# Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)

**Zhenyu Zhu, Fanghui Liu, Grigorios G Chrysos, Volkan Cevher**
EPFL, Switzerland
{[first name].[surname]}@epfl.ch

## Abstract

We study the average robustness notion in deep neural networks in (selected) wide and narrow, deep and shallow, as well as lazy and non-lazy training settings. We prove that in the under-parameterized setting, width has a negative effect while it improves robustness in the over-parameterized setting. The effect of depth closely depends on the initialization and the training mode. In particular, when initialized with LeCun initialization, depth helps robustness with lazy training regime. In contrast, when initialized with Neural Tangent Kernel (NTK) and He-initialization, depth hurts the robustness. Moreover, under non-lazy training regime, we demonstrate how the width of a two-layer ReLU network benefits robustness. Our theoretical developments improve the results by Huang et al. [2021], Wu et al. [2021] and are consistent with Bubeck and Sellke [2021], Bubeck et al. [2021].

## 1 Introduction

It is now well-known that deep neural networks (DNNs) are susceptible to adversarially chosen, albeit imperceptible, perturbations to their inputs [Goodfellow et al., 2015, Szegedy et al., 2014]. This lack of robustness is worrying as DNNs are now deployed in many real-world applications [Eykholt et al., 2018]. As a result, new algorithms are more and more being developed to defend against adversarial attacks to improve the DNN robustness. Among the current defense methods, the most commonly used and arguably the most successful method is adversarial training based minimax optimization [Athalye et al., 2018, Croce and Hein, 2020, Madry et al., 2018]. To study adversarial attacks and defenses, we need to throughout investigate the robustness of DNNs at first.

The literature focuses on several aspects of the robustness issue, from algorithms to their initialization as well as from width of neural networks to their depth (i.e., the architecture). On the practical side, Madry et al. [2018] advocate that adversarial training requires more parameters (e.g., width) for better performance in minimax optimization, which would fall into the so-called over-parameterized regime[1] [Zhang et al., 2017]. On the theory side, recent works suggest that over-parameterization may damage the adversarial robustness [Huang et al., 2021, Wu et al., 2021, Zhou and Schoellig, 2019, Hassani and Javanmard, 2022]. In stark contrast, Bubeck and Sellke [2021], Bubeck et al. [2021] argue that the robustness of DNNs needs enough parameters to be guaranteed. See a detailed discussion in Section 2.

Our work is motivated to investigate this apparent contradiction in theory, and to close the gap as much as possible. We begin with a definition of the *perturbation stability* of DNNs, which can be used to describe the robustness, following the spirit of Wu et al. [2021], Dohmatob and Bietti [2022]. Mathematically,

[1] Over-parameterized regime works in the setting that the number of parameters in DNN is (much) larger than the number of training data.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

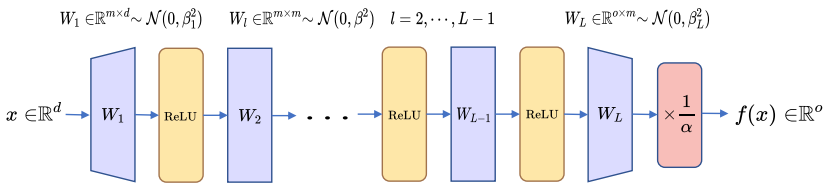



Figure 1: Schematic of our deep fully connected ReLU neural network.

Table 1: Comparison of the *perturbation stability* of a deep ReLU neural network (see Fig. 1) under three common Gaussian initializations with different variances. A formal definition of this neural network refers to Eq. (2).

| Initialization name | Initialization form | Our bound for $\mathscr{P}(\boldsymbol{f}, \epsilon)/\epsilon$ |
|---|---|---|
| LeCun et al. [2012] | $\beta_1 = \sqrt{\frac{1}{d}}, \beta = \beta_L = \sqrt{\frac{1}{m}}, \alpha = 1$ | $\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{o}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$ |
| He et al. [2015] | $\beta_1 = \sqrt{\frac{2}{d}}, \beta = \beta_L = \sqrt{\frac{2}{m}}, \alpha = 1$ | $\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{o}{d}}$ |
| Allen-Zhu et al. [2019] | $\beta = \beta_1 = \sqrt{\frac{2}{m}}, \beta_L = \sqrt{\frac{1}{o}}, \alpha = 1$ | $\sqrt{\frac{L^3 m}{o}} e^{-m/L^3} + 1$ |

**Definition 1** (*perturbation stability*)**.** *The perturbation stability of the neural network* $\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W}) : \mathbb{R}^d \mapsto \mathbb{R}^o$ *parameterized by the neural network parameter* $\boldsymbol{W}$ *under the data distribution* $\mathcal{D}_X$ *and a perturbation radius* $\epsilon$ *is defined as follows:*

$$\mathscr{P}(\boldsymbol{f}, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{W}} \left\| \nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W})^\top (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2 , \quad \forall \boldsymbol{x} \sim \mathcal{D}_X, \ \hat{\boldsymbol{x}} \sim \mathit{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})) , \tag{1}$$

*where* $\hat{\boldsymbol{x}}$ *is uniformly sampled from an* $\ell_2$ *norm ball of* $\boldsymbol{x}$ *with radius* $\epsilon$*, denoted as* $\mathit{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x}))$.

Since our definition of the *perturbation stability* takes the expectation to clean and adversarial data points, it is natural to describe the *average robustness* of a neural network. It can be noticed that the larger value of the *perturbation stability* means worse robustness in average, i.e., *average robustness*.

Based on the *perturbation stability*, we study the *average robustness* of neural networks under different initializations in (selected) wide and narrow, deep and shallow, as well as lazy and non-lazy[2] training settings. Generally, non-lazy training makes the analysis of neural networks intractable as DNNs in this regime cannot be simplified as a time-independent model [Chizat and Bach, 2018], and accordingly, the analysis in this regime is mainly restricted to the two-layer setting [Mei et al., 2018, 2019].

Overall, our results suggest that the width (*good*) helps *average robustness* in the over-parameterized regime but the depth (*bad*) can help only under certain initializations (*ugly*). To be specific, we make the following contributions and findings under the lazy/non-lazy training regimes, see Table 1.

In the **lazy training** regime, the derived upper-bounds for DNNs, suggest that

- along with the increase in width, the robustness first becomes worse in the under-parameterized regime and then gets better, and finally tends to be a constant in highly over-parameterized regimes, which implies the existence of phase transition.
- the depth has more complex tendency on robustness, which largely depends on the initialization and the training mode. It can be grouped into two main classes (*cf.*, Table 1): depth helps robustness in an exponential order under LeCun initialization [LeCun et al., 2012], whereas it hurts robustness in a polynomial order under He-initialization [He et al., 2015] and under Neural Tangent Kernel (NTK) initialization [Allen-Zhu et al., 2019].

Surprisingly, standard tools on training dynamics of neural networks [Allen-Zhu et al., 2019, Du et al., 2018] are sufficient to obtain our bounds, which explain the relationship between robustness

[2]Here the lazy/non-lazy training regime indicates that neural network parameters change little/much during training. These two phases are determined by different initializations [Woodworth et al., 2020, Luo et al., 2021].

and the structural/architectural parameters of neural network. Our theoretical developments improve the results by Huang et al. [2021], Wu et al. [2021], and are supported by empirical evidence.

In the **non-lazy** training regime, we derive upper-bounds for two-layer networks, suggesting that

- the width improves robustness under different initializations.
- the convergence rates of the average robustness is effected by several certain initializations.

We also derive a sufficient condition to identify when DNNs enter in this regime, as an initial but first attempt on understanding DNNs in this regime. Our technical contribution lies in connecting robustness to changes of neural network parameters during the early stages of training, which could expand the application scope of deep learning theory beyond *lazy training* analysis [Jacot et al., 2018, Allen-Zhu et al., 2019].

**Notations:** We use the shorthand $[n] := \{1, 2, \ldots, n\}$ for a positive integer $n$. We denote by $a(n) \lesssim b(n)$: there exists a positive constant $c$ independent of $n$ such that $a(n) \leqslant cb(n)$. The standard Gaussian distribution is $\mathcal{N}(0, 1)$ with the zero-mean and the identity variance. Uniform distribution inside the sphere is $\text{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x}))$ with the center $\boldsymbol{x}$ and radius $\epsilon$. We follow the standard Bachmann–Landau notation in complexity theory e.g., $\mathcal{O}$, $o$, $\Omega$, and $\Theta$ for order notation.

## 2 Related work

DNNs are demonstrated to be fragile, sensitive to adversarially chosen but undetectable noise both empirically [Szegedy et al., 2014] and theoretically [Huang et al., 2021, Bubeck and Sellke, 2021]. Adversarial training [Athalye et al., 2018, Croce and Hein, 2020, Zhang et al., 2020b] is a reliable way to obtain adversarially robust neural network. Nevertheless, improving the overall robustness of neural networks is still an unsolved problem in machine learning, especially when coupling with initializations and parameters.

**Over-parameterized neural networks under lazy/non-lazy training regimes:** Modern DNNs in practice [He et al., 2016] work under the setting where the number of parameters is (much) larger than the number of training data. Analysis of DNNs in terms of optimization [Safran et al., 2021, Zhou et al., 2021] and generalization [Cao and Gu, 2019] has received great attention in deep learning theory [Zhang et al., 2017].

In deep learning theory, neural tangent kernel (NTK) [Jacot et al., 2018] and mean field [Mei et al., 2018] analysis are two powerful tools for neural network analysis. To be specific, NTK builds an equivalence between training dynamics by gradient-based algorithms of DNNs and kernel regression under a specific initialization, and thus allows for deep networks analysis [Allen-Zhu et al., 2019, Du et al., 2019a, Chen et al., 2020]. However, this way actually requires neural networks falling in a *lazy training* regime [Chizat et al., 2019], where neural networks are able to achieve zero training loss but the parameters change little, or even remain unchanged during training. In contrast, mean-field theory establishes global convergence by casting network weights during training as an evolution in the space of probability distributions under some certain initialization to results [Mei et al., 2018, Chizat and Bach, 2018]. This strategy goes beyond lazy training regime, which allows for neural networks parameters changing in a constant order after training.

If neural networks parameters changes a lot after training, or even tend to infinity, then neural networks work in *non-lazy* training regime. Analysis of DNNs under this setting appears intractable and challenging, so current work mainly focus on two-layer neural networks [Maennel et al., 2018, Luo et al., 2021].

**Robustness and over-parameterization** Goodfellow et al. [2015] demonstrate that adversarial learning helps robustness and reduces overfitting, and in the following, many works focus on the trace the origin and influencing factors of adversarial examples and robustness for neural network [Schmidt et al., 2018, Zhang et al., 2020a, Allen-Zhu and Li, 2022]. The relation between model capacity and robustness is empirically investigated by Madry et al. [2018], i.e., neural network with insufficient capacity can seriously hurt the robustness. Bubeck et al. [2021] theoretically study the inherent trade-off between the size of neural networks and their robustness, and they claim that over-parameterization is necessary for the robustness of two-layer neural networks.

However, some recent works propose the opposite view. Under the lazy training regime, Huang et al. [2021] demonstrate that when over-parameterized neural networks get wider, the robustness will be

hurted in a polynomial order. At the same time, the depth hurts the robustness in an exponential order. Wu et al. [2021] affirm the view of Huang et al. [2021] on the width. However for depth, they derive a stronger bound that the robustness gets worse in a polynomial decay as depth increases, as suggested by Hassani and Javanmard [2022]: over-parameterization hurts robustness. In addition, Gao et al. [2019] also make a similar view: more model capacity (i.e., wider width and deeper depth) leads to the robustness of neural networks worse. Nevertheless, we remark that, the results of Hassani and Javanmard [2022] work in a slightly different setting with Bubeck et al. [2021] on data interpolation, which requires a careful comparison. Accordingly, we adopt a complementary view to these vast literature, provide an in-depth theoretical analysis to investigate this apparent contradiction in theory, and to close the gap as much as possible.

# 3 Problem setting

Let $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^o$ be compact metric spaces. We assume that the training set $\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ is drawn from a unknown probability measure $\mathcal{D}$ on $X \times Y$. Its marginal data distribution is denoted by $\mathcal{D}_X$. The goal of the classification task is to learn a neural network $\boldsymbol{f} : X \to Y$ such that $\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W})$ parameterized by $\boldsymbol{W}$ is a good approximation of the label $\boldsymbol{y} \in Y$ corresponding to a new sample $\boldsymbol{x} \in X$. In this paper, we use the empirical risk $L(\boldsymbol{W}) = \frac{1}{2n}\sum_{i=1}^n \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{W}) - \boldsymbol{y}_i\|_2^2$. Then we make the following assumptions.

**Assumption 1.** *We assume that the data satisfy* $\|\boldsymbol{x}\|_2 = 1$.

**Remark:** This assumption is standard in theory on over-parameterized neural networks and also commonly used in practice [Du et al., 2019b,a, Allen-Zhu et al., 2019, Oymak and Soltanolkotabi, 2020, Malach et al., 2020].

## 3.1 Network

We focus on the typical depth-$L$ fully-connected ReLU neural networks admitting the width $m_l$ of the $l$-th hidden layer, $\forall l \in [L]$ (*cf.*, Fig. 1):

$$\boldsymbol{h}_{i,0} = \boldsymbol{x}_i; \quad \boldsymbol{h}_{i,l} = \phi(\boldsymbol{W}_l \boldsymbol{h}_{i,l-1}); \quad \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{W}) = \boldsymbol{f}_i = \frac{1}{\alpha}\boldsymbol{W}_L \boldsymbol{h}_{i,L-1}; \quad \forall l \in [L-1]\,,\ i \in [n]\,, \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^o$, $\alpha$ is the scaling factor, and $\phi = \max(0, x)$ is ReLU activation function. The neural networks parameters formulate the tuple of weight matrices $\boldsymbol{W} := \{\boldsymbol{W}_i\}_{i=1}^L \in \{\mathbb{R}^{m\times d} \times (\mathbb{R}^{m\times m})^{L-2} \times \mathbb{R}^{o\times m}\}$. According to the property $\phi(x) = x\phi'(x)$ 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 2** (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 our setting, we consider the standard Gaussian initialization with different variances that includes three typical initialization schemes in practice.

**Initialization:** Let $m_0 = d$, $m_L = o$ and $m_2 = \cdots = m_{L-1} = m$, we make the standard random initialization $[\boldsymbol{W}_l]_{i,j} \sim \mathcal{N}(0, \beta_l^2)$ for every $(i,j) \in [m_l] \times [m_{l-1}]$ and $l \in [L]$. Choosing different variance, our work holds for three commonly used Gaussian initializations, i.e., LeCun initialization [LeCun et al., 2012], He-initialization [He et al., 2015] and Neural Tangent Kernel (NTK) initialization [Allen-Zhu et al., 2019], refer to the formal definition Table 1 in details.

## 3.2 Discussion on various robustness metrics

In Section 1, we have proposed the our robustness metric: perturbation stability (*cf.*, Definition 1). This metric can be viewed as an expectation of inner product of first-order approximation of adversarial risk [Madry et al., 2018] and the perturbations with uniform distribution, which measures the *average robustness* of the neural network. As we mentioned before, under the same perturbation radius $\epsilon$, smaller value of $\mathcal{P}(\boldsymbol{f}, \epsilon)$ implies better *perturbation stability*, that is better *average robustness*. Previous works [Hein and Andriushchenko, 2017, Weng et al., 2018, Wu et al., 2021, Bubeck and Sellke, 2021] use Lipschitzness to describe the robustness of the network, suggesting that smaller Lipschitzness leads to robust models. However, Lipschitzness is only a worst-case measure, and cannot reasonably describe the average changes of the entire dataset. Instead, we follow the measure

of Wu et al. [2021], Dohmatob and Bietti [2022], that aims to comprehensively considers the overall distribution of the data, not only the extreme case. Besides, the worst-case robustness can be extended to a probabilistic robustness view [Robey et al., 2022], which shares a similar spirit of our average robustness concept. Schmidt et al. [2018] presents another definition of robustness, depending on the misclassified error of an adversarial data point. Instead, our perturbation stability measures the function value changes at the clean data point via Taylor expansion which can exclude the influence of the learning capacity of the network.

# 4 Main results

In this section, we state the main theoretical results. Firstly, in Section 4.1 we provide the upper bound of the *perturbation stability* in lazy training regime for deep neural networks defined by Eq. (2). The sufficient condition that the neural network (2) is under non-lazy training regime is given in Section 4.2. Finally, in Section 4.3, we provide the upper bound on the *perturbation stability* during early training of two-layer network under non-lazy training regime.

## 4.1 Upper bound of the perturbation stability of DNNs under the lazy training regime

We are now ready to state the main results under the lazy training regime. The following theorem provides the upper bound of the *perturbation stability* and connects to the width, and the depth of a deep fully-connected neural network under different standard Gaussian initialization. The proof of Theorem 1 is deferred to Appendix B.

**Theorem 1.** *Given an $L$-layer neural network $\boldsymbol{f}$ defined by Eq.* (2) *trained by $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ satisfying Assumption 1, for the convenience of analysis, we set $\alpha = 1$ and $\beta := \beta_2 = \cdots = \beta_{L-1}$ , define a constant $\gamma := \beta/\sqrt{\frac{2}{m}}$, then under a small perturbation $\epsilon$ , we have the following:*

$$\frac{\mathscr{P}(\boldsymbol{f}, \epsilon)}{\epsilon} \lesssim \left(\sqrt{L^3 m^2 \beta_1^2 \beta_L^2} e^{-m/L^3} + \sqrt{mo\beta_1^2\beta_L^2}\right)\gamma^{L-2}\,. \tag{3}$$

**Remark:** Our results cover the effect of the width and depth of neural network on robustness under various common initializations depending on $\gamma \gtreqless 1$.

**Comparison with three commonly used initializations:** For the initializations used in practice, our theoretical results can be mainly divided into two classes. 1) Depth helps robustness in an exponential order under LeCun initialization: our Theorem 1 implies $\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{o}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$. 2) Depth hurts robustness in a polynomial order under He-initialization $\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{o}{d}}\right)$ and under NTK initialization $\left(\sqrt{\frac{L^3 m}{o}} e^{-m/L^3} + 1\right)$ derived by Theorem 1. When employing other initializations, robustness could be hurted in a exponential order. Below, we elaborate on these three initalizations:

**1) LeCun initialization** ($\gamma = \frac{\sqrt{2}}{2}$): The order has three main parts $\sqrt{\frac{L^3 m}{d}}$, $e^{-m/L^3}$ and $(\frac{\sqrt{2}}{2})^{L-2}$. Regarding the width $m$, the first part $\sqrt{\frac{L^3 m}{d}}$ is an increasing function of $m$ and the second part $e^{-m/L^3}$ is a decreasing function of $m$. Accordingly, in the under-parameterized region (e.g., $m$ is small), $\sqrt{\frac{L^3 m}{d}}$ plays a major role, so the stability will increase as $m$ increases. After a critical point, $e^{-m/L^3}$ plays a major role, so the stability will decrease as $m$ increases. When $m$ tends to infinity, the first term of bound tends to 0. Hence the *perturbation stability* tends to be a constant and independent with width $m$ as the width $m$ tends to infinity. That means, there exists a phase transition phenomenon between the *perturbation stability* and over-parameterization in terms of the width $m$.

Regarding the depth $L$, the constant $\gamma = \frac{\sqrt{2}}{2}$ implies that the third part has a faster decrease speed than the first and second part and play a major role in the tendency. The *perturbation stability* of the neural network exponentially decreases with respect to the depth. That means, for LeCun initialization, the deeper the network, the better robustness. Nevertheless, the energy of LeCun initialization decreases as the network depth increases due to the variance $\beta = \sqrt{\frac{1}{m}}$ and the activation function ReLU that

Table 2: Comparison of the orders of the proposed bound with other two recent works. Our results are general to cover both under- and over-parameterized regimes, which expands the application scope of previous results [Wu et al., 2021, Huang et al., 2021]. (The original result of [Wu et al., 2021] can be reduced to $\sqrt{mL}$ as the $\frac{m}{(\log m)^6} \geq L^{12}$ condition is required).

| Metrics | Our result | Wu et al. [2021] | Huang et al. [2021] |
|---|---|---|---|
| $\mathscr{P}(\boldsymbol{f}, \epsilon)/\epsilon$ | $\sqrt{\frac{L^3 m}{o}} e^{-m/L^3} + 1$ | $L^2 m^{1/3} \sqrt{\log m} + \sqrt{mL}$ | $2^{\frac{3L-5}{2}} \sqrt{m}$ |

loses half of energy every layer, which means that training deep network with LeCun initialization is very difficult. Hence we need a trade off between robustness and training difficulty regarding the network depth in practice for LeCun initialization.

**2) He initialization and NTK initialization** ($\gamma = 1$): the bounds for these two initializations are almost the same, and only differ in the feature dimension. We can see that phase transition phenomena exist under these two initializations regarding the width $m$, similar to LeCun initialization. Regarding the depth, when $L$ is large, the first part $\sqrt{L^3}$ plays a major role in the *perturbation stability*. So these two initializations hurt the robustness of the neural network at a polynomial order.

All of the three initializations admit $\gamma \leq 1$. If some initialization schemes admit $\gamma > 1$, then the depth $L$ will hurt robustness of the neural network at an exponentially increasing rate.

**Comparison with previous work:** Theorem 1 provides a new relationship between the robustness with width and depth of DNNs. We compare our result with [Wu et al., 2021, Huang et al., 2021] using a basic NTK initialization [Allen-Zhu et al., 2019] (suppose that $m \gg o$ and $m \gg d$). For better comparison, we derive their results under our robustness metric *perturbation stability*, reported by Table 2.

Our results indicate a behavior transition on the width. For the over-parameterized regime, the robustness of neural network only depends on the perturbation energy, and it is almost independent of the width $m$. The results on width are significantly better than the previous results increasing as the square root of $m$. For depth $L$, our results provide a tighter and more precise estimate as compared to [Wu et al., 2021] in a two-degree polynomial increasing order and [Huang et al., 2021] in an exponential increasing order.

Furthermore, compared with the results of Bubeck et al. [2021] showing that the robustness of the two-layer neural network becomes better with the increase of the number of neurons (i.e., width), we provide more detailed and refined results on the robustness of DNNs under different initializations and under-parameterized regimes.

### 4.2 Sufficient condition for neural network under non-lazy training regime

Beyond the lazy training regime, we turn our attention to the non-lazy training regime and provide a sufficient condition for a (well-chosen) initialization of neural networks when entering into the non-lazy training regime. This is a first attempt to understanding training dynamics of DNNs in this regime.

Our result requires a further assumption on the data and the empirical risk as follows.

**Assumption 2.** *For a single-output network defined in Eq.* (2)*, we assume that* $\max_{i \in [n]} y_i \geq C_1 > 0$ *for some constant* $C_1$*. We also assume that the neural network can be well-trained such that the empirical risk is* $\mathcal{O}(\frac{1}{n})$*.*

**Remark:** This is a common assumption in the field of optimization [Song et al., 2021] in the under- and over-parameterized regime, and we can even assume zero risk. Here we follow the specific assumption of Luo et al. [2021].

Now we are ready to present our result: a sufficient condition to identify when deep ReLU neural networks fall into non-lazy training regime, as a promising extension of Luo et al. [2021] on two-layer neural networks. To avoid cluttering the analysis, we assume a single-output i.e., $o = 1$. The proof of Theorem 2 is deferred to Appendix C.

**Theorem 2.** *Given an $L$-layer neural network $\boldsymbol{f}$ defined by Eq.* (2) *with $o = 1$, trained by $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, under Assumptions 1 and 2, suppose that $\alpha \gg (m^{3/2}\sum_{i=1}^L \beta_i)^L$ and $m \gg d$ in Eq.* (2)*, then for sufficiently large $m$, with probability at least $1 - (L-2)\exp(-\Theta(m^2)) - \exp(-\Theta(md)) - \exp(-\Theta(m))$ over the initialization, we have:*

$$\sup_{t\in[0,+\infty)} \frac{\|\boldsymbol{W}_l(t) - \boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}} \gg 1\,, \quad \forall l \in [L]\,.$$

**Remark:** The condition $\alpha \gg (m^{3/2}\sum_{i=1}^L \beta_i)^L$ implies that, a neural network falls in a non-lazy training regime when the variance of the Gaussian initialization is very small. It can be achieved by a typical case: taking $m \gg L^2$, choosing $\alpha = 1$ and $\forall l \in [L]$; $\beta_l = \frac{1}{m^2}$. Commonly used initializations such as NTK initialization, LeCun initialization, He's initialization lead to lazy training.

### 4.3 Upper bound of the perturbation stability for two-layer networks in non-lazy training

Unlike lazy training, weights of non-lazy training concentrate on few directions determined by the input data in the early stages of training [Luo et al., 2021]. The following theorem describes the neural network *perturbation stability* in the early training stage as a function of network width in the non-lazy training regime. For ease of description, here we consider a special initialization scheme under the non-lazy regime, the proof of Theorem 3 is deferred to Appendix D.

**Theorem 3.** *Given a two-layer neural network with single output $f_t$ defined by Eq.* (2) *and trained by $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ satisfying Assumption 1, using gradient descent under the squared loss, consider the following initialization in Eq.* (2) *: $L = 2$, $\alpha \sim 1$, $\beta_1 \sim \beta_2 \sim \beta \sim \frac{1}{m^c}$ with $c \geq 1.5$, $m \gg n^2$ and training time less than a constant that only depends on $n, m$ and $\lambda_0$ , then for a small range of perturbation $\epsilon$, with probability at least $1 - n\exp(-\frac{n}{2}) - \frac{3}{n}$ over initialization, we have the following:*

$$\frac{\mathscr{P}(f_t, \epsilon)}{\epsilon} \leq \Theta\left(\frac{\sqrt{n\log m} + n}{m^{c-1}}\left(\frac{1}{\sqrt{n^3 m}} + \frac{1}{m^{c-0.5}}\right)\right). \tag{4}$$

**Remark:** Under this setting of non-lazy training regime, the robustness and width of the neural network are positively correlated in the early stages of training. That is, as the width $m$ increases in the over-parameterized regime, a Gaussian initialization with smaller variance leads to the robustness increasing in a faster decay. Our result holds for other initialization schemes in the non-lazy training regime, e.g., $c = 2$ leads to $\mathscr{P}(f_t, \epsilon)/\epsilon \leq \Theta\left(\frac{\sqrt{n\log m}+n}{m^{2.5}}\right)$; and $c = 3$ leads to $\mathscr{P}(f_t, \epsilon)/\epsilon \leq \Theta\left(\frac{\sqrt{n\log m}+n}{m^{4.5}}\right)$.

## 5 Numerical evidence

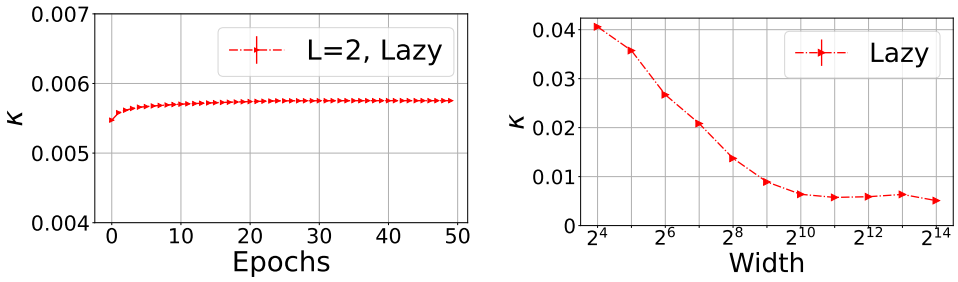
We validate our theoretical results with a series of experiments. In Section 5.2 we first verify that our initialization settings belong in the lazy and the non-lazy training regimes. In Section 5.3, we then explore the effect of varying widths from under-parameterized to over-parameterized regions on the *perturbation stability* of neural networks. In Section 5.4, we finally compare the effect of two different initializations and the network depth on the *perturbation stability*. Additional experimental results can be found in Appendix E.

### 5.1 Experimental settings

Here we present our experimental setting including models, hyper-parameters, the choice of width and depth, and initialization schemes. We use the popular datasets of MNIST [Lecun et al., 1998] and CIFAR-10 [Krizhevsky et al., 2014] for experimental validation.

**Models:** We report results using the following models: fully connected ReLU neural network named "FCN" in main paper and convolutional ReLU neural network named "CNN" in Appendix E.

**Hyper-parameters:** Unless mentioned otherwise, all models are trained for 50 epochs with a batch size of 64. The initial value of the learning rate is 0.001. After the first 25 epochs, the learning rate is




(a) (b)

Figure 2: (a) Tendency with respect to time (training epochs) and (b) relationship between width and lazy training ratio of neural networks. Fig. 2(a) shows that ratio $\kappa$ is small and almost unchanged, recognized as *lazy training*. In Fig. 2(b), we can see that the $\kappa$ decreases with the increasing width.

multiplied by a factor of $0.1$ every $10$ epochs. The SGD is used to optimize all the models, while the cross-entropy loss is used.

**Width and depth:** In order to verify our theoretical results, we conduct a series of experiments with different depths and widths of the same type neural network. Specifically, our experiments include $11$ different widths from $2^4$ to $2^{14}$, and four different choices of depths, i.e., $2, 4, 6, 8, 10$.

**Initialization:** We report results using the following initializations: 1) He initialization where $W_{ij} \sim \mathcal{N}(0, \frac{2}{m_{in}})$, 2) LeCun initialization where $W_{ij} \sim \mathcal{N}(0, \frac{1}{m_{in}})$ and 3) An initialization allows for non-lazy training regime on two-layer networks, i.e., $\beta_1 = \beta_2 = 1/m^2$ and $\alpha = 1$.

### 5.2 Validation for lazy and non-lazy training regimes

Before verifying our results, we need identify the lazy and non-lazy training regime under different initializations. To this end, we define a measure *lazy training ratio*, i.e., $\kappa = \frac{\sum_{l=1}^{L} \|\boldsymbol{W}_l(t) - \boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\sum_{l=1}^{L} \|\boldsymbol{W}_l(0)\|_{\mathrm{F}}}$ to measure whether the neural network is under the lazy training regime. A smaller $\kappa$ implies that the neural network is more close to lazy training.

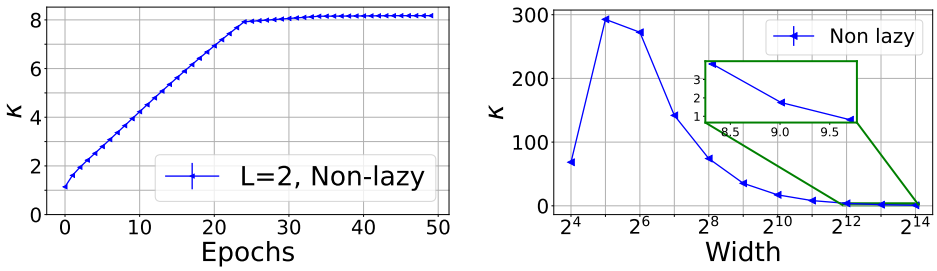
According to the theory, we employ the He initialization and the non lazy training initialization we state in Section 5.1 to conduct the experiment under two-layer neural networks to verify that their lazy training ratio matches the theoretical results of lazy training and non-lazy training (i.e. the experiment is under the correct regime). Fig. 2(a) and Fig. 2(b) show the tendency of ratio with respect to time (training epochs) and relationship between width and lazy training ratio of neural networks under lazy training regime, respectively. We can find that the ratio of lazy traing regime is almost a constant that does not change with time, and this constant decreases with the width of network increases. This is in line with what we know about lazy training [Chizat et al., 2019].

Likewise, Fig. 3 shows the ratio tendency with time and width under non-lazy training regime. The ratio increases almost linearly over time in Fig. 3(a). In epoch $25$ we decrease the learning rate, which results in the increase rate of $\kappa$ changes. At the same time Fig. 3(b) shows the similar tendency between width and lazy training ratio as lazy training. However, the value of $\kappa$ is much higher than that of lazy training regime. Combining the results about tendency with time, the ratio will be expected to increase as time at a slow decay and tend to infinity under highly wide width setting.

### 5.3 Validation for width

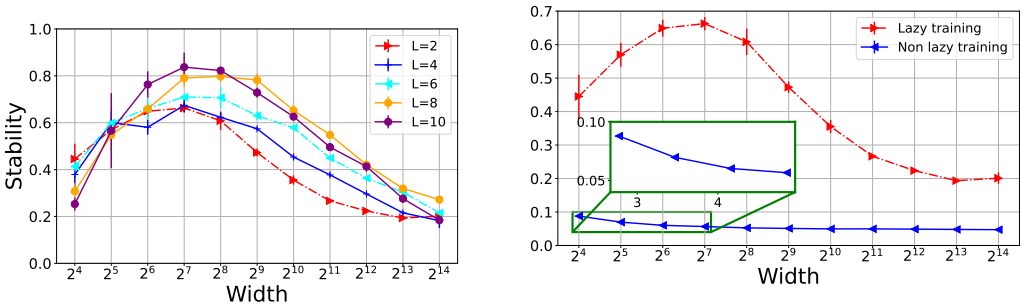
We verify the relationship between the *perturbation stability* and the width of network as illustrated by Eqs. (3) and (4). We conduct a series of experiments on MNIST dataset using FCN with different widths. Fig. 4 shows the relationship between the *perturbation stability* and width of FCN with different depths and training regimes. Here for lazy training and non-lazy training we use the same initialization as Section 5.2.

Fig. 4(a) exhibits the relationship between the *perturbation stability* and the width for neural networks with different depths of $L = 2, 4, 6, 8$, and $10$. All of the five curves show the phase transition with width, and the *perturbation stability* first increases and then decreases with width, which match our




(a) (b)

Figure 3: (a) Tendency with respect to time (training epochs) and relationship with width of non-lazy training ratio of neural networks. Fig. 3(a) shows that ratio $\kappa$ is changed a lot (increasing and then remains unchanged), recognized as *non-lazy training*. The tendency of $\kappa$ for non-lazy training is increasing with the width and then decreasing, i.e., a phase transition in Fig. 3(b).




(a) Lazy training with different widths and depths (b) Lazy and non-lazy training under different widths

Figure 4: Influence of width of neural network on the *perturbation stability*. (a) phase transition of the *perturbation stability vs.* width with five different depths under lazy training. (b) the difference between lazy training and non-lazy training regimes for two layer neural networks.

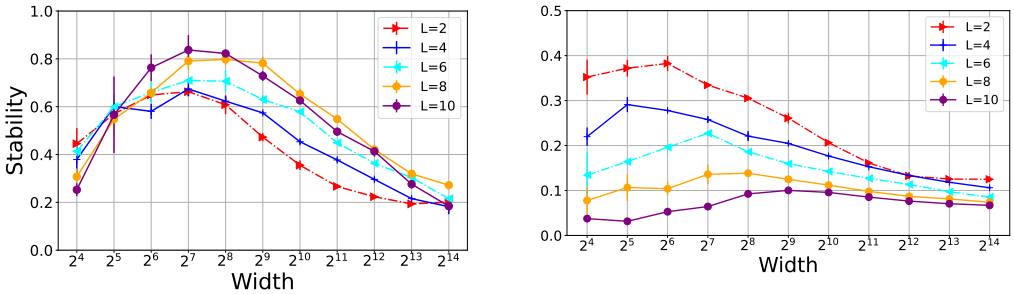
theoretical results. Fig. 4(b) shows the difference of the effect of width on the *perturbation stability* of lazy and non-lazy training for two-layer neural networks. First, the *perturbation stability* of non-lazy training is significantly smaller than that of lazy training, which means non-lazy training regime is more robust. Besides, the *perturbation stability* of non-lazy training decrease with the width of the neural network increases, which coincides with our theoretical result, i.e., no phase transition phenomenon.

### 5.4 Validation for depth and initialization

Let us explore the effect of depth on the *perturbation stability* under lazy training regime with different initializations in Fig. 5(a) and Fig. 5(b). Our results show the tendency of the *perturbation stability* for FCN with different widths and depths under He initialization and LeCun initialization, respectively. We observe a similar phase transition phenomenon, and find that, the *perturbation stability* under He initialization increases with depth, while the LeCun initialization shows the opposite tendency, which verified our theory.

## 6 Conclusions

In this work, we explore the interplay of the width, the depth and the initialization of neural networks on their average robustness with new theoretical bounds in an effort to address the apparent contradiction in the literature. Our theoretical results hold in both under- and over-parameterized regimes. Intriguingly, we find a change of behavior in average robustness with respect to the depth, initially exacerbating robustness and then alleviating it. We suspect that this could help explain the contradictory messages in the literature. We also characterize the average robustness in the non-lazy training regime for two layer neural networks and find that width always help, coinciding the with




(a) (b)

Figure 5: Relationship between the *perturbation stability* and depth of FCN under He initialization (a) and LeCun initialization (b) different depths of $L = 2, 4, 6, 8$ and $10$.

the results [Bubeck and Sellke, 2021, Bubeck et al., 2021]. We also provide numerical evidence to support the theoretical developments.

## Acknowledgements

We are also thankful to the reviewers for providing constructive feedback. Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-19-1-0404. 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). Corresponding authors: Fanghui Liu and Zhenyu Zhu.

## References


Z. Allen-Zhu and Y. Li. Feature purification: How adversarial training performs robust deep learning. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

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

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

A. Bietti and J. Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

S. Bubeck and M. Sellke. A universal law of robustness via isoperimetry. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

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.

L. Chen and S. Xu. Deep neural tangent kernel and laplace kernel have the same 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 and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020.

E. Dohmatob and A. Bietti. On the (non-)robustness of two-layer neural networks in different learning regimes, 2022.

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, W. Hu, and J. D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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.

K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

R. Gao, T. Cai, H. Li, C.-J. Hsieh, L. Wang, and J. D. Lee. Convergence of adversarial training in overparametrized neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A. Geifman, A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and R. Basri. On the similarity between the laplace and neural tangent kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

H. Hassani and A. Javanmard. The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression, 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)*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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.

A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

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*. 2012.

T. Luo, Z.-Q. J. Xu, Z. Ma, and Y. Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 2021.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

H. Maennel, O. Bousquet, and S. Gelly. Gradient descent quantizes relu network features, 2018.

E. Malach, G. Yehudai, S. Shalev-Schwartz, and O. Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning (ICML)*, 2020.

S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layers neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 2018.

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, 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.

S. Oymak and M. Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 2020.

A. Robey, L. F. O. Chamon, G. J. Pappas, and H. Hassani. Probabilistically robust learning: Balancing average- and worst-case performance. In *International Conference on Machine Learning (ICML)*, 2022.

I. M. Safran, G. Yehudai, and O. Shamir. The effects of mild over-parameterization on the optimization landscape of shallow relu neural networks. In *Conference on Learning Theory (COLT)*, 2021.

L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

C. Song, A. Ramezani-Kebrya, T. Pethick, A. Eftekhari, and V. Cevher. Subquadratic overparameterization for shallow neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. 2018.

T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations (ICLR)*, 2018.

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory (COLT)*, 2020.

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.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

X. Zhang, J. Chen, Q. Gu, and D. Evans. Understanding the intrinsic robustness of image distributions using conditional generative models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020a.

Y. Zhang, O. Plevrakis, S. S. Du, X. Li, Z. Song, and S. Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

M. Zhou, R. Ge, and C. Jin. A local convergence theory for mildly over-parameterized two-layer neural network. In *Conference on Learning Theory (COLT)*, 2021.

S. Zhou and A. P. Schoellig. An analysis of the expressiveness of deep neural network architectures based on their lipschitz constants, 2019.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We clearly discuss the limitation of this work in the Appendix F.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We clearly discuss the societal impact of this work in the Appendix G.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We have detailed the rationale of the applied assumptions in this paper.
   (b) Did you include complete proofs of all theoretical results? [Yes] We include the complete proofs of our theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The datasets we use in this work are all in the public domain and standard for image-related tasks. Thus, our setup can be reproduced by interested practitioners. The code will be open-sourced upon the acceptance of the paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Our experiments are mainly for validation our theory, all of the settings and hyper-parameters are mentioned in the main paper.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All our experiments are conducted on a **single** GPU in our internal cluster.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all the corresponding papers that provide the assets we use.
   (b) Did you mention the license of the assets? [Yes] All the datasets used in this work are publicly available datasets. All the datasets enable their use for research purposes.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets we utilize are publicly available and contain tens or hudrends of thousands of images. We refer to the authors original papers describing the datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Our experiments rely on a) synthetic data, b) public benchmarks that do not contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

## 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 the proofs and related lemmas of Theorem 1.
- In Appendix C, we provide the proofs of Theorem 2.
- In Appendix D, we provide the proofs and related lemmas of Theorem 3.
- In Appendix E, we detail our experimental settings and exhibit additional experimental results.
- In Appendix F, we discuss several limitations of this work.
- Finally, in Appendix G, we discuss the societal impact of this paper.

## A Symbols and Notation

In the paper, vectors are indicated with bold small letters, matrices with bold capital letters. To facilitate the understanding of our work, we include the some core symbols and notation in Table 3.

Table 3: Core symbols and notations used in this project.

| Symbol | Dimension(s) | Definition |
|---|---|---|
| $\mathcal{N}(\mu, \sigma)$ | - | Gaussian distribution of mean $\mu$ and variance $\sigma$ |
| $\mathrm{Ber}(m, p)$ | - | Bernoulli (Binomial) distribution with $m$ trials and $p$ success rate. |
| $\chi^2(\omega)$ | - | Chi-square distribution of degree $\omega$. |
| $\|\boldsymbol{v}\|_2$ | - | Euclidean norms of vectors $\boldsymbol{v}$ |
| $\|\boldsymbol{M}\|_2$ | - | Spectral norms of matrices $\boldsymbol{M}$ |
| $\|\boldsymbol{M}\|_{\mathrm{F}}$ | - | Frobenius norms of matrices $\boldsymbol{M}$ |
| $\|\boldsymbol{M}\|_*$ | - | Nuclear norms of matrices $\boldsymbol{M}$ |
| $\lambda(\boldsymbol{M})$ | - | Eigenvalues of matrices $\boldsymbol{M}$ |
| $\boldsymbol{M}^{[l]}$ | - | $l$-th row of matrices $\boldsymbol{M}$ |
| $\boldsymbol{M}_{i,j}$ | - | $(i, j)$-th element of matrices $\boldsymbol{M}$ |
| $\phi(x) = \max(0, x)$ | - | 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 $A$ |
| $n$ | - | Size of the dataset |
| $d$ | - | Input size of the network |
| $o$ | - | Output size of the network |
| $L$ | - | Depth of the network |
| $m$ | - | Width of intermediate layer |
| $\beta_l$ | - | Standard deviation of Gaussian initialization of $l$-th intermediate layer |
| $\alpha$ | - | Scale factor for the output layer |
| $\boldsymbol{x}_i$ | $\mathbb{R}^d$ | The $i$-th data point |
| $\boldsymbol{y}_i$ | $\mathbb{R}^o$ | The $i$-th target vector |
| $\mathcal{D}_X$ | - | Input data distribution |
| $\mathcal{D}_Y$ | - | Target data distribution |
| $\boldsymbol{W}_1$ | $\mathbb{R}^{m \times d}$ | Weight matrix for the input layer |
| $\boldsymbol{W}_l$ | $\mathbb{R}^{m \times m}$ | Weight matrix for the $l$-th hidden layer |
| $\boldsymbol{W}_L$ | $\mathbb{R}^{o \times m}$ | Weight matrix for the output layer |
| $\boldsymbol{h}_{i,l}$ | $\mathbb{R}^m$ | The $l$-th layer activation for input $\boldsymbol{x}_i$ |
| $\boldsymbol{f}_i$ | $\mathbb{R}^o$ | Output of network for input $\boldsymbol{x}_i$ |
| $\mathcal{O}$, $o$, $\Omega$ and $\Theta$ | - | Standard Bachmann–Landau order notation |
| $\mathbb{P}(A)$ | - | Probability of event $A$ |

## B Proof of upper bound of the Perturbation Stability in lazy training regime for deep neural network

We present the details of our results from Section 4.1 in this section. Firstly, we introduce some lemmas in Appendix B.1 to facilitate the proof of theorems. Then, in Appendix B.2 we provide the proof of Theorem 1.

### B.1 Relevant Lemmas

**Lemma 1.** *Let $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_n)$. Then, for two fixed non-zero vectors $\boldsymbol{h}_1 \in \mathbb{R}^n$ and $\boldsymbol{h}_2 \in \mathbb{R}^n$, define two random variables $X = (\boldsymbol{w}^\top \boldsymbol{h}_1 1\left\{\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0\right\})^2$ and $Y = s(\boldsymbol{w}^\top \boldsymbol{h}_1)^2$, where $s \sim \mathrm{Ber}(1, 1/2)$ follows a Bernoulli distribution with $1$ trial and $\frac{1}{2}$ success rate. Then $X$ and $Y$ have the same distribution, denoted as $X \overset{d}{=} Y$.*

*Proof.* Firstly, we derive the cumulative distribution function (CDF) of $X$. Obviously, $X$ is non-negative and $\boldsymbol{w}^\top \boldsymbol{h}_1 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \|\boldsymbol{h}_1\|_2^2 \mathbb{I}_n)$, and then we have:

$$\mathbb{P}(X = 0) = \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0) + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, \boldsymbol{w}^\top \boldsymbol{h}_1 = 0)\,,$$

which implies:

$$\mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0) \leq \mathbb{P}(X = 0) \leq \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0) + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_1 = 0) = \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0)\,,$$

leading to $\mathbb{P}(X = 0) = \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0) = 1/2$.

Accordingly, for $x \geq 0$ , we have:

$$\begin{aligned}
\mathbb{P}(X \leq x) &= \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 < 0) + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, -\sqrt{x} \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) \\
&= \frac{1}{2} + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, -\sqrt{x} \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) \\
&= \frac{1}{2} + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, -\sqrt{x} \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq 0) + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, 0 \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) \\
&= \frac{1}{2} + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \leq 0, 0 \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) + \mathbb{P}(\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0, 0 \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) \\
&= \frac{1}{2} + \mathbb{P}(0 \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) \\
&= \frac{1}{2} + \int_0^{\sqrt{x}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{t^2}{2\sigma_1^2}} \mathrm{d}t\,,
\end{aligned}$$

where we use the symmetry of the Gaussian random variable and $\sigma_1 = \|\boldsymbol{h}_1\|_2 \sigma$.

Then $X$ admits the following cumulative distribution function:

$$F(X \leq x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \\ \frac{1}{2} + \int_0^{\sqrt{x}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{t^2}{2\sigma_1^2}} dt & \text{if } x > 0\,. \end{cases} \tag{5}$$

We then derive the CDF of $Y$. Obviously, $Y$ is non-negative and $\mathbb{P}(Y = 0) = 1/2$, which holds by $Y = 0$ iff $s = 0$. Accordingly, for $x \geq 0$ , we have:

$$\begin{aligned}
\mathbb{P}(Y \leq x) = \mathbb{P}(s = 0) + \mathbb{P}(s = 1)\mathbb{P}(-\sqrt{x} \leq \boldsymbol{w}^\top \boldsymbol{h}_1 \leq \sqrt{x}) &= \frac{1}{2} + \frac{1}{2} \int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{t^2}{2\sigma_1^2}} \mathrm{d}t \\
&= \frac{1}{2} + \int_0^{\sqrt{x}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{t^2}{2\sigma_1^2}} \mathrm{d}t\,.
\end{aligned}$$

Then $Y$ has the following cumulative distribution function:

$$F(Y \leq x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \\ \frac{1}{2} + \int_0^{\sqrt{x}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{t^2}{2\sigma_1^2}} dt & \text{if } x > 0\,, \end{cases} \tag{6}$$

which implies $X \overset{d}{=} Y$ by comparing Eq. (5) and Eq. (6). □

**Lemma 2.** *Given two fixed non-zero vectors $\boldsymbol{h}_1 \in \mathbb{R}^p$ and $\boldsymbol{h}_2 \in \mathbb{R}^p$, let $\boldsymbol{W} \in \mathbb{R}^{q \times p}$ be random matrix with i.i.d. entries $\boldsymbol{W}_{i,j} \sim \mathcal{N}(0, 2/q)$ and a vector $\boldsymbol{v} = \phi'(\boldsymbol{W}\boldsymbol{h}_2)\boldsymbol{W}\boldsymbol{h}_1 \in \mathbb{R}^q$, then, we have $\frac{q\|\boldsymbol{v}\|_2^2}{2\|\boldsymbol{h}_1\|_2^2} \sim \chi^2(\varrho)$, where $\varrho \sim \mathrm{Ber}(q, 1/2)$.*

*Proof.* According to the definition of $\boldsymbol{v} = \phi'(\boldsymbol{W}\boldsymbol{h}_2)\boldsymbol{W}\boldsymbol{h}_1 \in \mathbb{R}^q$, we have:

$$\|\boldsymbol{v}\|_2^2 = \sum_{i=1}^{q} \left( \boldsymbol{D}_{i,i} \left\langle \boldsymbol{W}^{[i]}, \boldsymbol{h}_1 \right\rangle \right)^2,$$

where $\boldsymbol{D}_{i,i} = 1\left\{\left\langle \boldsymbol{W}^{[i]}, \boldsymbol{h}_2 \right\rangle \geq 0\right\}$, the $\boldsymbol{W}^{[i]}$ is defined in the second part of the Table 3.

Let $\varpi_i = \left\langle \boldsymbol{W}^{[i]}, \boldsymbol{h}_1 \right\rangle / \left( \sqrt{\frac{2\|\boldsymbol{h}_1\|_2^2}{q}} \right)$, then $\varpi_i \sim \mathcal{N}(0,1)$ independently. Accordingly, by Lemma 1, recall $s \sim \mathrm{Ber}(1, 1/2)$, we have:

$$\frac{q\,\|\boldsymbol{v}\|_2^2}{2\,\|\boldsymbol{h}_1\|_2^2} = \sum_{i=1}^{q} \left( 1\left\{ \left\langle \boldsymbol{W}^{[i]}, \boldsymbol{h}_2 \right\rangle \geq 0 \right\} \varpi_i \right)^2 \overset{d}{=} \sum_{i=1}^{q} s\varpi_i^2\,,$$

which implies $\frac{q\|\boldsymbol{v}\|_2^2}{2\|\boldsymbol{h}_1\|_2^2} \sim \chi^2(\varrho)$ with $\varrho \sim \mathrm{Ber}(q, 1/2)$ according to the definition of chi-square distribution.

□

**Lemma 3.** *(Dynamic equivalence under different scaling) Given an $L$-layer neural network $\boldsymbol{f}$ defined by Eq.* (2)*, as follows:*

$$\boldsymbol{f}(\boldsymbol{x}) = \widehat{\boldsymbol{W}}_L \phi(\widehat{\boldsymbol{W}}_{L-1} \cdots \phi(\widehat{\boldsymbol{W}_1}\boldsymbol{x}) \cdots)\,, \tag{7}$$

*where $[\widehat{\boldsymbol{W}}_l]_{i,j}$ satisfy the initialization in Section 3.1, i.e., $\beta := \beta_2 = \cdots = \beta_{L-1}$.*

*Scaling all weights of $\boldsymbol{f}$, than we get a new model $\widetilde{\boldsymbol{f}}$ as follows.*

$$\widetilde{\boldsymbol{f}}(\boldsymbol{x}) = \gamma^L \widetilde{\boldsymbol{W}}_L \phi(\widetilde{\boldsymbol{W}}_{L-1} \cdots \phi(\widetilde{\boldsymbol{W}_1}\boldsymbol{x}) \cdots)\,, \tag{8}$$

*where $[\widetilde{\boldsymbol{W}}_l]_{i,j} = [\widehat{\boldsymbol{W}}_l]_{i,j}/\gamma \quad \forall l \in [L]$.*

*Then if we choose an appropriate learning rate $\widetilde{\eta} := \frac{\eta}{\gamma^2}$ , $\boldsymbol{f}$ and $\widetilde{\boldsymbol{f}}$ will have the same dynamics.*

*Proof.* According to the chain rule, we have:

$$\frac{\mathrm{d}\widetilde{\boldsymbol{f}}}{\mathrm{d}\widetilde{\boldsymbol{W}}_l} = \gamma \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\widehat{\boldsymbol{W}}_l} \quad \forall l \in [L]\,.$$

If we choose learning rate $\widetilde{\eta} := \frac{\eta}{\gamma^2}$, then, we have:

$$\frac{\mathrm{d}\widetilde{\boldsymbol{W}}_l}{\mathrm{d}t} = \frac{1}{\gamma} \frac{\mathrm{d}\widehat{\boldsymbol{W}}_l}{\mathrm{d}t} \quad \forall l \in [L]\,.$$

Consider that $\widetilde{\boldsymbol{W}}_l(0) = \frac{1}{\gamma}\widehat{\boldsymbol{W}}_l(0) \quad \forall l \in [L]$, then, we have:

$$\widetilde{\boldsymbol{W}}_l(t) = \frac{1}{\gamma}\widehat{\boldsymbol{W}}_l(t) \quad \forall l \in [L]\,.$$

That means $\boldsymbol{f}(t) = \widetilde{\boldsymbol{f}}(t) \quad \forall t \geq 0$, which concludes the proof. □

**Lemma 4.** *Given an L-layer neural network $\boldsymbol{f}$ defined by Eq.* (2) *trained by $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, under a small perturbation $\epsilon$, we have:*

$$\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\| \nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})^\top (\boldsymbol{x} - \hat{\boldsymbol{x}}) - \boldsymbol{W}_L \boldsymbol{D}_{L-1} \boldsymbol{W}_{L-1} \cdots \boldsymbol{D}_1 \boldsymbol{W}_1 (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2 \leq \Theta\left( \epsilon \gamma^{L-2} \sqrt{\frac{\pi L^3 m^2 \beta_1^2 \beta_L^2}{8}} e^{-m/L^3} \right), \tag{9}$$

*where $[\boldsymbol{W}_l]_{i,j}$ satisfy the initialization in Section 3.1, $\boldsymbol{x} \sim \mathcal{D}_X$ and $\hat{\boldsymbol{x}} \sim$ Unif$(\mathbb{B}(\epsilon, \boldsymbol{x}))$.*

*Proof.* We set the weight of the neural network after training are $\widehat{\boldsymbol{W}}$. i.e.

$$\boldsymbol{f}(\boldsymbol{x}) = \widehat{\boldsymbol{W}}_L \phi(\widehat{\boldsymbol{W}}_{L-1} \cdots \phi(\widehat{\boldsymbol{W}_1} \boldsymbol{x}) \cdots).$$

According to the standard chain rule and Lemma 3, we have:

$$\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})^\top = \widehat{\boldsymbol{W}}_L \widehat{\boldsymbol{D}}_{L-1} \widehat{\boldsymbol{W}}_{L-1} \cdots \widehat{\boldsymbol{D}}_1 \widehat{\boldsymbol{W}}_1 = \gamma^L \widehat{\boldsymbol{W}}'_L \widehat{\boldsymbol{D}}_{L-1} \widehat{\boldsymbol{W}}'_{L-1} \cdots \widehat{\boldsymbol{D}}_1 \widehat{\boldsymbol{W}}'_1,$$

where $[\widehat{\boldsymbol{W}}'_l]_{i,j} = [\widehat{\boldsymbol{W}}_l]_{i,j}/\gamma \quad \forall l \in [L]$.

Assume that the perturbation matrices satisfy $\left\| \widehat{\boldsymbol{W}}_l - \boldsymbol{W}_l \right\|_2 \leq \omega$, $\forall l \in [L]$, where the parameter $\omega$ will be determined later. Then by Allen-Zhu et al. [2019, Lemma 7.4, Lemma 8.6, Lemma 8.7], we obtain that for any integer $s \in \left[\Omega(\frac{d}{\log m}), \mathcal{O}(\frac{m}{L^3 \log m})\right]$, for $d \leq \mathcal{O}(\frac{m}{L \log m})$, with probability at least $1 - \exp\Big( - \Omega(s \log m)\Big)$ over the randomness of $\{\boldsymbol{W}\}_{l=1}^L$, it holds that:

$$\left\| \widehat{\boldsymbol{W}}'_L \widehat{\boldsymbol{D}}_{L-1} \widehat{\boldsymbol{W}}'_{L-1} \cdots \widehat{\boldsymbol{D}}_1 \widehat{\boldsymbol{W}}'_1 - \boldsymbol{W}'_L \boldsymbol{D}_{L-1} \boldsymbol{W}'_{L-1} \cdots \boldsymbol{D}_1 \boldsymbol{W}'_1 \right\|_2 \leq \mathcal{O}\left( \sqrt{\frac{L^3 s \log m + \omega^2 L^3 m}{d}} \sqrt{\frac{dm}{2}} \frac{\beta_1 \beta_L}{\gamma^2} \right),$$

which implies that

$$\left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x})^\top - \boldsymbol{W}_L \boldsymbol{D}_{L-1} \boldsymbol{W}_{L-1} \cdots \boldsymbol{D}_1 \boldsymbol{W}_1 \right\|_2 \leq \mathcal{O}\left( \sqrt{\frac{L^3 s \log m + \omega^2 L^3 m}{d}} \sqrt{\frac{dm}{2}} \beta_1 \beta_L \gamma^{L-2} \right),$$

holds with probability at least $1 - \exp\Big( - \Omega(s \log m)\Big)$.

If we choose $s := \lfloor \frac{m}{L^3 \log m} + \frac{\omega^2}{\log m} \rfloor$, then, we have:

$$\left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x})^\top - \boldsymbol{W}_L \boldsymbol{D}_{L-1} \boldsymbol{W}_{L-1} \cdots \boldsymbol{D}_1 \boldsymbol{W}_1 \right\|_2 \leq \mathcal{O}\left( \sqrt{\frac{L^3 \omega^2 + m + \omega^2 L^3 m}{d}} \sqrt{\frac{dm}{2}} \beta_1 \beta_L \gamma^{L-2} \right),$$

with probability at least $1 - \exp\Big( - \Omega(\frac{m}{L^3} + \omega^2)\Big)$.

Let $\delta := \sqrt{\frac{L^3 \omega^2 + m + \omega^2 L^3 m}{d}} \sqrt{\frac{dm}{2}} \beta_1 \beta_L \gamma^{L-2}$, we have $\omega^2 = \frac{u\delta^2 - m}{L^3(m+1)}$, $u = \frac{2}{m \beta_1^2 \beta_L^2 \gamma^{2(L-2)}}$. We have the following probability inequality:

$$\mathbb{P}\Big( \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x})^\top - \boldsymbol{W}_L \boldsymbol{D}_{L-1} \boldsymbol{W}_{L-1} \cdots \boldsymbol{D}_1 \boldsymbol{W}_1 \right\|_2 > \delta \Big) \leq \exp\Big( - \frac{u\delta^2 - m}{L^3(m+1)} - \frac{m}{L^3} \Big) = \exp\Big( - \frac{\delta^2 u + m^2}{L^3(m+1)} \Big).$$

Then by the expectation integral equality [Vershynin, 2018, Lemma 1.2.1], the expectation is:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{W}}\left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})^{\top}-\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1\right\|_2 &= \int_0^{+\infty}\mathbb{P}\Big(\left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})^{\top}-\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1\right\|_2>\delta\Big)\mathrm{d}\delta\\ &\le \int_0^{+\infty}\exp\Big(-\frac{\delta^2 u+m^2}{L^3(m+1)}\Big)\mathrm{d}\delta\\ &= \sqrt{\frac{\pi L^3(m+1)}{4u}}\exp\Big(-\frac{m^2}{(m+1)L^3}\Big)\\ &= \Theta\Big(\gamma^{L-2}\sqrt{\frac{\pi L^3 m^2\beta_1^2\beta_L^2}{8}}e^{-m/L^3}\Big)\,.\end{aligned} \tag{10}$$

Finally, by the definition of $\hat{\boldsymbol{x}}$, we have:

$$\left\|\nabla_{\boldsymbol{x}}\boldsymbol{f}^{\top}(\boldsymbol{x})(\boldsymbol{x}-\hat{\boldsymbol{x}})-\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2\le\epsilon\left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})^{\top}-\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1\right\|_2\,. \tag{11}$$

By Eq. (10) and Eq. (11) and consider expectations for $\boldsymbol{x}$ and $\boldsymbol{x}'$, we finish the proof.

□

**Lemma 5.** *Given an $L$-layer neural network $\boldsymbol{f}$ defined by Eq.* (2) *trained by $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n$, under a small $\epsilon$, expectation over $\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}$, we have:*

$$\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}}\left\|\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2^2\le\frac{mo\beta_1^2\beta_L^2\gamma^{2(L-2)}}{2}\epsilon^2\,, \tag{12}$$

*where $[\boldsymbol{W}_l]_{i,j}$ satisfy the initialization in Section 3.1 and $\boldsymbol{x}\sim\mathcal{D}_X$, $\hat{\boldsymbol{x}}\sim$ Unif$(\mathbb{B}(\epsilon,\boldsymbol{x}))$.*

*Proof.* Define $\boldsymbol{t}_l=\boldsymbol{D}_l\boldsymbol{W}_l\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})$, then:

$$\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}}\left\|\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2^2=\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}}\left\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\right\|_2^2\,.$$

By Lemma 2, we have $\frac{\|\boldsymbol{t}_l\|_2^2}{\beta^2\|\boldsymbol{t}_{l-1}\|_2^2}\sim\chi^2(\varrho)$, where $\varrho\sim\mathrm{Ber}(m,1/2),\forall l=2,\cdots,L-1$. By the law of total expectation $\mathbb{E}[\mathbb{E}[X|Y]]=\mathbb{E}[X]$, we have

$$\mathbb{E}_{\boldsymbol{W}}\frac{\|\boldsymbol{t}_l\|_2^2}{\|\boldsymbol{t}_{l-1}\|_2^2}=\beta^2\mathbb{E}_{\varrho}\chi^2(\varrho)=\beta^2\mathbb{E}\varrho=\frac{m\beta^2}{2}=\gamma^2\,,\ \forall l=2,\cdots,L-1\,.$$

Similarly, we have:

$$\mathbb{E}_{\boldsymbol{W}}\frac{\|\boldsymbol{t}_1\|_2^2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2}=\frac{m\beta_1^2}{2}\,.$$

By the definition of chi-square distribution, we have $\frac{\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\|_2^2}{\beta_L^2\|\boldsymbol{t}_{L-1}\|_2^2}\sim\chi^2(o)$, which means $\mathbb{E}_{\boldsymbol{W}}\left\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\right\|_2^2/\left\|\boldsymbol{t}_{L-1}\right\|_2^2=o\beta_L^2$.

Then, according to the independence among $\frac{\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\|_2^2}{\|\boldsymbol{t}_{L-1}\|_2^2}$, $\frac{\|\boldsymbol{t}_1\|_2^2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2}$, $\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2$ and $\frac{\|\boldsymbol{t}_{l+1}\|_2^2}{\|\boldsymbol{t}_l\|_2^2}$ $\forall l\in[L-2]$, we have:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}}\left\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\right\|_2^2 &= \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}}\frac{\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\|_2^2}{\|\boldsymbol{t}_{L-1}\|_2^2}\frac{\|\boldsymbol{t}_{L-1}\|_2^2}{\|\boldsymbol{t}_{L-2}\|_2^2}\cdots\frac{\|\boldsymbol{t}_1\|_2^2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2}\left\|\hat{\boldsymbol{x}}-\boldsymbol{x}\right\|_2^2\\ &= \mathbb{E}_{\boldsymbol{W}}\frac{\|\boldsymbol{W}_L\boldsymbol{t}_{L-1}\|_2^2}{\|\boldsymbol{t}_{L-1}\|_2^2}\mathbb{E}_{\boldsymbol{W}}\frac{\|\boldsymbol{t}_{L-1}\|_2^2}{\|\boldsymbol{t}_{L-2}\|_2^2}\cdots\mathbb{E}_{\boldsymbol{W}}\frac{\|\boldsymbol{t}_1\|_2^2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2}\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}\left\|\hat{\boldsymbol{x}}-\boldsymbol{x}\right\|_2^2\\ &= \frac{mo\beta_1^2\beta_L^2\gamma^{2(L-2)}}{2}\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}\left\|\hat{\boldsymbol{x}}-\boldsymbol{x}\right\|_2^2\,,\end{aligned}$$

using the definition of $\hat{\boldsymbol{x}}$ which conclude the proof. □

### B.2 Proof of Theorem 1

*Proof.* According to the triangle inequality and the Jensen's inequality, we have:

$$\begin{aligned}
\mathscr{P}(\boldsymbol{f},\epsilon) &= \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\|\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2 \\
&\leq \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\|\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{x}-\hat{\boldsymbol{x}}) - \boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2 \\
&+ \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\|\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2 \\
&\leq \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\|\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{x}-\hat{\boldsymbol{x}}) - \boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2 \\
&+ \sqrt{\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{W}} \left\|\boldsymbol{W}_L\boldsymbol{D}_{L-1}\boldsymbol{W}_{L-1}\cdots\boldsymbol{D}_1\boldsymbol{W}_1(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\|_2^2} \\
&\lesssim \epsilon\left(\sqrt{L^3m^2\beta_1^2\beta_L^2}e^{-m/L^3} + \sqrt{mo\beta_1^2\beta_L^2}\right)\gamma^{L-2}\,,
\end{aligned}$$

where the last inequality using the results of Lemma 4 and Lemma 5.

□

## C Proof of sufficient condition for DNNs under the non-lazy training regime

In this section, we provide the proof of Theorem 2.

*Proof.* By Assumption 2 and by following the setting of Luo et al. [2021], without loss of generality, we have that there exits a $T^\star > 0$ such that $L(\boldsymbol{W}(T^\star)) \leq \frac{1}{32n}$ and $y_1 \geq \frac{1}{2}$. Therefore, we have:

$$\frac{1}{2n}(f_1(T^\star)-y_1)^2 \leq \frac{1}{2n}\sum_{i=1}^{n}(f_i(T^\star)-y_i)^2 \leq L(\boldsymbol{W}(T^\star)) \leq \frac{1}{32n}\,,$$

which means $|f_1(T^\star)-y_1| \leq \frac{1}{4}$. Accordingly, we conclude:

$$\begin{aligned}
\frac{1}{4} \leq y_1 - \frac{1}{4} &\leq f_1(T^\star) \\
&= \frac{1}{\alpha}\boldsymbol{W}_L(T^\star)\sigma(\boldsymbol{W}_{L-1}(T^\star)\cdots\sigma(\boldsymbol{W}_1(T^\star)\boldsymbol{x}_1)) \\
&= \frac{1}{\alpha}\boldsymbol{W}_L(T^\star)\boldsymbol{D}_{1,L-1}(T^\star)\boldsymbol{W}_{L-1}(T^\star)\cdots\boldsymbol{D}_{1,1}(T^\star)\boldsymbol{W}_1(T^\star)\boldsymbol{x}_1 \\
&\leq \frac{1}{\alpha}\left\|\boldsymbol{W}_L(T^\star)\right\|_2\left\|\boldsymbol{D}_{1,L-1}(T^\star)\right\|_2\left\|\boldsymbol{W}_{L-1}(T^\star)\right\|_2\cdots\left\|\boldsymbol{D}_{1,1}(T^\star)\right\|_2\left\|\boldsymbol{W}_1(T^\star)\right\|_2\left\|\boldsymbol{x}_1\right\|_2 \\
&\leq \frac{1}{\alpha}\left\|\boldsymbol{W}_L(T^\star)\right\|_2\left\|\boldsymbol{W}_{L-1}(T^\star)\right\|_2\cdots\left\|\boldsymbol{W}_1(T^\star)\right\|_2\,,
\end{aligned} \tag{13}$$

where the last inequality uses Assumption 1 and 1-Lipschitz of ReLU.

According to Du et al. [2018, Corollary 2.1] , we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\|\boldsymbol{W}_1\|_{\mathrm{F}}^2) = \frac{\mathrm{d}}{\mathrm{d}t}(\|\boldsymbol{W}_2\|_{\mathrm{F}}^2) = \cdots = \frac{\mathrm{d}}{\mathrm{d}t}(\|\boldsymbol{W}_L\|_{\mathrm{F}}^2)\,.$$

Then for any $l_1, l_2 \in [L]$, we have:

$$\|\boldsymbol{W}_{l_1}(T^\star)\|_{\mathrm{F}}^2 - \|\boldsymbol{W}_{l_1}(0)\|_{\mathrm{F}}^2 = \|\boldsymbol{W}_{l_2}(T^\star)\|_{\mathrm{F}}^2 - \|\boldsymbol{W}_{l_2}(0)\|_{\mathrm{F}}^2\,,$$

which implies:

$$\begin{aligned}
\|\boldsymbol{W}_{l_1}(T^\star)\|_2 &\leq \|\boldsymbol{W}_{l_1}(T^\star)\|_{\mathrm{F}} \\
&= \sqrt{\|\boldsymbol{W}_{l_1}(T^\star)\|_{\mathrm{F}}^2} \\
&= \sqrt{\|\boldsymbol{W}_{l_2}(T^\star)\|_{\mathrm{F}}^2 - \|\boldsymbol{W}_{l_2}(0)\|_{\mathrm{F}}^2 + \|\boldsymbol{W}_{l_1}(0)\|_{\mathrm{F}}^2} \\
&\leq \sqrt{\|\boldsymbol{W}_{l_2}(T^\star)\|_{\mathrm{F}}^2 + \|\boldsymbol{W}_{l_1}(0)\|_{\mathrm{F}}^2} \\
&\leq \|\boldsymbol{W}_{l_2}(T^\star)\|_{\mathrm{F}} + \|\boldsymbol{W}_{l_1}(0)\|_{\mathrm{F}}\,.
\end{aligned} \tag{14}$$

According to Luo et al. [2021, Proposition 16] and the relationship between $\ell_2$ norm and Frobenius norm, i.e. $\|\cdot\|_{\mathrm{F}} \leq \sqrt{r}\,\|\cdot\|_2$, where the $r$ is the rank of matrix, with probability at least $1-(L-2)\exp(-\Theta(m^2))-\exp(-\Theta(md))-\exp(-\Theta(m))$ over the initialization, we have $\|\boldsymbol{W}_1(0)\|_{\mathrm{F}} \leq \sqrt{d}\,\|\boldsymbol{W}_1(0)\|_2 \leq \sqrt{\frac{3md^2}{2}}\beta_1$, $\|\boldsymbol{W}_l(0)\|_{\mathrm{F}} \leq \sqrt{m}\,\|\boldsymbol{W}_l(0)\|_2 \leq \sqrt{\frac{3m^3}{2}}\beta_l$, $\forall l \in [L-1]$ and $\|\boldsymbol{W}_L(0)\|_{\mathrm{F}} = \|\boldsymbol{W}_L(0)\|_2 \leq \sqrt{\frac{3m}{2}}\beta_L$.

If we combine Eqs. (13) and (14), for any $l^\star \in [L]$, with probability at least $1-(L-2)\exp(-\Theta(m^2))-\exp(-\Theta(md))-\exp(-\Theta(m))$ over the initialization, we have:

$$
\begin{aligned}
\frac{1}{4} &\leq \frac{1}{\alpha}\,\|\boldsymbol{W}_L(T^\star)\|_2\,\|\boldsymbol{W}_{L-1}(T^\star)\|_2 \cdots \|\boldsymbol{W}_1(T^\star)\|_2 \\
&= \frac{1}{\alpha}\prod_{l=1}^{L}\Big(\,\|\boldsymbol{W}_{l^\star}(T^\star)\|_{\mathrm{F}} + \|\boldsymbol{W}_l(0)\|_{\mathrm{F}}\Big) \\
&\leq \frac{1}{\alpha}\Big(\,\|\boldsymbol{W}_{l^\star}(T^\star)\|_{\mathrm{F}} + \frac{1}{L}\sum_{l=1}^{L}\Big(\,\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}\Big)\Big)^L \\
&\leq \frac{1}{\alpha}\Big(\,\|\boldsymbol{W}_{l^\star}(T^\star)\|_{\mathrm{F}} + \sqrt{\frac{3m^3}{2L^2}}\sum_{l=1}^{L}\beta_l\Big)^L .
\end{aligned}
$$

Then with probability at least $1-(L-2)\exp(-\Theta(m^2))-\exp(-\Theta(md))-\exp(-\Theta(m))$ over the initialization, we have:

$$
\|\boldsymbol{W}_{l^\star}(T^\star)\|_{\mathrm{F}} \geq \left(\frac{\alpha}{4}\right)^{1/L} - \sqrt{\frac{3m^3}{2L^2}}\sum_{l=1}^{L}\beta_l\,. \tag{15}
$$

Therefore, with probability at least $1-(L-2)\exp(-\Theta(m^2))-\exp(-\Theta(md))-\exp(-\Theta(m))$ over the initialization, we have:

$$
\begin{aligned}
\sup_{t\in[0,+\infty)} \frac{\|\boldsymbol{W}_l(t)-\boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}} &\geq \frac{\|\boldsymbol{W}_l(T^\star)-\boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}} \\
&\geq \frac{\|\boldsymbol{W}_l(T^\star)\|_{\mathrm{F}}}{\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}} - 1 \\
&\geq \frac{(\frac{\alpha}{4})^{1/L} - \sqrt{\frac{3m^3}{2L^2}}\sum_{i=1}^{L}\beta_i}{\sqrt{\frac{3m^3}{2}}\beta_l} - 1 \\
&\geq \frac{(\frac{\alpha}{4})^{1/L} - \sqrt{\frac{3m^3}{2L^2}}\sum_{i=1}^{L}\beta_i}{\sqrt{\frac{3m^3}{2}}\sum_{i=1}^{L}\beta_i} - 1 \\
&= \frac{(\frac{\alpha}{4})^{1/L}}{\sqrt{\frac{3m^3}{2}}\sum_{i=1}^{L}\beta_i} - \frac{1}{L} - 1\,,
\end{aligned}
$$

where the second inequality uses triangle inequality and third inequality uses Eq. (15).

If $\alpha \gg (m^{3/2}\sum_{i=1}^{L}\beta_i)^L$, then with probability at least $1-(L-2)\exp(-\Theta(m^2))-\exp(-\Theta(md))-\exp(-\Theta(m))$ over the initialization, we have:

$$
\sup_{t\in[0,+\infty)} \frac{\|\boldsymbol{W}_l(t)-\boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\|\boldsymbol{W}_l(0)\|_{\mathrm{F}}} \gg 1\,.
$$

□

# D Proof of the perturbation stability in non-lazy training regime for two-layer networks

Without loss of generality, we consider two-layer neural networks with a scalar output without bias.

$$f(\boldsymbol{x}) = \frac{1}{\alpha}\sum_{r=1}^{m} a_r \sigma(\boldsymbol{w}_r^\top \boldsymbol{x})\,, \tag{16}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $f(\boldsymbol{x}) \in \mathbb{R}$, $\alpha$ is the scaling factor. The parameters are initialized by $a_r(0) \sim \mathcal{N}(0, \beta_2^2)$, $\boldsymbol{w}_r(0) \sim \mathcal{N}(0, \beta_1^2 \boldsymbol{I}_d)$. Our result can be extended with slight modification to the multiple-output case with bias setting.

Our proof requires some additional notation, which we establish below:

$$\boldsymbol{H}_{ij}^{\infty} = \frac{m}{\alpha^2}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(0,\beta_1^2\boldsymbol{I}_d),\boldsymbol{a}\sim\mathcal{N}(0,\beta_2^2)} a_r^2 \boldsymbol{x}_i^\top \boldsymbol{x}_j 1\left\{\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}_r^\top \boldsymbol{x}_j \geq 0\right\}\,,$$

$$\widetilde{\boldsymbol{H}}_{i,j}(t) = \frac{1}{\alpha^2}\sum_{r=1}^{m} a_r^2(t)\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(0,\beta_1^2\boldsymbol{I}_d)} \boldsymbol{x}_i^\top \boldsymbol{x}_j 1\left\{\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}_r^\top \boldsymbol{x}_j \geq 0\right\}\,,$$

$$\boldsymbol{H}_{i,j}(t) = \frac{1}{\alpha^2}\sum_{r=1}^{m} a_r(t)^2 \boldsymbol{x}_i^\top \boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(t)^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}_r(t)^\top \boldsymbol{x}_j \geq 0\right\}\,,$$

$$\widehat{\boldsymbol{H}}_{i,j} = \frac{1}{\alpha^2}\sum_{r=1}^{m} a_r(t)^2 \boldsymbol{x}_i^\top \boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(0)^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}_r(0)^\top \boldsymbol{x}_j \geq 0\right\}\,,$$

$$\boldsymbol{G}_{i,j}(t) = \frac{1}{\alpha^2}\sigma(\boldsymbol{w}_r(t)^\top \boldsymbol{x}_i)\sigma(\boldsymbol{w}_r(t)^\top \boldsymbol{x}_j)\,.$$

The minimum eigenvalue of $\boldsymbol{H}_{ij}^{\infty}$ is denoted as $\lambda_0$ and is assumed to be strictly greater than $0$, i.e.

$$\lambda_0 = \lambda_{\min}(\boldsymbol{H}^{\infty}) > 0\,.$$

**Remark:** This assumption follows Du et al. [2019b] but can be proved by Nguyen et al. [2021] under the NTK initialization. Moreover, Chen and Xu [2021], Geifman et al. [2020], Bietti and Mairal [2019] discuss this assumption in different settings.

The following two symbols are used to measure the weight changes during training:

$$R_a := \frac{\alpha}{n}\sqrt{\frac{\lambda_0}{8nm}} - \sqrt{\frac{2}{\pi}}\beta_2, \quad \text{and} \quad R_w := \frac{\alpha^2\lambda_0\sqrt{2\pi}\beta_1}{32n^3m(R_a(R_a+\sqrt{8/\pi}\beta_2)+\beta_2^2)}\,. \tag{17}$$

The last two symbols are used to characterize the early stages of neural network training:

$$t_1^\star = -\frac{2}{\lambda_0}\log\left(1 - \frac{R_w\lambda_0\alpha}{2\sqrt{n}(\sqrt{n}\beta_2 + R_a)\left\|\boldsymbol{y}-\boldsymbol{f}(0)\right\|_2}\right),$$

$$t_2^\star = -\frac{2}{\lambda_0}\log\left(1 - \frac{R_a\lambda_0\alpha}{2\sqrt{n}(3\beta_1\sqrt{\log(mn^2)} + R_w)\left\|\boldsymbol{y}-\boldsymbol{f}(0)\right\|_2}\right).$$

Then we present the details of our results on Section 4.3 in this section. Firstly, we introduce some lemmas in Appendix D.1 to facilitate the proof of theorems. Then in Appendix D.2 we provide the proof of Theorem 3.

### D.1 Relevant Lemmas

**Lemma 6.** *[Du et al., 2019b, Appendix A.1] Given a two-layer neural network $f$ defined by Eq.* (16) *and trained by $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ using gradient descent with the quadratic loss, let $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n$ be the label vector and $\boldsymbol{f}(t) = (f_1(t), \ldots, f_n(t)) \in \mathbb{R}^n$ be the output vector at time t, then, we have:*

$$\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = (\boldsymbol{H}(t) + \boldsymbol{G}(t))(\boldsymbol{y} - \boldsymbol{f}(t)) \,. \tag{18}$$

*Proof.* Our proof here just re-organizes Du et al. [2019b, Appendix A.1]. For making our manuscript self-contained, we provide a formal proof here.

We want to minimize the quadratic loss:

$$L(\boldsymbol{W}, \boldsymbol{a}) = \sum_{i=1}^n \frac{1}{2}[f(\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{x}_i) - y_i]^2 \,.$$

Using the gradient descent algorithm, the formula for update the weights is:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) - \eta \frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a}(t))}{\partial \boldsymbol{W}(t)} \,,$$

$$\boldsymbol{a}(t+1) = \boldsymbol{a}(t) - \eta \frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a}(t))}{\partial \boldsymbol{a}(t)} \,.$$

According to the standard chain rule, we have:

$$\frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a}(t))}{\partial \boldsymbol{W}(t)} = \frac{1}{\alpha} \sum_{i=1}^n [f(\boldsymbol{W}(t), \boldsymbol{a}(t), \boldsymbol{x}_i) - y_i] a_r(t) 1\left\{\boldsymbol{w}_r^\top(t)\boldsymbol{x}_i \geq 0\right\} \boldsymbol{x}_i \,,$$

$$\frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a}(t))}{\partial \boldsymbol{a}(t)} = \frac{1}{\alpha} \sum_{i=1}^n [f(\boldsymbol{W}(t), \boldsymbol{a}(t), \boldsymbol{x}_i) - y_i] \sigma(\boldsymbol{w}_r^\top(t)\boldsymbol{x}_i) \,.$$

Then, we have:

$$\begin{aligned}
\frac{\mathrm{d}f_i(t)}{\mathrm{d}t} &= \sum_{r=1}^m \left\langle \frac{\partial f_i(t)}{\partial \boldsymbol{w}_r(t)}, \frac{\partial \boldsymbol{w}_r(t)}{\partial t} \right\rangle + \sum_{r=1}^m \frac{\mathrm{d}f_i(t)}{\mathrm{d}a_r(t)} \frac{\mathrm{d}a_r(t)}{\mathrm{d}t} \\
&= \sum_{i=1}^n [y_i - f_i(t)][\boldsymbol{H}_{ij}(t) + \boldsymbol{G}_{ij}(t)] \,.
\end{aligned}$$

Written in vector form, we have:

$$\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = (\boldsymbol{H}(t) + \boldsymbol{G}(t))(\boldsymbol{y} - \boldsymbol{f}(t)) \,.$$

□

**Lemma 7.** *If $\alpha \geq \frac{16n\beta_2\sqrt{\log(2n^3)}}{\lambda_0}$, with probability at least $1 - \frac{1}{n}$, we have:*

$$\|\boldsymbol{H}(0) - \boldsymbol{H}^\infty\|_2 \leq \frac{\lambda_0}{4}, \quad \textit{and} \quad \lambda_{\min}(\boldsymbol{H}(0)) \geq \frac{3}{4}\lambda_0 \,,$$

**Remark:** This lemma is a modified version of Du et al. [2019b, Lemma 3.1], which differs in the initialization of $\boldsymbol{a}$ from Unif($\{-1, +1\}$) to Gaussian initialization. This makes our analysis relatively intractable due to their analysis based on $a_i^2 = 1,\ \forall i \in [m]$.

*Proof.* Firstly, for a fixed pair $(i,j)$, $\boldsymbol{H}_{ij}^{\infty}$ is an average of $\widetilde{\boldsymbol{H}}_{i,j}$ with respect to $a_r$ . By Bernstein's inequality [Vershynin, 2018, Chapter 2], with probability at least $1-\delta$, we have:

$$\left|\boldsymbol{H}_{ij}^{\infty}-\widetilde{\boldsymbol{H}}_{i,j}\right| \leq \frac{2\beta_2\sqrt{\log(\frac{1}{\delta})}}{\alpha}\,.$$

Then, for fixed pair $(i,j)$, $\widetilde{\boldsymbol{H}}_{i,j}$ is an average of $\boldsymbol{H}_{ij}(0)$ with respect to $\boldsymbol{w}_r$. By Hoeffding's inequality [Vershynin, 2018, Chapter 2], with probability at least $1-\delta'$, we have:

$$\left|\boldsymbol{H}_{ij}(0)-\widetilde{\boldsymbol{H}}_{i,j}\right| \leq \frac{2\beta_2\sqrt{\log(\frac{1}{\delta'})}}{\alpha}\,.$$

Choose $\delta := \delta' := \frac{1}{2n^3}$, we have with probability at least $1-\frac{1}{n^3}$, for fixed pair $(i,j)$:

$$\left|\boldsymbol{H}_{ij}(0)-\boldsymbol{H}_{ij}^{\infty}\right| \leq \frac{4\beta_2\sqrt{\log(2n^3)}}{\alpha}\,.$$

Consider the union bound over $(i,j)$ pairs, with probability at least $1-\frac{1}{n}$, we have:

$$\left|\boldsymbol{H}_{ij}(0)-\boldsymbol{H}_{ij}^{\infty}\right| \leq \frac{4\beta_2\sqrt{\log(2n^3)}}{\alpha}\,.$$

Thus, we have:

$$\|\boldsymbol{H}(0)-\boldsymbol{H}^{\infty}\|_2^2 \leq \|\boldsymbol{H}(0)-\boldsymbol{H}^{\infty}\|_{\mathrm{F}}^2 \leq \sum_{i,j}\left|\boldsymbol{H}_{ij}(0)-\boldsymbol{H}_{ij}^{\infty}\right|^2 \leq \frac{16n^2\beta_2^2\log(2n^3)}{\alpha^2}\,.$$

when $\alpha \geq \frac{16n\beta_2\sqrt{\log(2n^3)}}{\lambda_0}$, we have the desired result. □

**Lemma 8.** *With probability at least $1-\frac{2}{n}$ over initialization , if a set of weight vectors $\{\boldsymbol{w}_r\}_{r=1}^m$ and the output weight $\{a_r\}_{r=1}^m$ satisfy for all $r\in[m]$, $\|\boldsymbol{w}_r(t)-\boldsymbol{w}_r(0)\|_2 \leq R_w$ and $|a_r(t)-a_r(0)| \leq R_a$, then, we have:*

$$\|\boldsymbol{H}(t)-\boldsymbol{H}(0)\|_2 \leq \frac{\lambda_0}{4}, \quad \textit{and} \quad \lambda_{\min}(\boldsymbol{H}(t)) \geq \frac{\lambda_0}{2}\,.$$

*Proof.* Firstly, we can derive that:

$$\widehat{\boldsymbol{H}}_{i,j}(t)-\boldsymbol{H}_{i,j}(0) = \frac{1}{\alpha^2}\sum_{r=1}^{m}(a_r(t)^2-a_r(0)^2)\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i \geq 0, \boldsymbol{w}_r(0)^\top\boldsymbol{x}_j \geq 0\right\}\,,$$

$$\begin{aligned}\boldsymbol{H}_{i,j}(t)-\widehat{\boldsymbol{H}}_{i,j}(t) = &\frac{1}{\alpha^2}\sum_{r=1}^{m}a_r(t)^2\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(t)^\top\boldsymbol{x}_i \geq 0, \boldsymbol{w}_r(t)^\top\boldsymbol{x}_j \geq 0\right\}\\ &-\frac{1}{\alpha^2}\sum_{r=1}^{m}a_r(t)^2\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i \geq 0, \boldsymbol{w}_r(0)^\top\boldsymbol{x}_j \geq 0\right\}\,.\end{aligned}$$

Then we can compute the expectation of $\left|\widehat{\boldsymbol{H}}_{i,j}(t)-\boldsymbol{H}_{i,j}(0)\right|$:

$$
\begin{aligned}
\mathbb{E}\left|\widehat{\boldsymbol{H}}_{i,j}(t)-\boldsymbol{H}_{i,j}(0)\right| &= \mathbb{E}\left|\frac{1}{\alpha^2}\sum_{r=1}^{m}(a_r(t)^2-a_r(0)^2)\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i\geq 0,\boldsymbol{w}_r(0)^\top\boldsymbol{x}_j\geq 0\right\}\right| \\
&\leq \frac{m}{\alpha^2}\mathbb{E}\left|a_r(t)^2-a_r(0)^2\right| \\
&= \frac{m}{\alpha^2}\mathbb{E}\left|(a_r(t)-a_r(0))(a_r(t)+a_r(0))\right| \\
&\leq \frac{mR_a}{\alpha^2}\mathbb{E}\left|a_r(t)+a_r(0)\right| \\
&\leq \frac{mR_a}{\alpha^2}(R_a+2\mathbb{E}\left|a_r(0)\right|) \\
&\leq \frac{m(R_a+\mathbb{E}\left|a_r(0)\right|)^2}{\alpha^2} \\
&= \frac{m(R_a+\sqrt{\frac{2}{\pi}}\beta_2)^2}{\alpha^2}\,.
\end{aligned}
\tag{19}
$$

Then we define the event:

$$
A_{i,r}=\left\{\exists:\|\boldsymbol{w}_r(t)-\boldsymbol{w}_r(0)\|\leq R_w, 1\left\{\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i\geq 0\right\}\neq 1\left\{\boldsymbol{w}_r(t)^\top\boldsymbol{x}_i\geq 0\right\}\right\}\,.
$$

This event happens if and only if $\left|\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i\right|<R_t$. According to this, we can get $\mathbb{P}(A_{i,r})=\mathbb{P}_{z\sim\mathcal{N}(0,\beta_1^2)}(|z|\leq R_w)\leq\frac{2R_w}{\sqrt{2\pi}\beta_1}$, further:

$$
\begin{aligned}
\mathbb{E}\left|\boldsymbol{H}_{i,j}(t)-\widehat{\boldsymbol{H}}_{i,j}(t)\right| &= \frac{1}{\alpha^2}\mathbb{E}\Bigg|\sum_{r=1}^{m}a_r(t)^2\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(t)^\top\boldsymbol{x}_i\geq 0,\boldsymbol{w}_r(t)^\top\boldsymbol{x}_j\geq 0\right\} \\
&\quad -\sum_{r=1}^{m}a_r(t)^2\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{\boldsymbol{w}_r(0)^\top\boldsymbol{x}_i\geq 0,\boldsymbol{w}_r(0)^\top\boldsymbol{x}_j\geq 0\right\}\Bigg| \\
&\leq \frac{1}{\alpha^2}\sum_{r=1}^{m}\mathbb{E}\left(a_r(t)^2\boldsymbol{x}_i^\top\boldsymbol{x}_j 1\left\{A_{i,r}\cup A_{j,r}\right\}\right) \\
&\leq \frac{1}{\alpha^2}\sum_{r=1}^{m}\mathbb{E}\left(a_r(t)^2\frac{4R_w}{\sqrt{2\pi}\beta_1}\right) \\
&= \frac{4R_w}{\alpha^2\sqrt{2\pi}\beta_1}\sum_{r=1}^{m}\mathbb{E}(a_r(t)^2-a_r(0)^2+a_r(0)^2) \\
&\leq \frac{4R_w m}{\alpha^2\sqrt{2\pi}\beta_1}\left(R_a(R_a+\sqrt{\frac{8}{\pi}}\beta_2)+\beta_2^2\right),
\end{aligned}
\tag{20}
$$

where the last inequality uses the result of Eq. (19).

From Eqs. (19) and (20), using Markov's inequality. with probability at least $1-\frac{2}{n}$, we have:

$$
\left|\widehat{\boldsymbol{H}}_{i,j}(t)-\boldsymbol{H}_{i,j}(0)\right|\leq\frac{nm(R_a+\sqrt{\frac{2}{\pi}}\beta_2)^2}{\alpha^2}\,,
$$

$$
\left|\boldsymbol{H}_{i,j}(t)-\widehat{\boldsymbol{H}}_{i,j}(t)\right|\leq\frac{4R_w nm}{\alpha^2\sqrt{2\pi}\beta_1}\left(R_a(R_a+\sqrt{\frac{8}{\pi}}\beta_2)+\beta_2^2\right).
$$

Then, we have:

$$\begin{aligned}
\|\boldsymbol{H}(t)-\boldsymbol{H}(0)\|_2 &\le \|\boldsymbol{H}(t)-\boldsymbol{H}(0)\|_{\mathrm{F}} \\
&\le \sum_{(i,j)=(1,1)}^{(n,n)} |\boldsymbol{H}_{i,j}(t)-\boldsymbol{H}_{i,j}(0)| \\
&\le \sum_{(i,j)=(1,1)}^{(n,n)} \left( \left|\widehat{\boldsymbol{H}}_{i,j}(t)-\boldsymbol{H}_{i,j}(0)\right| + \left|\boldsymbol{H}_{i,j}(t)-\widehat{\boldsymbol{H}}_{i,j}(t)\right| \right) \\
&\le \frac{mn^3}{\alpha^2}\left( (R_a+\sqrt{\frac{2}{\pi}}\beta_2)^2 + \frac{4R_w}{\sqrt{2\pi}\beta_1}(R_a(R_a+\sqrt{\frac{8}{\pi}}\beta_2)+\beta_2^2) \right).
\end{aligned}$$

Then, by Eq. (17), we have:

$$\|\boldsymbol{H}(t)-\boldsymbol{H}(0)\|_2 \le \frac{\lambda_0}{4},$$

which implies:

$$\lambda_{\min}(\boldsymbol{H}(t)) \le \lambda_{\min}(\boldsymbol{H}(0)) - \frac{\lambda_0}{4} \le \frac{\lambda_0}{2}.$$

□

**Lemma 9.** *Suppose that for $0 \le s \le t$, $\lambda_{\min}(\boldsymbol{H}(s)) \ge \frac{\lambda_0}{2}$ and $|a_r(s)-a_r(0)| \le R_a$. Then with probability at least $1-n\exp(-n/2)$ over initialization, we have $\|\boldsymbol{w}_r(t)-\boldsymbol{w}_r(0)\|_2 \le R_w$ for all $r \in [m]$ and the $t \le t_1^\star$.*

*Proof.* By Lemma 6, we have $\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = (\boldsymbol{H}(t)+\boldsymbol{G}(t))(\boldsymbol{y}-\boldsymbol{f}(t))$. Then we can calculate the dynamics of risk function:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\|\boldsymbol{y}-\boldsymbol{f}(t)\|_2^2 &= -2(\boldsymbol{y}-\boldsymbol{f}(t))^\top(\boldsymbol{H}(t)+\boldsymbol{G}(t))(\boldsymbol{y}-\boldsymbol{f}(t)) \\
&\le -2(\boldsymbol{y}-\boldsymbol{f}(t))^\top(\boldsymbol{H}(t))(\boldsymbol{y}-\boldsymbol{f}(t)) \\
&\le -\lambda_0\|\boldsymbol{y}-\boldsymbol{f}(t)\|_2^2,
\end{aligned}$$

in the first inequality we use that the $\boldsymbol{G}(t)$ is Gram matrix thus it is positive. Then, we have $\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{\lambda_0 t}\|\boldsymbol{y}-\boldsymbol{f}(t)\|_2^2\right) \le 0$, then $e^{\lambda_0 t}\|\boldsymbol{y}-\boldsymbol{f}(t)\|_2^2$ is a decreasing function with respect to $t$. Thus, we can bound the risk:

$$\|\boldsymbol{y}-\boldsymbol{f}(t)\|_2^2 \le e^{-\lambda_0 t}\|\boldsymbol{y}-\boldsymbol{f}(0)\|_2^2. \tag{21}$$

Then we bound the gradient of $\boldsymbol{w}_r$. For $0 \le s \le t$, With probability at least $1-n\exp(-n/2)$, we have:

$$\begin{aligned}
\left\|\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{w}_r(s)\right\|_2 &= \left\|\frac{1}{\alpha}\sum_{i=1}^n [f(\boldsymbol{W}(s),\boldsymbol{a}(s),\boldsymbol{x}_i)-y_i]a_r(s)1\left\{\boldsymbol{w}_r^\top(s)\boldsymbol{x}_i \ge 0\right\}\boldsymbol{x}_i\right\|_2 \\
&\le \frac{1}{\alpha}\sum_{i=1}^n |f(\boldsymbol{W}(s),\boldsymbol{a}(s),\boldsymbol{x}_i)-y_i|\,|a_r(0)+R_a| \\
&\le \frac{\sqrt{n}}{\alpha}\|\boldsymbol{y}-\boldsymbol{f}(s)\|_2(\sqrt{n}\beta_2+R_a) \\
&\le \frac{\sqrt{n}}{\alpha}(\sqrt{n}\beta_2+R_a)e^{-\lambda_0 s/2}\|\boldsymbol{y}-\boldsymbol{f}(0)\|_2,
\end{aligned}$$

where the second inequality is because of $a_r(0) \sim \mathcal{N}(0, \beta_2^2)$, then with probability at least $1 - \exp(-n/2)$, we have $a_r(0) \leq \sqrt{n}\beta_2$. Then, we have:

$$\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq \int_0^t \left\| \frac{\mathrm{d}}{\mathrm{d}s} \boldsymbol{w}_r(s) \right\|_2 \mathrm{d}s \leq \frac{2\sqrt{n}}{\lambda_0 \alpha} (\sqrt{n}\beta_2 + R_a) \|\boldsymbol{y} - \boldsymbol{f}(0)\|_2 (1 - \exp(-\frac{\lambda_0 t}{2})) . \tag{22}$$

If we account for $t$, then we conclude the proof. □

**Lemma 10.** *Suppose that for* $0 \leq s \leq t$, $\lambda_{\min}(\boldsymbol{H}(s)) \geq \frac{\lambda_0}{2}$ *and* $\|\boldsymbol{w}_r(s) - \boldsymbol{w}_r(0)\|_2 \leq R_w$. *Then with probability at least* $1 - \frac{1}{n}$ *over initialization, we have* $|a_r(t) - a_r(0)| \leq R_a$ *for all* $r \in [m]$ *and the* $t \leq t_2^\star$.

*Proof.* Note for any $i \in [n]$ and $r \in [m]$, $\boldsymbol{w}_r^\top(0)\boldsymbol{x}_i \sim \mathcal{N}(0, \beta_1^2)$. Therefore applying Gaussian tail bound and union bound, we have with probability at least $1 - \frac{1}{n}$, for all $i \in [n]$ and $r \in [m]$, $\left|\boldsymbol{w}_r^\top(0)\boldsymbol{x}_i\right| \leq 3\beta_1 \sqrt{\log(mn^2)}$, That means for $0 \leq s \leq t$, With probability at least $1 - \frac{1}{n}$, we have:

$$\begin{aligned}
\left| \frac{\mathrm{d}}{\mathrm{d}s} a_r(s) \right| &= \left| \frac{1}{\alpha} \sum_{i=1}^n [f(\boldsymbol{W}(t), \boldsymbol{a}(t), \boldsymbol{x}_i) - y_i] \sigma(\boldsymbol{w}_r^\top(t)\boldsymbol{x}_i) \right| \\
&\leq \frac{\sqrt{n}}{\alpha} \|\boldsymbol{y} - \boldsymbol{f}(s)\|_2 \left(\left|\boldsymbol{w}_r^\top(0)\boldsymbol{x}_i\right| + R_w\right) \\
&\leq \frac{\sqrt{n}}{\alpha} e^{-\lambda_0 s/2} \|\boldsymbol{y} - \boldsymbol{f}(0)\|_2 \left(3\beta_1 \sqrt{\log(mn^2)} + R_w\right) .
\end{aligned}$$

Then, we have:

$$|a_r(t) - a_r(0)|_2 \leq \int_0^t \left| \frac{\mathrm{d}}{\mathrm{d}s} a_r(s) \right| \mathrm{d}s \leq \frac{2\sqrt{n}}{\lambda_0 \alpha} \left(3\beta_1 \sqrt{\log(mn^2)} + R_w\right) \|\boldsymbol{y} - \boldsymbol{f}(0)\|_2 (1 - \exp(-\frac{\lambda_0 t}{2})) . \tag{23}$$

Bring in $t$, then finish the proof.

□

**Lemma 11.** *Suppose* $0 \leq t \leq \min(t_1^\star, t_2^\star)$. *Then with probability at least* $1 - n\exp(-n/2) - \frac{3}{n}$ *over initialization, we have:* $\lambda_{\min}(\boldsymbol{H}(t)) \geq \frac{\lambda_0}{2}$,

$$|a_r(t) - a_r(0)| \leq \frac{2\sqrt{n}}{\lambda_0 \alpha} \left(3\beta_1 \sqrt{\log(mn^2)} + R_w\right) \|\boldsymbol{y} - \boldsymbol{f}(0)\|_2 (1 - \exp(-\frac{\lambda_0 t}{2})) := R_a^\star(t) ,$$

$$\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq \frac{2\sqrt{n}}{\lambda_0 \alpha} (\sqrt{n}\beta_2 + R_a) \|\boldsymbol{y} - \boldsymbol{f}(0)\|_2 (1 - \exp(-\frac{\lambda_0 t}{2})) := R_w^\star(t) ,$$

*for all* $r \in [m]$.

*Proof.* When $t = 0$, $\lambda_{\min}(\boldsymbol{H}(s)) \geq \frac{3}{4}\lambda_0$, $|a_r(t) - a_r(0)| = 0 < R_a$ and $\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 = 0 < R_w$. Using induction, combine Lemma 8, Lemma 9 and Lemma 10, we have the result. □

### D.2 Proof of Theorem 3

*Proof.* We can compute the gradient of the network that:

$$\nabla_{\boldsymbol{x}} f_t(\boldsymbol{x}) = \frac{1}{\alpha} \sum_{r=1}^m a_r(t) 1\left\{\boldsymbol{w}_r^\top(t)\boldsymbol{x} \geq 0\right\} \boldsymbol{w}_r^\top(t) .$$

□

Then we can derive that:

$$
\begin{aligned}
\mathscr{P}(f_t, \epsilon) &= \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}} \left| \frac{1}{\alpha} \sum_{r=1}^{m} a_r(t) 1 \left\{ \boldsymbol{w}_r^\top(t) \boldsymbol{x} \geq 0 \right\} \boldsymbol{w}_r^\top(t)(\boldsymbol{x} - \hat{\boldsymbol{x}}) \right| \\
&\leq \frac{1}{\alpha} \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}} \sum_{r=1}^{m} \left| a_r(t) \boldsymbol{w}_r^\top(t)(\boldsymbol{x} - \hat{\boldsymbol{x}}) \right| \\
&\leq \frac{1}{\alpha} \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}} \sum_{r=1}^{m} |a_r(t)| \left\| \boldsymbol{w}_r(t) \right\|_2 \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|_2 \\
&\leq \frac{\epsilon}{\alpha} \sum_{r=1}^{m} |a_r(t)| \left\| \boldsymbol{w}_r(t) \right\|_2 .
\end{aligned}
\tag{24}
$$

Then by Lemma 11, we have:

$$
|a_r(t)| \leq |a_r(t) - a_r(0)| + |a_r(0)| \leq R_a^\star(t) + |a_r(0)| .
$$

$$
\left\| \boldsymbol{w}_r(t) \right\|_2 \leq \left\| \boldsymbol{w}_r(t) - \boldsymbol{w}_r(0) \right\|_2 + \left\| \boldsymbol{w}_r(0) \right\|_2 \leq R_w^\star(t) + \left\| \boldsymbol{w}_r(0) \right\|_2 .
$$

From Eq. (19), we have $\mathbb{E}\left|a_r(0)\right| = \sqrt{\frac{2}{\pi}}\beta_2$. That means with probability at least $1 - \frac{1}{n}$ over initialization, we have $|a_r(0)| \leq \sqrt{\frac{2}{\pi}} n \beta_2$.

By Vershynin [2018, Chapter 3], with probability at least $1-\delta$ over initialization, we have $\left\| \boldsymbol{w}_r(0) \right\|_2 \leq 4\beta_1\sqrt{m} + 2\beta_1\sqrt{\log n}$.

By combining the results above with Eq. (24) and Lemma 11, with probability at least $1 - n\exp(-n/2) - \frac{3}{n}$ over initialization we obtain that:

$$
\begin{aligned}
\mathscr{P}(f_t, \epsilon) &\leq \frac{\epsilon}{\alpha} \sum_{r=1}^{m} |a_r(t)| \left\| \boldsymbol{w}_r(t) \right\|_2 \\
&\leq \frac{\epsilon m}{\alpha} (R_a^\star(t) + \sqrt{\frac{2}{\pi}} n \beta_2)(R_w^\star(t) + 4\beta_1\sqrt{m} + 2\beta_1\sqrt{\log n})
\end{aligned}
\tag{25}
$$

Suppose that $\alpha \sim 1$, $\beta_1 \sim \beta_2 \sim \beta \sim \frac{1}{m^c}$, $c \geq 1.5$, $m \gg n^2$. Then $R_a = \Theta(\frac{1}{\sqrt{n^3 m}})$, $R_w = \Theta(\frac{1}{m^c})$, $R_a^\star(t) = \Theta(\frac{\sqrt{n \log m}}{m^c})$ and $R_w^\star(t) = \Theta(\frac{1}{\sqrt{n^3 m}})$. Bring these results into Eq. (25), with probability at least $1 - n\exp(-\frac{n}{2}) - \frac{3}{n}$ over initialization, we have:

$$
\mathscr{P}(f_t, \epsilon) \leq \Theta\left( \epsilon \frac{\sqrt{n \log m} + n}{m^{c-1}} \left( \frac{1}{\sqrt{n^3 m}} + \frac{1}{m^{c-0.5}} \right) \right) .
$$

# E Additional Experiments

A number of additional experiments are conducted in this section. Unless explicitly mentioned otherwise, the experimental setup remains similar to the one in the main paper. The following experiments are conducted below:

1. In Appendix E.1, we compare the two different training regimes, lazy training and non-lazy training.
2. In Appendix E.2, we conduct some experiments to verify the reasonableness of using the early-stop training in the paper.
3. In Appendix E.3, we extend the experiments in Section 5.3 from fully connected network to Convolutional Neural Network.

Table 4: Comparison results of lazy training regime and non-lazy training regime of ResNet-110 under practical network/datasets.

| Dataset | Lazy training | Non-lazy training |
|---|---|---|
| CIFAR10 | $92.89\%$ | $92.14\%$ |
| CIFAR100 | $71.08\%$ | $70.55\%$ |

4. In Appendix E.4, we extend the experiments to some other initializations under non-lazy training regime.
5. In Appendix E.5, we extend the experiments from He and LeCun initialization to NTK initialization.
6. In Appendix E.6, we extend in Section 5.3 from fully connected network to ResNet.

### E.1 Comparison of Lazy training and Non-lazy training

In this section, we test the performance of lazy training regime and non-lazy training regime on practical task and networks. We choose the ResNet-110 model[3] for these experiments. We adopt a narrow model width for computational efficiency. We utilize the He initialization and the non-lazy training initialization as mentioned in Section 5.1 on two commonly used datasets CIFAR10 and CIFAR100, of which results are provided in Table 4. Notice that the non-lazy training regime achieves a similar performance to lazy training regime. This implies that non-lazy training regime is also needed for studying practical learning tasks.

### E.2 Ablation study for why we use early stopping (training 50 epochs)

In this section, we conduct some experiments to verify the reasonableness of using the early-stop training (50 epochs) method in the paper. The experimental results shown in Fig. 6 indicate that the loss and accuracy of the neural network almost not change from the 50th epoch to the 200th epoch under two different network settings: width = 32, depth = 4 and width = 64, depth = 8. Such results verify the rationality of our choice to run 50 epochs.

### E.3 Extend the experiment in Section 5.3 to CNN

In this section, we extend the experiments in Section 5.3 from fully connected networks to convolutional neural networks in Fig. 7. Compared with the fully connected network, the main difference of the convolutional neural network is that the gap between different depths is much larger than fully connected network, which is more in line with the relationship between robustness and depth under He initialization in Theorem 1.

### E.4 More experiments in non-lazy training regime

In this section, we extend the experiments of Fig. 4(b) to more initializations under non-lazy training regime (the variance of the initial weight are $\frac{1}{m^3}$ and $\frac{1}{m^4}$). Fig. 8 provides the relationship between robustness and width of neural network for these two initializations and shows that the robustness improves with the increase of the width of network which consistent with Theorem 3, but the difference between different initializations is not as large as our theoretical expectation, which may indicate that the bound in Theorem 3 is not tight enough.

### E.5 More experiments in under NTK initialization

In this section, we extend the experiments in Fig. 5 from He and LeCun initialization to NTK initialization in Fig. 9. Our experimental results show that, NTK initialization and He initialization yield similar curves, but differs in the curve of $L = 2$. This may be because the infinite-width NTK is equivalent to the linear model, and the large finite-width network approximates the linear model. This phenomenon can be more easily detected for two-layer neural networks when compared to deeper networks.

[3] `https://github.com/bearpaw/pytorch-classification`

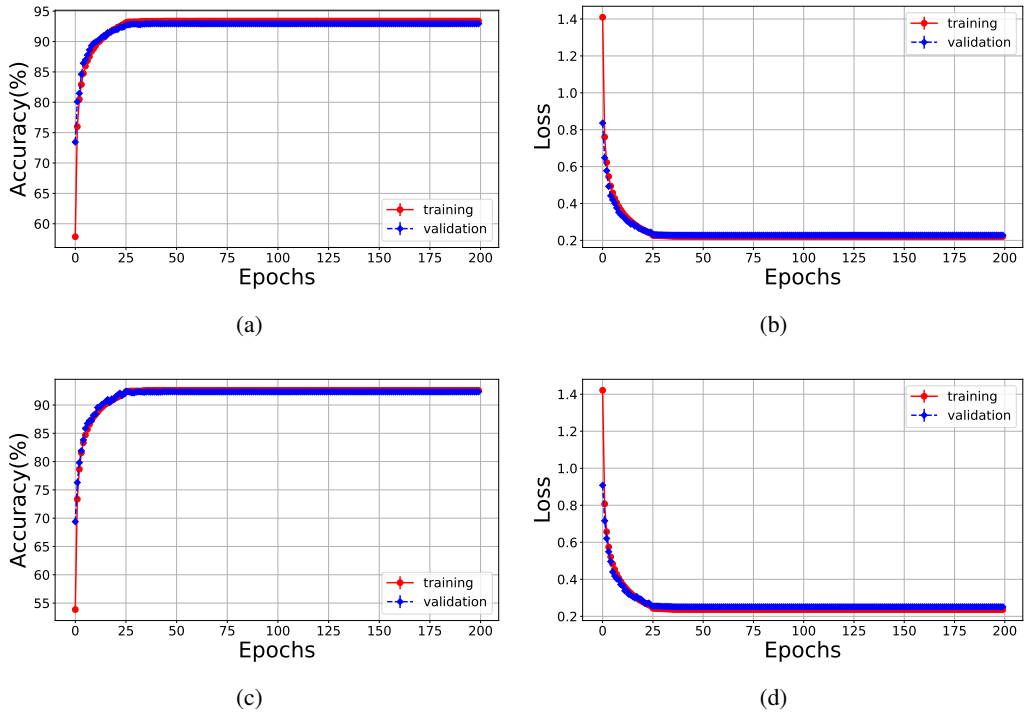



Figure 6: Verify the reasonableness of using the early-stop training, (a) accuracy and (b) loss for network with width=32 and depth=4, (c) accuracy and (d) loss for network with width=64 and depth=8

### E.6 More experiments on ResNet

In this section, we extend the experiments in Section 5.3 from fully connected networks to ResNet in Fig. 10. Compared with the fully connected network, the results of ResNet show similar characteristics to our theory on fully connected networks. Specifically, the perturbation stability increases with depth, and an insignificant phase transition can also be seen for width.

## F Limitation and discussion

The limitation of this work is mainly manifested in that Theorem 3 is built on two-layers neural networks. Extending this results to deep neural networks beyond lazy training regime is non-trivial. Firstly, the dynamics of the deep neural network and the bounds of the gap between the initialization and the expectation of the gram matrix will become more complex. Secondly, due to the coupling relationship between different layers, the critical change radius of the weight in Lemma 8 is also coupled with each other and is difficult to analyze. Then, due to the superposition of the previous two points, the relationship between the weights changing with time in the early stage of training (similar to Lemma 9) and the width and initialization of the neural network will be difficult to distinguish, which leads to the final result being complex, demanding and difficult to obtain a valid conclusion about width and initialization.

## G Societal impact

This is a theoretical work that explores the interplay of the width, the depth and the initialization of neural networks on their average robustness. Our goal is to deepen the understanding of the community in the factors that affect the robustness. We do not focus on obtaining any state-of-the-art results in a particular task, which means there are other works that can be used for forming strong adversarial attacks and can be used with malicious intent.

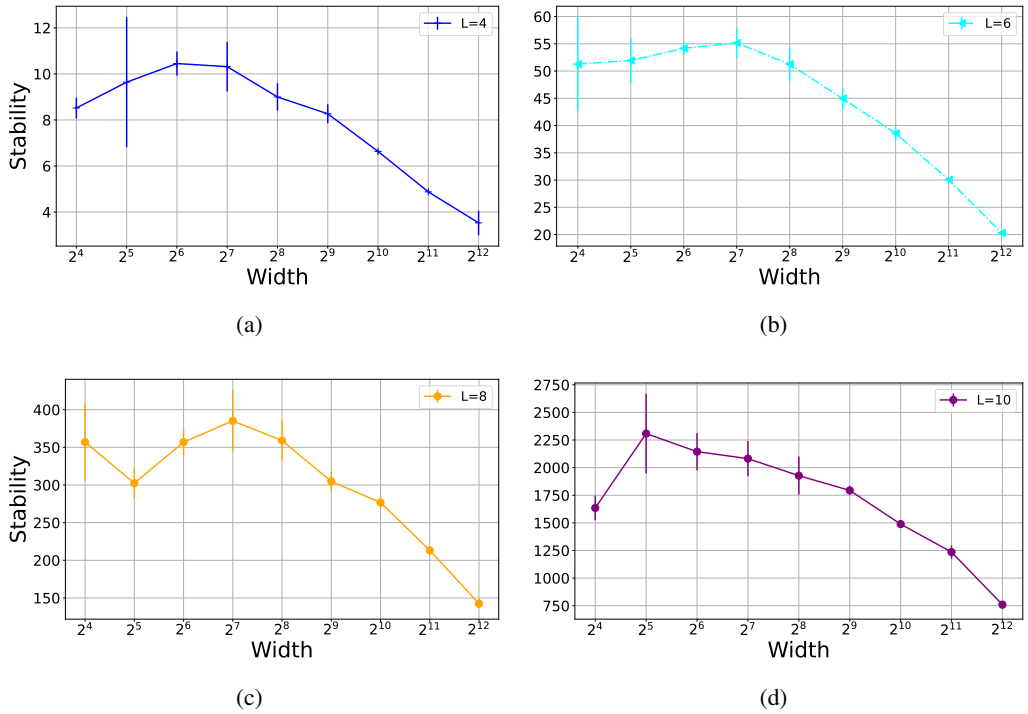



Figure 7: Relationship between the *perturbation stability* and width of CNN under He initialization for different depths of $L = 4, 6, 8$ and $10$.

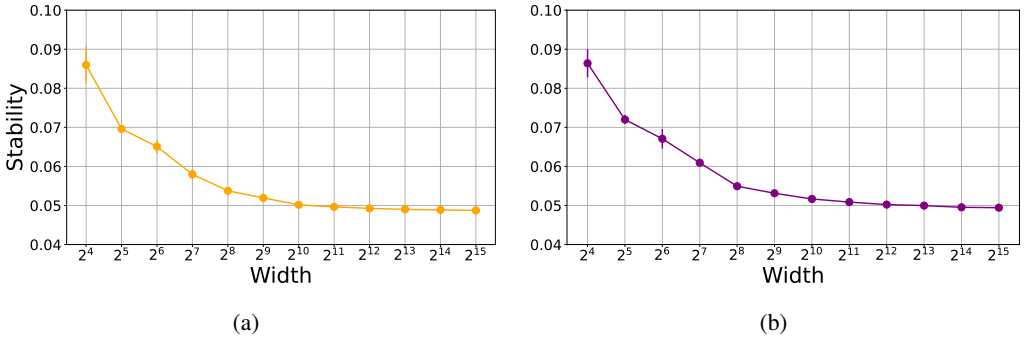



Figure 8: Influence of width of neural network on the perturbation stability under non-lazy training regime. (a) the variance of the initial weight is $\frac{1}{m^3}$. (b) the variance of the initial weight is $\frac{1}{m^4}$.

Despite the theoretical nature of our work, we encourage researchers to further investigate the impact of robustness on the society. We expect robustness to have a key role into a world where neural networks are increasingly deployed into real-world applications.

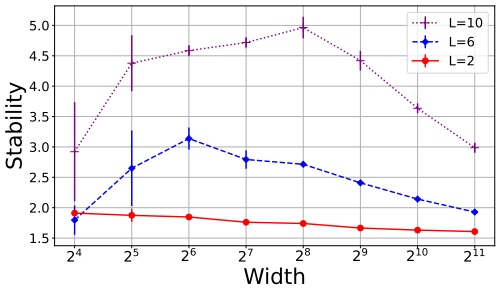



Figure 9: Relationship between the *perturbation stability* and depth of FCN under NTK initialization with different depths of $L = 2, 6$ and $10$.

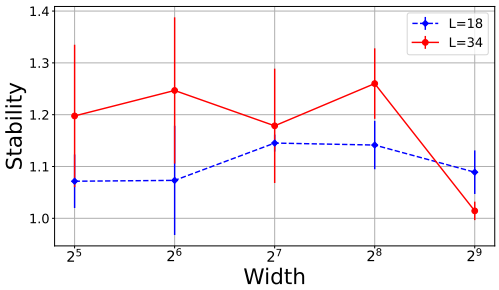



Figure 10: Relationship between the *perturbation stability* and width of ResNet-18 and ResNet-34.