Benign Overfitting in Deep Neural Networks under Lazy Training

Zhenyu Zhu [1] **Fanghui Liu** [1] **Grigorios G Chrysos** [1] **Francesco Locatello** [2] **Volkan Cevher** [1]

Abstract

This paper focuses on over-parameterized deep neural networks (DNNs) with ReLU activation functions and proves that when the data distribution is well-separated, DNNs can achieve *Bayes-optimal* test error for classification while obtaining (nearly) zero-training error under the lazy training regime. For this purpose, we unify three interrelated concepts of overparameterization, benign overfitting, and the Lipschitz constant of DNNs. Our results indicate that interpolating with smoother functions leads to better generalization. Furthermore, we investigate the special case where interpolating smooth ground-truth functions is performed by DNNs under the Neural Tangent Kernel (NTK) regime for generalization. Our result demonstrates that the generalization error converges to a constant order that only depends on label noise and initialization noise, which theoretically verifies benign overfitting. Our analysis provides a tight lower bound on the normalized margin under non-smooth activation functions, as well as the minimum eigenvalue of NTK under high-dimensional settings, which has its own interest in learning theory.

1. Introduction

Benign overfitting has attracted significant research interest recently in an effort to understand why predictors with zero training loss can still achieve counter-intuitively good generalization performance even in the presence of noise (Koehler et al., 2021; Zou et al., 2021; Chatterji and Long, 2022; Wang et al., 2022; Mei and Montanari, 2022). Current efforts on benign overfitting mainly focus on the finite sample behavior under linear regression (Bartlett et al., 2020; Chatterji et al., 2021; Zou et al., 2021), kernel-based estimators (Mei and Montanari, 2022; Liang et al., 2019), and logistic regression (Montanari et al., 2019; Wang et al., 2021).

To our knowledge, results on neural networks (NNs) are restricted to two-layer neural networks (Tsigler and Bartlett, 2020; Ju et al., 2021; Frei et al., 2022; Cao et al., 2022) and three-layer neural networks but only the last layer is trained (Ju et al., 2022). The extension from shallow NNs to deep neural networks (DNNs) is non-trivial: *under what conditions does benign overfitting occur in deep neural networks?* and *what makes them special?* are still open problems in both statistical learning theory and deep learning theory.

In this work, we address this open question in benign overfitting of deep ReLU NNs for binary classification under the lazy training regime. We assume the network is trained by stochastic gradient descent (SGD) on well-separated data under adversarially corrupted labels, following the standard problem setting of Frei et al. (2022). We prove that the ReLU DNN exhibits benign overfitting, i.e., obtaining *Bayes-optimal* test error while obtaining zero training error under the lazy training regime.

Our results establish a rigorous connection between the Lipschitz constants of DNNs and benign overfitting. We demonstrate that interpolating with (Lipschitz) smoother functions leads to a faster convergence rate on the generalization guarantees. Accordingly, for a better understanding of how the estimator by DNNs interpolates the ground-truth function, we also consider a regression task for DNNs from an approximation theory view (Cucker and Zhou, 2007), interpolating the smooth ground-truth function by DNNs under the neural tangent kernel (NTK) regime (Bach, 2017; Jacot et al., 2018).

Overall, we expect our results to foster a refined analysis of the generalization guarantees for large dimensional machine learning models, especially on DNNs.

1.1. Contributions and technical challenges

In this paper, we consider a finite sample behavior, in which the input dimension d can be large but fixed, or comparably large with the number of training data n and model parameters to obtain a dimension-free bound (Bartlett et al., 2020; Ju et al., 2022; Li et al., 2021). Our main contributions are

[1]Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland [2]Amazon Web Services (Work done outside of Amazon). Correspondence to: Zhenyu Zhu <zhenyu.zhu@epfl.ch>, Fanghui Liu <fanghui.liu@epfl.ch>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

summarized below:

- We adhere to the standard data setting with label noise that has been previously explored in two-layer networks (Frei et al., 2022), along with the model setting for multi-layer fully connected neural networks (Cao and Gu, 2019; Allen-Zhu et al., 2019b). Building upon the proof concept of (Frei et al., 2022), we extend our results to deep ReLU neural networks, which presents a considerable challenge in connecting the Bayes-optimal test error to the training dynamics of deep neural networks. Theorem 1 for binary classification shows that, under the lazy training regime, even though training on noisy data, DNNs can still obtain the *Bayes-optimal* test error, i.e., the error rate is less than the proportion of incorrect labels in the training set plus a generalization term that converges to zero. We also demonstrate that this term is positively correlated with the Lipschitz constants of DNNs, which implies that interpolating more smooth functions leads to a faster convergence rate.
- Theorem 2 provides the first lower bound on the minimum eigenvalues of the NTK matrix of DNNs in the high-dimensional setting and demonstrates its phase transition under different tendencies of the number of training data and input dimension. We believe that it has its own interest in learning theory.
- Theorem 3 builds the generalization guarantees of over-parameterized neural networks under the NTK regime in the high-dimensional setting to learn a ground-truth function in RKHS. Our result exhibits a phase transition on the excess risk (related to generalization performance) between the $n < d$ and $n > d$ case. It implies that the excess risk finally converges to a constant order only relying on the label noise and initialization noise, which theoretically verifies the benign overfitting.

Technical challenges. The main technical challenge of this paper is how to derive the lower bound of the non-smooth function in deep ReLU neural networks for the normalized margin on test points. In the context of lazy training (Chizat et al., 2019), the function of a neural network is nearly linear during the initial stages of training. By analyzing the accumulation of weights for each training step, we can establish a lower bound for the normalized margin on test points. By doing so, we transform the *Bayes-optimal* test error to the expected risk and Lipschitz constant of DNNs.

When compared with Frei et al. (2022) on shallow neural networks with smooth activation functions for binary classification, we extend their results to our deep ReLU neural networks, not limited to high-dimensional settings, and obtain a faster convergence rate with the number of data. The key difficulty lies in how to build the relationship between the *Bayes-optimal* test error and the training dynamics of DNNs. When compared to the generalization results of deep neural networks (DNNs) in the over-parameterized regime (Cao and Gu, 2019), our results focus on overfitted models that are trained by noisy data and achieve a faster convergence rate. Besides, Ju et al. (2022) present generalization guarantees on three-layer neural networks (only the last layer is training) for regression, which has the closed-form min L^2-norm solution. However, this nice property is invalid in our DNN setting. In this case, we build the connection between DNNs and kernel methods (e.g., NTK) in high dimensional settings for benign overfitting.

1.2. Related work

Benign overfitting: There has been a significant amount of research devoted to understanding the phenomenon of benign overfitting, with a particular emphasis on linear models, e.g., linear regression (Bartlett et al., 2020; Chatterji et al., 2021; Zou et al., 2021), sparse linear regression (Chatterji and Long, 2022; Koehler et al., 2021; Wang et al., 2022), logistic regression (Montanari et al., 2019; Wang et al., 2021), ridge regression (Tsigler and Bartlett, 2020) and kernel-based estimators (Mei and Montanari, 2022; Liang et al., 2019). Furthermore, the concept of benign overfitting can be extended to tempered or catastrophic based on various spectra of the kernel (ridge) regression (Mallinar et al., 2022).

For nonlinear models, Li et al. (2021) study the benign overfitting phenomenon of random feature models. Frei et al. (2022) prove that a two-layer fully connected neural network exhibits benign overfitting under certain conditions, e.g., well-separated log-concave distribution and smooth activation function. Then Xu and Gu (2023) extends the previous results to the non-smooth case. Similarly, Cao et al. (2022) focus on the benign overfitting of two-layer convolutional neural networks (CNN).

Mallinar et al. (2022) argue that many true interpolation methods (such as neural networks) for noisy data are not benign but tempered overfitting, and even catastrophic under various model capacities.

Generalization of NNs and Neural Tangent Kernel (NTK): The generalization ability of neural networks has been a core problem in machine learning theory. Brutzkus et al. (2017) show that SGD can learn an over-parameterized two-layer neural network with good generalization ability. Allen-Zhu et al. (2019a) study the generalization performance of SGD for 2- and 3-layer networks. Cao and Gu (2019) study the training and generalization of deep neural networks (DNNs) in the over-parameterized regime. Besides, Arora et al. (2019a); Cao and Gu (2020); E et al. (2020) provide the algorithm-dependent generalization bounds for different settings.

The NTK (Jacot et al., 2018) is a powerful tool for deep neural network analysis. Specifically, NTKs establish equivalence between the training dynamics of gradient-based algorithms for DNNs and kernel regression under specific initialization, so it can be considered as an intermediate step between simple linear models and DNNs (Allen-Zhu et al., 2019b; Du et al., 2019a; Chen et al., 2020). Besides, the convergence rate (Arora et al., 2019b) and generalization bound (Cao and Gu, 2019; Zhu et al., 2022; Nguyen et al., 2021; Bombari et al., 2022) can be linked to the minimum eigenvalue of the NTK matrix.

One can see that studying benign overfitting for DNNs is missing and it appears possible to borrow some ideas from deep learning theory, e.g., NTK. Nevertheless, we need to tackle the noisy training as well as the high-dimensional setting for benign overfitting, which is our main interest.

2. Problem Settings

In this section, we detail the problem setting for a deep ReLU neural network trained by SGD from the perspective of notations, neural network architecture, initialization schemes, and optimization algorithms.

2.1. Notation

In this paper, we use the shorthand $[n] := \{1, 2, \ldots, n\}$ for a positive integer n. We denote by $a(n) \gtrsim b(n)$: the inequality $a(n) \geq cb(n)$ that hides a positive constant c that is independent of n. Vectors (matrices) are denoted by boldface, lower-case (upper-case) letters. The standard Gaussian distribution is $\mathcal{N}(0, 1)$ with the zero-mean and the identity variance. We use the Lip_f to represent the Lipschitz constant of the function f. We follow the standard Bachmann–Landau notation in complexity theory e.g., $\mathcal{O}$, o, Ω, and Θ for order notation.

2.2. Network

Here we introduce the formulation of DNNs. We focus on the typical depth-L fully-connected ReLU neural networks with scalar output, width m on the hidden layers and n training data, $\forall i \in [n]$:

$$\begin{aligned} \boldsymbol{h}_{i,0} &= \boldsymbol{x}_i; \\ \boldsymbol{h}_{i,l} &= \phi(\boldsymbol{W}_l \boldsymbol{h}_{i,l-1}); \quad \forall l \in [L-1]; \\ f(\boldsymbol{x}_i; \boldsymbol{W}) &= \boldsymbol{W}_L \boldsymbol{h}_{i,L-1}; \end{aligned} \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the input, $f(\boldsymbol{x}_i; \boldsymbol{W}) \in \mathbb{R}$ is the neural network output, and $\phi = \max(0, x)$ is the ReLU activation function. The neural network parameters formulate the tuple of weight matrices $\boldsymbol{W} := \{\boldsymbol{W}_l\}_{l=1}^L \in \{\mathbb{R}^{m\times d} \times (\mathbb{R}^{m\times m})^{L-2} \times \mathbb{R}^{1\times m}\}$.

Initialization: We follow the standard Neural Tangent Kernel (NTK) initialization (Allen-Zhu et al., 2019b):

$$\begin{aligned} [\boldsymbol{W}_1]_{i,j} &\sim \mathcal{N}(0, \tfrac{2}{m}); \quad \forall i, j \in [m] \times [d]; \\ [\boldsymbol{W}_l]_{i,j} &\sim \mathcal{N}(0, \tfrac{2}{m}); \quad \forall i, j \in [m] \quad \text{and} \quad l \in [L-2]+1; \\ [\boldsymbol{W}_L]_{i,j} &\sim \mathcal{N}(0, 1); \quad \forall i, j \in [1] \times [m]. \end{aligned} \tag{2}$$

The related Neural Tangent Kernel (NTK) (Jacot et al., 2018) matrix of neural network f can be expressed as:

$$K_{\texttt{NTK}}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) := \mathbb{E}_{\boldsymbol{W}} \left\langle \frac{\partial f(\boldsymbol{x}; \boldsymbol{W})}{\partial \boldsymbol{W}}, \frac{\partial f(\widetilde{\boldsymbol{x}}; \boldsymbol{W})}{\partial \boldsymbol{W}} \right\rangle . \tag{3}$$

By virtue of $\phi(x) = x\phi'(x)$ of ReLU, we have $\boldsymbol{h}_{i,l} = \boldsymbol{D}_{i,l}\boldsymbol{W}_l\boldsymbol{h}_{i,l-1}$, where $\boldsymbol{D}_{i,l}$ is a diagonal matrix under the ReLU activation function defined as below.

Definition 1 (Diagonal sign matrix)**.** For each $i \in [n]$, $l \in [L-1]$ and $k \in [m]$, the diagonal sign matrix $\boldsymbol{D}_{i,l}$ is defined as: $(\boldsymbol{D}_{i,l})_{k,k} = 1\{(\boldsymbol{W}_l\boldsymbol{h}_{i,l-1})_k \geq 0\}$.

In addition, we define ω-neighborhood to describe the difference between two matrices.

For any $\boldsymbol{W} \in \mathcal{W}$, we define its ω-neighborhood as follows:

Definition 2 (ω-neighborhood)**.**

$$\mathcal{B}(\boldsymbol{W}, \omega) := \{\boldsymbol{W}' \in \mathcal{W} : \|\boldsymbol{W}_l' - \boldsymbol{W}_l\|_{\mathrm{F}} \leq \omega, l \in [L]\} .$$

2.3. Optimization algorithm

In our work, a deep ReLU neural network is trained by SGD on the training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ sampled from a joint distribution P. The data generation process is deferred to Section 3.1 for binary classification and Section 4 for regression. We employ the logistic loss for classification, which is defined as $\ell(z) = \log(1 + \exp(-z))$, and denote $g(z) := -\ell'(z) = \frac{1}{1+e^z}$ for notational simplicity.

The expected risk is defined as $\mathbb{E}_{(\boldsymbol{x},y)\sim P}\, \ell(yf(\boldsymbol{x}; \boldsymbol{W}))$. Denote the empirical risks under ℓ by: $\hat{L}(\boldsymbol{W}) := \frac{1}{n}\sum_{i=1}^n \ell(y_i f(\boldsymbol{x}_i; \boldsymbol{W}))$, we employ SGD to minimize $\hat{L}(\boldsymbol{W})$ initialized at $\boldsymbol{W}^{(0)}$ with fixed step-size $\alpha > 0$, as shown in Algorithm 1.

For notational simplicity, at step t, the neural network output is denoted as $f_i^{(t)} = f(\boldsymbol{x}_i; \boldsymbol{W}^{(t)})$ and the derivative of the loss function is realted to $g_i^{(t)} := g(y_i f_i^{(t)}) = g(y_i f(\boldsymbol{x}_i; \boldsymbol{W}^{(t)}))$.

Algorithm 1 SGD for training DNNs

Input: training data $\{(\boldsymbol{x}_i, y_i) \sim P\}_{i=1}^n$ and step size α.
Gaussian initialization: $\boldsymbol{W}_l^{(0)} \sim \mathcal{N}(0, 2/m)$, $l \in [L-1]$.
Gaussian initialization: $\boldsymbol{W}_L^{(0)} \sim \mathcal{N}(0, 1)$.
for $i = 1$ **to** n **do**
  Draw $(\boldsymbol{x}_i, y_i)$ from $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$.
  $\boldsymbol{W}^{(i)} = \boldsymbol{W}^{(i-1)} - \alpha \cdot \nabla_{\boldsymbol{W}} \ell\big(y_i f(\boldsymbol{x}_i; \boldsymbol{W}^{(i-1)})\big)$.
end for
Output $\boldsymbol{W}^{(n)}$ for the final network $f(\boldsymbol{x}; \boldsymbol{W}^{(n)})$.

3. Main Results on Binary Classification

In this section, we present our main result on benign overfitting of a ReLU DNN for binary classification under the lazy training regime. The data generation process is introduced in Section 3.1, and the related assumptions that are given in Section 3.2. Our main theory and proof sketch are presented in Section 3.3 and Section 3.4, respectively. We use NTK initialization (Allen-Zhu et al., 2019b) in this section, but the main result can be easily extended to more initializations, such as He (He et al., 2015) and LeCun (LeCun et al., 2012).

3.1. Data generation process

We consider a standard mixture model setting (Chatterji and Long, 2021; Frei et al., 2022) in benign overfitting for binary classification, where a joint distribution P is defined over $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \{\pm 1\}$ and samples from this distribution can have noisy labels. Following Frei et al. (2022), we first define the clean distribution $\widetilde{P}$ and then define the true distribution P based on $\widetilde{P}$:

1. Sample a clean label $\widetilde{y}$ uniformly at random, $\widetilde{y} \sim$ Uniform$(\{+1, -1\})$.
2. Sample $\boldsymbol{z} \sim P_{\text{clust}}$ that satisfy:
   - $P_{\text{clust}} = P_{\text{clust}}^{(1)} \times \cdots \times P_{\text{clust}}^{(d)}$ is a product distribution whose marginals are all mean-zero with the sub-Gaussian norm at most one;
   - P_{clust} is a λ-strongly log-concave distribution over $\mathbb{R}^d$ for some $\lambda > 0$;
   - For some κ, it holds that $\mathbb{E}_{\boldsymbol{z} \sim P_{\text{clust}}}(\|\boldsymbol{z}\|^2) > \kappa d$.
3. Generate $\widetilde{\boldsymbol{x}} = \boldsymbol{z} + \widetilde{y}\boldsymbol{\mu}$.
4. Then, given a noise rate $\eta \in [0, \frac{1}{2})$, P is any distribution over $\mathbb{R}^d \times \{\pm 1\}$ such that the marginal distribution of the features for P and $\widetilde{P}$ coincide, and the total variation distance between the two distributions satisfies $d_{\text{TV}}(\widetilde{P}, P) \leq \eta$. Specifically, P has the same marginal distribution over $\boldsymbol{x}$ as $\widetilde{P}$, but a sample $(\boldsymbol{x}, y) \sim P$ has a label equal to $\widetilde{y}$ with probability $1 - \eta$ and has a label equal to $-\widetilde{y}$ with probability η. That is, the labels are flipped with η ratio.

We denote by $\mathcal{C} \subset [n]$ the set of indices corresponding to samples with clean labels, and $\mathcal{C}'$ as the set of indices corresponding to noisy labels so that $i \in \mathcal{C}'$ implies $(\boldsymbol{x}_i, y_i) \sim P$ is such that $y_i = -\widetilde{y}_i$ using the notation above.

3.2. Assumptions

We make two assumptions about the data distribution.

Assumption 1 (Du et al. (2019a); Allen-Zhu et al. (2019b))**.** We assume that the data is bounded, i.e. there is a constant C_{norm} that satisfies $\|\boldsymbol{x}\|_2 \leq C_{\text{norm}}$.

Assumption 2. For two different data sample $(\boldsymbol{x}_1, \widetilde{y}_1), (\boldsymbol{x}_2, \widetilde{y}_2) \sim \widetilde{P}$, The NTK kernel defined in Eq. (3) satisfies that:

$$\begin{aligned}&\mathbb{E}_{(\boldsymbol{x}_1, \widetilde{y}_1), (\boldsymbol{x}_2, \widetilde{y}_2) \sim \widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1, \boldsymbol{x}_2) | \widetilde{y}_1 = \widetilde{y}_2\big] \\ -&\mathbb{E}_{(\boldsymbol{x}_1, \widetilde{y}_1), (\boldsymbol{x}_2, \widetilde{y}_2) \sim \widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1, \boldsymbol{x}_2) | \widetilde{y}_1 \neq \widetilde{y}_2\big] \\ \geq& C_N > 0\,.\end{aligned}$$

Remark: This assumption states that the NTK value for data points belonging to the same class is larger than that for a different class, in expectation. This makes sense in practice, since, as a kernel, the NTK is able to evaluate the similarity of two data points (Schölkopf et al., 2002): if they are from the same class, the similarity value is large and vice versa. To verify this assumption, we give an example of the two-layer NTK over a uniform distribution inside a multidimensional sphere such that $C_N = \Theta(1/\sqrt{d})$, refer to Appendix B.

Besides, we also empirically verify our assumption on MNIST (Lecun et al., 1998) with ten digits from 0 to 9. We randomly sample $1,000$ data for each digit and calculate the empirical mean (to approximate the expectation) of the two-layer NTK kernel value over these digit pairs. The experimental result is shown in Figure 1. We can see that the confusion matrix usually has a larger diagonal element than its non-diagonal element, which implies that the kernel value on the same class is often larger than that of different classes. This verifies the justification of our assumption.

3.3. Theoretical guarantees

Based on our assumption, we are ready to present our theoretical result that the test error of a ReLU DNN is close to the Bayes-optimal (noise rate).

Theorem 1. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1 with a step size* $\alpha \gtrsim L^{-2}(\log m)^{-5/2}$. *Then under Assumption 1 and 2, for* $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$ *and* $\lambda > 0$*, with probability at*

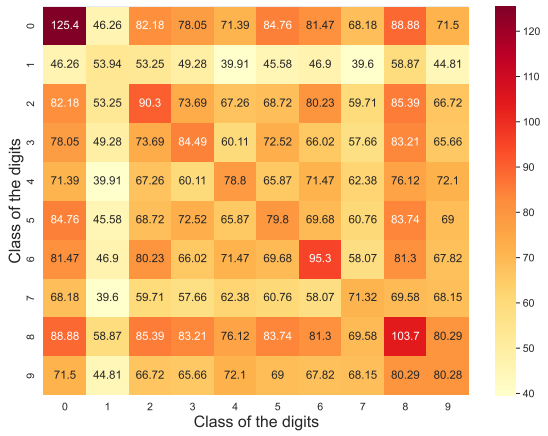

Figure 1. Averaged kernel values among 10 classes in MNIST, where a larger kernel value indicates a higher similarity between two data pairs.

least $1-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$*, we have:*

$$\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\neq \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(n)})))$$
$$\leq \eta+\exp\left(-\lambda\Theta\left(\frac{n\alpha(1-2\eta)C_N}{Lip_{f(\boldsymbol{x};\boldsymbol{W}^{(n)})}}\right)^2\right),$$

where the η is the noise rate defined in Section 3.1 and C_N is defined in Assumption 2.

Remark: Theorem 1 provides the upper bound on the test error rate, including two parts. The first part is the proportion of the wrong labels in the training data. The second part exponentially decreases with the square of the number of training samples n. Also, this term is positively correlated with the Lipschitz constants of DNNs after training, which implies that interpolating more smooth functions leads to a faster convergence rate. We take a closer look at this phenomenon in Section 4, analyzing how neural networks interpolate target functions in a regression setting. Overall, this bound shows that the models overfit the wrong or noisy data on the training set, but still achieve good generalization error on the testing set. This is consistent with previous work on broader settings of benign overfitting that are not limited to classification problems with label noise. For example, various regression problems (Bartlett et al., 2020; Zou et al., 2021; Chatterji and Long, 2022; Koehler et al., 2021; Tsigler and Bartlett, 2020), classification problems of 2-layer networks (Frei et al., 2022; Cao et al., 2022), 2-layer and 3-layer NTK networks (Ju et al., 2021; 2022).

Here we discuss the (nearly) zero-training loss and how the Lipschitz constant affects our error bounds.

SGD can obtain arbitrarily (nearly) zero-training errors on the training set: A lot of work has shown that deep neural networks trained with SGD can obtain zero training error on the training set and perfectly fit any training label in both classification and regression problems with mean squared loss or logistic loss (Du et al., 2019b;a; Chizat et al., 2019; Zou et al., 2020). These results for empirical loss need to be over-parameterized by the condition that $m=\text{poly}(n,L)$. In Appendix D, we provide proof that the loss can be arbitrarily small on the training set under the setting of Section 2. This indicates that when the training data has label noise, the neural network will learn all the noise, that is, overfitting. Combined with the bound of the test error rate in Theorem 1, we can say that the deep neural network has a benign overfitting phenomenon.

Lipschitz constant of the deep neural network: Theorem 1 shows that the convergence rate of the test error rate with the amount of data and is closely related to the Lipschitz constant of the neural network. The Lipschitz constant of DNNs has been widely studied in (Bubeck et al., 2021; Wu et al., 2021; Huang et al., 2021; Nguyen et al., 2021). For example, for ReLU DNNs, if we employ the result of Nguyen et al. (2021, Theorem 6.2): $\text{Lip}_f\lesssim\mathcal{O}\big((2\log m)^{L-1}\big)$, then our bound is

$$\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\neq \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(n)})))$$
$$\lesssim \eta+\exp\left(-\lambda\Theta\Big(\frac{n}{(2\log m)^{L-1}}\Big)^2\right),$$

which leads to a better convergence rate on generalization than the two-layer result Frei et al. (2022).

3.4. Proof sketch of Theorem 1

Let us first introduce a few relevant lemmas.

The first Lemma will follow by establishing a lower bound for the expected normalized margin on clean points, $\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W})]/\text{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}$.

Lemma 1. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1. For any $t\geq 0$, assuming $\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]\geq 0$, then we have:*

$$\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\neq \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(t)})))$$
$$\leq \eta+\exp\left(-\frac{\lambda}{4}\left(\frac{\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]}{Lip_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}}\right)^2\right).$$

We next introduce some structural results concerning the neural network optimization objective. The following lemma states that near initialization, the neural network function is almost linear in terms of its weights.

Lemma 2. *Let $\boldsymbol{W},\boldsymbol{W}'\in\mathcal{B}(\boldsymbol{W}^{(0)},\omega)$ with $\omega=\mathcal{O}(L^{-9/2}(\log m)^{-3})$, for any $\boldsymbol{x}\in\mathbb{R}^d$ that satisfy Assump-*

tion 1, with probability at least $1-\exp(-\Omega(m\omega^2\log m))-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$*, we have:*

$$\begin{aligned}&|f(\boldsymbol{x};\boldsymbol{W})-f(\boldsymbol{x};\boldsymbol{W}')-\langle\nabla f(\boldsymbol{x};\boldsymbol{W}'),\boldsymbol{W}-\boldsymbol{W}'\rangle|\\ \leq&\mathcal{O}(\sqrt{\omega^2L^3m\log m})\sum_{l=1}^{L-1}\|\boldsymbol{W}_l-\boldsymbol{W}_l'\|_2\ .\end{aligned}$$

The following lemma describes the change of $yf(\boldsymbol{x};\boldsymbol{W}^{(t+1)})$ from time t to $t+1$.

Lemma 3. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1. For any* $t\geq 0$ *and* $(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}$ *that satisfy Assumption 1, with* $\omega=\mathcal{O}(L^{-9/2}(\log m)^{-3})$*, with probability at least* $1-\exp(-\Omega(m\omega^2\log m))-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$*, we have:*

$$\begin{aligned}&\widetilde{y}[f(\boldsymbol{x};\boldsymbol{W}^{(t+1)})-f(\boldsymbol{x};\boldsymbol{W}^{(t)})]\\ \geq&\alpha g_i^{(t)}\left\langle\widetilde{y}\nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}),y_i\nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle\\ &-\mathcal{O}(\sqrt{\omega^2L^3m\log m})\sum_{l=1}^{L-1}\left\|\boldsymbol{W}_l^{(t+1)}-\boldsymbol{W}_l^{(t)}\right\|_2\ ,\end{aligned}$$

where $(\boldsymbol{x}_i,y_i)$ *is the random selected training sample at step* $t+1$.

Based on the previous lemmas, we can now derive a lower bound on the normalized margin. Note that this lower bound on the normalized margin in conjunction with Lemma 1 results in the test error bound for the main theorem.

Lemma 4. *Let us define a DNN using Eq.* (1) *and trained by Algorithm 1 with a step size* $\alpha\gtrsim L^{-2}(\log m)^{-5/2}$*. Then under Assumption 1 and 2, for any* $t\geq 0$*,* $\omega\leq\mathcal{O}(L^{-9/2}(\log m)^{-3})$*, with probability at least* $1-\exp(-\Omega(m\omega^2\log m))-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$*, we have:*

$$\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]\geq\Theta\big(t\alpha(1-2\eta)C_N\big)\ .$$

Now, we can prove Theorem 1.

Proof. According to Lemma 1, choosing $t:=n$, we have:

$$\begin{aligned}&\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\neq\operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(n)})))\\ \leq&\eta+\exp\left(-\frac{\lambda}{4}\left(\frac{\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(n)})]}{\operatorname{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(n)})}}\right)^2\right).\end{aligned}$$

Then, by Lemma 4, choosing $t:=n$ and $\alpha\gtrsim L^{-2}(\log m)^{-5/2}$, for $\omega\leq\mathcal{O}(L^{-9/2}(\log m)^{-3})$, with probability at least $1-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:

$$\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(n)})]\geq\Theta\big(n\alpha(1-2\eta)C_N\big)\ .$$

Combine the results, we have:

$$\begin{aligned}&\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\neq\operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(n)})))\\ \leq&\eta+\exp\left(-\lambda\Theta\left(\frac{n\alpha(1-2\eta)C_N}{\operatorname{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(n)})}}\right)^2\right).\end{aligned}$$

□

4. Interpolating Smooth Function by NTK

In this section, we take a closer look at the phenomenon of the relationship between Lipschitz constants of DNNs and convergence rate in Theorem 1, and accordingly analyze how neural networks interpolate smooth ground-truth functions in a regression setting from an approximation theory view (Cucker and Zhou, 2007). In this section, we will also follow the NTK initialization (Allen-Zhu et al., 2019b). For other different initialization (Cao and Gu, 2019; Arora et al., 2019b), similar conclusions will apply.

To be specific, let $X\subseteq\mathbb{R}^d$ be an input space, and $Y\subseteq\mathbb{R}$ be the output space, $f_\rho:X\to Y$ be the ground-truth function, that is smooth in RKHS, described by the source condition in Section 4.1. We assume that the data $(\boldsymbol{x},y)$ is sampled from an unknown distribution ρ, and ρ_X is the marginal distribution of ρ over X. The label is generated through $y=f_\rho(\boldsymbol{x})+\epsilon$, where ϵ is the noise. Accordingly, denote $L^2_{\rho_X}$ as the ρ_X weighted L^2-space and its norm $\|f\|^2_{L^2_{\rho_X}}=\int_X|f(\boldsymbol{x})|^2\,d\rho_X(\boldsymbol{x})$, we are interested in the excess risk $\|f(\boldsymbol{x};\boldsymbol{W}^{(t)})-f_\rho\|^2_{L^2_{\rho_X}}$, which describes how neural networks interpolate/approximate a smooth ground-truth function in a certain space (Cucker and Zhou, 2007; Bach, 2017). In this section, we use the standard NTK network and initialization, which is equivalent to Arora et al. (2019b) using the initialization with standard normal distribution together with the scale factor after each layer for the training dynamics.

4.1. Assumptions

We make the following assumptions:

Assumption 3 (High dimensionality (Liang and Rakhlin, 2020; Liu et al., 2021))**.** There exists universal constants $c_1,c_2\in(0,\infty)$ such that $c_1\leq\frac{d}{n}\leq c_2$.

Assumption 4 (Noise condition (Liang and Rakhlin, 2020; Liu et al., 2021))**.** There exists a $\sigma_\epsilon>0$ such that $\mathbb{E}[(f_\rho(\boldsymbol{x})-y)^2|\boldsymbol{x}]\leq\sigma_\epsilon^2$, almost surely.

Assumption 5 (Geifman et al. (2020); Chen and Xu (2021))**.** We assume that $\boldsymbol{x}_i,\forall i\in[n]$ are i.i.d. sampled from a uniform distribution on the d-dimensional unit sphere. i.e. $\boldsymbol{x}\sim\text{Unif}(\mathbb{S}^{d-1}(1))$, $\mathbb{S}^{d-1}(1):=\big\{\boldsymbol{x}\in\mathbb{R}^d\big|\,\|\boldsymbol{x}\|_2=1\big\}$.

Remark: The i.i.d unit sphere data assumption implies that the data $\boldsymbol{x}$ is isotropic *asymptotically* under our high-dimensional setting (Wainwright, 2019), i.e., $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{I}_d/d$. In fact, there is an alternative way in our proof by directly assuming $\boldsymbol{x}$ is sub-Gaussian and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{I}_d/d$.

Assumption 6 (Existence of f_ρ)**.** We assume the ground-truth function $f_\rho \in \mathcal{H}_{\text{NTK}}$, where $\mathcal{H}_{\text{NTK}}$ is the RKHS associated with the limiting NTK kernel.

Remark: This is a standard assumption in learning theory by assuming that the ground-truth function f_ρ is indeed realizable (Cucker and Zhou, 2007; Rudi and Rosasco, 2017; Liu et al., 2021). This assumption is a special case of the source condition (Cucker and Zhou, 2007) by taking certain values and can be easily extended to non-RKHS spaces or teacher-student settings (Hinton et al., 2015). For ease of analysis, we directly assume the ground-truth function in an RKHS.

4.2. Kernel regression estimator

Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ be a matrix, each column of which is the input of one training sample, $\boldsymbol{\epsilon} \in \mathbb{R}^{n\times 1}$ be the noise in the output of training data. The empirical risk minimization (ERM) is defined with the squared loss:

$$\hat{f}_{\boldsymbol{z}} = \arg\min_{f\in\mathcal{F}} \left\{ \frac{1}{2n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2 \right\}, \tag{4}$$

where the hypothesis space $\mathcal{F}$ can be defined properly. For example, if $\mathcal{F}$ is a RKHS $\mathcal{H}_{\boldsymbol{K}}$, Eq. (4) is formulated as a kernel regression. Denoting that $\boldsymbol{K}_{\texttt{ker}}(\boldsymbol{x}, \boldsymbol{X}) = [\boldsymbol{K}_{\texttt{ker}}(\boldsymbol{x}, \boldsymbol{x}_1), \boldsymbol{K}_{\texttt{ker}}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \boldsymbol{K}_{\texttt{ker}}(\boldsymbol{x}, \boldsymbol{x}_n)]^\top \in \mathbb{R}^n$, the closed form of the kernel regression estimator to Eq. (4) is given by:

$$f_{\texttt{ker}} = \boldsymbol{K}_{\texttt{ker}}(\boldsymbol{x}, \boldsymbol{X})^\top \boldsymbol{K}_{\texttt{ker}}^{-1}\boldsymbol{y}\,.$$

If we use a neural network as in Eq. (1) to solve Eq. (4), the corresponding hypothesis space $\mathcal{F}_{\texttt{nn}}$ is:

$$\mathcal{F}_{\texttt{nn}} := \Big\{ f(\boldsymbol{x}; \boldsymbol{W}) \text{ admits Eq. (1)} : \boldsymbol{x} \sim \text{Unif}(\mathbb{S}^{d-1}(1)), \\ \boldsymbol{W} \in \mathbb{R}^{m\times d} \times (\mathbb{R}^{m\times m})^{L-2} \times \mathbb{R}^{1\times m} \Big\},$$

which implies:

$$f_{\texttt{nn}} = \arg\min_{f\in\mathcal{F}_{\texttt{nn}}} \left\{ \frac{1}{2n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i; \boldsymbol{W}) - y_i)^2 \right\}.$$

In addition to the neural tangent kernel mentioned earlier Eq. (3), we will present some examples of the positive definite kernels to be studied in this paper.

Dot product kernel (Ghosh et al., 2022): The dot product kernels have the following forms:

$$K_{\texttt{dot}}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = k(\langle \boldsymbol{x}, \widetilde{\boldsymbol{x}}\rangle), \quad \forall \boldsymbol{x}, \widetilde{\boldsymbol{x}} \in \mathbb{S}^{d-1}(1)\,,$$

for some function $k : [-1, 1] \to \mathbb{R}$.

Laplace kernel (Geifman et al., 2020): The Laplace kernel is defined as:

$$K_{\texttt{Laplace}}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = e^{-c\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|_2}, \quad c > 0\,.$$

According to Assumption 5, we have:

$$K_{\texttt{Laplace}}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = e^{-c\sqrt{2(1-\boldsymbol{x}^\top\widetilde{\boldsymbol{x}})}} = e^{-\tilde{c}\sqrt{1-u}} \triangleq K_{\texttt{dot}}(u)\,, \tag{5}$$

where $u = \langle \boldsymbol{x}, \widetilde{\boldsymbol{x}}\rangle$.

4.3. The minimum eigenvalue of NTK matrix under the high dimensional setting

We are now ready to state the main result of a deep over-parameterized NTK network. We first provide the lower bounds of the minimum eigenvalue of NTK under the high dimensional setting.

Recall that the Neural Tangent Kernel (NTK) (Jacot et al., 2018) matrix of neural network f is defined in Eq. (3). When we focus on the infinite-width setting ($m \to \infty$), the NTK matrix for a neural network Eq. (1) is derived by the following regular chain rule.

Lemma 5 (Adapted from Lemma 3.1 in Nguyen et al. (2021))**.** *For any $l \in [3, L]$ and $s \in [2, L]$, denote*

$$\begin{aligned}
&\boldsymbol{G}^{(1)} = \boldsymbol{X}\boldsymbol{X}^\top\,,\\
&\boldsymbol{G}^{(2)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_d)}[\sigma_1(\boldsymbol{X}\boldsymbol{w})\sigma_1(\boldsymbol{X}\boldsymbol{w})^\top]\,,\\
&\boldsymbol{G}^{(l)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_N)}[\sigma_{l-1}(\sqrt{\boldsymbol{G}^{(l-1)}}\boldsymbol{w})\sigma_{l-1}(\sqrt{\boldsymbol{G}^{(l-1)}}\boldsymbol{w})^\top]\,,\\
&\dot{\boldsymbol{G}}^{(s)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_N)}[\sigma'_{s-1}(\sqrt{\boldsymbol{G}^{(s-1)}}\boldsymbol{w})\sigma'_{s-1}(\sqrt{\boldsymbol{G}^{(s-1)}}\boldsymbol{w})^\top]\,.
\end{aligned}$$

Then, the NTK for a L-layer neural network defined in Eq. (1) *can be written as*

$$\boldsymbol{K}_{\texttt{NTK}} = \boldsymbol{G}^{(L)} + \sum_{l=1}^{L-1} \boldsymbol{G}^{(l)} \circ \dot{\boldsymbol{G}}^{(l+1)} \circ \dot{\boldsymbol{G}}^{(l+2)} \circ \cdots \circ \dot{\boldsymbol{G}}^{(L)}\,,$$

where $\circ$ represents the element-wise Hadamard product.

Based on the formulation of NTK, we are ready to present the estimation of the minimum eigenvalue of NTK.

Theorem 2 (Minimum eigenvalue of NTK matrix)**.** *For a DNN defined by Eq.* (1)*, let $\boldsymbol{K}_{\texttt{NTK}}$ be the limiting NTK recursively defined in Lemma 5 and let λ_0 be the minimum*

eigenvalue of $\boldsymbol{K}_{\texttt{NTK}}$. Then, under Assumption 3 and 5, with probability at least $1-2e^{-n}$, we obtain that:

$$\lambda_0 \geq \begin{cases} 2\mu_1^2\frac{n}{d}\left(\frac{3}{4}-\frac{c}{4}\sqrt{\frac{d}{n}}\right)^2, & \text{if} \quad n \geq d, \\ 2\mu_1^2\frac{n}{d}\left(\sqrt{\frac{d}{n}}-\frac{c+6}{4}\right)^2, & \text{if} \quad n < d\,, \end{cases}$$

where we have an absolute constant $c = 2^{3.5}\sqrt{\log(9)} \approx 16.77$ and μ_1 is the 1-st Hermite coefficient of the ReLU activation function.

Remark: This theorem provides the upper bound of the minimum eigenvalue of the NTK matrix of the infinite-width neural network under the high-dimensional setting and can be easily extended to the finite-width setting. Note that our result under the high dimensional setting is different from previous work under the fixed d setting in Zhu et al. (2022). If we fix d and vary n from small to large, there exists a phase transition when n increases, see Table 1. If $n \ll d$, the lower bound of the minimum eigenvalue of NTK $\lambda_0 \geqslant \Omega(1)$, then when n increases, we can see that λ_0 decreases to a bottom and then increases until $n := d$. In the $n \geq d$ regime, there exists a similar trend to that of the $n \leq d$ regime: firstly decreasing and then increasing. When $n \gg d$, we have $\lambda_0 \geqslant \Omega(n)$.

Table 1. The trend of the bound with respect to n under different range of n values and a fixed d.

Range of n	Trend w.r.t. n	Limit bound
$n \ll d$	-	$2\mu_1^2$
$0 \leq n \leq (\frac{4}{c+6})^2 d$	$\searrow$	-
$(\frac{4}{c+6})^2 \leq n \leq d$	$\nearrow$	-
$d \leq n \leq \frac{c^2}{9}d$	$\searrow$	-
$\frac{c^2}{9}d \leq n$	$\nearrow$	-
$n \gg d$	-	$\frac{9}{8}\mu_1^2\frac{n}{d}$

4.4. Generalization error bound

Based on the aforementioned upper and lower bounds of the minimum eigenvalue of NTK under the high dimensional setting, we establish the relationship between the minimum eigenvalue of NTK and the generalization error of DNNs. We provide a bound on the norm of the difference between the network output and ground truth function under the weighted $L^2_{\rho_X}$ space.

Theorem 3 (An upper bound on the generalization error for deep over-parameterized NTK network)**.** *Let $\theta \in (0, 1/2]$, δ and c are some non-negative constants, the ground-truth function f_ρ lies in a RKHS by Assumption 6 and d large enough, under Assumption 3, 4 and 5, suppose that, $\omega \leq poly(1/n, \lambda_0, 1/L, 1/\log(m), \epsilon, 1/\log(1/\delta'), \kappa)$, $m \geq poly(1/\omega)$ and $\kappa = \mathcal{O}(\frac{\epsilon}{\log(n/\delta')})$. then for any given $\varepsilon > 0$, with high probability, we have:*

$$\mathbb{E}\left\|f_{\texttt{nn}} - f_\rho\right\|^2_{L^2_{\rho_X}} \lesssim \mathcal{O}\Bigg(n^{-\theta}\log^4(\frac{2}{\delta}) + \frac{\sigma_\epsilon^2}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}} + \frac{\sigma_\epsilon^2\log^{2+4\varepsilon}d}{d^{4\theta-1}} + \epsilon^2 + \frac{n}{\lambda_0^2}\omega^{2/3}L^5 m\log m + \frac{n^3}{\lambda_0^6\kappa^2}\Bigg),$$

where the λ_0 satisfies Theorem 2 and the effective dimension $\mathcal{N}_{\widetilde{\boldsymbol{X}}}$ is defined as:

$$\mathcal{N}_{\widetilde{\boldsymbol{X}}} := \sum_{i=0}^{n-1}\frac{\lambda_i(\widetilde{\boldsymbol{X}})}{(\lambda_i(\widetilde{\boldsymbol{X}}+\gamma))^2},$$

with $\widetilde{\boldsymbol{X}} := \beta\boldsymbol{X}\boldsymbol{X}^\top/d + \alpha\mathbf{1}\mathbf{1}^\top$ for some non-negative constants α, β, γ.

Remark:

This theorem builds a connection between DNNs and kernel methods in benign overfitting and gives the upper bound of the generalization error of the NTK network in the high-dimensional setting. To be specific,the first term is the upper bound of the bias of NTK regression, which decreases as the number of data increases. The second and the third terms jointly form the upper bound of variance of NTK regression, which is mainly affected by the effective dimension (eigenvalue decay) of the data. The fourth term is the error introduced by the initialization of the NTK neural network. The fifth and the sixth term reflect the difference between the finite-width NTK network and the infinite-width NTK network (neural tangent kernel regression), which decreases with the increase of the minimum eigenvalue of the NTK network we provide in Theorem 2.

Under refined assumptions, e.g., source condition, capacity condition (Cucker and Zhou, 2007), we can achieve $\theta = 1$ for a better convergence rate. Regarding the convergence properties, we need to make the following discussion.

The three non-negative constants α, β, and γ are related to the linearization of the kernel matrix in the high dimension setting, refer to Liu et al. (2021) for details. Here we give the following discussion on $\mathcal{N}_{\widetilde{\boldsymbol{X}}}$ under three typical eigenvalue decay of $\boldsymbol{X}\boldsymbol{X}^\top$ cases, and then discuss our generalization bound. Note that when $n > d$, the sample matrix $\boldsymbol{X}\boldsymbol{X}^\top$ has at most d eigenvalues, so we can directly have $\mathcal{N}_{\widetilde{\boldsymbol{X}}} \leq \mathcal{O}(d)$. Accordingly, here we present the results on the $n < d$ case.

- **Harmonic decay:** $\lambda_i(\widetilde{\boldsymbol{X}}) \propto n/i, \forall i \in \{1, 2, \ldots, r_\star\}$ and $\lambda_i(\widetilde{\boldsymbol{X}}) = 0, \forall i \in \{r_\star + 1, \ldots, n\}$.

We have: $\mathcal{N}_{\boldsymbol{X}} = \mathcal{O}(n)$, then the term $\frac{\sigma_\epsilon^2}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}} \leq \mathcal{O}(\frac{\sigma_\epsilon^2}{d}n)$.

- **Polynomial decay:** $\lambda_i(\widetilde{\boldsymbol{X}}) \propto ni^{-2a}$ with $a > 1/2, \forall i \in \{1, 2, \ldots, r_\star\}$ and $\lambda_i(\widetilde{\boldsymbol{X}}) = 0, \forall i \in \{r_\star + 1, \ldots, n\}$.

  We have: $\mathcal{N}_{\boldsymbol{X}} = \mathcal{O}(n^{1/2a})$, then the term $\frac{\sigma_\epsilon^2}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}} \leq \mathcal{O}(\frac{\sigma_\epsilon^2}{d}n^{1/2a}) \leq \mathcal{O}(\frac{\sigma_\epsilon^2}{d}n)$.

- **Exponential decay:** $\lambda_i(\widetilde{\boldsymbol{X}}) \propto ne^{-ai}$ with $a > 0, \forall i \in \{1, 2, \ldots, r_\star\}$ and $\lambda_i(\widetilde{\boldsymbol{X}}) = 0, \forall i \in \{r_\star + 1, \ldots, n\}$.

  We have: $\mathcal{N}_{\widetilde{\boldsymbol{X}}} = \frac{1}{a}\big(\frac{1}{\gamma + n\exp(-a(r_\star+1))} - \frac{1}{\gamma + n\exp(-a)}\big)$, then the term $\frac{\sigma_\epsilon^2}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}} \leq \mathcal{O}(\frac{\sigma_\epsilon^2}{d}\frac{e^{ar_\star}}{n})$.

Based on our discussion on the eigenvalue decay, we are ready to discuss our generalization bound in Theorem 3. When n, d are comparably large enough, e.g., $n \geq \frac{c^2}{9}d$, in this case, we have $\mathcal{N}_{\widetilde{\boldsymbol{X}}} \leq \mathcal{O}(d)$, according to Theorem 2 for λ_0, three terms $n^{-\theta}\log^4(\frac{2}{\delta})$, $\frac{n}{\lambda_0^2}\omega^{2/3}L^5 m \log m$ and $\frac{n^3}{\lambda_0^6\kappa^2}$ convergence to 0. The term $\frac{\sigma_\epsilon^2 \log^{2+4\varepsilon} d}{d^{4\theta-1}}$ also converges to 0 for a large enough d. Accordingly, for large enough n and d, we have

$$\mathbb{E}\left\|f_{\text{nn}} - f_\rho\right\|_{L^2_{\rho_X}}^2 \lesssim \mathcal{O}\big(\sigma_\epsilon^2 + \epsilon^2\big)\,, w.h.p\,,$$

which show that the bound only depends on the noise and random initialization term, and thus coincide with previous work on benign overfitting (Frei et al., 2022; Cao et al., 2022; Ju et al., 2022; Arora et al., 2019b).

Besides, we can also find that the phase transition exists in the minimum eigenvalue λ_0 and the effective dimension $\mathcal{N}_{\widetilde{\boldsymbol{X}}}$ from $n < d$ and $n > d$. This also leads to a phase transition on the excess risk. Roughly speaking, the excess risk firstly increases with n until $n := d$ and then decreases with n when $n > d$.

5. Conclusion and limitations

In this work, we present a theoretical analysis of benign overfitting for deep ReLU NNs. For binary classification, our results demonstrate that DNNs under the lazy training regime obtain the *Bayes-optimal* test error with a better convergence rate than Frei et al. (2022). For regression, our results exhibit a phase transition on the excess risk from $n < d$ to $n > d$, of which the excess risk converges to a constant order $\mathcal{O}(1)$ that only depends on label noise and initialization noise. The above two results theoretically validate the benign overfitting of DNNs.

We need to mention that, our results are only applicable to lazy training regimes and appear difficult to be extended to the non-lazy training regime, commonly used in practice. This is because DNN cannot be linearly approximated well under the non-lazy training regime. We leave this as a future work. Besides, an interesting direction is, extending our data-generating distribution assumption from log-concave distribution to a general one, as we require it to ensure the output of neural networks is sub-Gaussian.

Acknowledgements

We are thankful to the reviewers for providing constructive feedback. This work was supported by Hasler Foundation Program: Hasler Responsible AI (project number 21043). This work was supported by SNF project – Deep Optimisation of the Swiss National Science Foundation (SNSF) under grant number 200021_205011. This work was supported by Zeiss. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 - time-data).

References

Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems (NeurIPS)*, 2019a.

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning (ICML)*, 2019b.

S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, 2019a.

S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in neural information processing systems (NeurIPS)*, 2019b.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 2017.

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

A. Bietti and J. Mairal. On the inductive bias of neural tangent kernels. In *Advances in neural information processing systems (NeurIPS)*, 2019.

S. Bombari, M. H. Amani, and M. Mondelli. Memorization and optimization in deep neural networks with minimum

over-parameterization. In *Advances in neural information processing systems (NeurIPS)*, 2022.

A. Brutzkus, A. Globerson, E. Malach, and S. Shalev-Shwartz. Sgd learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations (ICLR)*, 2017.

S. Bubeck, Y. Li, and D. M. Nagaraj. A law of robustness for two-layers neural networks. In *Conference on Learning Theory*, 2021.

T. Cai, J. Fan, and T. Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 2013.

Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2019.

Y. Cao and Q. Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *AAAI Conference on Artificial Intelligence*, 2020.

Y. Cao, Z. Chen, M. Belkin, and Q. Gu. Benign overfitting in two-layer convolutional neural networks, 2022.

N. S. Chatterji and P. M. Long. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *Journal of Machine Learning Research*, 2021.

N. S. Chatterji and P. M. Long. Foolish crowds support benign overfitting. *Journal of Machine Learning Research*, 2022.

N. S. Chatterji, P. M. Long, and P. L. Bartlett. The interplay between implicit bias and benign overfitting in two-layer linear networks, 2021.

L. Chen and S. Xu. Deep neural tangent kernel and laplace kernel have the same $\{\text{rkhs}\}$. In *International Conference on Learning Representations (ICLR)*, 2021.

Z. Chen, Y. Cao, D. Zou, and Q. Gu. How much over-parameterization is sufficient to learn deep ReLU networks? In *International Conference on Learning Representations (ICLR)*, 2020.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in neural information processing systems (NeurIPS)*, 2019.

F. Cucker and D. X. Zhou. *Learning theory: an approximation theory viewpoint*. Cambridge University Press, 2007.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning (ICML)*, 2019a.

S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019b.

W. E, C. Ma, and L. Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics*, 2020.

S. Frei, N. S. Chatterji, and P. Bartlett. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. In *Conference on Learning Theory*, 2022.

A. Geifman, A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and B. Ronen. On the similarity between the laplace and neural tangent kernels. In *Advances in neural information processing systems (NeurIPS)*, 2020.

N. Ghosh, S. Mei, and B. Yu. The three stages of learning dynamics in high-dimensional kernel methods. In *International Conference on Learning Representations (ICLR)*, 2022.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

H. Huang, Y. Wang, S. M. Erfani, Q. Gu, J. Bailey, and X. Ma. Exploring architectural ingredients of adversarially robust deep neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2021.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2018.

P. Ju, X. Lin, and N. Shroff. On the generalization power of overfitted two-layer neural tangent kernel models. In *International Conference on Machine Learning (ICML)*, 2021.

P. Ju, X. Lin, and N. Shroff. On the generalization power of the overfitted three-layer neural tangent kernel model. In *Advances in neural information processing systems (NeurIPS)*, 2022.

F. Koehler, L. Zhou, D. J. Sutherland, and N. Srebro. Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting. In *Advances in neural information processing systems (NeurIPS)*, 2021.

I. Kuzborskij, C. Szepesvari, O. Rivasplata, A. Rannen-Triki, and R. Pascanu. On the role of optimization in double descent: A least squares study. In *Advances in neural information processing systems (NeurIPS)*, 2021.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

Z. Li, Z.-H. Zhou, and A. Gretton. Towards an understanding of benign overfitting in neural networks, 2021.

T. Liang and A. Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 2020.

T. Liang, A. Rakhlin, and X. Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels, 2019.

F. Liu, Z. Liao, and J. Suykens. Kernel regression in high dimensions: Refined analysis beyond double descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

N. R. Mallinar, J. B. Simon, A. Abedsoltan, P. Pandit, M. Belkin, and P. Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. In *Advances in neural information processing systems (NeurIPS)*, 2022.

S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 2022.

A. Montanari, F. Ruan, Y. Sohn, and J. Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime, 2019.

Q. Nguyen, M. Mondelli, and G. F. Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning (ICML)*, 2021.

A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in neural information processing systems (NeurIPS)*, 2017.

B. Schölkopf, A. J. Smola, F. Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

A. Tsigler and P. L. Bartlett. Benign overfitting in ridge regression, 2020.

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

G. Wang, K. Donhauser, and F. Yang. Tight bounds for minimum ell_1-norm interpolation of noisy data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

K. Wang, V. Muthukumar, and C. Thrampoulidis. Benign overfitting in multiclass classification: All roads lead to interpolation. In *Advances in neural information processing systems (NeurIPS)*, 2021.

B. Wu, J. Chen, D. Cai, X. He, and Q. Gu. Do wider neural networks really help adversarial robustness? In *Advances in neural information processing systems (NeurIPS)*, 2021.

X. Xu and Y. Gu. Benign overfitting of non-smooth neural networks beyond lazy training. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Z. Zhu, F. Liu, G. Chrysos, and V. Cevher. Generalization properties of NAS under activation and skip connection search. In *Advances in neural information processing systems (NeurIPS)*, 2022.

D. Zou, Y. Cao, D. Zhou, and Q. Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 2020.

D. Zou, J. Wu, V. Braverman, Q. Gu, and S. Kakade. Benign overfitting of constant-stepsize sgd for linear regression. In *Conference on Learning Theory*, 2021.

Appendix introduction

The Appendix is organized as follows:

- In Appendix A, we state the symbols and notation used in this paper.
- In Appendix B, we provide a example to verify the Assumption 2.
- In Appendix C, we provide the proof for the lemmas in Section 3.4.
- In Appendix D, we provide the theorem and its proof of the optimization result for the classification problem.
- In Appendix E, we provide the proof for the Theorem 2.
- In Appendix F, we provide the proof for the Theorem 3.

A. Symbols and Notation

In the paper, vectors are indicated with bold small letters and matrices with bold capital letters. To facilitate the understanding of our work, we include some core symbols and notation in Table 2.

Table 2. Core symbols and notations used in this project.

Symbol	Dimension(s)	Definition
$\mathcal{N}(\mu, \sigma)$	-	Gaussian distribution of mean μ and variance σ
$\mathcal{B}(\boldsymbol{W}, \cdot)$	-	Neighborhood of matrix $\boldsymbol{W}$
$\lambda(\boldsymbol{M})$	-	Eigenvalues of matrices $\boldsymbol{M}$
$\lambda_{\min}(\boldsymbol{M})$	-	Minimum eigenvalue of matrices $\boldsymbol{M}$
λ_0	-	Minimum eigenvalue NTK matrix
$\phi(x) = \max(0, x)$	-	ReLU activation function for scalar
$\phi(\boldsymbol{v}) = (\phi(v_1), \ldots, \phi(v_m))$	-	ReLU activation function for vectors
$1\{A\}$	-	Indicator function for event A
$\mathrm{sgn}(\cdot)$	-	Sign function
$\mathrm{Lip}_{(\cdot)}$	-	Lipschitz constant of a function
n	-	Size of the dataset
d	-	Input size of the network
L	-	Depth of the network
m	-	Width of intermediate layer
$\boldsymbol{x}_i$	$\mathbb{R}^d$	The i-th data point
y_i	$\{\pm 1\}$	The i-th clean label
$\widetilde{y}_i$	$\{\pm 1\}$	The i-th training label
P	-	Clean data distribution that $(\boldsymbol{x}_i, y_i) \sim P$
$\widetilde{P}$	-	Training data distribution that $(\boldsymbol{x}_i, \widetilde{y}_i) \sim \widetilde{P}$
P_{clust}	-	Cluster distribution for generate data
$\mathcal{C}$	-	A subset of training data for clean labels
$\mathcal{C}'$	-	A subset of training data for noisy labels
α	-	Step size of SGD
η	-	Noise rate
ω	-	Lazy training rate
λ	-	Strongly log-concave rate of distribution P_{clust}
ℓ	-	Logistic loss function
$\hat{L}$	-	Empirical risks
$g_i^{(t)}$	-	Value of g for input $\boldsymbol{x}_i$ at time t, where $g(z) := -\ell'(z)$
$f_i^{(t)}$	-	Output of neural network for input $\boldsymbol{x}_i$ at time t
f_ρ	-	Ground-truth function
$\boldsymbol{K}_{\texttt{NTK}}$, $\boldsymbol{K}_{\texttt{Laplace}}$, $\boldsymbol{K}_{\texttt{dot}}$	$\mathbb{R}^{n\times n}$	Three different kernel matrices
$\mathcal{H}_{\texttt{NTK}}$, $\mathcal{H}_{\texttt{Laplace}}$, $\mathcal{H}_{\texttt{dot}}$	-	RKHS of the kernel
$\boldsymbol{W}_1$	$\mathbb{R}^{m\times d}$	Weight matrix for the input layer
$\boldsymbol{W}_l$	$\mathbb{R}^{m\times m}$	Weight matrix for the l-th hidden layer
$\boldsymbol{W}_L$	$\mathbb{R}^{1\times m}$	Weight matrix for the output layer
$\boldsymbol{h}_{i,l}$	$\mathbb{R}^m$	The l-th layer activation for input $\boldsymbol{x}_i$
$\boldsymbol{D}_{i,l}$	$\mathbb{R}^{m\times m}$	Diagonal sign matrix of l-th layer input $\boldsymbol{x}_i$

B. A example of Assumption 2

Proposition 1. *For two different data samples* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d \sim \widetilde{P} := \mathit{Unif}(\mathbb{S}^{d-1}(C_{\mathit{norm}}))$, $y = \begin{cases} 1 & \text{if } x_i > 0, \forall i \in [d], \\ -1 & \text{if } x_i \le 0, \forall i \in [d], \end{cases}$ *the* 2*-layer NTK kernel defined in Eq.* (3) *with* $L = 2$ *satisfy that:*

$$\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1 = \widetilde{y}_2\big] - \mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1 \neq \widetilde{y}_2\big] \ge C_N = \Theta(1)\,.$$

Proof. According to Bietti and Mairal (2019), we have for 2-layer neural network:

$$K_{\texttt{NTK}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \kappa\left(\frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2}\right),$$

where $\kappa(u) := u\kappa_0(u) + \kappa_1(u)$ with $\kappa_0(u) := \frac{1}{\pi}(\pi - \arccos(u))$ and $\kappa_1(u) := \frac{1}{\pi}(u(\pi - \arccos(u)) + \sqrt{1-u^2})$. Denote $\cos(\theta) := \frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2}$, we have:

$$\begin{aligned}
K_{\texttt{NTK}}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \kappa\big(\frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle}{\|\boldsymbol{x}_1\|_1 \|\boldsymbol{x}_2\|_2}\big) \\
&= \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \kappa\big(\cos(\theta)\big) \\
&= \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \big(\cos(\theta)\kappa_0(\cos(\theta)) + \kappa_1(\cos(\theta))\big) \\
&= \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \big(\cos(\theta)\frac{1}{\pi}(\pi-\theta) + \frac{1}{\pi}(\cos(\theta)(\pi-\theta) + |\sin(\theta)|)\big) \\
&= \|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2 \big(\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}\big).
\end{aligned}$$

According to $y = \begin{cases} 1 & \text{if } x_1 > 0 \\ -1 & \text{if } x_1 \le 0 \end{cases}$, we have if $\boldsymbol{x}_3 = -\boldsymbol{x}_2$ then $y_3 = -y_2$ and $\cos(\theta') := \frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_3\rangle}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_3\|_2} = -\cos(\theta)$, so $\theta' = \pi - \theta$ and$|\sin(\theta')| = |\sin(\theta)|$.

According to the symmetry of $\widetilde{P}$, we can compute that:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}|\widetilde{y}_1 = \widetilde{y}_2\big] \\
=&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_3,\widetilde{y}_3)\sim\widetilde{P}}\big[\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}|\widetilde{y}_1 = \widetilde{y}_3\big] \\
=&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_3,\widetilde{y}_3)\sim\widetilde{P}}\big[-\cos(\theta')\frac{2}{\pi}\theta' + \frac{|\sin(\theta')|}{\pi}|\widetilde{y}_1 = \widetilde{y}_3\big]
\end{aligned}$$

According to the isotropy of $\widetilde{P}$, we have:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1 = \widetilde{y}_2\big] - \mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1 \neq \widetilde{y}_2\big] \\
=&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1)\sim\widetilde{P}}\|\boldsymbol{x}_1\|_2\,\mathbb{E}_{(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\|\boldsymbol{x}_2\|_2 \\
\times&\Big(\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}|\widetilde{y}_1 = \widetilde{y}_2\big] - \mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}|\widetilde{y}_1 \neq \widetilde{y}_2\big]\Big) \\
=&\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1)\sim\widetilde{P}}\|\boldsymbol{x}_1\|_2\,\mathbb{E}_{(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\|\boldsymbol{x}_2\|_2 \\
\times&\Big(\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)\frac{2}{\pi}(\pi-\theta) + \frac{|\sin(\theta)|}{\pi}|\widetilde{y}_1 = \widetilde{y}_2\big] - \mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_3,\widetilde{y}_3)\sim\widetilde{P}}\big[-\cos(\theta')\frac{2}{\pi}\theta' + \frac{|\sin(\theta')|}{\pi}|\widetilde{y}_1 = \widetilde{y}_3\big]\Big) \\
=&2\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1)\sim\widetilde{P}}\|\boldsymbol{x}_1\|_2\,\mathbb{E}_{(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\|\boldsymbol{x}_2\|_2\,\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)|\widetilde{y}_1 = \widetilde{y}_2\big] \\
=&2C_{\text{norm}}^2\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[\cos(\theta)|\widetilde{y}_1 = \widetilde{y}_2\big] \\
\geq&2C_{\text{norm}}^2\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}|\cos(\theta)|,
\end{aligned}$$

where the last inequality holds by the fact that the inner product of two data points from the same class is always non-negative in our problem setting. We know that the distribution of the angle between the vectors uniformly distributed on a d-dimensional sphere is (Cai et al., 2013):

$$p(\theta) = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})\sqrt{\pi}}\sin^{d-2}\theta, \quad \theta \in [0,\pi].$$

Then we use substitution that $x = \cos\theta$, then we have the distribution of the cosine of the angle between the vectors uniformly distributed on a d-dimensional sphere is:

$$p(x) = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})\sqrt{\pi}}(1-x^2)^{(d-3)/2}, \quad x \in [-1,1]\,.$$

Then we have:

$$\begin{aligned}
\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\,|\cos(\theta)| &= \int_{-1}^{1}|x|\,p(x)\mathrm{d}x \\
&= 2\frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})\sqrt{\pi}}\int_0^1 x(1-x^2)^{(d-3)/2}\mathrm{d}x \\
&= \frac{2\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})\sqrt{\pi}(d-1)} \\
&= \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d+1}{2})\sqrt{\pi}}\,.
\end{aligned}$$

Then, we have:

$$\mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\mathtt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1=\widetilde{y}_2\big] - \mathbb{E}_{(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}}\big[K_{\mathtt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)|\widetilde{y}_1\neq\widetilde{y}_2\big] \geq \frac{2C_{\mathrm{norm}}^2\Gamma(\frac{d}{2})}{\Gamma(\frac{d+1}{2})\sqrt{\pi}} = \Theta\left(\frac{1}{\sqrt{d}}\right).$$

When we fix the data dimension d, it is a constant order. □

C. Proof for Lemmas in Section 3.4

C.1. Proof of Lemma 1

Let us restate Lemma 1 as below:

Lemma 1. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1. For any $t \geq 0$, assuming $\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})] \geq 0$, then we have:*

$$\begin{aligned}
&\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y \neq \mathrm{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(t)}))) \\
\leq&\eta + \exp\left(-\frac{\lambda}{4}\left(\frac{\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]}{\mathit{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}}\right)^2\right).
\end{aligned}$$

Proof. According to the proof of Chatterji and Long (2021, Lemma 9) and Frei et al. (2022, Lemma 3), we have:

$$\begin{aligned}
\mathbb{P}_{(\boldsymbol{x},y)\sim P}(y \neq \mathrm{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(t)}))) &= \mathbb{P}_{(\boldsymbol{x},y)\sim P}(y\mathrm{sgn}(f(\boldsymbol{x};\boldsymbol{W}^{(t)})) < 0) \\
&\leq \eta + \mathbb{P}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}(\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)}) < 0)\,.
\end{aligned} \tag{6}$$

Denoting the Lipschitz constant of the neural network as $\mathrm{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}$, since P_{clust} is λ-strongly log-concave, then according to Wainwright (2019, Theorem 3.16), for any $t > 0$, we have:

$$\mathbb{P}(|\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)}) - \mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]| \geq t) \leq 2\exp\left(-\frac{\lambda}{4}\left(\frac{t}{\mathrm{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}}\right)^2\right).$$

Choosing $t := \mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]$, we have:

$$\mathbb{P}(|\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)}) - \mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]| \geq \mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]) \leq 2\exp\bigg(-\frac{\lambda}{4}\bigg(\frac{\mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]}{\mathrm{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}}\bigg)^2\bigg),$$

which implies:

$$\begin{aligned}\mathbb{P}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}(\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)}) < 0) &= \mathbb{P}(\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)}) - \mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})] < -\mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]) \\ &\leq \exp\bigg(-\frac{\lambda}{4}\bigg(\frac{\mathbb{E}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]}{\mathrm{Lip}_{f(\boldsymbol{x};\boldsymbol{W}^{(t)})}}\bigg)^2\bigg).\end{aligned}$$

Incorporating it into Eq. (6), we conclude the proof.

□

C.2. Proof of Lemma 2

Let us restate Lemma 2 as below:

Lemma 2. *Let $\boldsymbol{W}, \boldsymbol{W}' \in \mathcal{B}(\boldsymbol{W}^{(0)}, \omega)$ with $\omega = \mathcal{O}(L^{-9/2}(\log m)^{-3})$, for any $\boldsymbol{x} \in \mathbb{R}^d$ that satisfy Assumption 1, with probability at least $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:*

$$\begin{aligned}&|f(\boldsymbol{x};\boldsymbol{W}) - f(\boldsymbol{x};\boldsymbol{W}') - \langle \nabla f(\boldsymbol{x};\boldsymbol{W}'), \boldsymbol{W} - \boldsymbol{W}' \rangle| \\ \leq &\mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \|\boldsymbol{W}_l - \boldsymbol{W}_l'\|_2 .\end{aligned}$$

Proof. We can directly calculate that:

$$\begin{aligned}&f(\boldsymbol{x};\boldsymbol{W}) - f(\boldsymbol{x};\boldsymbol{W}') - \langle \nabla f(\boldsymbol{x};\boldsymbol{W}'), \boldsymbol{W} - \boldsymbol{W}' \rangle \\ =&\boldsymbol{W}_L(\boldsymbol{h}_{i,L-1} - \boldsymbol{h}'_{i,L-1}) - \sum_{l=1}^{L-1} \boldsymbol{W}'_L\bigg(\prod_{r=l+1}^{L-1}(\boldsymbol{D}'_{i,r}\boldsymbol{W}'_r)\bigg)\boldsymbol{D}'_{i,l}(\boldsymbol{W}_l - \boldsymbol{W}'_l)\boldsymbol{h}'_{i,l-1} .\end{aligned}$$

By Allen-Zhu et al. (2019b, Claim 11.2), when $\boldsymbol{W}, \boldsymbol{W}' \in \mathcal{B}(\boldsymbol{W}^{(0)}, \omega)$, there exist diagonal matrices $\boldsymbol{D}''_{i,l} \in \mathbb{R}^{m\times m}$ with entries in $\{+1, -1\}$ such that $\left\|\boldsymbol{D}''_{i,l}\right\|_0 \leq \mathcal{O}(m\omega^{2/3}L)$ and:

$$\boldsymbol{h}_{i,L-1} - \boldsymbol{h}'_{i,L-1} = \sum_{l=1}^{L-1}\bigg(\prod_{r=l+1}^{L-1}(\boldsymbol{D}_{i,r} + \boldsymbol{D}''_{i,r})\boldsymbol{W}_r\bigg)(\boldsymbol{D}_{i,l} + \boldsymbol{D}''_{i,l})(\boldsymbol{W}_l - \boldsymbol{W}'_l)\boldsymbol{h}'_{i,l-1}, \quad \forall i \in [n] .$$

Therefore, we have:

$$\begin{aligned}&f(\boldsymbol{x};\boldsymbol{W}) - f(\boldsymbol{x};\boldsymbol{W}') - \langle \nabla f(\boldsymbol{x};\boldsymbol{W}'), \boldsymbol{W} - \boldsymbol{W}' \rangle \\ =&\sum_{l=1}^{L-1} \boldsymbol{W}_L\bigg(\prod_{r=l+1}^{L-1}(\boldsymbol{D}_{i,r} + \boldsymbol{D}''_{i,r})\boldsymbol{W}_r\bigg)(\boldsymbol{D}_{i,l} + \boldsymbol{D}''_{i,l})(\boldsymbol{W}_l - \boldsymbol{W}'_l)\boldsymbol{h}'_{i,l-1} \\ &- \sum_{l=1}^{L-1} \boldsymbol{W}'_L\bigg(\prod_{r=l+1}^{L-1}(\boldsymbol{D}'_{i,r}\boldsymbol{W}'_r)\bigg)\boldsymbol{D}'_{i,l}(\boldsymbol{W}_l - \boldsymbol{W}'_l)\boldsymbol{h}'_{i,l-1} .\end{aligned} \tag{7}$$

According to Cao and Gu (2019, Lemma B.1), for $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$, then with probability at least $1 - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:

$$\frac{1}{2}\|\boldsymbol{x}_i\|_2 \leq \left\|\boldsymbol{h}'_{i,l-1}\right\|_2 \leq \frac{3}{2}\|\boldsymbol{x}_i\|_2 \leq \frac{3}{2}C_{\text{norm}} = \Theta(1), \forall i \in [n], l \in [L-1]\,. \tag{8}$$

According to Allen-Zhu et al. (2019b, Lemma 8.7), for $s \in \left[\Omega\left(\frac{1}{\log m}\right), \mathcal{O}\left(\frac{m}{L^3 \log m}\right)\right]$, $\omega = \mathcal{O}(L^{-3/2})$, with probability at least $1 - \exp(-\Omega(s \log m))$, we have:

$$\left\|\boldsymbol{W}_L\Big(\prod_{r=l+1}^{L-1}(\boldsymbol{D}_{i,r} + \boldsymbol{D}''_{i,r})\boldsymbol{W}_r\Big)(\boldsymbol{D}_{i,l} + \boldsymbol{D}''_{i,l}) - \boldsymbol{W}'_L\Big(\prod_{r=l+1}^{L-1}(\boldsymbol{D}'_{i,r}\boldsymbol{W}'_r)\Big)\boldsymbol{D}'_{i,l}\right\|_2 \leq \mathcal{O}(\sqrt{L^3 s \log m + \omega^2 L^3 m})\,.$$

Then, taking $s := \Theta(m\omega^2)$, we have $m\omega^2 \leq \mathcal{O}\left(\frac{m}{L^3 \log m}\right)$, that is $\omega \leq \mathcal{O}(L^{-3/2}(\log m)^{-1/2})$, with probability at least $1 - \exp(-\Omega(m\omega^2 \log m))$, we have:

$$\left\|\boldsymbol{W}_L\Big(\prod_{r=l+1}^{L-1}(\boldsymbol{D}_{i,r} + \boldsymbol{D}''_{i,r})\boldsymbol{W}_r\Big)(\boldsymbol{D}_{i,l} + \boldsymbol{D}''_{i,l}) - \boldsymbol{W}'_L\Big(\prod_{r=l+1}^{L-1}(\boldsymbol{D}'_{i,r}\boldsymbol{W}'_r)\Big)\boldsymbol{D}'_{i,l}\right\|_2 \leq \mathcal{O}(\sqrt{\omega^2 L^3 m \log m})\,. \tag{9}$$

Take Eq. (8) and Eq. (9) into Eq. (7), for $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$, then with probability at least $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:

$$|f(\boldsymbol{x}; \boldsymbol{W}) - f(\boldsymbol{x}; \boldsymbol{W}') - \langle \nabla f(\boldsymbol{x}; \boldsymbol{W}'), \boldsymbol{W} - \boldsymbol{W}' \rangle| \leq \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \|\boldsymbol{W}_l - \boldsymbol{W}'_l\|_2\,,$$

which concludes the proof. □

C.3. Proof of Lemma 3

Let us restate Lemma 3:

Lemma 3. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1. For any $t \geq 0$ and $(\boldsymbol{x}, \widetilde{y}) \sim \widetilde{P}$ that satisfy Assumption 1, with $\omega = \mathcal{O}(L^{-9/2}(\log m)^{-3})$, with probability at least $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:*

$$\begin{aligned}
&\widetilde{y}[f(\boldsymbol{x}; \boldsymbol{W}^{(t+1)}) - f(\boldsymbol{x}; \boldsymbol{W}^{(t)})] \\
\geq &\alpha g_i^{(t)} \left\langle \widetilde{y}\nabla f(\boldsymbol{x}; \boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i; \boldsymbol{W}^{(t)}) \right\rangle \\
&- \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\|\boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)}\right\|_2\,,
\end{aligned}$$

where $(\boldsymbol{x}_i, y_i)$ is the random selected training sample at step $t+1$.

Proof. By Lemma 2, for $\omega = \mathcal{O}(L^{-9/2}(\log m)^{-3})$, with probability at least $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:

$$\left|f(\boldsymbol{x}; \boldsymbol{W}^{(t+1)}) - f(\boldsymbol{x}; \boldsymbol{W}^{(t)}) - \left\langle \nabla f(\boldsymbol{x}; \boldsymbol{W}^{(t)}), \boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)} \right\rangle\right| \leq \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\|\boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)}\right\|_2\,.$$

Since $\widetilde{y} \in \{\pm 1\}$, we can calculate that:

$$
\begin{aligned}
&\widetilde{y}[f(\boldsymbol{x};\boldsymbol{W}^{(t+1)}) - f(\boldsymbol{x};\boldsymbol{W}^{(t)})] \\
\geq&\widetilde{y}\left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), \boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)} \right\rangle - \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\| \boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)} \right\|_2 \\
\overset{(a)}{=}&\widetilde{y}\alpha g_i^{(t)} \left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) \right\rangle - \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\| \boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)} \right\|_2 \\
=&\alpha g_i^{(t)} \left\langle \widetilde{y} \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) \right\rangle - \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\| \boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)} \right\|_2 ,
\end{aligned}
$$

where $(\boldsymbol{x}_i, y_i)$ is the random selected training sample at step $t+1$, and (a) uses Algorithm 1 and definition of $g_i^{(t)}$ that:

$$
\begin{aligned}
\boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)} &= -\alpha \cdot \nabla \ell\big(y_i f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\big) \\
&= -\alpha \ell'\big(y_i f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\big) \cdot \nabla\big(y_i f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\big) \\
&= \alpha g_i^{(t)} y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) .
\end{aligned}
$$

Finally we conclude the proof. □

C.4. Proof of Lemma 4

Let us restate Lemma 4 as below:

Lemma 4. *Let us define a DNN using Eq.* (1) *and trained by Algorithm 1 with a step size* $\alpha \gtrsim L^{-2}(\log m)^{-5/2}$. *Then under Assumption 1 and 2, for any* $t \geq 0$, $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$, *with probability at least* $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, *we have:*

$$
\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y} f(\boldsymbol{x};\boldsymbol{W}^{(t)})] \geq \Theta\big(t\alpha(1-2\eta)C_N\big) .
$$

Proof. According to the Lemma 3, $\forall t \geq 0$, for $\omega = \mathcal{O}(L^{-9/2}(\log m)^{-3})$, with probability at least $1 - \exp(-\Omega(m\omega^2 \log m)) - \mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, we have:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}(f(\boldsymbol{x};\boldsymbol{W}^{(t+1)}) - f(\boldsymbol{x};\boldsymbol{W}^{(t)}))] \\
\geq&\alpha g_i^{(t)} \mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}} \left\langle \widetilde{y} \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) \right\rangle - \mathcal{O}(\sqrt{\omega^2 L^3 m \log m}) \sum_{l=1}^{L-1} \left\| \boldsymbol{W}_l^{(t+1)} - \boldsymbol{W}_l^{(t)} \right\|_2 \\
\geq&\alpha g_i^{(t)} \mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}} \left\langle \widetilde{y} \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) \right\rangle - \mathcal{O}(\sqrt{\omega^2 L^5 \log m}) \\
:=&\alpha g_i^{(t)} \mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}} \left\langle \widetilde{y} \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}) \right\rangle - \varepsilon ,
\end{aligned}
\tag{10}
$$

where the second inequality use the result of lazy-training that $\boldsymbol{W}^{(t+1)} \in \mathcal{B}(\boldsymbol{W}^{(t)}, \frac{1}{\sqrt{m}})$ and $(\boldsymbol{x}_i, y_i)$ is the random selected training sample at step $t+1$.

By the definition of ε we have:

$$
\varepsilon = \mathcal{O}(\sqrt{\omega^2 L^3 \log m}) = \mathcal{O}(L^{-2}(\log m)^{-5/2}) .
$$

According to Assumption 2, we have:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left\langle \widetilde{y}\nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i\nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle \\
=&\mathbb{P}(i\in\mathcal{C})\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \widetilde{y}\nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i\nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C}\right] \\
&+\mathbb{P}(i\in\mathcal{C}')\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \widetilde{y}\nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i\nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C}'\right] \\
=&\mathbb{P}(i\in\mathcal{C})\mathbb{P}(\widetilde{y}=y_i)\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C},\widetilde{y}=y_i\right] \\
&-\mathbb{P}(i\in\mathcal{C})\mathbb{P}(\widetilde{y}\neq y_i)\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C},\widetilde{y}\neq y_i\right] \\
&+\mathbb{P}(i\in\mathcal{C}')\mathbb{P}(\widetilde{y}=y_i)\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C}',\widetilde{y}=y_i\right] \\
&-\mathbb{P}(i\in\mathcal{C}')\mathbb{P}(\widetilde{y}\neq y_i)\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left[\left\langle \nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), \nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle | i\in\mathcal{C}',\widetilde{y}\neq y_i\right] \\
\geq&(1-\eta)\times\frac{1}{2}\times C_N+[\eta\times\frac{1}{2}\times(-C_N)] \\
=&\frac{1-2\eta}{2}C_N\,,
\end{aligned}
\tag{11}
$$

where the $\mathcal{C}$ and $\mathcal{C}'$ represent the subset of training data for clean labels and noisy labels respectively. And $(\boldsymbol{x}_i, y_i)$ is the random selected training sample at step $t+1$.

According to Cao and Gu (2019, Lemma B.1). For $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$, then with probability at least $1-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$, $\frac{1}{2}\leq\left|f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}))\right|\leq\frac{3}{2}\left\|\boldsymbol{x}_i\right\|_2\leq\frac{3}{2}C_{\text{norm}}, \forall i\in[n]$.

That means:

$$
g_i^{(t)}=-\ell'(y_i f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}))=\frac{1}{1+\exp\left(y_i f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)}\right)}\geq\frac{1}{1+\exp(\frac{3}{2}C_{\text{norm}})}\,,\quad\forall i\in[n]\,. \tag{12}
$$

Take Eq. (11) and Eq. (12) into Eq. (10), we have for $\omega \leq \mathcal{O}(L^{-9/2}(\log m)^{-3})$, then with probability at least $1-\exp(-\Omega(m\omega^2\log m))-\mathcal{O}(nL^2)\exp(-\Omega(m\omega^{2/3}L))$:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}(f(\boldsymbol{x};\boldsymbol{W}^{(t+1)})-f(\boldsymbol{x};\boldsymbol{W}^{(t)}))] \\
\geq&\alpha g_i^{(t)}\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}\left\langle \widetilde{y}\nabla f(\boldsymbol{x};\boldsymbol{W}^{(t)}), y_i\nabla f(\boldsymbol{x}_i;\boldsymbol{W}^{(t)})\right\rangle-\varepsilon \\
\geq&\alpha g_i^{(t)}\frac{1-2\eta}{2}C_N-\varepsilon \\
\geq&\frac{\alpha}{1+\exp(\frac{3}{2}C_{\text{norm}})}\frac{1-2\eta}{2}C_N-\varepsilon \\
=&\Theta\big(\alpha(1-2\eta)C_N\big)\,,
\end{aligned}
$$

where the last inequality hold for $\alpha\gtrsim\varepsilon=\mathcal{O}(L^{-2}(\log m)^{-5/2})$.

Since the symmetry of the random Gaussian initialization and zero-mean of $\tilde{y}$, we have $\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(0)})]=0$, then we have:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(t)})]\\
=&\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}f(\boldsymbol{x};\boldsymbol{W}^{(0)})]+\sum_{s=0}^{n-1}\mathbb{E}_{(\boldsymbol{x},\widetilde{y})\sim\widetilde{P}}[\widetilde{y}(f(\boldsymbol{x};\boldsymbol{W}^{(s+1)})-f(\boldsymbol{x};\boldsymbol{W}^{(s)}))]\\
\geq&0+\sum_{t=0}^{t-1}\Theta\big(\alpha(1-2\eta)C_N\big)\\
=&\Theta\big(t\alpha(1-2\eta)C_N\big)\,.
\end{aligned}
$$

□

D. Optimization result for the classification problem

Before presenting the optimization result, we first introduce the following assumption:

Assumption 7. For $(\boldsymbol{x}_1,\widetilde{y}_1),(\boldsymbol{x}_2,\widetilde{y}_2)\sim\widetilde{P}$, if $y_1\neq y_2$, then $\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2\geq\phi$ for some $\phi>0$.

Then we present the theorem and its proof.

Theorem 4. *Given a DNN defined by Eq.* (1) *and trained by Algorithm 1 with the training data satisfy Assumption 1 and Assumption 7, then for the step size $\alpha=\mathcal{O}(n^{-3}L^{-9}m^{-1})$, the width $m=\widetilde{\Omega}(poly(n,\phi^{-1},L))\Omega(1/\delta)$ and the maximum number of iteration $t=\mathcal{O}\big(poly(n,\phi^{-1},L)\big)\mathcal{O}(1/\delta)$ then with high probability, Algorithm 1 can find a point $\boldsymbol{W}^{(t)}$ such that $\hat{L}(\boldsymbol{W}^{(t)})\leq\delta$.*

Proof. Recall our loss function is:

$$
\ell(z)=\log(1+\exp(-z))\,.
$$

And we can derive that:

$$
\ell'(z)=-\frac{1}{1+e^z},\quad \ell''(z)=\frac{e^z}{(1+e^z)^2}\,.
$$

It is easy to verify that:

$$
\ell'(z)<0,\quad \lim_{z\to\infty}\ell(z)=0,\quad \lim_{z\to\infty}\ell'(z)=0,\quad -\ell'(z)\geq\min\left\{\frac{1}{2},\frac{1}{2}\ell(z)\right\},\quad |\ell''(z)|\leq\frac{1}{4}\,.
$$

Then according to Zou et al. (2020, Theorem 4.1), we conclude the proof. □

E. Proof of Theorem 2

Before proving Theorem 2, we first introduce the following lemmas.

Lemma 6 (Minimum eigenvalue of sample covariance matrix. Adapted from Lemma 1 in Kuzborskij et al. (2021))**.** *Let $\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n]^\top\in\mathbb{R}^{n\times d}$ be a matrix with i.i.d. columns that satisfy Assumptions 5, and let $\hat{\boldsymbol{\Sigma}}=\boldsymbol{X}^\top\boldsymbol{X}/n$, and $\boldsymbol{\Sigma}=\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$. Then, for every $s\geq0$, with probability at least $1-2e^{-n}$, we have:*

$$
\lambda_{\min}(\hat{\boldsymbol{\Sigma}})\geq\lambda_{\min}(\boldsymbol{\Sigma})\left(\frac{3}{4}-\frac{c}{4}\sqrt{\frac{d}{n}}\right)^2,\quad \textit{if}\quad n\geq d\,,
$$

and

$$
\lambda_{\min}(\hat{\boldsymbol{\Sigma}})\geq\lambda_{\min}(\boldsymbol{\Sigma})\left(\sqrt{\frac{d}{n}}-\frac{c+6}{4}\right)^2,\quad \textit{if}\quad n<d\,,
$$

where we have an absolute constant $c=2^{3.5}\sqrt{\log(9)}$.

Proof. According to Assumption 5, we have $\max_i \|\boldsymbol{x}_i\|_{\psi_2}$ is bounded and $\|\boldsymbol{x}_i\|_{\boldsymbol{\Sigma}^\dagger} = \sqrt{d}$ almost sure for all $i \in [n]$. Without loss of generality, take $\max_i \|\boldsymbol{x}_i\|_{\psi_2} \leq \frac{1}{2}$ into Kuzborskij et al. (2021, Lemma 1), then choosing $s := n$,which conclude the proof. □

Then we are ready to prove Theorem 2.

Proof of Theorem 2. According to Lemma 5 and Zhu et al. (2022, Theorem 1), we have:

$$\lambda_0 \geq 2\mu_1^2 \lambda_{\min}(\boldsymbol{X}\boldsymbol{X}^\top)\,,$$

where μ_1 is the 1-st Hermite coefficient of the ReLU activation function.

We know that, $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$ have the same non-zero eigenvalues. Then according to Lemma 6 and Assumption 5, with probability at least $1 - 2e^{-n}$, we have:

Case 1: $n \geq d$

$$\begin{aligned}
\lambda_0 &\geq 2\mu_1^2 \lambda_{\min}(\boldsymbol{X}\boldsymbol{X}^\top) \\
&= 2\mu_1^2 n \lambda_{\min}(\hat{\boldsymbol{\Sigma}}) \\
&\geq 2\mu_1^2 \frac{n}{d}\left(\frac{3}{4} - \frac{c}{4}\sqrt{\frac{d}{n}}\right)^2, \quad \text{if} \quad n \geq d\,.
\end{aligned}$$

Case 2: $n < d$

$$\begin{aligned}
\lambda_0 &\geq 2\mu_1^2 \lambda_{\min}(\boldsymbol{X}\boldsymbol{X}^\top) \\
&= 2\mu_1^2 n \lambda_{\min}(\hat{\boldsymbol{\Sigma}}) \\
&\geq 2\mu_1^2 \frac{n}{d}\left(\sqrt{\frac{d}{n}} - \frac{c+6}{4}\right)^2, \quad \text{if} \quad n < d\,.
\end{aligned}$$

where we have an absolute constant $c = 2^{3.5}\sqrt{\log(9)} \approx 16.77$. □

F. Supplementary proofs for Theorem 3

In this section, we present the proofs of Theorem 3 in Section 4.

F.1. A Precise Form of the Theorem 3

Theorem 5 (Precise form of Theorem 3)**.** *Let α, β and γ be three non-negative parameters depends on the Laplace kernel, under Assumption 3, 4 and 5, let $0 < \delta < \frac{1}{2}$, $0 < \theta \leq 1/2$, the ground-truth function f_ρ lies in a RKHS by Assumption 6 and d large enough, suppose that, $\omega \leq poly(1/n, \lambda_0, 1/L, 1/\log(m), \epsilon, 1/\log(1/\delta'), \kappa)$, $m \geq poly(1/\omega)$ and $\kappa = \mathcal{O}(\frac{\epsilon}{\log(n/\delta')})$. then for any given $\varepsilon > 0$, it holds with probability at least $1 - 2\delta - \delta' - d^{-2} - 2e^{-n}$.*

$$\mathbb{E}\,\|f_{\text{nn}} - f_\rho\|^2_{L^2_{\rho_X}} \leq n^{-\theta}\log^4(\frac{2}{\delta}) + \frac{\sigma_\epsilon^2\beta}{d}\mathcal{N}^\gamma_{\widetilde{\boldsymbol{X}}} + \frac{\sigma_\epsilon^2\log^{2+4\varepsilon}d}{\gamma^2 d^{4\theta-1}} + \mathcal{O}\left(\left(\epsilon + \frac{\sqrt{n}}{\lambda_0}\omega^{1/3}L^{5/2}\sqrt{m\log m} + \frac{n^{3/2}}{\lambda_0^3\kappa}\right)^2\right),$$

where:

$$\lambda_0 \geq \begin{cases} 2\mu_1^2\frac{n}{d}\left(\frac{3}{4} - \frac{c}{4}\sqrt{\frac{d}{n}}\right)^2, & \text{if} \quad n \geq d, \\ 2\mu_1^2\frac{n}{d}\left(\sqrt{\frac{d}{n}} - \frac{c+6}{4}\right)^2, & \text{if} \quad n < d\,. \end{cases}$$

where we have an absolute constant $c = 2^{3.5}\sqrt{\log(9)}$, $\widetilde{\boldsymbol{X}} := \beta\boldsymbol{X}\boldsymbol{X}^\top/d + \alpha\mathbf{1}\mathbf{1}^\top$,$\mathcal{N}^\gamma_{\widetilde{\boldsymbol{X}}} = \sum_{i=0}^{n-1}\frac{\lambda_i(\widetilde{\boldsymbol{X}})}{(\lambda_i(\widetilde{\boldsymbol{X}})+\gamma))^2}$, $\|f\|^2_{L^2_{\rho_X}} = \int_X |f(\boldsymbol{x})|^2\, d\rho_X(\boldsymbol{x})$, and μ_1 is the 1-st Hermite coefficient of the ReLU activation function.

Remark: The three non-negative parameters α, β, and γ depend on the linearization of the Laplace kernel in the high dimension setting, refer to (Liu et al., 2021) for details.

F.2. Propositions

We present several propositions that are needed for our Theorem 3 as below.

Proposition 2 (Convergence to the NTK at initialization. Adapted from Theorem 3.1 in Arora et al. (2019b)). *Fix $\epsilon > 0$ and $\delta \in (0,1)$. Suppose that $m \geq \Omega(\frac{L^6}{\epsilon^4}\log(L/\delta))$. Then for any inputs $\|\boldsymbol{x}_1\| \leq 1$, $\|\boldsymbol{x}_2\| \leq 1$, with probability at least $1-\delta$ we have:*

$$\left|\left\langle \frac{\partial f(\boldsymbol{x}_1;\boldsymbol{W})}{\partial \boldsymbol{W}}, \frac{\partial f(\boldsymbol{x}_2;\boldsymbol{W})}{\partial \boldsymbol{W}} \right\rangle - K_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)\right| = \mathcal{O}(\epsilon L)\,.$$

Proposition 3 (Equivalence between trained neural network and kernel regression). *Suppose that, $\omega \leq poly(1/n, \lambda_0, 1/L, 1/\log(m), \epsilon, 1/\log(1/\delta), \kappa)$, $m \geq poly(1/\omega)$, $\boldsymbol{x}_{te}$ satisfy Assumption 5 and $\kappa = \mathcal{O}(\frac{\epsilon}{\log(n/\delta)})$. Then w.p. at least $1-\delta$ over random initialization, we have:*

$$|f_{\texttt{nn}}(\boldsymbol{x}_{te}) - f_{\texttt{NTK}}(\boldsymbol{x}_{te})| \leq \mathcal{O}\big(\epsilon + \frac{\sqrt{n}}{\lambda_0}\omega^{1/3}L^{5/2}\sqrt{m\log m} + \frac{n^{3/2}}{\lambda_0^3\kappa}\big)\,.$$

Proposition 4 (Adapted from Theorem 2 in Liu et al. (2021)). *Let α, β and γ be three non-negative parameters depends on the laplace kernel, under Assumption 3, 4 and 5 let $0 < \delta < \frac{1}{2}$, $\theta = \frac{1}{2} - \frac{2}{8+m}$, ground-truth function f_ρ lies in a RKHS and d large enough, then for any given $\varepsilon > 0$, it holds with probability at least $1 - 2\delta - d^{-2}$.*

$$\mathbb{E}\,\|f_{\texttt{NTK}} - f_\rho\|^2_{L^2_{\rho_X}} \leq n^{-2\theta r}\log^4(\frac{2}{\delta}) + \frac{\sigma_\epsilon^2\beta}{d}\mathcal{N}^\gamma_{\widetilde{\boldsymbol{X}}} + \frac{\sigma_\epsilon^2\log^{2+4\varepsilon}d}{\gamma^2 d^{4\theta-1}}\,,$$

where $\widetilde{\boldsymbol{X}} := \beta\boldsymbol{X}\boldsymbol{X}^\top/d + \alpha\mathbf{1}\mathbf{1}^\top, \mathcal{N}^\gamma_{\widetilde{\boldsymbol{X}}} = \sum_{i=0}^{n-1}\frac{\lambda_i(\widetilde{\boldsymbol{X}})}{(\lambda_i(\widetilde{\boldsymbol{X}})+\gamma))^2}$, $\|f\|^2_{L^2_{\rho_X}} = \int_X |f(\boldsymbol{x})|^2\, d\rho_X(\boldsymbol{x})$.

F.3. Proof of Proposition 3

Before proving Proposition 3, we need the following lemmas:

Lemma 7 (Gradient Perturbation $\rightarrow$ Kernel Perturbation). *For any two data point $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ that satisfy Assumption 5 and the neural network defined in Eq. (1), if $\left\|\frac{\partial f(\boldsymbol{x}_1;\boldsymbol{W}^{(t)})}{\partial \boldsymbol{W}} - \frac{\partial f(\boldsymbol{x}_1;\boldsymbol{W}^{(0)})}{\partial \boldsymbol{W}}\right\| \leq \epsilon$ and $\left\|\frac{\partial f(\boldsymbol{x}_2;\boldsymbol{W}^{(t)})}{\partial \boldsymbol{W}} - \frac{\partial f(\boldsymbol{x}_2;\boldsymbol{W}^{(0)})}{\partial \boldsymbol{W}}\right\| \leq \epsilon$, we have*

$$\left|K^{(t)}_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2) - K^{(0)}_{\texttt{NTK}}(\boldsymbol{x}_1,\boldsymbol{x}_2)\right| = \mathcal{O}(\epsilon)\,,$$

where the $\boldsymbol{K}^{(t)}_{\texttt{NTK}}$ means the NTK kernel defined in Eq. (3) for neural networks Eq. (1) at training time t.

Proof. According to Lemma 5 in Zhu et al. (2022) and our network Eq. (1), we have:

$$\frac{\partial f(\boldsymbol{x};\boldsymbol{W}^{(0)})}{\partial \boldsymbol{W}} = \Theta(1)\,.$$

Then we use triangle inequality, which concludes the proof. □

Lemma 8 (Adapted from Lemma 8.2 in Allen-Zhu et al. (2019b)). *Suppose that $\omega = \mathcal{O}(\frac{1}{L^{9/2}(\log m)^3})$, then w.p. at least $1 - \exp(-\Omega(m\omega^{2/3}L))$ over random initialization, if $\|\boldsymbol{W}_l - \boldsymbol{W}'_l\|_2 \leq \omega, \forall l \in [L]$, we have $\left\|\boldsymbol{W}_l\boldsymbol{h}_{i,l-1} - \boldsymbol{W}'_l\boldsymbol{h}'_{i,l-1}\right\|_2 = \mathcal{O}(\omega L^{5/2}\sqrt{\log m}), \forall l \in [L]$.*

For notational simplicity, we define the notation $\boldsymbol{b}_l$:

$$\boldsymbol{b}_l = \begin{cases} 1 \quad \text{if } l = L+1, \\ \boldsymbol{D}_l(\boldsymbol{W}_{l+1})^\top\boldsymbol{b}_{l+1} \quad \text{otherwise.} \end{cases}$$

Lemma 9 (Adapted from Lemma 8.7 in Allen-Zhu et al. (2019b)). *Suppose that $\omega = \mathcal{O}(\frac{1}{L^6(\log m)^{3/2}})$, then with probability at least $1 - \exp(-\Omega(\omega^{2/3} m L \log m))$ over random initialization, if $\|\boldsymbol{W}_l - \boldsymbol{W}_l'\|_2 \leq \omega, \forall l \in [L]$, we have $\|\boldsymbol{b}_l - \boldsymbol{b}_l'\|_2 = \mathcal{O}(\omega^{1/3} L^2 \sqrt{m \log m}), \forall l \in [L]$.*

Proof. According to Allen-Zhu et al. (2019b, Lemma 8.7), choose $s := m\omega^{2/3} L$, which concludes the proof. □

Lemma 10. *Suppose that $\omega = \mathcal{O}(\frac{1}{L^6(\log m)^3})$, then with probability at least $1 - \exp(-\Omega(\omega^{2/3} m L))$ over random initialization, if $\|\boldsymbol{W}_l - \boldsymbol{W}_l'\|_2 \leq \omega, \forall l \in [L]$, we have:*

$$\left\|\boldsymbol{b}_l'(\boldsymbol{W}_{l-1}' \boldsymbol{h}_{i,l-2}')^\top - \boldsymbol{b}_l(\boldsymbol{W}_{l-1}\boldsymbol{h}_{i,l-2})^\top\right\|_{\mathrm{F}} = \mathcal{O}(\omega^{1/3} L^{5/2} \sqrt{m \log m}), \quad \forall l \in [L].$$

Proof. We use Lemmas 8 and 9 and the triangle inequality:

$$\begin{aligned}
&\left\|\boldsymbol{b}_l'(\boldsymbol{W}_{l-1}' \boldsymbol{h}_{i,l-2}')^\top - \boldsymbol{b}_l(\boldsymbol{W}_{l-1}\boldsymbol{h}_{i,l-2})^\top\right\|_{\mathrm{F}} \\
\leq &\left\|\boldsymbol{b}_l'(\boldsymbol{W}_{l-1}' \boldsymbol{h}_{i,l-2}')^\top - \boldsymbol{b}_l(\boldsymbol{W}_{l-1}'\boldsymbol{h}_{i,l-2}')^\top\right\|_{\mathrm{F}} + \left\|\boldsymbol{b}_l(\boldsymbol{W}_{l-1}' \boldsymbol{h}_{i,l-2}')^\top - \boldsymbol{b}_l(\boldsymbol{W}_{l-1}\boldsymbol{h}_{i,l-2})^\top\right\|_{\mathrm{F}} \\
\leq &\mathcal{O}(\omega^{1/3} L^{5/2} \sqrt{m \log m}).
\end{aligned}$$

□

Lemma 11 (Adapted from Lemma F.9 in Arora et al. (2019b)). *Let $\omega \leq$ poly$(\epsilon, L, \lambda_0, \frac{1}{\log(m)}, \frac{1}{\log(1/\delta)}, \kappa, \frac{1}{n})$. If $m \geq$ poly$(1/\omega)$, then with probability at least $1 - \delta$ over random initialization, we have:*

$$\left\|\boldsymbol{W}_l^{(t)} - \boldsymbol{W}_l^{(0)}\right\|_{\mathrm{F}} \leq \omega, \quad \forall t \geq 0, \forall l \in [L],$$

and

$$f_{\mathtt{nn}}^{(t)}(\boldsymbol{x}) - y \leq \exp(-\frac{1}{2}\kappa^2 \lambda_0 t)(f_{\mathtt{nn}}^{(0)}(\boldsymbol{x}) - y).$$

Lemma 12 (Kernel Perturbation During Training). *Suppose that, $\omega \leq$ poly$(1/n, \lambda_0, 1/L, 1/\log(m), \epsilon, 1/\log(1/\delta))$, $m \geq$ poly$(1/\omega)$ and $\kappa \leq 1$. Then with probability at least $1 - \delta$ over random initialization, we have for all $t \leq 0$, $\forall (\boldsymbol{x}_1, \boldsymbol{x}_2)$:*

$$\left|K_{\mathtt{NTK}}^{(t)}(\boldsymbol{x}_1, \boldsymbol{x}_2) - K_{\mathtt{NTK}}^{(0)}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right| \leq \mathcal{O}(\omega^{1/3} L^{5/2} \sqrt{m \log m}).$$

Proof. By Lemma 11, we know that for $t \to \infty$, $\left\|\boldsymbol{W}_l^{(t)} - \boldsymbol{W}_l^{(0)}\right\|_{\mathrm{F}} \leq \omega$, by Lemma 10, we know that on the gradient, there is only a small perturbation, then the perturbation of kernel value is small by Lemma 7. □

Lemma 13 (Kernel Perturbation → Output Perturbation. Adapted from Lemma F.1 in Arora et al. (2019b)). *Fix $\epsilon_{\boldsymbol{H}} \leq \frac{1}{2}\lambda_0$. Suppose $\left|f(\boldsymbol{x}_i; \boldsymbol{W}^{(0)})\right| \leq \epsilon_0, \forall i \in [n]$, $\left|f(\boldsymbol{x}_{te}; \boldsymbol{W}^{(0)})\right| \leq \epsilon_0$ and $f_{\mathtt{nn}}^{(0)}(\boldsymbol{x}) - y = \mathcal{O}(\sqrt{n})$. Furthermore, if $\forall t \geq 0$, $\left\|\boldsymbol{K}_{\mathtt{NTK}}^{(t)}(\boldsymbol{x}_{te}, \boldsymbol{X}) - \boldsymbol{K}_{\mathtt{NTK}}^{(0)}(\boldsymbol{x}_{te}, \boldsymbol{X})\right\|_2 \leq \epsilon_{te}$ and $\left\|\boldsymbol{K}_{\mathtt{NTK}}^{(0)} - \boldsymbol{K}_{\mathtt{NTK}}^{(t)}\right\|_2 \leq \epsilon_{\boldsymbol{H}}$, then we have:*

$$|f_{\mathtt{nn}}(\boldsymbol{x}_{te}) - f_{\mathtt{NTK}}(\boldsymbol{x}_{te})| \leq \mathcal{O}\left(\epsilon_0 + \frac{\sqrt{n}}{\lambda_0}\epsilon_{te} + \frac{\sqrt{n}}{\lambda_0^2}\log\left(\frac{n}{\epsilon_{\boldsymbol{H}}\lambda_0\kappa}\right)\epsilon_{\boldsymbol{H}}\right).$$

Then we are ready to prove Proposition 3.

Proof of Proposition 3. According to Lemma 13, we have:

$$|f_{\mathtt{nn}}(\boldsymbol{x}_{te}) - f_{\mathtt{NTK}}(\boldsymbol{x}_{te})| \leq \mathcal{O}\left(\epsilon + \frac{\sqrt{n}}{\lambda_0}\epsilon_{te} + \frac{\sqrt{n}}{\lambda_0^2}\log\left(\frac{n}{\epsilon_{\boldsymbol{H}}\lambda_0\kappa}\right)\epsilon_{\boldsymbol{H}}\right).$$

Note that the function $g(x) := x\log(\frac{n}{x\lambda_0\kappa})$ achieves its maximum $g(x)_{\max} = \frac{n}{e\lambda_0\kappa}$.

Combine this and Lemma 12, we have:

$$|f_{\mathtt{nn}}(\boldsymbol{x}_{te}) - f_{\mathtt{NTK}}(\boldsymbol{x}_{te})| \leq \mathcal{O}\left(\epsilon + \frac{\sqrt{n}}{\lambda_0}\omega^{1/3}L^{5/2}\sqrt{m\log m} + \frac{n^{3/2}}{\lambda_0^3\kappa}\right).$$

□

F.4. Proof of Proposition 4

Before proving Proposition 4, we first introduce the following lemmas:

Lemma 14 (Adapted from Theorem 1 in Chen and Xu (2021))**.** *Let $\mathcal{H}_{\mathtt{Lap}}$ and $\mathcal{H}_{\mathtt{NTK}}$ be the RKHS associated with the Laplace kernel and the neural tangent kernel of a L-layer fully connected ReLU network. Both kernels are restricted to the sphere $\mathbb{S}^{d-1}$. Then the two spaces include the same set of functions:*

$$\mathcal{H}_{\mathtt{Lap}} = \mathcal{H}_{\mathtt{NTK}}.$$

Then we are ready to prove Proposition 4.

Proof of Proposition 4. According to Lemma 14, we have $\mathcal{H}_{\mathtt{Lap}} = \mathcal{H}_{\mathtt{NTK}}$, that means, the estimators of Eq. (4) under two RKHS corresponding to the NTK kernel and Laplace kernel are the same:

$$f_{\mathtt{NTK}}(\boldsymbol{x}_{te}) = f_{\mathtt{Laplace}}(\boldsymbol{x}_{te}), \tag{13}$$

where $f_{\mathtt{NTK}}$, $f_{\mathtt{Laplace}}$ are the estimators of Eq. (4) in $\mathcal{H}_{\mathtt{NTK}}$ and $\mathcal{H}_{\mathtt{Laplace}}$, respectively.

According to Eq. (5), when the Laplace kernel is restricted to the sphere $\mathbb{S}^{d-1}$, it is a dot product kernel.

Combine Eq. (13) and Liu et al. (2021, Theorem 2), we get the result.

□

F.5. Proof of Theorem 5

Proof. Using triangle inequality, we have:

$$\begin{aligned}
\mathbb{E}\left\|f_{\mathtt{nn}} - f_\rho\right\|_{L^2_{\rho_X}}^2 \leq& \mathbb{E}\left\|f_{\mathtt{nn}} - f_{\mathtt{ntk}}\right\|_{L^2_{\rho_X}}^2 + \mathbb{E}\left\|f_{\mathtt{ntk}} - f_\rho\right\|_{L^2_{\rho_X}}^2 \\
\leq& n^{-2\theta}\log^4(\frac{2}{\delta}) + \frac{\sigma_\epsilon^2\beta}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}}^{\gamma} + \frac{\sigma_\epsilon^2\log^{2+4\varepsilon}d}{\gamma^2 d^{4\theta-1}} \\
&+\mathbb{E}\int_X \left|f_{\mathtt{ntk}}(\boldsymbol{x}) - f_\rho(\boldsymbol{x})\right|^2 d\rho_X(\boldsymbol{x}) \\
\leq& n^{-2\theta}\log^4(\frac{2}{\delta}) + \frac{\sigma_\epsilon^2\beta}{d}\mathcal{N}_{\widetilde{\boldsymbol{X}}}^{\gamma} + \frac{\sigma_\epsilon^2\log^{2+4\varepsilon}d}{\gamma^2 d^{4\theta-1}} \\
&+\mathcal{O}\left((\epsilon + \frac{\sqrt{n}}{\lambda_0}\omega^{1/3}L^{5/2}\sqrt{m\log m} + \frac{n^{3/2}}{\lambda_0^3\kappa})^2\right),
\end{aligned} \tag{14}$$

with probability at least $1-2\delta-\delta'-d^{-2}$, where the first inequality use Proposition 4 and second inequality use Proposition 3.

Then According to Theorem 2, we have:

$$\lambda_0 \geq \begin{cases} 2\mu_1^2\frac{n}{d}\left(\frac{3}{4} - \frac{c}{4}\sqrt{\frac{d}{n}}\right)^2, & \text{if} \quad n \geq d, \\ 2\mu_1^2\frac{n}{d}\left(\sqrt{\frac{d}{n}} - \frac{c+6}{4}\right)^2, & \text{if} \quad n < d. \end{cases} \tag{15}$$

with probability at least $1 - 2e^{-n}$.

Combine Eq. (14) and Eq. (15), we conclude the proof. □