Clarifying the Behavior and the Difficulty of Adversarial Training

Xu Cheng[1,2], **Hao Zhang**[2], **Yue Xin**[2], **Wen Shen**[2], **Quanshi Zhang**[2*]

[1]Nanjing University of Science and Technology,
[2]Shanghai Jiao Tong University

Abstract

Adversarial training is usually difficult to optimize. This paper provides conceptual and analytic insights into the difficulty of adversarial training via a simple theoretical study, where we derive an approximate dynamics of a recursive multi-step attack in a simple setting. Despite the simplicity of our theory, it still reveals verifiable predictions about various phenomena in adversarial training under real-world settings. First, compared to vanilla training, adversarial training is more likely to boost the influence of input samples with large gradient norms in an exponential manner. Besides, adversarial training also strengthens the influence of the Hessian matrix of the loss *w.r.t.* network parameters, which is more likely to make network parameters oscillate and boosts the difficulty of adversarial training.

1 Introduction

Although deep neural networks (DNNs) have shown promise in different tasks, the DNN was generally fooled by specific imperceptible perturbations of the input data (Goodfellow, Shlens, and Szegedy 2014; LeCun, Bengio, and Hinton 2015), which were termed *adversarial examples*. Adversarial training (Kurakin, Goodfellow, and Bengio 2016; Madry et al. 2018) is the most widely used strategy to defend against adversarial examples. Despite the effectiveness of adversarial training, extensive experiments have shown that adversarial training is considerably more difficult to optimize than vanilla training. Previous studies have explained this from various perspectives, such as sharp loss landscapes (Liu et al. 2020; Kanai et al. 2021; Wu, Xia, and Wang 2020; Yamada et al. 2021), obfuscated gradients (Athalye, Carlini, and Wagner 2018), and inhomogeneous data distributions (Sinha et al. 2017; Zhang and Wang 2019; Miyato et al. 2018).

Unlike previous research, we make a first attempt to provide conceptual and analytic insights into the difficulty of adversarial training from the perspective of the dynamics of generating multi-step perturbations. However, it is a significant challenge to solve the exact dynamics of multi-step perturbations analytically. Thus, we derive an exceedingly simple theory to approximate the dynamics of multi-step perturbations on a two-layer ReLU network under three simplifying assumptions (cf. A1-A3 in Section 3.1).

More crucially, this approximate-yet-analytic dynamics of perturbations provides new insights into different findings in adversarial training, which are responsible for the difficulty of adversarial training.

- **Finding 1.** The dynamics of the adversarial perturbation reveals that the perturbation strengthens gradient components along a few top-ranked eigenvectors of the Hessian matrix of the loss *w.r.t.* the input.
- **Finding 2.** Based on the above dynamics, we infer that adversarial training is more influenced by a few input samples with large gradient norms, compared to vanilla training. This unbalanced influence on adversarial training over different samples boosts the difficulty of adversarial training. According to our analysis, the normalization/regularization of perturbations in ℓ_2 attacks and ℓ_∞ attacks usually can alleviate such an imbalance.
- **Finding 3.** Adversarial training usually strengthens the influence of Hessian matrix of the loss *w.r.t.* network parameters, which makes network parameters more likely to oscillate and increases the difficulty of adversarial training.

Although our simple theory is derived on a two-layer ReLU network, our findings can still predict the imbalance problem and the oscillation problem in adversarial training on deeper and more complex networks in experiments, which account for the difficulty of adversarial training. Our simple theory also reveals interesting verifiable predictions about the dynamics of perturbations on deeper networks.

2 Related Work

Previous studies have analyzed the difficulty of adversarial training from different perspectives. Specifically, some works (Liu et al. 2020; Kanai et al. 2021; Wu, Xia, and Wang 2020; Yamada et al. 2021; Yu et al. 2018) considered that the sharp loss landscape *w.r.t.* network parameters resulted in the difficulty of adversarial training. Kurakin, Goodfellow, and Bengio (2016) demonstrated that label leaking hindered adversarial training. Tsipras et al. (2019) had proven compared to vanilla training, adversarial training relied on robust

*Quanshi Zhang is the corresponding author. He is with the Department of Computer Science and Engineering, the John Hopcroft Center, at the Shanghai Jiao Tong University, China. Correspondence to: Quanshi Zhang $<$ zqs1022@sjtu.edu.cn$>$.

features and did not use non-robust features for inference, which resulted in the inferior classification performance. The gradient-masking phenomenon (Papernot et al. 2017; Athalye, Carlini, and Wagner 2018; Tramèr et al. 2018) led to a false sense of security in defenses against adversarial examples. Please see Appendix A for detailed discussions.

Unlike previous works, this paper makes a first attempt to formulate the dynamics of perturbations in a simple setting. Despite the simplicity of our theory, it can still provide conceptual and analytic insights into the difficulty of adversarial training on DNNs in real-world settings.

3 Explaining Adversarial Perturbations and Adversarial Training

First, let us revisit adversarial training. Given a DNN f_θ parameterized by θ and an input sample $x \in \mathbb{R}^n$ with its true label y, an adversarial attack adds a human-imperceptible perturbation δ to fool the DNN with an adversarial example $x+\delta$, whose objective is generally formulated as follows.

$$\max_{\delta} L(f_\theta(x+\delta), y), \quad \text{s.t.} \quad \|\delta\|_p \le \epsilon, \tag{1}$$

where $f_\theta(x+\delta)$ denotes the network output, and $L(f_\theta(x+\delta), y)$ represents the loss function. ϵ is the constraint of the ℓ_p norm of the adversarial perturbation. To defend against adversarial attacks, adversarial training is generally formulated as a min-max game (Madry et al. 2018).

$$\min_{\theta} \mathbb{E}_{\{x,y\}}\big[\max_{\delta} L(f_\theta(x+\delta), y)\big], \quad \text{s.t.} \quad \|\delta\|_p \le \epsilon. \tag{2}$$

3.1 Analysis of Adversarial Perturbations

Generally speaking, it is a significant challenge to solve the dynamics of adversarial perturbations analytically. Thus, we analyze the following two-layer ReLU network f in a simple setting, so as to obtain an analytic approximation of the dynamics of the perturbation in a multi-step attack.

$$\begin{aligned} &h(x) = W_1^T x + b_1, \\ &z(x) = W_2^T \text{ReLU}(h(x)) + b_2 = W_2^T \Sigma\, h(x) + b_2, \\ &f(x) = \text{softmax}(z(x)) \text{ or } \text{sigmoid}(z(x)), \end{aligned} \tag{3}$$

where $W_1 \in \mathbb{R}^{n\times D}$ and $b_1 \in \mathbb{R}^D$. The diagonal matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_D) \in \mathbb{R}^{D\times D}$ represents the binary gating states of the ReLU layer, $\sigma_d \in \{0, 1\}$.

Then, the adversarial perturbation $\delta^{(m)}$ generated on the ReLU network f after m steps can be written as

$$\delta^{(m)} = \sum\nolimits_{t=0}^{m-1} \alpha \cdot g_{x+\delta^{(t)}}, \tag{4}$$

where α indicates the step size. Intuitively, the most straightforward method to craft a multi-step adversarial attack is to set $g_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$. For the widely used ℓ_2 attack and ℓ_∞ attack (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2018), the gradient is regularized or normalized as $g^{(\ell_2)}_{x+\delta^{(t)}} = g_{x+\delta^{(t)}}/\|g_{x+\delta^{(t)}}\|$, and $g^{(\ell_\infty)}_{x+\delta^{(t)}} = \text{sign}(g_{x+\delta^{(t)}})$.

However, the exact dynamics of perturbations in Eq. (4) is difficult to solve analytically. Therefore, to approximate the first analytic dynamics of perturbations, we make the following three simplifying assumptions (A1-A3). Intriguingly, our theory based on these simplifying assumptions provides an approximate-yet-analytic prediction about adversarial perturbations. We find, nicely, that this prediction still hold in experiments on deeper networks.

(A1) *We assume the perturbation is generated by the gradient ascent $g_{x+\delta^{(t)}}$ without regularization/normalization.*

Thus, we obtain a reduced dynamics for the adversarial perturbation generated by the gradient ascent, which is the most straightforward method to craft an attack. While A1 is a simplification, as we shall see, it can enable us to provide an interesting insight into the ℓ_2 attack or the ℓ_∞ attack. Please see Section 3.2 for details.

(A2) *We assume that a small perturbation does not significantly change gating states of the ReLU layer in Eq.* (3), *i.e., we assume $\widetilde{W}^T = W_2^T \Sigma W_1^T$ as a constant matrix during the attacking process, so that $z(x) = (\widetilde{W})^T x + \tilde{b}$.*

Appendix J shows a small approximation error based on this assumption[1]. In fact, many previous studies have made similar assumptions to ignore the change of gating states or remove ReLU layers (Tian, Chen, and Ganguli 2021; Kumar et al. 2022; Arora, Cohen, and Hazan 2018), because the change of gating states is usually quite unpredictable/chaotic. Since our fundamental goal is to obtain an analytic understanding of the dynamics of perturbations, it is useful to achieve this in the simple setting. Interestingly, we discover our final conclusions can generalize to deep nets.

Theorem 1 (Dynamics of perturbations of the m-step attack, proven in Appendix B)**.** *Let us fix a* ***small constant*** *β to reflect the overall adversarial strength. The step size is $\alpha = \beta/m$, where the step number m is a large integer. Based on assumptions A1 and A2, the adversarial perturbation $\delta^{(m)}$ can be approximated as*

$$\begin{aligned} \delta^{(m)} &= \sum\nolimits_{i=1}^{n} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \boldsymbol{\rho}, \\ g_{x+\delta^{(m)}} &= \sum\nolimits_{i=1}^{n} (1+\alpha\lambda_i)^m \gamma_i v_i, \end{aligned} \tag{5}$$

where λ_i and v_i denote the i-th largest eigenvalue of the matrix $\bar{H}_x = \widetilde{W} \bar{H}_z (\widetilde{W})^T$ and its corresponding eigenvector, respectively. The matrix[2] $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x+\delta^{(t)}), y)$, where $\Delta x^{(t)} = \alpha \cdot g_{x+\delta^{(t-1)}}$; $\gamma_i = g_x^T v_i \in \mathbb{R}$; $g_x = \frac{\partial}{\partial x} L(f(x), y)$. Each element $\boldsymbol{\rho}_i \in \mathbb{R}$ of the residual term $\boldsymbol{\rho} \in \mathbb{R}^n$ in Taylor expansion is proven to be of the order $O(1/m)$.

In Theorem 1, the matrix[2] $\bar{H}_x$ is used to approximate the second derivative of the loss *w.r.t.* the input sample x. Note that the second-order derivative of the output z of a ReLU network mainly comes from the sigmoid/softmax

[1] Appendix J has shown that the change of the largest eigenvalue of $\bar{H}_x$ during an attack is generally at the level of 10^{-4}—10^{-2}.

[2] To obtain an approximate analytic understanding of the dynamics of $\delta^{(m)}$, we use the matrix $\bar{H}_z$ to approximate the equivalent Hessian matrix. Intriguingly, we find that our theory under this approximation can still reveals verifiable predictions about the perturbation generated on deeper networks, *i.e.,* the error between the real perturbation and the theoretically derived one was at the level of 10^{-8}—10^{-4} in Table 1.

	3-layer MLP	4-layer MLP	5-layer MLP	3-layer CNN	4-layer CNN	5-layer CNN	3-layer ResCNN	4-layer ResCNN	5-layer ResCNN	10-layer ResCNN
Error κ	1.5×10^{-5}	3.5×10^{-6}	6.6×10^{-7}	3.4×10^{-7}	5.1×10^{-8}	4.7×10^{-8}	1.3×10^{-5}	1.5×10^{-5}	3.7×10^{-5}	1.7×10^{-4}

Table 1: The difference (*i.e.,* the error κ) between the derived perturbation $\hat{\delta}$ in Theorem 2 and the real perturbation generated on different ReLU networks. The small error κ verified Theorem 2, *i.e.,* the theoretical perturbation fitted well with the real one.

function in the end of the network. Thus, we can write $\bar{H}_x = \widetilde{W}\bar{H}_z(\widetilde{W})^T$, where z is the input of the sigmoid/-softmax function. Moreover, if the step number m is large enough, the residual term $\boldsymbol{\rho}$ is negligible.

(A3) *In the following manuscript, we assume that the adversarial perturbation is generated via an infinite-step attack with an infinitesimal step size.*

Assumption A3 is motivated by the fact that different settings of the step number m and the step size $\alpha = \beta/m$ slightly influence perturbations, when the overall adversarial strength β is given. Thus, we propose A3 to remove side effects of the step size and the step number in multi-step attacks, and simplify the story.

In this way, given a fixed adversarial strength β, the multi-step attack in Theorem 1 can be extended to a more idealized case of the infinite-step attack with the step number $m \to +\infty$ and the step size $\alpha = \beta/m \to 0$. This infinite-step perturbation $\hat{\delta} = \lim_{m\to+\infty} \alpha \sum_{t=0}^{m-1} \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$ further enables us to provide interesting verifiable insights into the difficulty of adversarial training.

Theorem 2 (Perturbations of the infinite-step attack, proven in Appendix C)**.** *Based on assumptions A1-A3, the infinite-step perturbation $\hat{\delta}$ can be re-written as follows.*

$$\hat{\delta} = \sum_{i=1}^{n} \frac{\exp(\beta\lambda_i) - 1}{\lambda_i} \gamma_i v_i, \;\; g_{x+\hat{\delta}} = \sum_{i=1}^{n} \exp(\beta\lambda_i)\gamma_i v_i. \qquad (6)$$

λ_i denotes the i-th largest eigenvalue of $\bar{H}_x$. $\bar{H}_x$ is defined in Theorem 1 and computed at a condition $m \to +\infty$.

Particularly, the residual term $\boldsymbol{\rho}$ in Theorem 1 is eliminated in Theorem 2. This is because $\boldsymbol{\rho}_i$ is proven to be on the order of $O(1/m)$, which means $\boldsymbol{\rho} \to \mathbf{0}$, subject to $m \to +\infty$.

Theorems 1 and 2 show the following two conclusions.

(C. 1) The adversarial perturbation strengthens gradient components in g_x along a few eigenvectors with large eigenvalues λ_i of the matrix $\bar{H}_x$ exponentially. A larger adversarial strength β constrains the perturbation along very few top-ranked eigenvectors more significantly.

(C. 2) Both the gradient norm $\|g_{x+\hat{\delta}}\|$ *w.r.t.* the perturbation and the perturbation norm $\|\hat{\delta}\|$ increase exponentially with the overall adversarial strength $\beta = \alpha m$.

- ***Experimental verification 1 of Theorem 2***. Although Theorem 2 was derived on a simple two-layer network, we tested whether our theory could predict the dynamics of adversarial perturbations on **deep** networks. That is, we checked whether $\hat{\delta}$ derived in Theorem 2 fitted well with the real perturbation δ^*. Specifically, we calculated the metric $\kappa = \mathbb{E}_x[\|\delta^* - \hat{\delta}\|]/\mathbb{E}_x[\|\delta^*\|]$ to evaluate the error between the derived perturbation $\hat{\delta}$ and the real perturbation δ^{*}[3]. To this end, we crafted perturbations δ^* on different ReLU networks with more than two linear layers for the MNIST dataset (LeCun et al. 1998). We followed the settings in (Ren et al. 2022) to construct various MLPs, CNNs, and CNNs with skip connections (namely ResCNNs), respectively.

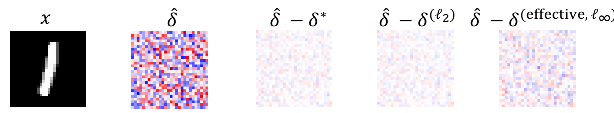

Figure 1: Visualization of the difference between the theoretical perturbation $\hat{\delta}$ and the real perturbation δ^*, the difference between $\hat{\delta}$ and the perturbation $\delta^{(\ell_2)}$ of the ℓ_2-PGD attack, and the difference between $\hat{\delta}$ and the effective component $\delta^{(\text{effective},\ell_\infty)}$ of the perturbation of the ℓ_∞-PGD attack. We found that $\hat{\delta}$ fitted well with δ^*, $\delta^{(\ell_2)}$, and $\delta^{(\text{effective},\ell_\infty)}$. All perturbations were generated using a 3-layer MLP. The magnitudes of the perturbations were enlarged for clarity.

Table 1 shows that the error κ for each network was small, *i.e.,* at the level of 10^{-8}—10^{-4}, which indicated that the theoretical perturbation $\hat{\delta}$ well fitted the real one. Thus, Theorem 2 was verified. Additionally, Fig. 1 shows that the theoretical perturbation $\hat{\delta}$ and real one δ^* were quite similar.

As a supplementary to the above experiment, Appendix J shows that adversarial perturbations do not significantly changed gating states Σ or the equivalent weight $\widetilde{W}^1$.

- ***Experimental verification 2 of Theorem 2***. We conducted experiments to check whether Theorem 2 derived on a simple two-layer network could predict the conclusion **(C. 2)** on **deep** networks. That is, we examined whether both the gradient $\|g_{x+\hat{\delta}}\|$ on the adversarial example and the perturbation $\|\hat{\delta}\|$ had exponentially increasing norms *w.r.t.* the overall adversarial strength $\beta \propto m$ (α is fixed here). Specifically, we generated perturbations $\hat{\delta}$ in Theorem 2 based on VGG-11 (Simonyan and Zisserman 2014), AlexNet (Krizhevsky, Sutskever, and Hinton 2012), and ResNet-18 (He et al. 2016), which were all learned using the MNIST dataset. The perturbation $\hat{\delta}$ was crafted by the gradient $g_{x+\hat{\delta}^{(t)}} = \frac{\partial}{\partial x} L(f(x+\hat{\delta}^{(t)}), y)$. Besides, we also generated two baseline perturbations via the ℓ_2 attack and the ℓ_∞ attack for comparison, *i.e.,* applying $g^{(\ell_2)}_{x+\delta^{(t)}}$, and $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ defined under Eq. (4). Please see Appendix L for the hyper-parameters of the attack. Considering that different samples were successfully attacked at different steps (denoted by m_{success}), we normalized the step number to generate the relative progress

[3] Table 1 reports results generated under assumption A2. Appendix J shows results generated without assumption A2.

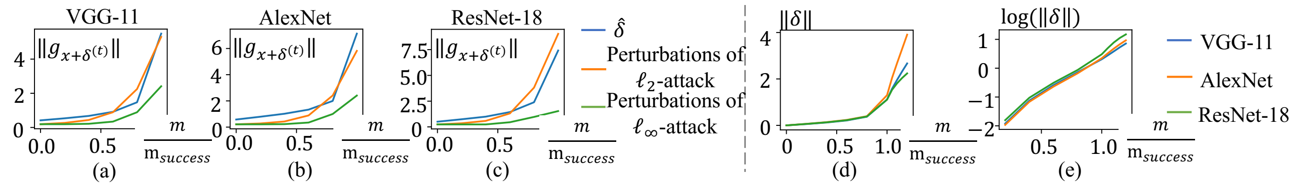

Figure 2: The increase of the overall adversarial strength $\beta \propto m$ (because α was fixed here) boosted both perturbation norms $\|\hat{\delta}\|$ and gradient norms $\|g_{x+\hat{\delta}^{(t)}}\|$ exponentially. In subfigures (a-d) , the exponential increase was verified in experiments without assuming gating states unchanged (A2). Whereas, in subfigure (e), we controlled the gating states of each ReLU layer in each step of the adversarial attack, to remove side effects caused by chaotic gating states, so that subfigure (e) exhibited a more clearly exponential increase of $\|\hat{\delta}\|$ *w.r.t.* m.

rate m/m_{success} as the horizontal axis in Fig. 2. This relative progress rate m/m_{success} was used to align the progress of the attack on different samples. Fig. 2 shows both $\|g_{x+\hat{\delta}}\|$ and $\|\hat{\delta}\|$ increased exponentially with $\beta \propto m$ (because α was fixed here), which verified conclusion **(C. 2)**.

3.2 Discussions on ℓ_2 Attacks and ℓ_∞ Attacks

Intriguingly, we find that our simple theory can also provide a new insight into ℓ_2 attacks and ℓ_∞ attacks. Thus, in this subsection, we discuss whether in the specific scenario of the infinite-step attack with the infinitesimal step size, the perturbation $\hat{\delta}$ in Theorem 2 can be used to analyze perturbations generated by the ℓ_2 attack and the ℓ_∞ attack.

For ℓ_2 attack. Let $\hat{\delta}^{(\ell_2)}$ denote the perturbation generated by the infinite-step ℓ_2 attack with the infinitesimal step size. *We have proven in Appendix D that based on assumption A2, $\hat{\delta}^{(\ell_2)}$ equals to $\hat{\delta}$, in the specific scenario of the infinite-step attack.* Furthermore, we examine whether the perturbation $\hat{\delta}$ in Theorem 2 can well approximate $\delta^{(\ell_2)}$ generated by the ℓ_2-PGD attack (Madry et al. 2018) with a few steps. Appendix D shows that the matching error $1 - \cos(\hat{\delta}, \delta^{(\ell_2)})$ between $\hat{\delta}$ and $\delta^{(\ell_2)}$ is at the level of 10^{-6}—10^{-4} for different networks. Additionally, Fig. 1 also illustrates the good fitness between the theoretically derived perturbation and real perturbations of the ℓ_2 attack and ℓ_∞ attack.

For ℓ_∞ attack. Let $\hat{\delta}^{(\ell_\infty)} = \sum_t \alpha \cdot g^{(\ell_\infty)}_{x+\delta^{(t)}} = \sum_t \alpha \cdot \text{sign}(g_{x+\delta^{(t)}})$ denote the perturbation of the infinite-step ℓ_∞ attack. We first disentangle the gradient $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ as $g^{(\ell_\infty)}_{x+\delta^{(t)}} = g^{(\text{effective})}_{x+\delta^{(t)}} + g^{(\text{ineffective})}_{x+\delta^{(t)}}$, where $g^{(\text{effective})}_{x+\delta^{(t)}} = o_x^T g^{(\ell_\infty)}_{x+\delta^{(t)}} o_x$ represents the gradient component along $g_{x+\delta^{(t)}}$, subject to $o_x = g_{x+\delta^{(t)}}/\|g_{x+\delta^{(t)}}\|$. In fact, we roughly consider $g^{(\text{effective})}_{x+\delta^{(t)}}$ is effective on ℓ_∞ attacks, while $g^{(\text{ineffective})}_{x+\delta^{(t)}}$ has negligible effects. Because $g^{(\text{effective})}_{x+\delta^{(t)}}$ is parallel to the exact gradient $g_{x+\delta^{(t)}}$ of the loss, $\hat{\delta}^{(\text{effective},\ell_\infty)} = \sum_t \alpha \cdot g^{(\text{effective})}_{x+\delta^{(t)}} = \sum_t \alpha \cdot o_x^T \text{sign}(g_{x+\delta^{(t)}}) o_x$ denotes the effective component *w.r.t.* the adversarial utility, which is disentangled from $\hat{\delta}^{(\ell_\infty)}$.

In this way, let us check whether we can roughly use $C_{\ell_\infty} \cdot \hat{\delta}/\|\hat{\delta}\|$ to approximate $\hat{\delta}^{(\text{effective},\ell_\infty)}$ in the infinite-step ℓ_∞ attack based on assumption A2, although there may be some errors. Here, $C_{\ell_\infty} \in \mathbb{R}$ reflects the total adversarial strength of the ℓ_∞ attack. To this end, we experimentally test the similarity between $C_{\ell_\infty} \cdot \hat{\delta}/\|\hat{\delta}\|$ and $\delta^{(\text{effective},\ell_\infty)}$ generated by the ℓ_∞-PGD attack with a few steps in Appendix D, which shows that the average matching error $1 - \cos(C_{\ell_\infty} \cdot \hat{\delta}/\|\hat{\delta}\|, \delta^{(\text{effective},\ell_\infty)})$ is as small as 3.4×10^{-5}.

Notice that $\hat{\delta}^{(\ell_2)}$ of the infinite-step ℓ_2 attack equals $\hat{\delta}$ under assumption A2. Thus, we use the notation $\hat{\delta}^{(\text{norm})} = C \cdot \hat{\delta}/\|\hat{\delta}\| = C \cdot \sum_{i=1}^n \frac{\exp(\beta\lambda_i)-1}{\lambda_i} \gamma_i v_i / \sqrt{\sum_{i=1}^n (\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i)^2}$ as a roughly unified approximation of ℓ_2 attacks and the effective component in ℓ_∞ attacks, where we set $C = \|\hat{\delta}\|$ for ℓ_2 attacks and set $C = C_{\ell_\infty}$ for ℓ_∞ attacks.

(C. 3) Above approximation based on $\hat{\delta}^{(\text{norm})}$ shows that a weak adversarial strength β makes the perturbation $\hat{\delta}^{(\ell_2)}$ (or $\hat{\delta}^{(\text{effective},\ell_\infty)}$) approximately parallel to the gradient g_x, due to $g_x = \sum_{i=1}^n \gamma_i v_i$. Whereas, a large adversarial strength β makes perturbations $\hat{\delta}^{(\ell_2)}$ (or $\hat{\delta}^{(\text{effective},\ell_\infty)}$) approximately parallel to the eigenvector v_1 *w.r.t.* the largest eigenvalue.

Constraint of adversarial perturbations. It is a significant challenge to derive the exact dynamics of perturbations analytically. To simplify the problem setting, we follow (Wang et al. 2021) to ignore the clip operation. Experiments in Appendix D and Appendix J show that our simple theory derived under a simple setting can still well predict dynamics of perturbations generated with the clip operation.

3.3 Difficulty of Adversarial Training

The dynamics of adversarial perturbations enables us to provide conceptual insights into the difficulty of adversarial training. Specifically, we analyze the effects of adversarial perturbations on weight optimization in adversarial training based on a simple two-layer ReLU network f. Intriguingly, we find that our analysis under this simple setting can still reveal verifiable predictions about adversarial training on deeper and more complex networks in later experiments.

Let $g_W = \frac{\partial}{\partial W} L(f(x), y)$ denote the gradient of the loss *w.r.t.* the weight of the first layer $W \overset{\text{def}}{=} W_1$ in Eq. (3)[4], when we use vanilla training to fine-tune the network on the original input sample x for a single step. In comparison, let $g_W^{(\text{adv})} = \frac{\partial}{\partial W} L(f(x+\hat{\delta}), y)$ denote the gradient of the loss *w.r.t.* W, when we train the network on the adversarial example $x + \hat{\delta}$ for a single step. Thus, $\Delta g_W = g_W^{(\text{adv})} - g_W$ denotes

[4]We can use W to approximate an equivalent weight matrix of multiple layers, because $\hat{\delta}$ does not significantly change most gating states of ReLU layers, according to assumption A2.

	3-layer MLP	4-layer MLP	5-layer MLP	3-layer CNN	4-layer CNN	5-layer CNN	3-layer ResCNN	4-layer ResCNN	5-layer ResCNN	10-layer ResCNN
Error $\kappa\prime$	3.9×10^{-5}	8.8×10^{-6}	1.5×10^{-6}	8.5×10^{-7}	1.3×10^{-7}	1.2×10^{-7}	3.4×10^{-5}	3.9×10^{-5}	9.0×10^{-5}	1.9×10^{-4}

Table 2: The difference (*i.e.,* the error $\kappa\prime$) between the theoretical effect $\hat{\phi}$ of $\tilde{g}_x^T\Delta\tilde{g}_x$ derived in Theorem 3 and the real effect ϕ^* measured in experiments. The small error $\kappa\prime$ verified Theorem 3.

additional effects of adversarial training on the gradient.

$$\Delta g_W = g_W^{(\text{adv})} - g_W = \frac{\partial}{\partial W}L(f(x+\hat{\delta}),y) - \frac{\partial}{\partial W}L(f(x),y). \tag{7}$$

Similarly, $\Delta g_W^{(\text{norm})} = g_W^{(\text{adv,norm})} - g_W$ represents the additional effects on the gradient brought by adversarial training, when we use the perturbation $\hat{\delta}^{(\text{norm})}$ (related to ℓ_2 and ℓ_∞ attacks).

$$\begin{aligned}\Delta g_W^{(\text{norm})} &= g_W^{(\text{adv,norm})} - g_W \\ &= \frac{\partial}{\partial W}L(f(x+\hat{\delta}^{(\text{norm})}),y) - \frac{\partial}{\partial W}L(f(x),y).\end{aligned} \tag{8}$$

Lemma 1 (proven in Appendix F)**.** *Let us focus on the cross-entropy loss $L(f(x),y)$. When the classification is based on a softmax operation, then the Hessian matrix $H_z = \frac{\partial^2}{\partial z\partial z^T}L(f(x),y)$ is positive semi-definite. When the classification is based on a sigmoid operation, the scalar $H_z \geq g_z^2 \geq 0$, if $z(x)\cdot y > 0, y \in \{-1,+1\}$ (i.e., the attacking has not completed). Here, $g_z = \frac{\partial}{\partial z}L(f(x),y) \in \mathbb{R}$.*

Theorems 3 and 4 yield insights into how perturbations $\hat{\delta}$ in Theorem 2 make effects on adversarial training.

Theorem 3 (proven in Appendix G)**.** *Based on Lemma 1 and assumption A2, let us focus on the binary classification based on a sigmoid function. Then, the effect of the adversarial perturbation $\hat{\delta}$ in Eq.* (6) *on the change of the gradient $\tilde{g}_x$ is formulated as follows.*

$$\begin{aligned}\tilde{g}_x^T\Delta\tilde{g}_x &= -\eta\tilde{g}_x^T\Delta g_W\tilde{g}_h \\ &= (e^{\mathcal{A}}-1)\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}),\end{aligned} \tag{9}$$

where η denotes the learning rate to update the weight; $\Delta\tilde{g}_x \overset{def}{=} -\eta\Delta g_W\tilde{g}_h$; $\tilde{g}_x = \frac{\partial z(x)}{\partial x}$; $\tilde{g}_h = \frac{\partial z(x)}{\partial h}$, $h = W^Tx + b_1$; $\Delta\tilde{g}_x^{(ori)} \overset{def}{=} -\eta g_W\tilde{g}_h$; $\mathcal{A} = \beta\bar{H}_z\|\tilde{g}_x\|^2 \in \mathbb{R}$.

In Theorem 3, $\Delta\tilde{g}_x = -\eta\Delta g_W\tilde{g}_h$ represents the additional effects of adversarial training on changing the gradient $\tilde{g}_x$, which are owing to the additional change $-\eta\Delta g_W$ on W[5] made by adversarial training. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x$ measures the significance of these additional changes along the direction of the gradient $\tilde{g}_x$. Similarly, $\Delta\tilde{g}_x^{(\text{ori})} = -\eta g_W\tilde{g}_h$ measures effects of vanilla training on changing $\tilde{g}_x$ in the current back-propagation[5].

Theorem 4 (proven in Appendix H)**.** *Based on Lemma 1 and assumption A2, let us focus on the binary classification based on a sigmoid function. Then, we derived the following equation w.r.t. adversarial training based on the perturbation $\hat{\delta}$ in Theorem 2, where $\Delta\tilde{g}_x^{(adv)} \overset{def}{=} -\eta g_W^{(adv)}\tilde{g}_h$.*

$$\begin{aligned}\tilde{g}_x^T\Delta\tilde{g}_x^{(adv)} &= -\eta\tilde{g}_x^T g_W^{(adv)}\tilde{g}_h \\ &= e^{\mathcal{A}}\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - \frac{\eta g_z^2(e^{2\mathcal{A}} - e^{\mathcal{A}})}{\bar{H}_z}\|\tilde{g}_h\|^2.\end{aligned} \tag{10}$$

In Theorem 4, $\Delta\tilde{g}_x^{(\text{adv})} = -\eta g_W^{(\text{adv})}\tilde{g}_h$ reflects effects of adversarial training on changing the gradient $\tilde{g}_x$[5]. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{adv})}$ represents the significance of these effects along the direction of gradient $\tilde{g}_x$.

A common understanding of adversarial training is to alleviate the current gradient g_x, *i.e.,* having a trend towards $g_x^T\Delta\tilde{g}_x < 0$, so as to boost the adversarial robustness. Then, **Theorems 3 and 4 reveal the following two conclusions**.

(C. 4) Adversarial training has a potential to reduce the significance of the current gradient. More importantly, if vanilla training has already alleviated the current gradient g_x (*i.e.,* $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{ori})} < 0$), then adversarial training will further strengthen such an alleviation exponentially.

The conclusion **(C. 4)** is obtained based on the following analysis. Because the second terms in Eq. (9) and Eq. (10) are both non-positive (owing to $\bar{H}_z > 0$ in Lemma 1), adversarial training tends to push $\tilde{g}_x^T\Delta\tilde{g}_x$ and $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{adv})}$ towards negative values, *i.e.,* alleviating the gradient g_x.

(C. 5) Adversarial training makes **additional effects** beyond vanilla training on strengthening the influence of a few samples with large $\bar{H}_z \in \mathbb{R}$ and large gradient norms $\|\tilde{g}_x\|$ exponentially. To be precise, these samples in adversarial training have about $\exp(\mathcal{A})$ times larger influence than vanilla training. We consider it as an imbalance over different samples in adversarial training.

The conclusion **(C. 5)** is obtained because $\tilde{g}_x^T\Delta\tilde{g}_x$ and $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{adv})}$ have exponential relation with $\mathcal{A} = \beta\bar{H}_z\|\tilde{g}_x\|^2$. These mechanisms make adversarial training more likely to oscillate in directions of a few samples (cf. Theorem 6), which increases the difficulty of adversarial training.

Besides, the derived imbalance of influence $\mathcal{A}$ on adversarial training over different samples provides new insights into the selection of an optimal step number for attacking in adversarial training. Please see Appendix M for details.

- *Experimental verification 1 of Theorem 3.* Although Theorem 3 was derived on a simple two-layer network, we checked whether our theory could predict the effects of adversarial perturbations on training **deep networks** adversarially. That is, we examined whether the theoretical derivation $\hat{\phi}$ computed according to the right side of Eq. (9) fitted well with the real values of $\phi^* = \tilde{g}_x^T\Delta\tilde{g}_x$ measured in experiments. Thus, we calculated a metric $\kappa\prime = \mathbb{E}_x[\|\phi^* - \hat{\phi}\|]/\mathbb{E}_x[\|\phi^*\|]$ to evaluate the fitness between the theoretical derivation $\hat{\phi}$ and the real effect ϕ^*, where ϕ^* was

[5]It is because adversarial training changes W by $-\eta g_W^{(\text{adv})}$, and vanilla training changes W by $-\eta g_W$, $\eta > 0$.

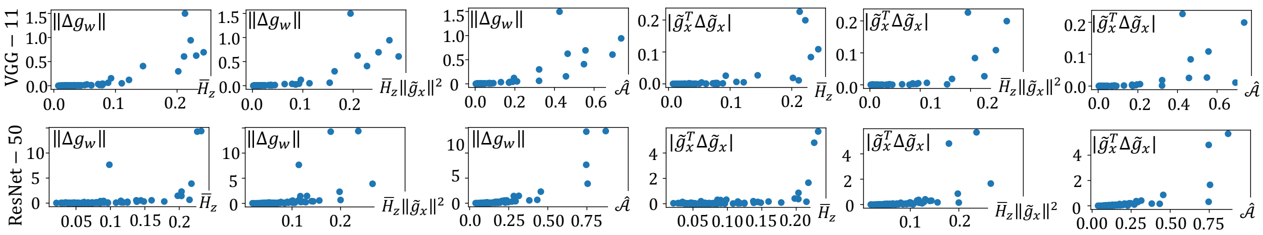

Figure 3: Illustration of the additional effects (quantified as $\|\Delta g_W\|$ and $|\tilde{g}_x^T \Delta \tilde{g}_x|$) of adversarial training on each sample (each dot) beyond vanilla training. This figure verified the conclusion **(C. 5)** that adversarial training boosted the influence of samples with large $\bar{H}_z$, $\bar{H}_z\|\tilde{g}_x\|^2$, and $\hat{\mathcal{A}}$ values, *i.e.*, these samples usually yielded larger $|\tilde{g}_x^T \Delta \tilde{g}_x|$ and $\|\Delta g_W\|$ values.

computed using real measurements of $\tilde{g}_x, \eta, g_W^{(\text{adv})}, g_W$, and $\tilde{g}_h$ on a ReLU network under assumption A2. In this way, we learned three types of ReLU networks on the MNIST dataset through adversarial training, following settings in (Ren et al. 2022) to construct MLPs, CNNs, and ResCNNs. Please see Appendix L for hyper-parameters of the attack in testing. Table 2 shows that for each ReLU network, the error $\kappa\prime$ was small, which indicates that the derived training effect $\hat{\phi}$ well matched the real effect ϕ^*. Thus, Theorem 3 was verified.

• *Experimental verification 2 of Theorem 3.* Here, we conducted experiments to check whether Theorem 3 (conclusion **(C. 5)**) derived on a simple two-layer network could predict the behavior of learning **deep networks** adversarially. We examined whether samples with large $\bar{H}_z$, large $\bar{H}_z\|\tilde{g}_x\|^2$ values, and large $\mathcal{A}$ values had large impacts $|\tilde{g}_x^T \Delta \tilde{g}_x|$ and $\|\Delta g_W\|$, *i.e.*, whether adversarial training boosted the influence of such samples (concluded from Theorem 3). Note that in real applications, the $\mathcal{A}$ value changed at each step of the attack, because the step-wise perturbation sometimes changed the matrix $\bar{H}_z$ and the gradient $\tilde{g}_x$. Thus, to be precise, we estimated the real $\mathcal{A}$ value in Theorem 3 as $\hat{\mathcal{A}} = \sum_{t=1}^m \alpha \bar{H}_z \|\tilde{g}_{x+\hat{\delta}^{(t)}}\|^2$, subject to $\tilde{g}_{x+\hat{\delta}^{(t)}} = \frac{\partial}{\partial x} z(x+\hat{\delta}^{(t)})$. To this end, we learned AlexNet, VGG-11, and ResNet-50 on the MNIST dataset via adversarial training on PGD attack, respectively. Fig. 3 shows that input samples with larger values of $\bar{H}_z$, $\bar{H}_z\|\tilde{g}_x\|^2$, and $\hat{\mathcal{A}}$ generally yielded larger $|\tilde{g}_x^T \Delta \tilde{g}_x|$ and $\|\Delta g_W\|$ values, which indicated adversarial training strengthened the influence of these samples. Thus, conclusion **(C. 5)** was verified on deep networks.

Additionally, Appendix K also visualized samples with large gradients $\|\tilde{g}_x\|$, and samples with small gradients $\|\tilde{g}_x\|$.

• *Experimental verification 3 of Theorem 3.* We also obtained the conclusion **(C. 5)** from Theorem 3 that the optimization direction of adversarial training was dominated by a few samples with large $\mathcal{A} = \beta \bar{H}_z \|\tilde{g}_x\|^2$ values. We conducted experiments to verify this conclusion on deep neural networks. Specifically, let $\Delta g_W = g_W^{(\text{adv})} - g_W$ denote the additional effect of adversarial training on a specific sample x beyond vanilla training. Based on the adversarially trained networks in *experimental verification 2 of Theorem 3*, we measured the cosine similarity $\cos(\Delta g_W, \Delta \bar{g}_W)$ between the training effect Δg_W on a single adversarial example and the average effect $\Delta \bar{g}_W = \mathbb{E}_{x+\hat{\delta}}[\Delta g_W]$ over different adversarial examples. Please see Appendix L for hyper-parameters

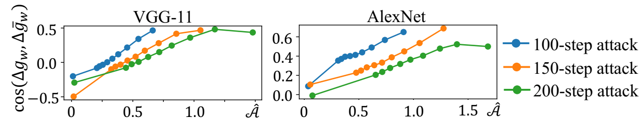

Figure 4: Verifying that the optimization direction of adversarial training was dominated by samples with large $\hat{\mathcal{A}}$ values (the conclusion **(C. 5)**). We divided samples into 10 groups with different ranges of $\hat{\mathcal{A}}$ values. We found that the average cosine similarity $\mathbb{E}_{x\in X_{\text{group}}}[\cos(\Delta g_W|_x, \Delta \bar{g}_W)]$ between $\Delta \bar{g}_W = \mathbb{E}_{x\in X_{\text{all}}}[\Delta g_W]$ and each sample's effect Δg_W in the group increased along with the value of $\hat{\mathcal{A}}$, which verified the conclusion **(C. 5)**.

of the attack in testing. Fig. 4 shows that the direction of the average effect $\Delta \bar{g}_W$ was similar to (dominated by) training effects of a few samples with large $\hat{\mathcal{A}}$ values (the real $\mathcal{A}$ calculated in experiments), which verified conclusion **(C. 5)**. We also conducted experiments on CIFAR-10 dataset in Appendix K, which shows the same phenomenon as in Fig. 4.

Effects of ℓ_2 attacks and ℓ_∞ attacks on adversarial training. Section 3.2 reveals that in our simple setting, we can roughly use to approximate the infinite-step ℓ_2 attack and the the effective component in ℓ_∞ attack. Thus, we further analyze the effects of the perturbation $\hat{\delta}^{(\text{norm})}$ on adversarial training, so as to approximate the effects of ℓ_2 attack and the ℓ_∞ attack on adversarial training.

Theorem 5 (proven in Appendix I)**.** *Based on Lemma 1 and assumption A2, let us focus on the binary classification based on a sigmoid function. We derive the following equation w.r.t. adversarial training on the perturbation* $\hat{\delta}^{(norm)}$.

$$\tilde{g}_x^T \Delta \tilde{g}_x^{(norm)} = C \cdot \Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|}\Big) \tilde{g}_x^T \Delta \tilde{g}_x^{(ori)} - C \cdot \frac{\eta g_z^2 \, \|\tilde{g}_h\|^2}{\bar{H}_z} \left(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|} + C \cdot \Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|}\Big)^2 \right), \tag{11}$$

where $\Delta \tilde{g}_x^{(norm)} \overset{def}{=} -\eta \Delta g_W^{(norm)} \tilde{g}_h = -\eta (g_W^{(adv,\, norm)} - g_W) \tilde{g}_h$.

In Theorem 5, $\Delta \tilde{g}_x^{(\text{norm})} = -\eta \Delta g_W^{(\text{norm})} \tilde{g}_h$ represents the additional effects of adversarial training on changing $\tilde{g}_x$, which are owing to the additional change $-\eta \Delta g_W^{(\text{norm})}$ on W^4 made by adversarial training. Thus, $\tilde{g}_x^T \Delta \tilde{g}_x^{(\text{norm})} = -\eta \tilde{g}_x^T \Delta g_W^{(\text{norm})} \tilde{g}_h$ reflects the significance of such additional effects along the direction of the gradient $\tilde{g}_x$.

According to Lemma 1, compared with the term $1/\|\hat{\delta}\|$ in Eq. (11), we prove that the strength of the training effect $\tilde{g}_x^T \Delta \tilde{g}_x^{(\text{norm})}$ is mainly determined by the term $\exp(\mathcal{A})/\|\hat{\delta}\| = \exp(\beta \bar{H}_z \|\tilde{g}_x\|^2)/\|\hat{\delta}\|$, owing to $\bar{H}_z \geq 0$. Moreover, given a relatively strong attack, the dominant term can be approximately represented as $\exp(\mathcal{A})/\|\hat{\delta}\| \approx \|g_x\| \cdot \exp(\beta \|\tilde{g}_x\|^2 (\bar{H}_z - g_z^2))$. It is because Theorem 2 indicates that a relatively strong adversarial strength β generally results in $\|\hat{\delta}\| \rightarrow \exp(\beta \|\tilde{g}_x\|^2 g_z^2)/\|g_x\|$ with an exponential strength (proven in Appendix I). Hence, we obtain following two conclusions.

(C. 6) Adversarial training based on $\hat{\delta}^{(\text{norm})}$ makes **additional effects** beyond vanilla training on strengthening influences of a specific set of samples, which must satisfy two requirements, *i.e.,* (1) they have large gradient norms $\|\tilde{g}_x\|$; (2) they are neither $z(x) \cdot y \rightarrow \infty$ nor $z(x) \cdot y \rightarrow 0$. Precisely, these samples in adversarial training on $\hat{\delta}^{(\text{norm})}$ have about $[\exp(\mathcal{A}) - 1]/\|\hat{\delta}\|$ times larger influence than vanilla training.

This is because, according to Lemma 1, as long as the attack has not yet succeeded, we have $\bar{H}_z - g_z^2 > 0$. However, for samples *s.t.* $z(x) \cdot y \rightarrow \infty$ or $z(x) \cdot y \rightarrow 0$, we get $\bar{H}_z - g_z^2 \rightarrow 0$, thereby obtaining a small value of $\exp(\mathcal{A})/\|\hat{\delta}\|$.

(C. 7) Unlike adversarial training based on perturbations $\hat{\delta}$ focusing on a few samples with large $\bar{H}_z$ and large gradient $\|\tilde{g}_x\|$ (cf. Theorems 3 and 4), the perturbation $\hat{\delta}^{(\text{norm})}$ alleviates the imbalance between different samples, but such an imbalance is still larger than vanilla training.

Oscillation of network parameters. Above proofs provide insights into that adversarial training makes network parameters oscillate in very few directions, which is considered as a common phenomenon in adversarial training. This insight is based on a typical claim in optimization (Cohen et al. 2021; Wu, Ma et al. 2018) that if the largest eigenvalue of the Hessian matrix of the loss *w.r.t* network parameters is sufficiently large, network parameters will oscillate along the eigenvector corresponding to the largest eigenvalue.

Here, although we do not directly prove that adversarial training can boost the largest eigenvalue of the Hessian matrix $\frac{\partial^2}{\partial W \partial W^T} L(f(x), y)$, Theorems 1 and 2 show that training on adversarial examples is somewhat equivalent to boosting the influence of the Hessian matrix.

Specifically, given a two-layer network f and an adversarial example $x + \hat{\delta}$ for adversarial training, let us consider the Hessian matrix $H_h \overset{\text{def}}{=} \frac{\partial^2}{\partial h \partial h^T} L(f(x), y)$ *w.r.t* $h = W^T x + b_1$. We use the second-order Taylor expansion to decompose the loss on adversarial examples $L(f(x + \hat{\delta}), y) = \text{Loss}(h + \Delta h)$, where $\Delta h = W^T \hat{\delta} \in \mathbb{R}^{D \times 1}$ denotes the change of the intermediate-layer feature h caused by $\hat{\delta}$. In this way, the loss function can be decomposed into $\text{Loss}(h + \Delta h) = \text{Loss}(h) + g_h^T \Delta h + \frac{1}{2!} \Delta h^T H_h \Delta h + R_2(\Delta h) = \text{Loss}(h) + g_h^T (W^T \hat{\delta}) + \frac{1}{2!} (W^T \hat{\delta})^T H_h (W^T \hat{\delta}) + R_2(\Delta h)$, where $g_h = \partial L(f(x), y)/\partial h$, and $R_2(\Delta h)$ indicates terms higher than the second order.

Theorem 6. *Let $\hat{\delta}_i \in \mathbb{R}$ denote the i-th dimension of $\hat{\delta}$. Then, the loss function $Loss(h + \Delta h)$ can be represented as*

$$Loss(h + \Delta h) = \tau + [\hat{\delta}_i \, g_{h,i}^T] w_i^T + w_i [\frac{1}{2!} \hat{\delta}_i^2 H_h] w_i^T, \quad (12)$$

where w_i denotes the i-th row of the weight matrix W, and τ is a constant w.r.t the change of w_i.

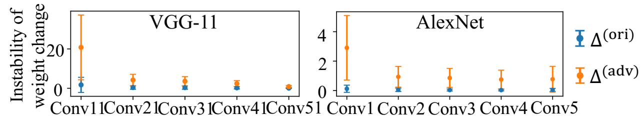

Figure 5: Comparison of the instability of weight gradients between vanilla training and adversarial training, which proved that adversarial training was more likely to make network parameters oscillation (the conclusion **(C. 8)**). $\Delta^{(\text{adv})}$ measured the instability of weight gradients in adversarial training, and $\Delta^{(\text{ori})}$ estimated the instability of weight gradients in vanilla training. We found that the value of $\Delta^{(\text{adv})}$ was larger than that of $\Delta^{(\text{ori})}$, which verified conclusion **(C. 8)**.

(C. 8) Adversarial training is more likely to make network parameters oscillate than vanilla training.

This conclusion is obtained because Theorem 6 shows that adversarial training is equivalent to setting the Hessian matrix $\frac{\partial^2}{\partial w_i \partial w_i^T} L(f(x), y)$ proportional to $\hat{\delta}_i^2 H_h$. Thus, the exponential increase of the perturbation $\hat{\delta}$ (shown in Eq. (6)) makes adversarial training more likely to oscillate.

- *Experimental verification of Theorem 6.* We checked whether the conclusion **(C. 8)** could well generalize to **deep networks**. Specifically, we trained AlexNet and VGG-11 on the MNIST dataset, and measured the effects of adversarial examples on the optimization of network parameters. To this end, we used an original input sample x and its corresponding adversarial example $x + \delta$ to update the weight $W_j \in \mathbb{R}^{D \times D}$ in each layer by the length $\|\Delta W_j\|$ and $\|\Delta W_j^{(\text{adv})}\|$, respectively.

In this way, the instability of the weight gradients in vanilla training could be measured as $\Delta^{(\text{ori})} = \|(\partial L(f(x|W_j + \Delta W_j), y)/\partial W_j) - (\partial L(f(x|W_j), y)/\partial W_j)\| / (D\|\Delta W_j\|)$. Similarly, the instability of the weight gradients in adversarial training could be estimated as $\|(\partial L(f(x + \delta|W_j + \Delta W_j^{(\text{adv})}), y)/\partial W_j) - (\partial L(f(x + \delta|W_j), y)/\partial W_j)/(D\|\Delta W_j^{(\text{adv})}\|)$. Here, $f(x|W_j + \Delta W_j)$ denotes the output of the ReLU network f, when the weight of the j-th linear layer was updated to $W_j + \Delta W_j$. Please see Appendix L for more details regarding experimental settings.

Fig. 5 compares the instability of weight gradients between vanilla training $\Delta^{(\text{ori})}$ and adversarial training $\Delta^{(\text{adv})}$. We discovered that adversarial training exhibited much higher instability than vanilla training, which demonstrated that adversarial training boosted the influence of Hessian matrix *w.r.t.* the network parameters. This verified the conclusion **(C. 8)**.

4 Conclusion and Discussion

This paper makes a first attempt to derive an approximate-yet-analytic dynamics of perturbations on a simple two-layer ReLU network. Based on this, we provide conceptual insights into the difficulty of adversarial training. Although our theory is derived under simplifying assumptions, it can still reveal verifiable predictions about dynamics of perturbations, the imbalance problem, and the oscillation problem in adversarial training under real-world settings.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (2021ZD0111602), the National Nature Science Foundation of China (62276165), Shanghai Natural Science Foundation (21JC1403800,21ZR1434600). This work is also partially supported by Huawei Technologies Inc.

References

Arora, S.; Cohen, N.; and Hazan, E. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, 244–253. PMLR.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, 274–283. PMLR.

Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; and Wang, Z. 2020. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*.

Cohen, J.; Kaur, S.; Li, Y.; Kolter, J. Z.; and Talwalkar, A. 2021. Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. In *International Conference on Learning Representations*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Kanai, S.; Yamada, M.; Takahashi, H.; Yamanaka, Y.; and Ida, Y. 2021. Smoothness Analysis of Adversarial Training. *arXiv preprint arXiv:2103.01400*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kumar, A.; Raghunathan, A.; Jones, R. M.; Ma, T.; and Liang, P. 2022. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, C.; Huang, Z.; Salzmann, M.; Zhang, T.; and Süsstrunk, S. 2021. On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training. *arXiv preprint arXiv:2112.07324*.

Liu, C.; Salzmann, M.; Lin, T.; Tomioka, R.; and Süsstrunk, S. 2020. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, 33: 21476–21487.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Qian, Z.; Zhang, S.; Huang, K.; Wang, Q.; Gu, B.; Xiong, H.; and Yi, X. 2022. Perturbation Diversity Certificates Robust Generalisation.

Ren, J.; Li, M.; Zhou, M.; Chan, S.-H.; and Zhang, Q. 2022. Towards Theoretical Analysis of Transformation Complexity of ReLU DNNs. In *International Conference on Machine Learning*, 18537–18558. PMLR.

Rice, L.; Wong, E.; and Kolter, Z. 2020. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, 8093–8104. PMLR.

Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; and Madry, A. 2018. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sinha, A.; Namkoong, H.; Volpi, R.; and Duchi, J. 2017. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Stutz, D.; Hein, M.; and Schiele, B. 2020. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, 9155–9166. PMLR.

Tian, Y.; Chen, X.; and Ganguli, S. 2021. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, 10268–10278. PMLR.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *International Conference on Learning Representations*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Wang, X.; Ren, J.; Lin, S.; Zhu, X.; Wang, Y.; and Zhang, Q. 2021. A Unified Approach to Interpreting and Boosting Adversarial Transferability. In *International Conference on Learning Representations*.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969.

Wu, L.; Ma, C.; et al. 2018. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31.

Yamada, M.; Kanai, S.; Iwata, T.; Takahashi, T.; Yamanaka, Y.; Takahashi, H.; and Kumagai, A. 2021. Adversarial Training Makes Weight Loss Landscape Sharper in Logistic Regression. *arXiv preprint arXiv:2102.02950*.

Yu, F.; Liu, C.; Wang, Y.; Zhao, L.; and Chen, X. 2018. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*.

Zhai, R.; Cai, T.; He, D.; Dan, C.; He, K.; Hopcroft, J.; and Wang, L. 2019. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*.

Zhang, H.; and Wang, J. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems*, 32.