \widehat{h}_{1,\tilde{\theta}}(u) = u \in [-T, T]$, we obtain

$$\begin{aligned} \sum_{k=1}^{2^\ell} \|\widehat{h}_{k,\theta} - \widehat{h}_{k,\tilde{\theta}}\|_{L^\infty[-T,T]} & = \sum_{k=2}^{2^\ell} \|\widehat{h}_{k,\theta} - \widehat{h}_{k,\tilde{\theta}}\|_{L^\infty[-T,T]} \leq \sum_{j=1}^{\ell} 2^{j-1} B_j \\ & \leq \sum_{j=1}^{\ell} 2^{j-1} 2^j (T+1)(m+1)(20^{m+1})^j \eta \\ & = \frac{(T+1)(m+1)}{2} \sum_{j=1}^{\ell} (4(20^{m+1}))^j \eta \\ & \leq (T+1)(m+1)(4(20^{m+1}))^\ell \eta. \end{aligned}$$

We then have

$$\begin{aligned}
 & \left\| \sum_{k=1}^{2^\ell} (c_k \widehat{h}_{k,\theta} + c_0 - \widetilde{c}_k \widehat{h}_{k,\widetilde{\theta}} - \widetilde{c}_0) \right\|_{L^\infty[-T,T]} \\
 &= \left\| \sum_{k=1}^{2^\ell} (c_k \widehat{h}_{k,\theta} - \widetilde{c}_k \widehat{h}_{k,\theta}) + \sum_{k=1}^{2^\ell} (\widetilde{c}_k \widehat{h}_{k,\theta} - \widetilde{c}_k \widehat{h}_{k,\widetilde{\theta}}) + (c_0 - \widetilde{c}_0) \right\|_{L^\infty[-T,T]} \\
 &\leq \max_{1 \leq k \leq 2^\ell} \|\widehat{h}_{k,\theta}\|_{L^\infty[-T,T]} \eta + 2^\ell C_K (T+1)(m+1)(4(20^{m+1}))^\ell \eta \\
 &\leq 20^{\ell(m+1)} \eta + 2^\ell C_K (T+1)(m+1)(4(20^{m+1}))^\ell \eta \\
 &\leq C(T+1)(m+1)(8(20^{m+1}))^\ell \eta,
 \end{aligned}$$

where $C = 1 + C_K$ is a constant greater than 1.

Next, recall the function σ_λ defined in Definition 4 for $0 < \lambda \leq 1$. We observe that

$$|\sigma_\lambda(u) - \sigma_\lambda(v)| \leq \frac{1}{\lambda} |u - v|, \quad \forall u, v \in \mathbb{R}. \quad (72)$$

Note that each $f_{\mathcal{H}} \in \mathcal{H}$ has the form $\sigma_\lambda \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\theta} h_{k,j,\theta}^*(x) + c_{0,\theta} \right)$. Let $f_{\mathcal{H},\theta}$ and $f_{\mathcal{H},\widetilde{\theta}}$ represents two functions in \mathcal{H} with the set of parameters $\theta = \{\dot{W}, \dot{b}, \mathbf{W}, \mathbf{b}, a, c\} \in \Theta$ and $\widetilde{\theta} = \{\widetilde{W}, \widetilde{b}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}, \widetilde{a}, \widetilde{c}\} \in \widetilde{\Theta}$ respectively. To estimate $\mathcal{N}(\epsilon, \mathcal{H})$ for any $0 < \epsilon \leq 1$, we need to find a set of functions in \mathcal{H} that forms an ϵ -net.

To obtain such a function set, we choose $\widetilde{\Theta}$ in such a way that for any $\theta \in \Theta$, there exists $\widetilde{\theta} \in \widetilde{\Theta}$ such that

$$\begin{aligned}
 \|\dot{W} - \widetilde{W}\|_\infty &\leq \eta, & \|\dot{b} - \widetilde{b}\|_\infty &\leq \eta, & \|\mathbf{W} - \widetilde{\mathbf{W}}\|_{\infty,\infty} &\leq \eta, \\
 \|\mathbf{b} - \widetilde{\mathbf{b}}\|_\infty &\leq \eta, & \|a - \widetilde{a}\|_1 &\leq \eta, & \|c - \widetilde{c}\|_1 &\leq \eta.
 \end{aligned} \quad (73)$$

Note that for $x \in [-b-1, b+1]^d$,

$$\begin{aligned}
 |\widehat{r}_{i,j,\theta}(x) - \widehat{r}_{i,j,\widetilde{\theta}}(x)| &= |\dot{W}_{i,j,\theta} \cdot x + \dot{b}_{i,j,\theta} - \dot{W}_{i,j,\widetilde{\theta}} \cdot x - \dot{b}_{i,j,\widetilde{\theta}}| \\
 &\leq |(\dot{W}_{i,j,\theta} - \dot{W}_{i,j,\widetilde{\theta}}) \cdot x| + |\dot{b}_{i,j,\theta} - \dot{b}_{i,j,\widetilde{\theta}}| \leq d(b+1)\eta + \eta,
 \end{aligned}$$

which implies by (65) in Lemma 20 that

$$\left| \widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\theta}(x)) - \widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\widetilde{\theta}}(x)) \right| \leq 20^{m+1}(d(b+1) + 1)\eta.$$

Observe that $|\widehat{r}_{i,j}(x)| = |\dot{W}_{i,j} \cdot x + \dot{b}_{i,j}| \leq C_K^0 \sqrt{d}(b+1) + C_K^1$ for $x \in [-b-1, b+1]^d$. Then by (66) in Lemma 20 with $T_1 = \max\{C_K^0 \sqrt{d}(b+1) + C_K^1, 1\}$, we have

$$\begin{aligned}
 & \left| \widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\theta}(x)) - \widehat{\Phi}_{\widetilde{\theta}}(\Psi_b(x), \widehat{r}_{i,j,\widetilde{\theta}}(x)) \right| \\
 &\leq 20^{m+1}(d(b+1) + 1)\eta + 2(T_1 + 1)(m+1)20^m \eta \\
 &\leq (20(d(b+1) + 1) + (2C_K^0 \sqrt{d}(b+1) + 2C_K^1 + 4)(m+1))20^m \eta.
 \end{aligned}$$

Also, $|\widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\theta}(x))| \leq T_1 \cdot 20^{m+1}$ by (64) of Lemma 20 which implies

$$\begin{aligned} \left| \widehat{f}_{m,\theta} \left(\left| \widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\theta}(x)) \right| \right) \right| &\leq 20 \|H_\theta^{(m)}\|_{L^\infty[-T_1 \cdot 20^{m+1}, T_1 \cdot 20^{m+1}]} \\ &\leq 20 \left(20^m (T_1 \cdot 20^{m+1}) + 4 \left(\frac{20^m - 1}{20 - 1} \right) \right) \leq (T_1 + 1) 20^{2m+2} =: T_2. \end{aligned}$$

It follows from (72), (69), and (63) that for $x \in [-b-1, b+1]^d$,

$$\begin{aligned} &|f_{\mathcal{H},\theta}(x) - f_{\mathcal{H},\tilde{\theta}}(x)| \\ &= \left| \sigma_\lambda \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\theta} h_{k,j,\theta}^*(x) + c_{0,\theta} \right) - \sigma_\lambda \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\tilde{\theta}} h_{k,j,\tilde{\theta}}^*(x) + c_{0,\tilde{\theta}} \right) \right| \\ &\leq \frac{1}{\lambda} \left| \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\theta} h_{k,j,\theta}^*(x) + c_{0,\theta} \right) - \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\tilde{\theta}} h_{k,j,\tilde{\theta}}^*(x) + c_{0,\tilde{\theta}} \right) \right| \\ &\leq \frac{1}{\lambda} KC(T_2 + 1)(m+1)(8(20^{m+1}))^\ell \eta + \left| \frac{1}{\lambda} \sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\tilde{\theta}} \left\{ \widehat{h}_{k,j,\tilde{\theta}} \left(\frac{1}{d} \sum_{i=1}^d \widehat{f}_{m,\theta} \left(\left| \widehat{\Phi}_\theta(\Psi_b(x), \widehat{r}_{i,j,\theta}(x)) \right| \right) \right) \right. \right. \\ &\quad \left. \left. - \widehat{h}_{k,j,\tilde{\theta}} \left(\frac{1}{d} \sum_{i=1}^d \widehat{f}_{m,\tilde{\theta}} \left(\left| \widehat{\Phi}_{\tilde{\theta}}(\Psi_b(x), \widehat{r}_{i,j,\tilde{\theta}}(x)) \right| \right) \right) \right\} \right| \\ &\leq \frac{1}{\lambda} KC(T_2 + 1)(m+1)(8(20^{m+1}))^\ell \eta + \frac{1}{\lambda} K 2^{\ell+1} C_K 20^{\ell(m+1)} ((T_1 \cdot 20^{m+1} + 1)(m+1) 20^m \eta \\ &\quad + 20^{m+1}(20(d(b+1) + 1) + (2C_K^0 \sqrt{d}(b+1) + 2C_K^1 + 4)(m+1)) 20^m \eta) \\ &\leq \frac{K}{\lambda} C'_K (m+1)(b+1) 20^{(\ell+2)(m+2)} (C + C_K) \eta, \end{aligned}$$

where $C'_K > 0$ is a constant depending on d, C_K^0, C_K^1 .

When $x \notin [-b-1, b+1]^d$, we have $\Psi_b(x) = 0$ and thereby $h_{k,j}^*(x) = \widehat{h}_k \left(\frac{1}{d} \sum_{i=1}^d \widehat{f}_m(0) \right) = \widehat{h}_k(\widehat{f}_m(0))$ and by (59) with $T = 1$ and the bound for

$$\begin{aligned} &|f_{\mathcal{H},\theta}(x) - f_{\mathcal{H},\tilde{\theta}}(x)| \\ &\leq \frac{1}{\lambda} \left| \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\theta} \widehat{h}_{k,\theta}(\widehat{f}_{m,\theta}(0)) + c_{0,\theta} \right) - \left(\sum_{j=1}^K \sum_{k=1}^{2^\ell} c_{k,j,\tilde{\theta}} \widehat{h}_{k,\tilde{\theta}}(\widehat{f}_{m,\tilde{\theta}}(0)) + c_{0,\tilde{\theta}} \right) \right| \\ &\leq \frac{1}{\lambda} KC(2 \cdot 20^{m+1})(m+1)(8(20^{m+1}))^\ell \eta + \frac{1}{\lambda} \sum_{j=1}^K \sum_{k=1}^{2^\ell} |c_{k,j,\tilde{\theta}}| \left| \widehat{h}_{k,\tilde{\theta}}(\widehat{f}_{m,\theta}(0)) - \widehat{h}_{k,\tilde{\theta}}(\widehat{f}_{m,\tilde{\theta}}(0)) \right| \\ &\leq \frac{1}{\lambda} KC(2 \cdot 20^{m+1})(m+1)(8(20^{m+1}))^\ell \eta + \frac{1}{\lambda} K 2^\ell C_K 2 \cdot 20^{\ell(m+1)} |f_{m,\theta}(0) - f_{m,\tilde{\theta}}(0)| \\ &\leq \frac{1}{\lambda} KC(2 \cdot 20^{m+1})(m+1)(8(20^{m+1}))^\ell \eta + \frac{1}{\lambda} K 2^\ell C_K 2 \cdot 20^{\ell(m+1)} (2m+2) 20^m \eta \\ &\leq \frac{K}{\lambda} (2 \cdot 20^{m+1})(m+1) 20^{(\ell+1)(m+2)} (C + C_K) \eta. \end{aligned}$$

Therefore, $\|f_{\mathcal{H},\theta} - f_{\mathcal{H},\tilde{\theta}}\|_{\infty} \leq \frac{K}{\lambda}(C'_K + 1)(2m + 2)(b + 1)20^{(\ell+2)(m+2)}(C + C_K)\eta$. Hence, $\|f_{\mathcal{H},\theta} - f_{\mathcal{H},\tilde{\theta}}\|_{\infty} \leq \epsilon$ if

$$\eta = \frac{\lambda\epsilon}{K(b+1)(C'_K+1)(C+C_K)(2m+2)20^{(\ell+2)(m+2)}}. \quad (74)$$

Since $0 < \lambda \leq 1, 0 < \epsilon \leq 1, C > 1, C_K, C'_K > 0, b > 1, m, \ell, K \in \mathbb{N}$, we can see that $0 < \eta \leq 1$.

Functions in \mathcal{H} can be implemented by a neural network consisting of a preprocessing subnetwork, a stack of dK product gates Φ , and a stack of K EBTnets. For more details, please refer to Remark 5.

The preprocessing subnetwork consists of dK product gates $\hat{\Phi}$ and dK units of $\mathcal{F}(2, (2, 1))$ that is equipped with one $\dot{W} \in \mathbb{R}^d$ with $\|\dot{W}\| \leq C_K^0$, and one $\dot{b} \in \mathbb{R}$ with $|\dot{b}| \leq C_K^1$. This implies that the preprocessing subnetwork consists of $2dK$ $\mathbf{W} \in \mathbf{W}^*$, $\mathbf{b} \in \mathbf{b}^*$, $a \in a^*$, and dK \dot{W} and dK \dot{b} .

Each EBTnet consists of $1 + 2 + \dots + 2^{\ell-1} = 2^{\ell} - 1$ product gates $\hat{\Phi}$, and each $\hat{\Phi}$ consists of two \hat{f}_m . Each \hat{f}_m consists of one $\mathbf{W} \in \mathbf{W}^*$, one $\mathbf{b} \in \mathbf{b}^*$ and one $a \in a^*$. This implies that each EBTnet is equipped with $2(2^{\ell} - 1)\mathbf{W} \in \mathbf{W}^*$, $\mathbf{b} \in \mathbf{b}^*$, $a \in a^*$. Finally, we see that such a neural network is equipped with $2K(2^{\ell} + 2d - 1)\mathbf{W} \in \mathbf{W}^*$, $\mathbf{b} \in \mathbf{b}^*$, $a \in a^*$, and $Kc \in c^*$, and dK \dot{W} and dK \dot{b} .

To satisfy the requirements in (73), it suffices to choose a set $\tilde{\Theta}$ which, with $\tilde{C} = C_K^0/\sqrt{d}$, has cardinality at most

$$\begin{aligned} & \left[\frac{20}{\eta} \right]^{5(2K)(2^{\ell}+2d-1)} \left[\frac{4}{\eta} \right]^{5m(2K)(2^{\ell}+2d-1)} \left[\frac{20}{\eta} \right]^{25m(2K)(2^{\ell}+2d-1)} \left[\frac{2^{\ell}C_K}{\eta} \right]^{K2^{\ell}} \left[\frac{\tilde{C}}{\eta} \right]^{d^2K} \left[\frac{C_K^1}{\eta} \right]^{dK} \\ & \leq \left(\frac{21}{\eta} \right)^{(10+50m)K(2^{\ell}+2d)} \left(\frac{5}{\eta} \right)^{10mK(2^{\ell}+2d)} \left(\frac{2^{\ell}(C_K+1)}{\eta} \right)^{K2^{\ell}} \left(\frac{\tilde{C}+1}{\eta} \right)^{d^2K} \left(\frac{C_K^1+1}{\eta} \right)^{dK} \\ & \leq 21^{(60m)K(2^{\ell}+2d)} 5^{10mK(2^{\ell}+2d)} \left(2^{\ell}K2^{\ell} \right) (C_K+1)^{\ell K2^{\ell}} (\tilde{C}+1)^{d^2K} (C_K^1+1)^{d^2K} \eta^{-K(70m(2^{\ell}+2d)+2^{\ell}+2d^2)} \\ & \leq (21^{60}5^{10})^{mK(2^{\ell}+2d)} (2(C_K+1))^{\ell K2^{\ell}} ((\tilde{C}+1)(C_K^1+1))^{d^2K} \left(\frac{1}{\eta} \right)^{71mK(2^{\ell}+2d^2)}. \end{aligned}$$

Then, plugging in η given in (74), we obtain

$$\begin{aligned} \mathcal{N}(\epsilon, \mathcal{H}) & \leq (21^{60}5^{10})^{mK(2^{\ell}+2d)} (2(C_K+1))^{\ell K2^{\ell}} ((\tilde{C}+1)(C_K^1+1))^{d^2K} \\ & \quad \left(\frac{K(b+1)(C'_K+1)(C+C_K)(2m+2)20^{(\ell+2)(m+2)}}{\lambda\epsilon} \right)^{71mK(2^{\ell}+2d^2)}. \end{aligned}$$

Then, we have

$$\begin{aligned} & \frac{1}{K} \log \mathcal{N}(\epsilon, \mathcal{H}) \\ & \leq m(2^{\ell} + 2d) \log(21^{60}5^{10}) + 4\ell 2^{\ell} \log(C_K) + d^2 \log((\tilde{C} + 1)(C_K^1 + 1)) \end{aligned}$$

$$\begin{aligned}
& +71m(2^\ell + 2d^2) \left(\log(96KC'_K) + \log\left(\frac{bmC_K}{\lambda\epsilon}\right) \right) + 639m^2\ell(2^\ell + 2d^2) \log(20) \\
\leq & 72m(2^\ell)(2d^2) \left(\log(21^{60}5^{10}96KC'_K(\tilde{C} + 1)(C_K^1 + 1)) + \log\left(\frac{bmC_K}{\lambda\epsilon}\right) \right) \\
& + 4\ell 2^\ell \log(C_K) + 639 \log(20)m^2\ell(2^\ell)(2d^2) \\
\leq & \frac{C'}{K}m2^\ell \log\left(\frac{bmC_K}{\lambda\epsilon}\right) + 4\ell 2^\ell \log(C_K) + 639 \log(20)m^2\ell(2^\ell)(2d^2) \\
\leq & \frac{C'}{K}m2^\ell \log\left(\frac{bC_K}{\lambda\epsilon}\right) + 4\ell 2^\ell \log(C_K) + \frac{C''}{K}m^2\ell(2^\ell),
\end{aligned}$$

where C', C'' are positive constants independent of ℓ, m, b, λ, C_K or ϵ . ■

References

- Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608 – 633, 2007.
- Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- Konstantinos Blekas, Aristidis Likas, Nikolas P Galatsanos, and Isaac E Lagaris. A spatially constrained mixture model for image segmentation. *IEEE Transactions on Neural Networks*, 16(2): 494–498, 2005.
- Thijs Bos and Johannes Schmidt-Hieber. Convergence rates of deep ReLU networks for multiclass classification. *Electronic Journal of Statistics*, 16(1):2724–2773, 2022.
- Colin Campbell and Yiming Ying. Learning with support vector machines. *Synthesis lectures on artificial intelligence and machine learning*, 5(1):1–95, 2011.
- Di-Rong Chen, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Support vector machine soft margin classifiers: Error Analysis. *The Journal of Machine Learning Research*, 5:1143–1175, 2004.
- Andreas Christmann. Classification based on the Support Vector Machine and on regression depth. In *Statistical Data Analysis Based on the L 1-Norm and Related Methods*, pages 341–352. Springer, 2002.
- Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- Felipe Cucker and Ding Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.
- Weinan E, Chao Ma, and Lei Wu. The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022.
- Han Feng, Shuo Huang, and Ding-Xuan Zhou. Generalization analysis of CNNs for classification on spheres. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6200–6213, 2021.
- Erin George, Michael Murray, William Swartworth, and Deanna Needell. Training shallow ReLU networks on noisy data using hinge loss: When do we overfit and is it benign? *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.
- László Györfi, Michael Köhler, Adam Krzyżak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*, volume 1. Springer, 2002.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- Shouyou Huang, Yunlong Feng, and Qiang Wu. Fast rates of Gaussian empirical gain maximization with heavy-tailed noise. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):6038–6043, 2022.
- Shirin Jalali, Carl Nuzman, and Iraj Saniee. Efficient deep approximation of GMMs. *Advances in Neural Information Processing Systems*, 32, 2019.
- JooSeuk Kim and Clayton Scott. L_2 kernel classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1822–1831, 2009.
- Yongdai Kim, Ilsang Ohn, and Dongha Kim. Fast convergence rates of deep neural networks for classification. *Neural Networks*, 138:179–197, 2021.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.
- Yunwen Lei, Urun Dogan, Alexander Binder, and Marius Kloft. Multi-class SVMs: From tighter data-dependent generalization bounds to novel algorithms. *Advances in Neural Information Processing Systems*, 28, 2015.
- Tianyang Li, Adarsh Prasad, and Pradeep K Ravikumar. Fast classification rates for high-dimensional Gaussian generative models. *Advances in Neural Information Processing Systems*, 28, 2015.
- Tianyang Li, Xinyang Yi, Constantine Carmanis, and Pradeep Ravikumar. Minimax Gaussian classification & clustering. In *Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2017.
- Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.
- Denali Molitor, Deanna Needell, and Rachel Ward. Bias of homotopic gradient descent for the Hinge loss. *Applied Mathematics & Optimization*, 84:621–647, 2021.
- Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 14, 2001.
- Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.
- Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.
- Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

- Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.
- Uri Shaham, Alexander Cloninger, and Ronald R Coifman. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018.
- Guohao Shen, Yuling Jiao, Yuanyuan Lin, and Jian Huang. Approximation with CNNs in Sobolev Space: with applications to classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:2876–2888, 2022.
- Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93, 2001.
- Ingo Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18(3):768–791, 2002.
- Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.
- Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.
- Namjoon Suh, Tian-Yi Zhou, and Xiaoming Huo. Approximation and non-parametric estimation of functions over high-dimensional spheres via deep ReLU networks. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.
- Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.
- Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & business media, 1999.
- Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Multi-kernel regularized classifiers. *Journal of Complexity*, 23(1):108–134, 2007.
- Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.
- William Henry Young. On classes of summable functions and their Fourier series. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 87(594):225–229, 1912.
- Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.
- Tian-Yi Zhou and Xiaoming Huo. Learning ability of interpolating deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 68:101582, 2024.