WHY ADVERSARIAL TRAINING OF RELU NETWORKS IS DIFFICULT?

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper mathematically derives an analytic solution of the adversarial perturbation on a ReLU network, and theoretically explains the difficulty of adversarial training. Specifically, we formulate the dynamics of the adversarial perturbation generated by the multi-step attack, which shows that the adversarial perturbation tends to strengthen eigenvectors corresponding to a few top-ranked eigenvalues of the Hessian matrix of the loss *w.r.t.* the input. We also prove that adversarial training tends to strengthen the influence of unconfident input samples with large gradient norms in an exponential manner. Besides, we find that adversarial training strengthens the influence of the Hessian matrix of the loss *w.r.t.* network parameters, which makes the adversarial training more likely to oscillate along directions of a few samples, and boosts the difficulty of adversarial training. Crucially, our proofs provide a unified explanation for previous findings in understanding adversarial training (Liu et al., 2020; Kanai et al., 2021; Wu et al., 2020; Yamada et al., 2021; Athalye et al., 2018; Tsipras et al., 2019; Ilyas et al., 2019; Liu et al., 2021; Chen et al., 2020; Rice et al., 2020).

1 INTRODUCTION

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

Unlike previous research, this paper aims to derive an approximate analytic solution to adversarial perturbations on a ReLU network, and further theoretically proves why adversarial training is difficult. However, considering adversarial training is a complex algorithm with lots of tricks, we summarize common settings in adversarial training into four assumptions (cf. A1-A4 in Section 2), so as to simplify the proof. Particularly, we have also conducted experiments in Section 2 to verify that our theorems can well explain adversarial training in real applications.

Then, based on the derived analytic solution to the adversarial perturbation of the multi-step attack, we further explain their effects on adversarial training. Hence, we obtain the following three conclusions.

(1) The adversarial perturbation strengthens eigenvectors corresponding to a few top-ranked eigenvalues of the Hessian matrix of the loss *w.r.t.* the input.

(2) Adversarial training mainly focuses on a few unconfident input samples with large gradient norms. Furthermore, we prove that the normalization/regularization of perturbations in ℓ_2 attacks and ℓ_∞ attacks alleviate such an imbalance.

(3) Adversarial training strengthens the influence of the Hessian matrix of the loss *w.r.t.* network parameters. Hence, adversarial training is more likely to make network parameters oscillate, which explains the difficulty of adversarial training, as well.

More crucially, our theoretical proof also provides a theoretical foundation, which may explain various previous findings/understandings of adversarial training (Liu et al., 2020; Kanai et al., 2021; Wu et al., 2020; Yamada et al., 2021; Athalye et al., 2018; Tsipras et al., 2019; Ilyas et al., 2019; Liu et al., 2021; Chen et al., 2020; Rice et al., 2020).

Contributions of this paper are summarized as follows. (1) We derive an analytic solution that explains the dynamics of the adversarial perturbation. (2) We prove that adversarial training strengthens the influence of a few input samples, and increases the likelihood of the oscillation of network parameters, which boosts the difficulty of adversarial training. (3) Our proofs can explain the benefit of the normalization/regularization of perturbations in ℓ_2 attacks and ℓ_∞ attacks, and can provide a unified view to understand a total of ten previous studies in adversarial training.

2 EXPLAINING ADVERSARIAL PERTURBATIONS AND ADVERSARIAL TRAINING

Let us first revisit adversarial training. Given a DNN f_θ parametrized by θ and an input sample $x \in \mathbb{R}^n$ with its true label y, the adversarial attack adds a human-imperceptible perturbation δ to fool the DNN with the adversarial example $x + \delta$, whose objective is usually 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 often formulated as a min-max game (Madry et al., 2018).

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

2.1 ANALYSIS OF ADVERSARIAL PERTURBATIONS

To analyze the dynamics of adversarial perturbations, let us consider the multi-step attack as follows, where $\delta^{(t)}$ is referred to as the perturbation generated after attacking for t steps; m represents the total number of steps; α denotes the step size.

$$\delta^{(m)} = \sum\nolimits_{t=0}^{m-1} \alpha \cdot g_{x+\delta^{(t)}}. \tag{3}$$

To simplify the story, we first analyze the most straightforward solution to the multi-step adversarial attack, $g_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$. Then, we will extend the analysis to the widely-used ℓ_2 attack and the ℓ_∞ attack (Dong et al., 2018; Goodfellow et al., 2014; Madry et al., 2018), where they regularize or normalize the gradient as $g^{(\ell_2)}_{x+\delta^{(t)}} = g_{x+\delta^{(t)}} / \|g_{x+\delta^{(t)}}\|$, and $g^{(\ell_\infty)}_{x+\delta^{(t)}} = \text{sign}(g_{x+\delta^{(t)}})$.

Without loss of generality, let us consider a ReLU network f and an input sample x. $z(x)$ denotes the input feature of the top layer (*e.g.* a softmax layer $f(x) = \text{softmax}(z(x))$, or a sigmoid layer $f(x) = \text{sigmoid}(z(x))$). The following equation formulates how the network uses the feature h of the j-th linear layer to compute $z(x)$.

$$z(x) = W_l^T(\dots \Sigma_{j+1}(W_{j+1}^T \Sigma_j h + b_{j+1}) \dots) + b_l, \tag{4}$$

where $h = W_j^T x' + b_j$ denotes the linear transformation in the j-th layer, subject to $x' = \Sigma_{j-1}(W_{j-1}^T(\dots \Sigma_1(W_1^T x + b_1) \dots) + b_{j-1})$. W_j and b_j denote the weight and bias of the j-th linear layer, respectively. The matrix $\Sigma_j = diag(\sigma_{j,1}, \sigma_{j,2}, \dots, \sigma_{j,D}) \in \mathbb{R}^{D \times D}$ represents gating states of the j-th gating layer (*e.g.* a ReLU layer, or a MaxPooling layer), $\sigma_{j,d} \in \{0, 1\}$.

To simplify the proof for the analytic solution to adversarial perturbations, we summarize common settings in adversarial training into the following assumptions, without hurting the trustworthiness.
(A1) We assume that the constraint of adversarial perturbations can be ignored. It is because there exists a common fact in adversarial training that people usually learn a robust network on relatively weak adversarial perturbations (Wong et al., 2020), which often have not reached the constraint $\|\delta\|_p < \epsilon$ for perturbations δ. This has been widely considered as an effective trick to reduce the optimization difficulty.
(A2) To simplify the proof, we assume that the adversarial perturbation is generated by the most straightforward method, *i.e.,* gradient ascent without regularization/normalization, although many attacking methods (Dong et al., 2018; Goodfellow et al., 2014; Madry et al., 2018) regularize or

normalize the gradient as a trick to speed up the multi-step attack. Experimental results in Appendix C have shown that the normalized perturbation in Remark 1 can approximately explain the ℓ_2 attack.
(A3) Because the change of gating states in multi-step attacks is usually chaotic and unpredictable for analysis, it is difficult to theoretically model the unpredictable change of gating states during attacking. Moreover, the chaotic change of gating states over numerous neurons may have mutually offsetting effects on adversarial training, to some extent. Thus, we make the following assumption.

Assumption 1. *We simplify our research into an idealized adversarial attack, whose adversarial perturbation does not significantly change gating states in gating layers. In this scenario, we approximate the ReLU network f to a linear model, i.e., $z(x) \approx (\tilde{W})^T x + \tilde{b}$, $\tilde{W}^T = W_l^T \Sigma_{l-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$.*

Before the later analysis of the ℓ_2 attack and the ℓ_∞ attack, we first focus on the original form of the multi-step attack, *i.e.* perturbation generated via $g_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$.

Theorem 1 (Dynamics of perturbations of the m-step attack, proven in Appendix A)**.** *Let us assume that the gradient $g_{x+\delta^{(t)}}$ is a Lipschitz function with the Lipschitz constant K, $\|g_{x+\delta^{(t)}} - g_x\| \leq K \cdot \|\delta^{(t)}\|$. Then, based on Assumption 1, the adversarial perturbation $\delta^{(m)}$ can be approximated as follows, where the overall adversarial strength $\beta = \alpha m$ is a small constant, and m is a large integer.*

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

Here, λ_i and v_i denote the i-th largest eigenvalue of the matrix $\bar{H}_x = \tilde{W} \bar{H}_z (\tilde{W})^T$ and its corresponding eigenvector, respectively, where $\bar{H}_x$ is used to approximate[1] the second derivative of the loss w.r.t. the input sample x. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ 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 $\Delta x^{(t)} = \alpha \cdot g_{x+\delta^{(t-1)}}$ denotes the perturbation updated at the t-th step. $\gamma_i = g_x^T v_i \in \mathbb{R}$ represents the projection of the gradient $g_x = \frac{\partial}{\partial x} L(f(x), y)$ on the eigenvector v_i. Particularly, if the step number m is large, then the residual term in the Taylor expansion $\boldsymbol{\rho} \in \mathbb{R}^n$ is ignorable, since each element $\boldsymbol{\rho}_i \in \mathbb{R}$ is proven to be the order of $O(1/m)$.

(A4) Notice that different parameter settings of multi-step attacks (such as the step size or the step number) may make slightly different influences on adversarial perturbations. Thus, to remove side effects of such settings and simplify the story, in the following manuscript, we assume the adversarial perturbation in adversarial training is generated via the infinite-step attack with the infinitesimal step size. In this way, the m-step attack in Theorem 1 can be extended to a more idealized case of the infinite-step attack as follows, which is further used to analyze adversarial training.

Theorem 2 (Perturbations of the infinite-step attack, proven in Appendix B)**.** *$\beta = \alpha m$ reflects the overall adversarial strength of the infinite-step attack with the step number $m \to +\infty$ and the step size $\alpha = \beta/m \to 0$. Then, based on Assumption 1, this infinite-step adversarial perturbation $\hat{\delta} = \lim_{m\to+\infty} \alpha \sum_{t=0}^{m-1} \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$ can be re-written as follows.*

$$\hat{\delta} = \sum\nolimits_{i=1}^{n} \frac{\exp(\beta\lambda_i) - 1}{\lambda_i} \gamma_i v_i + \hat{\boldsymbol{\rho}}, \quad g_{x+\hat{\delta}} = \sum\nolimits_{i=1}^{n} \exp(\beta\lambda_i) \gamma_i v_i. \tag{6}$$

Here, $\hat{\boldsymbol{\rho}} \in \mathbb{R}^n$ denotes an ignorable residual term in the Taylor expansion, because each element $\hat{\boldsymbol{\rho}}_i \in \mathbb{R}$ is proven to be the order of $O(1/m)$.

Theorem 1 and Theorem 2 show the following two conclusions.
(C. 1) The adversarial perturbation strengthens gradient components in g_x along eigenvectors corresponding to a few top-ranked eigenvalues λ_i of the matrix $\bar{H}_x$ exponentially. Furthermore, a larger adversarial strength β, such as attacking for more steps, is more likely to force the perturbation to change along fewer top-ranked eigenvectors.
(C. 2) Both the gradient norm $\|g_{x+\hat{\delta}}\|$ *w.r.t.* the adversarial perturbation, and the perturbation norm $\|\hat{\delta}\|$ increase along with the overall adversarial strength $\beta = \alpha m$ exponentially.

- *Experimental verification 1 of Theorem 2*. We have derived an analytic solution to the perturbation in Theorem 2. Hence, we conducted experiments to verify the trustworthiness of Theorem 2, *i.e.,*

[1]Theoretically, it is very hard to derive the analytic solution to the adversarial perturbation $\delta^{(m)}$ without such an approximation. Hence, we use the matrix $\bar{H}_z$ to approximate the equivalent Hessian matrix, which allows us to derive the first analytic solution to the adversarial perturbation of the multi-step attack. More crucially, experimental results in Table 1 verified the trustworthiness of such an approximation, *i.e.,* the error between the real perturbation and the theoretically derived solution is at the level of 10^{-8}—10^{-5}.

Table 1: The error κ between the derived analytic solution $\hat{\delta}$ in Theorem 2 and the real perturbation generated on different ReLU networks. The small error κ successfully verified Theorem 2.

	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
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}

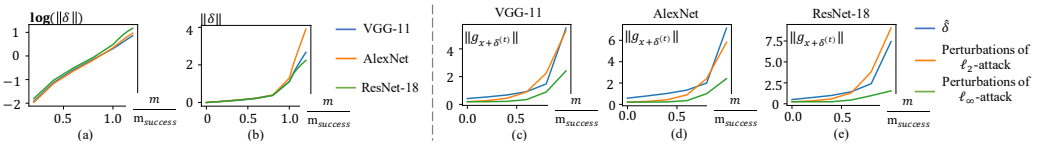

Figure 1: Exponential increases of perturbation norms $\|\hat{\delta}\|$ and gradient norms $\|g_{x+\hat{\delta}^{(t)}}\|$ with the overall adversarial strength $\beta \propto m$ (because α was fixed here). Note that the instability of gating states might bring in uncertainty and lead to an unclean phenomenon of an exponential increase. Whereas, in subfigure (a), we controlled the gating states of each ReLU layer in each step of the adversarial attack, in order to remove side effects brought by the chaotic gating states. Hence, subfigure (a) exhibited a more clearly exponential increase of $\|\hat{\delta}\|$ *w.r.t.* m.

checking whether the solution $\hat{\delta}$ derived in Theorem 2 well fitted the real perturbation δ^* measured in practice. Specifically, we calculated the metric $\kappa = \mathbb{E}_x[\|\delta^* - \hat{\delta}\|]/\mathbb{E}_x[\|\delta^*\|]$ to evaluate the error between the derived solution $\hat{\delta}$ and the real perturbation δ^*. To this end, we generated adversarial perturbations on different ReLU networks, where we followed settings in (Ren et al., 2022) to construct various MLPs, CNNs, and CNNs with skip connections (namely ResCNNs), respectively. Table 1 reports the error κ, which was small for each network, *i.e.,* at the level of 10^{-8}—10^{-5}. Thus, the theoretically derived perturbation $\hat{\delta}$ on Assumption 1 well fitted the real one, which successfully verified Theorem 2. In other words, various assumptions[1] made ignorable impacts on the trustworthiness of Theorem 2, *i.e.,* the derived solution could reflect the real dynamics of adversarial perturbations. Please see Appendix J for more results and experimental settings.

- *Experimental verification 2 of Theorem 2.* Theorem 2 indicates that 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). Here, we conducted experiments to verify this conclusion. Specifically, we generated perturbations $\hat{\delta}$ in Theorem 2 based on VGG-11 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), and ResNet-18 (He et al., 2016), which were learned on the MNIST dataset (LeCun et al., 1998), respectively. Then, 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. (3). Please see Appendix L for more details of experimental settings. Considering different samples were successfully attacked at different steps, we normalized the step number, m/m_{success}, as the horizontal axis in Fig. 1. Here, the relative progress rate m/m_{success} was used to align the progress of the adversarial attacking on different samples. Fig. 1 shows that both the gradient norm $\|g_{x+\hat{\delta}}\|$, and the perturbation norm $\|\hat{\delta}\|$ increased exponentially with $\beta \propto m$ (because α was fixed here), which verified Theorem 2.

Approximation for ℓ_2 attacks and ℓ_∞ attacks. As two typical attacking methods, the ℓ_2 attack and the ℓ_∞ attack usually regularize/normalize the adversarial strength in each step by applying $g^{(\ell_2)}_{x+\delta^{(t)}} = g_{x+\delta^{(t)}}/\|g_{x+\delta^{(t)}}\|$ and $g^{(\ell_\infty)}_{x+\delta^{(t)}} = \text{sign}(g_{x+\delta^{(t)}})$, respectively. In fact, for the ℓ_∞ attack, we can roughly consider that only the gradient component $o_x^T g^{(\ell_\infty)}_{x+\delta^{(t)}} o_x$ disentangled from $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ along $\frac{\partial}{\partial x} L(f(x), y)$ is effective, where $o_x = \frac{\partial}{\partial x} L(f(x), y)/\|\frac{\partial}{\partial x} L(f(x), y)\|$ is the unit vector in the direction of $\frac{\partial}{\partial x} L(f(x), y)$. However, it is quite complex to analyze the exact attacking behavior. **Therefore, in Remark 1, we just brutally normalize the perturbation in Theorem 2 to roughly approximate the regularization/normalization of perturbations in ℓ_2 attacks and ℓ_∞ attacks.** Nevertheless, the trustworthiness of the approximation in Remark 1 was experimentally verified. Table 3 in Appendix C shows that the matching error between $\hat{\delta}^{(\text{norm})}$ and the real perturbation generated via ℓ_2 attack was at the level of 10^{-6}—10^{-4}, which successfully verified the trustworthiness of Remark 1.

Remark 1 (Normalized perturbation of the infinite-step attack)**.** *Based on Theorem 2, we ignore residual terms $\hat{\boldsymbol{\rho}}$, where $\hat{\boldsymbol{\rho}}_i$ is proven to be the order of $O(1/m)$. Then, the perturbation of the infinite-*

step ℓ_2 attack generated via $g^{(\ell_2)}_{x+\delta^{(t)}}$, and the perturbation of the infinite-step ℓ_∞ attack generated via $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ can be approximated as follows.

$$\hat{\delta}^{(norm)} \approx C \cdot \hat{\delta}/\|\hat{\delta}\| = C \cdot \sum_{i=1}^{n} \frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i v_i \Big/ \sqrt{\sum_{i=1}^{n}(\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i)^2}, \tag{7}$$

where $C \in \mathbb{R}$ reflects the total adversarial strength of the ℓ_2 attack or the ℓ_∞ attack.

(C. 3) Remark 1 reveals that a weak adversarial strength β makes the normalized perturbation $\hat{\delta}^{(\text{norm})}$ approximately parallel to the gradient g_x. Whereas, a large adversarial strength makes the normalized perturbation $\hat{\delta}^{(\text{norm})}$ approximately parallel to the eigenvector v_1 *w.r.t.* the largest eigenvalue.

2.2 Explaining the difficulty of adversarial training

In this subsection, we explain the effects of adversarial perturbations on weight optimization in adversarial training. Without loss of generality, we analyze the learning dynamics of the j-th linear layer of the ReLU network f. Specifically, if we use vanilla training to fine-tune the network on the original input sample x for a single step, then the gradient of the loss *w.r.t.* the weight of the j-th layer $W^T = W_j^T\Sigma_{j-1}\cdots\Sigma_2 W_2^T\Sigma_1 W_1^T$[2] is given as $g_W = \frac{\partial}{\partial W}L(f(x),y)$. In comparison, if we train the network on the adversarial example $x+\hat{\delta}$ for a single step, then we will get the gradient $g_W^{(\text{adv})} = \frac{\partial}{\partial W}L(f(x+\hat{\delta}),y)$. In this way, $\Delta g_W = g_W^{(\text{adv})} - g_W$ denotes additional effects of adversarial training on the gradient.

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

Similarly, $\Delta g_W^{(\text{norm})} = g_W^{(\text{adv,norm})} - g_W$ represents additional effects on the gradient brought by adversarial training, when we use the normalized perturbation $\hat{\delta}^{(\text{norm})}$ in Remark 1 (related to the ℓ_2 attack and the ℓ_∞ attack).

$$\Delta g_W^{(\text{norm})} = g_W^{(\text{adv,norm})} - g_W = \frac{\partial}{\partial W}L(f(x+\hat{\delta}^{(\text{norm})}),y) - \frac{\partial}{\partial W}L(f(x),y). \tag{9}$$

Assumption 2 (proven in Appendix D)**.** *The analysis of binary classification based on a sigmoid function, $f(x) = \frac{1}{1+exp(-z(x))}, z(x) \in \mathbb{R}$, can also explain the multi-category classification with a softmax function, $f(x) = \frac{exp(z'_1)}{\sum_{i=1}^{c} exp(z'_i)}, z' \in \mathbb{R}^c$, if the second-best category is much stronger than other categories. In this case, attacks on the multi-category classification can be approximated by attacks on the binary classification between the best and the second-best categories, i.e., $f(x) \approx \frac{1}{1+exp(-z)}$, subject to $z = z'_1 - z'_2 \in \mathbb{R}$. z'_1 and z'_2 are referred to as network outputs corresponding to the best category and the second-best category, respectively.*

Lemma 1 (proven in Appendix E)**.** *Let us focus on the cross-entropy loss $L(f(x),y)$. If 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. If the classification is based on a sigmoid operation, the scalar $H_z \geq g_z^2 \geq 0$, as long as the attacking has not finished (still $z(x)\cdot y > 0, y \in \{-1,+1\}$). Here, $g_z = \frac{\partial}{\partial z}L(f(x),y) \in \mathbb{R}$.*

Theorems 3 and 4 explain training effects of the perturbation $\hat{\delta}$ in Theorem 2 on adversarial training.

Theorem 3 (proven in Appendix F)**.** *Based on Assumptions 1 and 2, 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 $\tilde{g}_x = \frac{\partial z(x)}{\partial x}$ is formulated as follows. $\Delta\tilde{g}_x = -\eta\Delta g_W \tilde{g}_h$ represents the additional effects of adversarial training on changing $\tilde{g}_x$, because adversarial training makes an additional change $-\eta\Delta g_W$ on W[3]. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x$ measures the significance of such additional changes along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T\Delta\tilde{g}_x = -\eta\tilde{g}_x^T\Delta g_W\tilde{g}_h = (e^{\mathcal{A}}-1)\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}}-e^{\mathcal{A}}), \tag{10}$$

where $\tilde{g}_h = \frac{\partial z(x)}{\partial h}$, $\mathcal{A} = \beta\bar{H}_z\|\tilde{g}_x\|^2 \in \mathbb{R}$, and η denotes the learning rate to update the weight. Considering the footnote[3], $\Delta\tilde{g}_x^{(ori)} = -\eta g_W\tilde{g}_h$ measures the effects of vanilla training on changing $\tilde{g}_x$ in the current back-propagation.

[2]For simplicity, we analyze the equivalent weight W for all the first j linear layers, but actually W has similar behaviors as W_j, without hurting the generality of the analysis. Please see Appendix F for discussion.

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

Table 2: Experimental verification of Theorem 3 on different adversarially trained ReLU networks. The small error κ verified Theorem 3.

	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
Error κ	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}

Theorem 4 (proven in Appendix G)**.** *Based on Assumptions 1 and 2, 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 perturbations $\hat{\delta}$ in Theorem 2. Considering the footnote[3], $\Delta\tilde{g}_x^{(adv)} = -\eta g_W^{(adv)}\tilde{g}_h$ reflects effects of adversarial training on changing the gradient $\tilde{g}_x$. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x^{(adv)}$ represents the significance of such effects along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T\Delta\tilde{g}_x^{(adv)} = -\eta\tilde{g}_x^T g_W^{(adv)}\tilde{g}_h = e^{\mathcal{A}}\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - \frac{\eta g_z^2(e^{2\mathcal{A}} - e^{\mathcal{A}})}{\bar{H}_z}\|\tilde{g}_h\|^2. \tag{11}$$

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. In this scenario, Theorem 3 and Theorem 4 reveal the following two conclusions.

(C. 4) Adversarial training usually has a potential of decreasing the significance of the current gradient, *i.e.,* pushing $\tilde{g}_x^T\Delta\tilde{g}_x$ and $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{adv})}$ towards negative values. It is because the second term in Eq. (10) and Eq. (11) is non-positive, due to $\bar{H}_z > 0$ in Lemma 1. More crucially, 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 in an exponential manner.

(C. 5) Adversarial training exponentially strengthens the influence of a few unconfident input samples with large values of $\bar{H}_z \in \mathbb{R}$ and large gradient norms $\|\tilde{g}_x\|$. Such mechanisms make the adversarial training more likely to oscillate in directions of a few samples (cf. Theorem 6), which boosts the difficulty of adversarial training, as well.

- *Experimental verification 1 of Theorem 3.* For verification, we conducted experiments to examine whether the theoretical solution $\hat{\phi}$ computed according to the right side of Eq. (10) well fitted the real values of $\phi^* = \tilde{g}_x^T\Delta\tilde{g}_x$ measured in experiments. To this end, we calculated the metric $\kappa = \mathbb{E}_x[\|\phi^* - \hat{\phi}\|]/\mathbb{E}_x[\|\phi^*\|]$ to evaluate the fitness between the theoretical derivation $\hat{\phi}$ and the real effect ϕ^*, where ϕ^* was computed using real measurements of $\tilde{g}_x, \eta, g_W^{(\text{adv})}, g_W$, and $\tilde{g}_h$ on a ReLU network. In this way, we learned three types of ReLU networks on the MNIST dataset via adversarial training, where we followed settings in (Ren et al., 2022) to construct MLPs, CNNs, and ResCNNs, respectively. Please see Appendix K for more details of experimental settings. Table 2 shows that for each ReLU network, the error κ was small, which meant that the derived training effect $\hat{\phi}$ well matched the real effect ϕ^*. Thus, Theorem 3 was verified.

- *Experimental verification 2 of Theorem 3.* Based on Theorem 3, we obtained the conclusion that adversarial training strengthened the influence of input samples with large $\bar{H}_z$ values and large gradient norms $\|\tilde{g}_x\|$. Here, we conducted experiments to verify this conclusion. Specifically, we examined whether input 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 $\|\Delta g_W^{(\text{adv})}\|$, *i.e.,* whether adversarial training boosted the influence of such samples. Note that in real applications, the $\mathcal{A}$ value changed in each step of the adversarial 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 and VGG-11 on the MNIST dataset via adversarial training on PGD, respectively. Please see Appendix L for more details of experimental settings. Fig. 2 shows that input samples with larger values of $\bar{H}_z$, $\bar{H}_z\|\tilde{g}_x\|^2$, and $\hat{\mathcal{A}}$ usually yielded larger $\|\tilde{g}_x^T\Delta\tilde{g}_x\|$ and $\|\Delta g_W^{(\text{adv})}\|$ values, which indicated that adversarial training strengthened the influence of these samples. Thus, the conclusion **C. 5** was verified.

- *Experimental verification 3 of Theorem 3.* We also obtained the conclusion from Theorem 3 that the optimization direction of adversarial training was dominated by a few input samples with large $\mathcal{A} = \beta\bar{H}_z\|\tilde{g}_x\|^2$ values. Here, we conducted experiments to verify this conclusion. 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. Then, 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

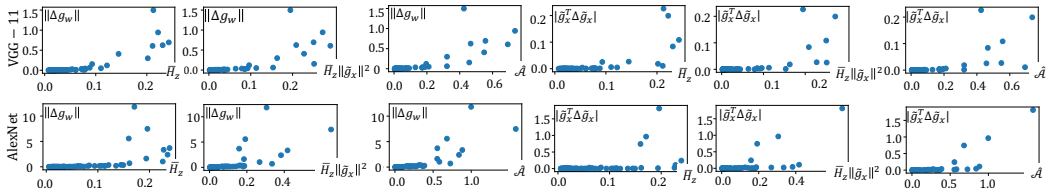

Figure 2: Impacts $\|\Delta g_W\|$ and $|\tilde{g}_x^T \Delta \tilde{g}_x|$ of different input samples on adversarial training. Adversarial training boosted the influence of input samples with large $\bar{H}_z$, $\bar{H}_z \|\tilde{g}_x\|^2$, and $\hat{\mathcal{A}}$ values.

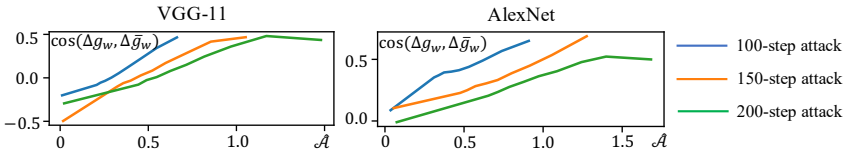

Figure 3: Average cosine similarity $\mathbb{E}_x[\cos(\Delta g_W|_x, \Delta \bar{g}_W)]$ between $\Delta \bar{g}_W$ and each sample x with a specific $\hat{\mathcal{A}}$ value. $\Delta \bar{g}_W$ was similar to the direction of Δg_W *w.r.t.* samples with large $\hat{\mathcal{A}}$ values.

examples. Please see Appendix M and Appendix L for more results and experimental settings, respectively. Fig. 3 shows that the direction of the average effect $\Delta \bar{g}_W$ was similar to (dominated by) training effects of a few input samples with large $\hat{\mathcal{A}}$ values (the real $\mathcal{A}$ calculated in experiments), which verified Theorem 3.

Effects of normalized perturbations. As aforementioned, the ℓ_2 attack and the ℓ_∞ attack can be roughly considered as the regularization/normalization of adversarial perturbations. In this way, we analyze the effects of the normalized perturbation $\hat{\delta}^{(\text{norm})}$ on adversarial training, which approximately explains adversarial training based on perturbations of the ℓ_2 attack and the ℓ_∞ attack.

Theorem 5 (proven in Appendix H)**.** *Based on Assumptions 1 and 2, 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 normalized perturbations $\hat{\delta}^{(norm)}$ in Remark 1. Considering the footnote[3], $\Delta \tilde{g}_x^{(norm)} = -\eta \Delta g_W^{(norm)} \tilde{g}_h = -\eta (g_W^{(adv,\ norm)} - g_W) \tilde{g}_h$ represents additional effects of adversarial training on changing $\tilde{g}_x$. In this way, $\tilde{g}_x^T \Delta \tilde{g}_x^{(norm)} = -\eta \tilde{g}_x^T \Delta g_W^{(norm)} \tilde{g}_h$ reflects the significance of such additional effects along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T \Delta \tilde{g}_x^{(norm)} = C \cdot \Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|}\Big) \tilde{g}_x^T \Delta \tilde{g}_x^{(ori)} - C \cdot \frac{\eta g_z^2 \, \|\tilde{g}_h\|^2}{\bar{H}_z} \Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|} + C \cdot \big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|}\big)^2 \Big). \quad (12)$$

It is because Theorem 2 shows that an extremely weak adversarial strength $\beta \to 0$ usually yields $\|\hat{\delta}\| \to \|g_x\|$, and a relatively strong adversarial strength β usually makes $\|\hat{\delta}\| \to \exp(\beta \|\tilde{g}_x\|^2 g_z^2)/\|g_x\|$ with an exponential strength. In this way, given a relatively strong attack, we can ignore the term $1/\|\hat{\delta}\| \to 0$ in Eq. (12), and prove that the strength of the training effect $\tilde{g}_x^T \Delta \tilde{g}_x^{(\text{norm})}$ is mainly determined by the term $\exp(\mathcal{A})/\|\hat{\delta}\| \approx \|g_x\| \cdot \exp(\beta \|\tilde{g}_x\|^2 (\bar{H}_z - g_z^2))$. Please see Appendix H.2 for the proof. Besides, according to Lemma 1, as long as the attack has not succeeded yet, we have $\bar{H}_z - g_z^2 > 0$, but for too confident samples $z(x) \to \infty$ or too unconfident samples $z(x) = 0$, we get $\bar{H}_z - g_z^2 = 0$. Hence, we obtain the following two conclusions.

(C. 6) Adversarial training on the normalized perturbations strengthens the influence of a few input samples with large gradient norms $\|\tilde{g}_x\|$, which are neither too confident nor too unconfident.

(C. 7) Compared to Theorems 3 and 4, the normalized perturbation $\hat{\delta}^{(\text{norm})}$ in Eq. (7) alleviates the imbalance between different samples, which proves the benefits of ℓ_2 attacks and ℓ_∞ attacks.

Oscillation of network parameters. Above proofs can explain that adversarial training makes network parameters oscillate in very few directions, which is considered as a common phenomenon in adversarial training. Such an explanation is based on a typical claim in optimization (Cohen et al., 2021; Wu et al., 2018) that if the largest eigenvalue of the Hessian matrix of the loss *w.r.t* network parameters is large enough, 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 of the loss *w.r.t* network parameters, Theorems 1 and 2 show that training on adversarial examples is somewhat equivalent to boosting the influence of the Hessian matrix.

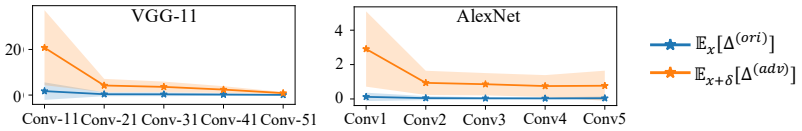

Figure 4: Influence of weight changes on gradients of the loss function *w.r.t.* network parameters (weights). The weight change in adversarial training made more significant impacts $\Delta^{(\text{adv})}$ on gradients than that in vanilla training $\Delta^{(\text{ori})}$, which verified Theorem 6.

Specifically, given a ReLU network f and an adversarial example $x + \hat{\delta}$ for adversarial training, let us temporarily consider the Hessian matrix $H_h \overset{\text{def}}{=} \frac{\partial^2}{\partial h \partial h^T} L(f(x), y)$ *w.r.t* the output h of the j-th linear layer. Then, the loss function on adversarial examples $L(f(x + \hat{\delta}), y)$ can be represented as follows.

Theorem 6. *Let $\Delta h = W^T \hat{\delta} \in \mathbb{R}^{D\times 1}$ denote the change of the intermediate-layer feature h caused by the perturbation $\hat{\delta}$, and Loss$(h + \Delta h) = L(f(x + \hat{\delta}), y)$ represents the loss function on the adversarial example $x + \hat{\delta}$. Then, we use the second-order Taylor expansion to decompose the loss, i.e., Loss$(h + \Delta h)$ = Loss$(h) + g_h^T \Delta h + \frac{1}{2!}\Delta h^T H_h \Delta h + R_2(\Delta h)$ = 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 $g_h = \partial L(f(x), y)/\partial h$ represents the gradient of the loss function $L(f(x), y)$ w.r.t. the intermediate-layer feature h, and $R_2(\Delta h)$ indicates terms higher than the second order. In this way, if we focus on the i-th dimension of $\hat{\delta}$, $\hat{\delta}_i \in \mathbb{R}$, the loss can be re-written as follows, where w_i denotes a row vector corresponding to the i-th row of the weight matrix W, and τ is a constant w.r.t the change of w_i.*

$$\textit{Loss}(h + \Delta h) = \tau + [\hat{\delta}_i \, g_{h,i}^T] w_i^T + w_i [\frac{1}{2!} \hat{\delta}_i^2 H_h] w_i^T. \tag{13}$$

(C. 8) Theorem 6 reveals that adversarial training can be roughly considered to boost the influence of the Hessian matrix *w.r.t.* network parameters w_i, *i.e.,* proportional to $\hat{\delta}_i^2 H_h$, because the perturbation $\hat{\delta}$ increases exponentially along with the overall adversarial strength $\beta = \alpha m$, according to Theorems 1 and 2. Adversarial training is more likely to make network parameters oscillate than vanilla training.

• *Experimental verification of Theorem 6.* Theorem 6 shows that adversarial training boosted the influence of Hessian matrix *w.r.t.* the network parameters. Here, we conducted experiments to verify this conclusion. Specifically, we learned AlexNet and VGG-11 on the MNIST dataset, and measured 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. Thus, vanilla training's influence and adversarial training's influence of such weight changes on the gradient could be estimated as $\Delta^{(\text{ori})} = \frac{1}{D\|\Delta W_j\|} \cdot \|(\partial L(f(x|W_j + \Delta W_j), y)/\partial W_j) - (\partial L(f(x|W_j), y)/\partial W_j)\|$, and $\Delta^{(\text{adv})} = \frac{1}{D\|\Delta W_j^{(\text{adv})}\|} \cdot \|(\partial L(f(x + \delta|W_j + \Delta W_j^{(\text{adv})}), y)/\partial W_j) - (\partial L(f(x + \delta|W_j), y)/\partial W_j)\|$, respectively. 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 of experimental settings. Fig. 4 compares the influence of weight changes on gradients *w.r.t.* network parameters. We discovered that compared to vanilla training, the weight change with a fixed strength in adversarial training usually affected the gradient much more significantly. Such a phenomenon demonstrated that adversarial training boosted the influence of Hessian matrix *w.r.t.* the network parameters, which verified Theorem 6.

3 Related work: A unified analysis of previous findings in adversarial training

In this section, we use our theorems to theoretically explain or provide a new perspective to understand previous findings in adversarial training. In fact, some studies are not directly related to our theorems, and we put the discussions on them in Appendix I.

• Many previous studies (Liu et al., 2020; Kanai et al., 2021; Wu et al., 2020; Yamada et al., 2021) considered that the difficulty of adversarial training was caused by the sharp loss landscape *w.r.t* network parameters. To this end, Theorem 6 verifies such an explanation. Specifically, we have

proven that adversarial training can be considered to strengthen the influence of the Hessian matrix of the loss *w.r.t.* network parameters, which is equivalent to sharpening the loss landscape.

• Athalye et al. (2018) discovered that obfuscated gradients led to a false sense of security in defenses against adversarial examples, which hindered adversarial training (Zhang & Wang, 2019a). To this end, Theorem 3 and Theorem 4 explain the third type of obfuscated gradients in (Athalye et al., 2018), *i.e.,* vanishing gradients. Specifically, we have proven that adversarial training significantly strengthens the influence of a few unconfident samples, and neglects the influence of many confident samples, which makes the training process more likely to oscillate in directions of a few unconfident samples. Such oscillation along optimization directions of a few hard samples usually significantly increases norms of weights along such directions, and causes over-confident predictions on some easy samples. These over-confident predictions on easy samples may lead to vanishing gradients.

• Tsipras et al. (2019) clarified that compared to vanilla training, adversarial training mainly relied on robust features and did not use non-robust features for inference, which caused inferior classification performance. To this end, Theorems 3 and 4 verify such a finding. Specifically, we have proven that adversarial training is mainly dominated by a few samples, which easily makes network parameters oscillate in very few directions. In other words, the training of non-robust features, or more precisely, training on samples with significant $\hat{A}$ values that are easily attacked, is hard to converge.

• Ilyas et al. (2019) demonstrated that adversarial examples were attributed to the presence of highly predictive but non-robust features. To this end, Theorems 1 and 2 verify such a finding, which reveals that in the multi-category classification, the direction of the largest eigenvalue of the Hessian matrix H_x suppresses features related to the target category, and promotes features related to the second-best category. Here, the eigenvector *w.r.t.* the largest eigenvalue corresponds to non-robust features.

• Liu et al. (2021) considered that the robust overfitting was caused by the fitting of hard samples, under the assumption that all training samples followed a Gaussian mixture distribution in a logistic regression problem. To this end, Theorem 3 and Theorem 4 explain such a finding in a more generic classification task without assuming the data distribution. Specifically, we have proven that compared to vanilla training, the adversarially trained network is more likely to be over-fitted to a few unconfident samples, which correspond to hard samples in adversarial training.

• Chen et al. (2020) discovered that the overfitting in adversarial training was because the network overfitted to adversarial examples generated in the early stage of adversarial training, and failed to generalize to adversarial examples generated in the late stage. To this end, we provide a deeper insight into such a phenomenon. Specifically, according to Theorem 3 and Theorem 4, only a few unconfident samples with large gradient norms $\|\tilde{g}_x\|$ influence the adversarial training. In fact, the imbalance of the sample influence can easily make unconfident samples with large $\bar{H}_z$ values and large gradient norms $\|\tilde{g}_x\|$ in the early stage of adversarial training significantly different from those in the late stage. Such mechanisms lead to the overfitting in adversarial training.

• Rice et al. (2020) demonstrated that early stopping could effectively reduce overfitting in adversarial training. To this end, Theorem 3 and Theorem 4 also explain the effectiveness of the early stopping. Specifically, during adversarial training process, the network becomes robust, and the number of unconfident samples decreases. Because adversarially trained networks mainly focus on unconfident samples, the decreasing number of unconfident samples boosts the significance of overfitting. In this way, early stopping can effectively reduce overfitting.

4 Conclusion and discussion

This paper theoretically analyzes the dynamics of adversarial perturbations via an analytic solution. We also prove that adversarial training strengthens the influence of a few input samples, which boosts the difficulty of adversarial training. Crucially, our proofs provide a theoretical explanation for previous studies in understanding adversarial training. Note that our analysis is all based on the assumption that adversarial perturbations cannot significantly change the gating states of the ReLU network. Despite this, experimental results show that our analysis can still explain most adversarial perturbations generated in real cases, when gating states change. Besides, in this paper, we use the normalized perturbations to approximate adversarial perturbations of the ℓ_2 attack and the ℓ_∞ attack, instead of deriving an exact formulation for these perturbations. Nevertheless, experimental results show that our analysis can well explain the ℓ_2 attack and the ℓ_∞ attack, to some extent.

ETHIC STATEMENT

This paper theoretically analyzes the dynamics of adversarial perturbations, and further theoretically explains the difficulty of adversarial training. There are no ethic issues with this paper.

REPRODUCIBILITY STATEMENT

Proofs for all theorems and assumptions are provided in Appendix A, B, D, E, F, G, and H. Details of experimental settings are provided in Appendix C, J, L, and K. Besides, we will release the code when the paper is accepted.

REFERENCES

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

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Sekitoshi Kanai, Masanori Yamada, Hiroshi Takahashi, Yuki Yamanaka, and Yasutoshi Ida. Smoothness analysis of adversarial training. *arXiv preprint arXiv:2103.01400*, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, 33:21476–21487, 2020.

Chen Liu, Zhichao Huang, Mathieu Salzmann, Tong Zhang, and Sabine Süsstrunk. On the impact of hard adversarial instances on overfitting in adversarial training. *arXiv preprint arXiv:2112.07324*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Zhuang QIAN, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, Bin Gu, Huan Xiong, and Xinping Yi. Perturbation diversity certificates robust generalisation, 2022. URL https://openreview.net/forum?id=jm1RxJFQdDN.

Jie Ren, Mingjie Li, Meng Zhou, Shih-Han Chan, and Quanshi Zhang. Towards theoretical analysis of transformation complexity of relu dnns. *arXiv preprint arXiv:2205.01940*, 2022.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

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

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, pp. 9155–9166. PMLR, 2020.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

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

Masanori Yamada, Sekitoshi Kanai, Tomoharu Iwata, Tomokatsu Takahashi, Yuki Yamanaka, Hiroshi Takahashi, and Atsutoshi Kumagai. Adversarial training makes weight loss landscape sharper in logistic regression. *arXiv preprint arXiv:2102.02950*, 2021.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*, 2018.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/d8700cbd38cc9f30cecb34f0c195b137-Paper.pdf.

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

A PROOF OF THEOREM 1

In this section, we prove Theorem 1 in Section 2.1 of the main paper, which analyzes the dynamics of perturbations of the m-step attack.

Let us focus on the most straightforward solution to the multi-step adversarial attack. In this scenario, given a ReLU network f and an input sample $x \in \mathbb{R}^n$, the perturbation generated after attacking for m steps is formulated as follows.

$$\delta^{(m)} = \sum\nolimits_{t=0}^{m-1} \alpha \cdot g_{x+\delta^{(t)}}, \tag{14}$$

where $g_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$ represents the gradient of the loss *w.r.t.* the input sample x, and m denotes the step size. Furthermore, we define the update of the perturbation at each step t as follows.

$$\Delta x^{(t)} \stackrel{\text{def}}{=} \alpha \cdot g_{x+\delta^{(t-1)}}, \tag{15}$$

In this way, the perturbation $\delta^{(m)}$ generated after the m-step attack can be re-written as

$$\delta^{(m)} = \Delta x^{(1)} + \Delta x^{(2)} + \cdots + \Delta x^{(m)}. \tag{16}$$

Then, in order to derive the analytic solution to the adversarial perturbation $\delta^{(m)}$ in Eq. (14), we use the quadratic Taylor approximation (LeCun et al., 2012; Cohen et al., 2021) to re-write the loss function as follows.

$$\begin{aligned} L(f(x+\delta^{(t)}), y) &= L(f(x+\delta^{(t-1)} + \Delta x^{(t)}), y) \\ &= L(f(x+\delta^{(t-1)}), y) + (\Delta x^{(t)})^T g_{x+\delta^{(t-1)}} + \frac{1}{2}(\Delta x^{(t)})^T H_x^{(t-1)} (\Delta x^{(t)}) + R_2(\Delta x^{(t)}), \end{aligned} \tag{17}$$

where $g_{x+\delta^{(t-1)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t-1)}), y)$ represents the gradient of the loss function *w.r.t.* the adversarial example $x+\delta^{(t-1)}$. $H_x^{(t-1)} = \frac{\partial^2}{\partial x \partial x^T} L(f(x+\delta^{(t-1)}), y)$ represents the Hessian matrix of the loss function *w.r.t.* the adversarial example $x+\delta^{(t-1)}$. $R_2(\Delta x^{(t)})$ is referred to as terms of higher than the second order in the Taylor series *w.r.t* the perturbation $\Delta x^{(t)}$.

Note that the order of $\Delta x^{(t)}$ is $O(1/m)$. Hence, if the step number m is large enough, the perturbation $\Delta x^{(t)}$ is ignorable. Moreover, each dimension $R_2^i(\Delta x^{(t)})$ of the residual term $R_2(\Delta x^{(t)}) \in \mathbb{R}^n$ in Eq. (17) is proven to be the order of $O(1/m^2)$. Hence, the residual term $R_2(\Delta x^{(t)})$ in Eq. (17) is also ignorable without hurting the subsequent proofs, if the step number m is large enough. Please see Section A.2 for the detailed analysis.

In this way, based on Eq. (17), the gradient of the loss function *w.r.t* the adversarial example $x+\delta^{(t)}$ can be represented as

$$\begin{aligned} g_{x+\delta^{(t)}} &= \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y) \\ &= \frac{\partial}{\partial x} L(f(x+\delta^{(t-1)} + \Delta x^{(t)}), y) \\ &= \frac{\partial}{\partial x} \left[L(f(x+\delta^{(t-1)}), y) + (\Delta x^{(t)})^T g_{x+\delta^{(t-1)}} + \frac{1}{2}(\Delta x^{(t)})^T H_x^{(t-1)} (\Delta x^{(t)}) + R_2(\Delta x^{(t)}) \right] \\ &\qquad\qquad // \text{ According to Eq. (17)} \\ &= g_{x+\delta^{(t-1)}} + H_x^{(t-1)} \Delta x^{(t)} + \frac{\partial}{\partial x} R_2(\Delta x^{(t)}). \end{aligned} \tag{18}$$

Lemma 2 (in Appendix)**.** *Based on Assumption 1, the update of the perturbation with the multi-step attack at step t can be represented as $\Delta x^{(t)} = \alpha(I + \alpha \bar{H}_x)^{t-1} g_x + \psi^{(t)}$, where $g_x = \frac{\partial}{\partial x} L(f(x), y)$. $\bar{H}_x = \tilde{W} \bar{H}_z (\tilde{W})^T$ is used to approximate[4] the second derivative of the loss w.r.t. the input sample*

[4]Theoretically, it is very hard to derive the analytic solution to the adversarial perturbation $\delta^{(m)}$ without such an approximation. Hence, we use the matrix $\bar{H}_z$ to approximate the equivalent Hessian matrix, which allows us to derive the first analytic solution to the adversarial perturbation of the multi-step attack. More crucially, experimental results in Table 1 verified the trustworthiness of such an approximation, *i.e.,* the error between the real perturbation and the theoretically derived solution is at the level of 10^{-8}—10^{-5}.

x, where $\tilde{W}^T = W_l^T \Sigma_{l-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x+\delta^{(t)}), y)$. $\psi^{(t)} \in \mathbb{R}^n$ denotes an ignorable residual term, because each dimension $\psi_i^{(t)}$ is the order of $O(1/m^2)$, if the step number m is sufficiently large.

Proof. If the step $t = 1$, according to Eq. (15), we have $\Delta x^{(1)} = \alpha g_x$.

For $\forall t > 1$, the perturbation of the t-th step attack is defined as $\Delta x^{(t)} = \alpha \cdot g_{x+\delta^{(t-1)}}$ in Eq. (15). Based on Eq. (18), the perturbation $\Delta x^{(t)}$ can be re-written as

$$
\begin{aligned}
g_{x+\delta^{(t-1)}} &= g_{x+\delta^{(t-2)}} + H_x^{(t-2)} \Delta x^{(t-1)} + \frac{\partial}{\partial x} R_2(\Delta x^{(t-1)}) \quad \text{// According to Eq. (18)} \\
&= g_{x+\delta^{(t-3)}} + H_x^{(t-3)} \Delta x^{(t-2)} + \frac{\partial}{\partial x} R_2(\Delta x^{(t-2)}) + H_x^{(t-2)} \Delta x^{(t-1)} + \frac{\partial}{\partial x} R_2(\Delta x^{(t-1)}) \\
&\cdots \\
&= g_x + \sum\nolimits_{t'=1}^{t-1} H_x^{(t'-1)} \Delta x^{(t')} + \sum_{t'=1}^{t-1} \frac{\partial}{\partial x} R_2(\Delta x^{(t')}) \\
&= g_x + \sum\nolimits_{t'=1}^{t-1} \tilde{W} H_z^{(t'-1)} (\tilde{W})^T \Delta x^{(t')} + \sum\nolimits_{t'=1}^{t-1} \frac{\partial}{\partial x} R_2(\Delta x^{(t')}) \quad \text{// According to Eq. (20)} \\
&\approx g_x + \sum\nolimits_{t'=1}^{t-1} \tilde{W} \bar{H}_z (\tilde{W})^T \Delta x^{(t')} + \sum\nolimits_{t'=1}^{t-1} \frac{\partial}{\partial x} R_2(\Delta x^{(t')}) \\
&= g_x + \sum\nolimits_{t'=1}^{t-1} \bar{H}_x \Delta x^{(t')} + \sum\nolimits_{t'=1}^{t-1} \tilde{R}_2(\Delta x^{(t')}).
\end{aligned}
\tag{19}
$$

Here, we use $\bar{H}_x = \tilde{W} \bar{H}_z (\tilde{W})^T$ to approximate[5] the second derivative of the loss *w.r.t.* the input sample x, where $\tilde{W}^T = W_l^T \Sigma_{l-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$ based on Assumption 1. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x + \delta^{(t)}), y)$. For simplicity, let $\tilde{R}_2(\Delta x^{(t')}) = \frac{\partial}{\partial x} R_2(\Delta x^{(t')})$.

Based on Assumption 1, we have

$$
\begin{aligned}
H_x^{(t)} &= \frac{\partial^2 L(f(x+\delta^{(t)}), y)}{\partial x \partial x^T} = \frac{\partial (\frac{\partial L(f(x+\delta^{(t)}), y)}{\partial z(x)} \frac{\partial z(x)}{\partial x^T})^T}{\partial x^T} \\
&= \frac{(\frac{\partial z(x)}{\partial x^T})^T}{\partial x^T} \cdot \frac{\partial L(f(x+\delta^{(t)}), y)}{\partial z(x)} + (\frac{\partial z(x)}{\partial x^T})^T \cdot \frac{\partial (\frac{\partial L(f(x+\delta^{(t)}), y)}{\partial z(x)})}{\partial x^T} \\
&= (\frac{\partial z(x)}{\partial x^T})^T \frac{\partial^2 L(f(x+\delta^{(t)}), y)}{\partial z(x) \partial z(x)} \frac{\partial z(x)}{\partial x^T} \\
&= \tilde{W} H_z^{(t)} (\tilde{W})^T. \quad \text{// According to Assumption 1 in the main paper.}
\end{aligned}
\tag{20}
$$

Note that the order of $\Delta x^{(t)}$ is $O(1/m)$. If the step number m is large enough, the perturbation $\Delta x^{(t)}$ is ignorable. Moreover, each dimension $\sum_{t'=1}^{t-1} R_2^i(\Delta x^{(t')}) \in \mathbb{R}$ of the residual term $\sum_{t'=1}^{t-1} R_2(\Delta x^{(t')}) \in \mathbb{R}^n$ in Eq. (19) is proven to be the order of $O(1/m)$. Hence, such a residual term $\sum_{t'=1}^{t-1} \tilde{R}_2(\Delta x^{(t')})$ is small enough to be ignored without hurting the trustworthiness of further analysis, if the step number m is large. Please see Section A.2 for the detailed analysis.

Substituting Eq. (19) back to Eq. (15), the perturbation $\Delta x^{(t)}$ can be re-written as

$$
\begin{aligned}
\Delta x^{(t)} &= \alpha \cdot g_{x+\delta^{(t-1)}} \\
&\approx \alpha \cdot g_x + \alpha \cdot \sum_{t'=1}^{t-1} \bar{H}_x \Delta x^{(t')} + \alpha \cdot \sum_{t'=1}^{t-1} \tilde{R}_2(\Delta x^{(t')}).
\end{aligned}
\tag{21}
$$

[5]Theoretically, it is very hard to derive the analytic solution to the adversarial perturbation $\delta^{(m)}$ without such an approximation. Hence, we use the matrix $\bar{H}_z$ to approximate the equivalent Hessian matrix, which allows us to derive the first analytic solution to the adversarial perturbation of the multi-step attack. More crucially, experimental results in Table 1 verified the trustworthiness of such an approximation, *i.e.,* the error between the real perturbation and the theoretically derived solution is at the level of 10^{-8}—10^{-5}.

In this way, we apply the mathematical induction to prove Lemma 2 in Appendix, *i.e.,* $\forall 1 \leq t \leq m, \Delta x^{(t)} = \alpha(I + \alpha\bar{H}_x)^{t-1} g_x + \psi^{(t)}$, where $\psi^{(t)} = \alpha \sum_{t'=1}^{t-1} (I + \alpha\bar{H}_x)^{t-1-t'} \tilde{R}_2(\Delta x^{(t')})$.

Base case: When $t = 1$, we have $\Delta x^{(1)} = \alpha \cdot g_x = \alpha \cdot (I + \alpha\bar{H}_x)^0 g_x$.

Inductive step:
For $t > 1$, assuming $\Delta x^{(t-1)} = \alpha(I + \alpha\bar{H}_x)^{t-2} g_x + \alpha \sum_{t'=1}^{t-2} (I + \alpha\bar{H}_x)^{t-2-t'} \tilde{R}_2(\Delta x^{(t')})$, we have

$$
\begin{aligned}
\Delta x^{(t)} &= \alpha \cdot \left(g_x + \bar{H}_x \sum\nolimits_{t'=1}^{t-1} \Delta x^{(t')} + \sum\nolimits_{t'=1}^{t-1} \tilde{R}_2(\Delta x^{(t')}) \right) \quad // \quad \text{According to Eq. (21)} \\
&= \alpha \cdot \left[g_x + \bar{H}_x \sum\nolimits_{t'=1}^{t-2} \Delta x^{(t')} + \sum\nolimits_{t'=1}^{t-2} \tilde{R}_2(\Delta x^{(t')}) \right] + \alpha \cdot \bar{H}_x \Delta x^{(t-1)} + \alpha \cdot \tilde{R}_2(\Delta x^{(t-1)}) \\
&= \Delta x^{(t-1)} + \alpha \cdot \bar{H}_x \Delta x^{(t-1)} + \alpha \cdot \tilde{R}_2(\Delta x^{(t-1)}) \quad // \quad \text{According to Eq. (21)} \\
&= (I + \alpha \cdot \bar{H}_x) \Delta x^{(t-1)} + \alpha \cdot \tilde{R}_2(\Delta x^{(t-1)}) \\
&= (I + \alpha \cdot \bar{H}_x)\, \alpha \cdot \left[(I + \alpha\bar{H}_x)^{(t-2)} g_x + \sum_{t'=1}^{t-2} (I + \alpha\bar{H}_x)^{t-2-t'} \tilde{R}_2(\Delta x^{(t')}) \right] + \alpha \cdot \tilde{R}_2(\Delta x^{(t-1)}) \\
&= \alpha \cdot (I + \alpha\bar{H}_x)^{t-1} g_x + \alpha \sum_{t'=1}^{t-1} (I + \alpha\bar{H}_x)^{t-1-t'} \tilde{R}_2(\Delta x^{(t')}) \\
&= \alpha \cdot (I + \alpha\bar{H}_x)^{t-1} g_x + \psi^{(t)},
\end{aligned}
\tag{22}
$$

where $\tilde{R}_2(\Delta x^{(t-1)}) = \frac{\partial}{\partial x} R_2(\Delta x^{(t-1)})$, and $R_2(\Delta x^{(t-1)})$ is referred to as the term of the perturbation $\Delta x^{(t-1)}$ higher than the second order.

Conclusion: Since both the base case and the inductive step have been proven to be true, we have $\Delta x^{(t)} = \alpha(I + \alpha\bar{H}_x)^{t-1} g_x + \psi^{(t)}$, where $\psi^{(t)} = \alpha \sum_{t'=1}^{t-1} (I + \alpha\bar{H}_x)^{t-1-t'} \tilde{R}_2(\Delta x^{(t')})$.

Here, $\psi^{(t)} \in \mathbb{R}^n$ denotes an ignorable residual term, because each dimension $\psi_i^{(t)}$ is the order of $O(1/m^2)$, if the step number m is sufficiently large. Please see Section A.2 for the detailed analysis.

Thus, Lemma 2 in Appendix is proven. □

A.1 Proof of Theorem 1

Theorem 1 (Dynamics of perturbations of the m-step attack). *Let us assume that the gradient $g_{x+\delta^{(t)}}$ is a Lipschitz function with the Lipschitz constant K, $\|g_{x+\delta^{(t)}} - g_x\| \leq K \cdot \|\delta^{(t)}\|$. Then, based on Assumption 1, the adversarial perturbation $\delta^{(m)}$ can be approximated as follows, where the overall adversarial strength $\beta = \alpha m$ is a small constant, and the step number m is a large integer.*

$$
\delta^{(m)} = \sum_{i=1}^{n} \frac{(1 + \alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \boldsymbol{\rho}, \quad g_{x+\delta^{(m)}} = \sum_{i=1}^{n} (1 + \alpha\lambda_i)^m \gamma_i v_i. \tag{23}
$$

Here, λ_i and v_i denote the i-th largest eigenvalue of the matrix $\bar{H}_x = \tilde{W}\bar{H}_z(\tilde{W})^T$ and its corresponding eigenvector, respectively, where $\bar{H}_x$ is used to approximate[6] the second derivative of the loss w.r.t. the input sample x. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ 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 $\Delta x^{(t)} = \alpha \cdot g_{x+\delta^{(t-1)}}$ denotes the perturbation updated at the t-th step. $\gamma_i = g_x^T v_i \in \mathbb{R}$ represents the projection of the gradient $g_x = \frac{\partial}{\partial x} L(f(x), y)$ on the eigenvector v_i. If the step number m is large, then the residual term in the Taylor expansion $\boldsymbol{\rho} \in \mathbb{R}^n$ is ignorable, since each element $\boldsymbol{\rho}_i \in \mathbb{R}$ is proven to be the order of $O(1/m)$.

[6]Theoretically, it is very hard to derive the analytic solution to the adversarial perturbation $\delta^{(m)}$ without such an approximation. Hence, we use the matrix $\bar{H}_z$ to approximate the equivalent Hessian matrix, which allows us to derive the first analytic solution to the adversarial perturbation of the multi-step attack. More crucially, experimental results in Table 1 verified the trustworthiness of such an approximation, *i.e.,* the error between the real perturbation and the theoretically derived solution is at the level of 10^{-8}—10^{-5}.

Proof. According to Eq. (16) and Lemma 2 in Appendix, the perturbation $\delta^{(m)}$ generated after the m-step attack can be re-written as

$$\begin{aligned}\delta^{(m)} &= \Delta x^{(1)} + \Delta x^{(2)} + \cdots + \Delta x^{(m)} \\ &= \alpha[I + (I + \alpha\bar{H}_x) + \cdots + (I + \alpha\bar{H}_x)^{m-1}]g_x + \sum_{t=1}^{m}\psi^{(t)}.\end{aligned} \tag{24}$$

Because each Hessian matrix $H_x^{(t)}$ in matrix $\bar{H}_x = \frac{1}{\sum_{t=1}^{m-1}\|\Delta x^{(t)}\|}\sum_{t=1}^{m-1}\|\Delta x^{(t)}\|H_x^{(t)}$ is a real-valued symmetric matrix, this matrix $\bar{H}_x$ is also a real-valued symmetric matrix. In this way, we can use the eigenvalue decomposition to decompose $\bar{H}_x$ as $\bar{H}_x = V\Lambda V^{-1}$. Here, $\Lambda = diag[\lambda_1, \lambda_2, \cdots, \lambda_p]$ is a diagonal matrix, whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$. The square matrix $V = [v_1, v_2, \cdots, v_n] \in \mathbb{R}^{n\times n}$ contains n linearly independent eigenvectors v_i, *i.e.,* $\forall i \neq k, v_i^T v_k = 0$, where v_i is the eigenvector corresponding to the eigenvalue λ_i. Without loss generality, we normalize these n eigenvectors v_i, thereby $V^T V = I$. In this scenario, the Hessian matrix $\bar{H}_x$ can be decomposed as $\bar{H}_x = V\Lambda V^T$, and the perturbation $\delta^{(m)}$ can be represented as

$$\begin{aligned}\delta^{(m)} &= \alpha[I + (I + \alpha V\Lambda V^T) + \cdots + (I + \alpha V\Lambda V^T)^{m-1}]g_x + \sum_{t=1}^{m}\psi^{(t)} \\ &= \alpha[VV^T + (VV^T + \alpha V\Lambda V^T) + \cdots + (VV^T + \alpha V\Lambda V^T)^{m-1}]g_x + \sum_{t=1}^{m}\psi^{(t)} \\ &= \alpha[VIV^T + V(I + \alpha\Lambda)V^T + \cdots + [V(I + \alpha\Lambda)V^T]^{m-1}]g_x + \sum_{t=1}^{m}\psi^{(t)} \\ &= \alpha[VIV^T + V(I + \alpha\Lambda)V^T + \cdots + V(I + \alpha\Lambda)^{m-1}V^T]g_x + \sum_{t=1}^{m}\psi^{(t)} \\ &= \alpha V[I + (I + \alpha\Lambda) + \cdots + (I + \alpha\Lambda)^{m-1}]V^T g_x + \sum_{t=1}^{m}\psi^{(t)}. \\ &= \alpha VDV^T g_x + \sum_{t=1}^{m}\psi^{(t)}.\end{aligned} \tag{25}$$

For simplicity, let $D = \alpha(I + (I + \alpha\Lambda) + \cdots + (I + \alpha\Lambda)^{m-1})$, which is a diagonal matrix, since I, $I + \alpha\Lambda$, ..., $(I + \alpha\Lambda)^{m-1}$ are all diagonal matrices. In this way, let us focus on the k-th diagonal element $D_{kk} \in \mathbb{R}$.

$$\begin{aligned}D_{kk} &= \alpha(1 + (1 + \alpha\lambda_k) + \cdots + (1 + \alpha\lambda_k)^{m-1}) \\ &= \alpha(1 \times \frac{1 - (1 + \alpha\lambda_k)^m}{1 - (1 + \alpha\lambda_k)}) \\ &= \frac{(1 + \alpha\lambda_k)^m - 1}{\lambda_k}.\end{aligned} \tag{26}$$

Then, combining Eq. (25) and Eq. (26), the perturbation $\delta^{(m)}$ can be written as follows. Here, considering that n eigenvectors of the Hessian matrix form a set of unit orthogonal basis, the gradient

g_x can be represented as $g_x = \sum_{i=1}^{n} \gamma_i v_i$, where γ_i is referred to as the projection length of g_x on v_i.

$$\begin{aligned}
\delta^{(m)} &= VDV^T g_x + \sum_{t=1}^{m} \psi^{(t)} \\
&= VDV^T (\sum_{i=1}^{n} \gamma_i v_i) + \sum_{t=1}^{m} \psi^{(t)} \\
&= \sum_{i=1}^{n} D_{ii} v_i v_i^T \sum_{k=1}^{n} \gamma_k v_k + \sum_{t=1}^{m} \psi^{(t)} \\
&= \sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \alpha \sum_{t=1}^{m} \sum_{t'=1}^{t-1} (I + \alpha \bar{H}_x)^{t-1-t'} \tilde{R}_2(\Delta x^{(t')}) \\
&= \sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \boldsymbol{\rho},
\end{aligned} \tag{27}$$

where we use $\boldsymbol{\rho} \in \mathbb{R}^n$ to denote the residual term $\sum_{t=1}^{m} \psi^{(t)}$. **In Lemma 3 in Appendix, we have proven that each dimension $\boldsymbol{\rho}_i \in \mathbb{R}$ of the residual term $\boldsymbol{\rho}$ is the order of $O(1/m)$. Thus, this residual term $\boldsymbol{\rho}$ can be ignored, without hurting the trustworthiness of the analysis of the adversarial perturbation, if the step number m is large.**

Based on Eq. (19), the gradient $g_{x+\delta^{(m)}}$ of the loss *w.r.t.* the adversarial example $x + \delta^{(m)}$ can be re-written as follows. Here, $R_2(\Delta x^{(m)})$ denotes terms of the perturbation $\Delta x^{(m)}$ higher than the second order.

$$\begin{aligned}
g_{x+\delta^{(m)}} &= g_x + \sum_{t=1}^{m} \bar{H}_x \Delta x^{(t)} + \sum_{t=1}^{m} \frac{\partial}{\partial x} R_2(\Delta x^{(t)}) \\
&= g_x + \bar{H}_x \delta^{(m)} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)}).
\end{aligned} \tag{28}$$

Substituting Eq. (27) back to Eq. (28), the gradient $g_{x+\delta^{(m)}}$ can be written as

$$\begin{aligned}
g_{x+\delta^{(m)}} &= g_x + \bar{H}_x \left(\sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \boldsymbol{\rho} \right) + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)}) \\
&= \sum_{i=1}^{n} \gamma_i v_i + \bar{H}_x \sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i v_i + \bar{H}_x \boldsymbol{\rho} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)}) \\
&= \sum_{i=1}^{n} \gamma_i v_i + \sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i (\bar{H}_x v_i) + \bar{H}_x \boldsymbol{\rho} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)}) \\
&= \sum_{i=1}^{n} \gamma_i v_i + \sum_{i} \frac{(1+\alpha\lambda_i)^m - 1}{\lambda_i} \gamma_i (\lambda_i v_i) + \bar{H}_x \boldsymbol{\rho} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)}). \\
&\approx \sum_{i} (1+\alpha\lambda_i)^m \gamma_i v_i.
\end{aligned} \tag{29}$$

We have proven in Section A.2 that each dimension of the residual term $\bar{H}_x \boldsymbol{\rho} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)})$ in Eq. (29) is the order of $O(1/m)$. Then, when the step number m is large enough, the residual term $\bar{H}_x \boldsymbol{\rho} + \sum_{t=1}^{m} \tilde{R}_2(\Delta x^{(t)})$ is small enough to be ignored, without hurting the trustworthiness of the analysis of the gradient. Please see Section A.2 for the detailed analysis.

Hence, Theorem 1 is proven. □

A.2 Reason for ignoring the residual term in Theorem 1

In this subsection, we clarify the reason why the residual term for the perturbation $\delta^{(m)}$ in Theorem 1 and the residual term for the gradient $g_{x+\delta^{(m)}}$ in Theorem 1 can be ignored.

Lemma 3 (in Appendix)**.** *Each dimension $\boldsymbol{\rho}_i \in \mathbb{R}$ of the residual term $\boldsymbol{\rho} = \sum_{t=1}^{m} \psi^{(t)} \in \mathbb{R}^n$ is the order of $O(1/m)$, where m represents the total number of steps. $\psi^{(t)} = \alpha \sum_{t'=1}^{t-1} (I + \alpha \bar{H}_x)^{t-1-t'} \tilde{R}_2(\Delta x^{(t')})$,*

and each dimension of $\psi^{(t)}$ is the order of $O(1/m^2)$. $\tilde{R}_2(\Delta x^{(t')}) = \frac{\partial}{\partial x} R_2(\Delta x^{(t')})$, and $R_2(\Delta x^{(t')})$ denotes terms of $\Delta x^{(t')}$ higher than the second order in Taylor expansion.

Proof. Without loss of generality, let us focus on the i-th dimension of the residual term $\boldsymbol{\rho} = \sum_{t=1}^{m} \psi^{(t)}$. For convenience, we use $\boldsymbol{\rho}_i \in \mathbb{R}$, $\psi_i^{(t)} \in \mathbb{R}$, and $R_2^i(\Delta x^{(t')}) \in \mathbb{R}$ to denote the i-th dimension of $\boldsymbol{\rho}$, $\psi^{(t)}$, and $R_2(\Delta x^{(t')})$, respectively. Then, the i-th dimension of the residual term $\boldsymbol{\rho}$ can be re-written as

$$\begin{aligned}
\boldsymbol{\rho}_i &= \sum_{t=1}^{m} \psi_i^{(t)} \\
&= \alpha \sum_{t=1}^{m} \sum_{t'=1}^{t-1} \left((I + \alpha \bar{H}_x)^{t-1-t'} \right)_i \tilde{R}_2^i(\Delta x^{(t')}) \\
&= \alpha \sum_{t=1}^{m} \sum_{t'=1}^{t-1} (1 + \alpha \lambda_i)^{t-1-t'} \tilde{R}_2^i(\Delta x^{(t')}). \quad // \quad \text{According to Eq. (25)}
\end{aligned} \tag{30}$$

In the following manuscript, we will prove that each dimension $\boldsymbol{\rho}_i$ of the residual term $\boldsymbol{\rho}$ is the order of $O(1/m^2)$ step by step.

• **Each dimension of the perturbation $\Delta x^{(m)}$ is the order of $O(1/m)$.** Specifically, according to Lemma 2 in Appendix, the perturbation $\Delta x^{(m)}$ can be represented as

$$\begin{aligned}
\Delta x^{(m)} &= \alpha \cdot g_{x+\delta^{(m-1)}} \\
&\approx \alpha (I + \alpha \bar{H}_x)^{m-1} g_x \\
&= \alpha V (I + \alpha \Lambda)^{m-1} V^T g_x. \quad // \quad \text{According to Eq. (25)}
\end{aligned} \tag{31}$$

Then, each dimension $\Delta x_i^{(m)}$ of the perturbation $\Delta x^{(m)}$ can be represented as

$$\Delta x_i^{(m)} = \alpha (1 + \alpha \lambda_i)^{m-1} \gamma_i v_i. \tag{32}$$

We notice that there exists a limit formula $\lim_{a \to +\infty} (1 + \frac{1}{a})^a = \exp(1)$. Then based on this limit formula, the above equation can be further re-written as follows, when the step number m is sufficiently large.

$$\begin{aligned}
\lim_{m \to +\infty} \Delta x_i^{(m)} &= \lim_{m \to +\infty} \alpha (1 + \alpha \lambda_i)^{m-1} \gamma_i v_i \\
&\leq \lim_{m \to +\infty} \alpha (1 + \alpha \lambda_i)^{m} \gamma_i v_i \\
&= \lim_{m \to +\infty} \alpha (1 + \frac{\alpha m}{m} \lambda_i)^{m} \gamma_i v_i \\
&= \alpha \exp(\alpha m \lambda_i) \gamma_i v_i \\
&= \alpha \exp(\beta \lambda_i) \gamma_i v_i \\
&= A \cdot B \cdot C.
\end{aligned} \tag{33}$$

Here, $A = \alpha = \beta / m$ is the order of $O(1/m)$, since β is a small constant, and m is a large enough constant. Besides, $B = \exp(\beta \lambda_i)$ is a constant, and $C = \gamma_i v_i$ is also a constant. Hence, the order of $A \cdot B \cdot C$, *i.e.* each dimension $\Delta x_i^{(m)}$ of the perturbation $\Delta x^{(m)}$, is $O(1/m)$, when the step number m is large enough.

• **Each dimension $R_2^i(\Delta x^{(t)})$ of the term $R_2(\Delta x^{(t)})$ in the residual term $\boldsymbol{\rho}$ is the order of $O(1/m^2)$.** Because $R_2(\Delta x^{(t)})$ denotes terms of $\Delta x^{(t)}$ higher than the second order in Taylor expansion, each dimension $R_2^i(\Delta x^{(t)})$ of the term $R_2(\Delta x^{(t)})$ is the order of $O(1/m^3) + O(1/m^4) + O(1/m^5) + \cdots$. Note that, when the step number m is large enough, $O(1/m^3) + O(1/m^4) + \cdots \leq mO(1/m^3)$, which is the order of $O(1/m^2)$. Hence, each dimension $R_2^i(\Delta x^{(t)})$ of the term $R_2(\Delta x^{(t)})$ is the order of $O(1/m^2)$.

• **Each dimension $\tilde{R}_2^i(\Delta x^{(t)})$ of the term $\tilde{R}_2(\Delta x^{(t)})$ in the residual term $\boldsymbol{\rho}$ is the order of $O(1/m^2)$.** Considering the assumption in Theorem 1 that the gradient $g_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$ is a Lipschitz

function with the Lipschitz constant K, $\|g_{x+\delta^{(t)}} - g_x\| \leq K \cdot \|\delta^{(t)}\|$, and the perturbation $\Delta x^{(t)}$ is small, each dimension $\tilde{R}_2^i(\Delta x^{(t)}) = \frac{\partial}{\partial x} R_2^i(\Delta x^{(t)})$ of the term $\tilde{R}_2(\Delta x^{(t)})$ is also the order of $O(1/m^2)$.

• **Each dimension $\psi_i^{(t)}$ of the term $\psi^{(t)}$ in the residual term $\boldsymbol{\rho}$ is the order of $O(1/m^2)$.** Note that the term $(1+\alpha\lambda_i)^{t-1-t'}$ in Eq. (30) is bounded by $0 \leq (1+\alpha\lambda_i)^{t-1-t'} \leq (1+\alpha\lambda_i)^m \leq \lim_{m\to+\infty}(1+\alpha\lambda_i)^m = \exp(\alpha \cdot m \cdot \lambda_i) = \exp(\beta\lambda_i)$. In this way, $\psi_i^{(t)}$ can be bounded as follows, where $\beta = \alpha m$ is a small constant.

$$\begin{aligned}
\psi_i^{(t)} = \alpha \sum_{t'=1}^{t-1} (1+\alpha\lambda_i)^{t-1-t'} \ \tilde{R}_2^i(\Delta x^{(t')}) \ &\leq \alpha \cdot m \cdot (1+\alpha\lambda_i)^m \ \tilde{R}_2^i(\Delta x^{(t')}) \\
&\leq \alpha \cdot m \cdot \exp(\beta\lambda_i) \ \tilde{R}_2^i(\Delta x^{(t')}) \\
&\leq \beta \cdot \exp(\beta\lambda_i) \ O(\frac{1}{m^2}) \\
&= O(\frac{1}{m^2}).
\end{aligned} \tag{34}$$

• **Each dimension $\boldsymbol{\rho}_i$ of the residual term $\boldsymbol{\rho}$ is the order of $O(1/m)$.**

$$\begin{aligned}
\boldsymbol{\rho}_i &= \sum_{t=1}^{m} \psi_i^{(t)} \\
&\leq m \ O(\frac{1}{m^2}) \quad // \quad \text{According to Eq. (34)} \\
&= O(\frac{1}{m}).
\end{aligned} \tag{35}$$

□

Reason for ignoring the residual term $\boldsymbol{\rho}$ for the perturbation $\delta^{(m)}$ in Theorem 1. According to Lemma 3 in Appendix, $\boldsymbol{\rho}_i$ is the order of $O(1/m)$. When the step number m is large enough, the residual term $\boldsymbol{\rho}$ is small enough to be ignored, without hurting the trustworthiness of the analysis of adversarial perturbations and adversarial training in Theorems 3, 4, 5, and 6.

Moreover, we have conducted experiments to verify that the residual term $\boldsymbol{\rho}$ made an ignorable influence on the adversarial perturbation, *i.e.,* checking whether the theoretically derived solution $\hat{\delta}$ well fitted the real perturbation in practice. Table 4 shows that for each network, the solution $\hat{\delta}$ well fitted the real one. Such a phenomenon successfully verified that the residual term could be ignored, without hurting the trustworthiness of analyzing the adversarial perturbation. Please see Section J for details.

Reason for ignoring the residual term $\bar{H}_x\boldsymbol{\rho} + \sum_{t=1}^m \tilde{R}_2(\Delta x^{(t)})$ for the gradient $g_{x+\delta^{(m)}}$ in Theorem 1. According to Lemma 3 in Appendix, each dimension in the term $\bar{H}_x\boldsymbol{\rho}$ is the order of $O(1/m)$. Moreover, $\sum_{t=1}^m \tilde{R}_2^i(\Delta x^{(t)})$ is the order of $m \cdot O(1/m^2) = O(1/m)$. Hence, each dimension in the residual term $\bar{H}_x\boldsymbol{\rho} + \sum_{t=1}^m \tilde{R}_2(\Delta x^{(t)})$ for the gradient $g_{x+\delta^{(m)}}$ in Theorem 1 is the order of $O(1/m)$. Then, when the step number m is large enough, the residual term $\bar{H}_x\boldsymbol{\rho} + \sum_{t=1}^m \tilde{R}_2(\Delta x^{(t)})$ is small enough to be ignored, without hurting the trustworthiness of the analysis of the gradient.

B PROOF OF THEOREM 2

In this section, we prove Theorem 2 in Section 2.1 of the main paper, which analyzes the adversarial perturbation of the infinite-step attack.

B.1 PROOF OF THEOREM 2

Theorem 2 (Perturbations of the infinite-step attack). *$\beta = \alpha m$ reflects the overall adversarial strength of the infinite-step attack with the step number $m \to +\infty$ and the step size $\alpha = \beta/m \to 0$. Then, based on Assumption 1, this infinite-step adversarial perturbation $\hat{\delta} = \lim_{m\to+\infty} \alpha \sum_{t=0}^{m-1} \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y)$ can be re-written as follows.*

$$\hat{\delta} = \sum_{i=1}^{n} \frac{\exp(\beta\lambda_i) - 1}{\lambda_i} \gamma_i v_i + \hat{\boldsymbol{\rho}}, \quad g_{x+\hat{\delta}} = \sum_{i=1}^{n} \exp(\beta\lambda_i)\gamma_i v_i. \tag{36}$$

Here, $\hat{\boldsymbol{\rho}} \in \mathbb{R}^n$ denotes an ignorable residual term in the Taylor expansion, because $\hat{\boldsymbol{\rho}}_i \in \mathbb{R}$ is proven to be the order of $O(1/m)$.

Proof. According to Eq. (16) and Lemma 2 in Appendix, when the step number $m \to +\infty$, the infinite-step adversarial perturbation $\hat{\delta}$ can be represented as

$$\begin{aligned} \hat{\delta} &= \lim_{m\to+\infty} \Delta x^{(1)} + \Delta x^{(2)} + \cdots + \Delta x^{(m)} \\ &= \lim_{m\to+\infty} \alpha[I + (I + \alpha\bar{H}_x) + \cdots + (I + \alpha\bar{H}_x)^{m-1}]g_x + \lim_{m\to+\infty} \sum_{t=1}^{m} \psi^{(t)}. \end{aligned} \tag{37}$$

Because the Hessian matrix $\bar{H}_x$ is a real-valued symmetric matrix, we can use the eigenvalue decomposition to decompose $\bar{H}_x$ as $\bar{H}_x = V\Lambda V^{-1} = V\Lambda V^T$. In this scenario, the perturbation $\hat{\delta}$ can be further simplified as

$$\begin{aligned} \hat{\delta} &= \lim_{m\to+\infty} \alpha[I + (I + \alpha\bar{H}_x) + \cdots + (I + \alpha\bar{H}_x)^{m-1}]g_x + \lim_{m\to+\infty} \sum_{t=1}^{m} \psi^{(t)} \\ &= \lim_{m\to+\infty} \alpha[I + (I + \alpha V\Lambda V^T) + \cdots + (I + \alpha V\Lambda V^T)^{m-1}]g_x + \lim_{m\to+\infty} \sum_{t=1}^{m} \psi^{(t)} \\ &= \lim_{m\to+\infty} \alpha V[I + (I + \alpha\Lambda) + \cdots + (I + \alpha\Lambda)^{m-1}]V^T g_x + \lim_{m\to+\infty} \sum_{t=1}^{m} \psi^{(t)} \\ &= \lim_{m\to+\infty} \alpha V D V^T g_x + \lim_{m\to+\infty} \sum_{t=1}^{m} \psi^{(t)}, \end{aligned} \tag{38}$$

where we use $D = \alpha(I + (I + \alpha\Lambda) + \cdots + (I + \alpha\Lambda)^{m-1})$ for simplicity. Then, when the step number $m \to +\infty$, the k-th diagonal element $\lim_{m\to+\infty} D_{kk}$ can be written as

$$\begin{aligned} \lim_{m\to+\infty} D_{kk} &= \lim_{m\to+\infty} \left[\alpha(1 + (1 + \alpha\lambda_k) + \cdots + (1 + \alpha\lambda_k)^{m-1})\right] \\ &= \frac{\lim_{m\to+\infty}(1 + \alpha\lambda_k)^m - 1}{\lambda_k} \\ &= \frac{\lim_{m\to+\infty}(1 + \frac{\alpha m\lambda_k}{m})^m - 1}{\lambda_k} \\ &= \frac{\exp(\alpha m\lambda_k) - 1}{\lambda_k} \\ &= \frac{\exp(\beta\lambda_k) - 1}{\lambda_k}. \end{aligned} \tag{39}$$

Then, combining Eq. (39) and Eq. (37), the perturbation $\hat{\delta}$ can be written as

$$\begin{aligned}\hat{\delta} &= \lim_{m\to+\infty} VDV^T g_x + \lim_{m\to+\infty}\sum_{t=1}^{m}\psi^{(t)} \\ &= \lim_{m\to+\infty} VDV^T(\sum_{i=1}^{n}\gamma_i v_i) + \lim_{m\to+\infty}\sum_{t=1}^{m}\psi^{(t)} \\ &= \sum_{i=1}^{n}\lim_{m\to+\infty} D_{ii}v_i v_i^T\sum_{k=1}^{n}\gamma_k v_k + \lim_{m\to+\infty}\alpha\sum_{t=1}^{m}\sum_{t'=1}^{t-1}(I+\alpha\bar{H}_x)^{t-1-t'}\tilde{R}_2(\Delta x^{(t')}) \\ &= \sum_i \frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i v_i + \hat{\boldsymbol{\rho}}.\end{aligned} \tag{40}$$

Here, we use $\hat{\boldsymbol{\rho}} \in \mathbb{R}^n$ to denote the residual term $\lim_{m\to+\infty}\sum_{t=1}^{m}\psi^{(t)}$. **In Lemma 3, we have proven that each dimension $\hat{\boldsymbol{\rho}}_i \in \mathbb{R}$ of the residual term $\hat{\boldsymbol{\rho}}$ is the order of $O(1/m)$. Thus, this residual term $\hat{\boldsymbol{\rho}}$ can be ignored, without hurting the trustworthiness of the analysis of the adversarial perturbation, since the step number m is infinite.** Please see Section B.2 for the detailed discussion.

Based on Eq. (19), the gradient $g_{x+\hat{\delta}}$ of the loss *w.r.t.* the adversarial example $x+\hat{\delta}$ can be re-written as follows. Here, $R_2(\Delta x^{(m)})$ denotes terms of the perturbation $\Delta x^{(m)}$ higher than the second order.

$$\begin{aligned}g_{x+\hat{\delta}} &= g_x + \lim_{m\to+\infty}\sum_{t=1}^{m}\bar{H}_x\Delta x^{(t)} + \lim_{m\to+\infty}\sum_{t=1}^{m}\frac{\partial}{\partial x}R_2(\Delta x^{(t)}) \\ &= g_x + \bar{H}_x\,\hat{\delta} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)}).\end{aligned} \tag{41}$$

Substituting Eq. (40) back to Eq. (41), the gradient $g_{x+\hat{\delta}}$ can be written as

$$\begin{aligned}g_{x+\hat{\delta}} &= g_x + \bar{H}_x\Big(\sum_i \frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i v_i + \hat{\boldsymbol{\rho}}\Big) + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)}) \\ &= \sum_{i=1}^{n}\gamma_i v_i + \bar{H}_x\sum_i\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i v_i + \bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)}) \\ &= \sum_{i=1}^{n}\gamma_i v_i + \sum_i\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i(H_x v_i) + \bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)}) \\ &= \sum_{i=1}^{n}\gamma_i v_i + \sum_i\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i(\lambda_i v_i) + \bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)}) \\ &\approx \sum_i \exp(\beta\lambda_i)\,\gamma_i v_i.\end{aligned} \tag{42}$$

Based on Lemma 3 in Appendix, we have proven that each dimension of the residual term $\bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)})$ in Eq. (42) is the order of $O(1/m)$. Then, considering the step number m is infinite, the residual term $\bar{H}_x\boldsymbol{\rho} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)})$ is small enough to be ignored, without hurting the trustworthiness of the analysis of the gradient. Please see Section B.2 for the detailed analysis.

Hence, Theorem 2 is proven. □

B.2 Reason for ignoring the residual term in Theorem 2

In this subsection, we clarify the reason why the residual term for the perturbation $\hat{\delta}$ in Theorem 2 and the residual term for the gradient $g_{x+\hat{\delta}}$ in Theorem 2 can be ignored.

Reason for ignoring the residual term $\hat{\boldsymbol{\rho}}$ for the perturbation $\hat{\delta}$ in Theorem 2. According to Lemma 3 in Appendix, $\hat{\boldsymbol{\rho}}_i$ is the order of $O(1/m)$. Since the step number m is infinite, the residual

term $\hat{\boldsymbol{\rho}}$ is small enough to be ignored, without hurting the trustworthiness of the analysis of adversarial perturbations and adversarial training in Theorems 3, 4, 5, and 6.

Moreover, we have conducted experiments to verify that the residual term $\hat{\boldsymbol{\rho}}$ made an ignorable influence on the adversarial perturbation, *i.e.,* checking whether the theoretically derived solution $\hat{\delta}$ well fitted the real perturbation in practice. Table 4 shows that for each network, the solution $\hat{\delta}$ well fitted the real one. Such a phenomenon successfully verified that the residual term could be ignored, without hurting the trustworthiness of analyzing the adversarial perturbation. Please see Section J for details.

Reason for ignoring the residual term $\bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)})$ for the gradient $g_{x+\hat{\delta}}$ in Theorem 2. According to Lemma 3 in Appendix, each dimension in the term $\bar{H}_x\hat{\boldsymbol{\rho}}$ is the order of $O(1/m)$. Moreover, $\sum_{t=1}^{m}\tilde{R}_2^i(\Delta x^{(t)})$ is the order of $m{\cdot}O(1/m^2) = O(1/m)$. Hence, each dimension in the residual term $\bar{H}_x\hat{\boldsymbol{\rho}}+\lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)})$ for the gradient $g_{x+\delta^{(m)}}$ in Theorem 2 is the order of $O(1/m)$. Since the step number m is infinite, the residual term $\bar{H}_x\hat{\boldsymbol{\rho}} + \lim_{m\to+\infty}\sum_{t=1}^{m}\tilde{R}_2(\Delta x^{(t)})$ is small enough to be ignored, without hurting the trustworthiness of the analysis of the gradient.

C DETAILED EXPLANATION FOR REMARK 1

In this section, we consider ℓ_2 attacks and ℓ_∞ attacks. As two typical attacking methods, the ℓ_2 attack and the ℓ_∞ attack usually regularize/normalize the adversarial strength in each step by applying $g^{(\ell_2)}_{x+\delta^{(t)}} = \frac{\partial}{\partial x}L(f(x+\delta^{(t)}),y)/\|\frac{\partial}{\partial x}L(f(x+\delta^{(t)}),y)\|$, and $g^{(\ell_\infty)}_{x+\delta^{(t)}} = \text{sign}(\frac{\partial}{\partial x}L(f(x+\delta^{(t)}),y))$, respectively. In fact, for the ℓ_∞ attack, we can roughly consider that only the gradient component $o_x^T g^{(\ell_\infty)}_{x+\delta^{(t)}} o_x$ disentangled from $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ along $\frac{\partial}{\partial x}L(f(x),y)$ is effective, where $o_x = \frac{\partial}{\partial x}L(f(x),y)/\|\frac{\partial}{\partial x}L(f(x),y)\|$ is the unit vector in the direction of $\frac{\partial}{\partial x}L(f(x),y)$. However, it is quite complex to analyze the exact attacking behavior. Therefore, in Remark 1, we just normalize the perturbation in Theorem 2 to roughly approximate the regularization/normalization of perturbations in ℓ_2 attacks and ℓ_∞ attacks.

Remark 1 (Normalized perturbation of the infinite-step attack). *Based on Theorem 2, we ignore residual terms $\hat{\boldsymbol{\rho}}$, where $\hat{\boldsymbol{\rho}}_i$ is proven to be the order of $O(1/m)$. Then, the perturbation of the infinite-step ℓ_2 attack generated via $g^{(\ell_2)}_{x+\delta^{(t)}}$, and the perturbation of the infinite-step ℓ_∞ attack generated via $g^{(\ell_\infty)}_{x+\delta^{(t)}}$ can be approximated as follows.*

$$\hat{\delta}^{(norm)} \approx C\cdot\hat{\delta}/\|\hat{\delta}\| = C\cdot\sum\nolimits_{i=1}^{n}\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i v_i \Big/ \sqrt{\sum\nolimits_{i=1}^{n}(\frac{\exp(\beta\lambda_i)-1}{\lambda_i}\gamma_i)^2}, \tag{43}$$

where $C\in\mathbb{R}$ reflects the total adversarial strength of the ℓ_2 attack or the ℓ_∞ attack.

- ***Experimental verification 1 of Remark 1.*** Although Remark 1 is a brutal approximation of the ℓ_2 attack and ℓ_∞ attack, we conducted experiments to verify the trustworthiness of Remark 1, *i.e.,* checking whether the approximate perturbation $\hat{\delta}^{(\text{norm})}$ in Remark 1 well matched the real perturbation $\delta^{(\ell_2)}$ generated via the ℓ_2 attack. To this end, we calculated the cosine similarity $\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$ to evaluate the error between the theoretical perturbation $\hat{\delta}^{(\text{norm})}$ in Remark 1 and the real perturbation $\delta^{(\ell_2)}$ measured in practice.

Specifically, we learned three types of ReLU networks, including MLPs, CNNs, and MLPs with skip connections (namely ResMLP), on the MNIST dataset. The specific architectures of these three types of ReLU networks were introduced in Section J.

Then, based on each network, we followed the setting in (Wu et al., 2020) to generate the adversarial perturbation $\delta^{(\ell_2)}$ via the ℓ_2 attack, and set the ℓ_2-norm constraint of the adversarial perturbation as $\epsilon = 128/255$ for fair comparison.

Table 3 reports the cosine similarity $\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$ for each network, which was averaged over 40 randomly-selected training samples. We discovered that the cosine similarity $\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$ approximated to 1, which indicated that the theoretically derived perturbations $\hat{\delta}^{(\text{norm})}$ in Remark 1 well matched the real perturbation $\delta^{(\ell_2)}$ of the ℓ_2 attack. Such a phenomenon successfully verified trustworthiness of Remark 1.

Table 3: Cosine similarity $\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$ between the approximate perturbation $\hat{\delta}^{(\text{norm})}$ in Remark 1 and the real perturbation $\delta^{(\ell_2)}$ of the ℓ_2 attack. The cosine similarity $\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$ approximated to 1, which successfully verified trustworthiness of Remark 1.

	1-layer MLP	3-layer MLP	3-layer ResMLP	3-layer CNN
$\cos(\hat{\delta}^{(\text{norm})},\delta^{(\ell_2)})$	0.999285	0.999995	0.999908	0.999999

D PROOF OF ASSUMPTION 2 IN MAIN PAPER

In this section, we prove Assumption 2 in Section 2.2 of the main paper.

Assumption 2 in main paper. *The analysis of binary classification based on a sigmoid function,* $f(x) = \frac{1}{1+exp(-z(x))}, z(x) \in \mathbb{R}$*, can also explain the multi-category classification with a softmax function,* $f(x) = \frac{exp(z'_1)}{\sum_{i=1}^{c} exp(z'_i)}, z' \in \mathbb{R}^c$*, if the second-best category is much stronger than other categories. In this case, attacks on the multi-category classification can be approximated by attacks on the binary classification between the best and the second-best categories, i.e.,* $f(x) \approx \frac{1}{1+exp(-z)}$*, subject to* $z = z'_1 - z'_2 \in \mathbb{R}$*.* z'_1 *and* z'_2 *are referred to as network outputs corresponding to the best category and the second-best category, respectively.*

Proof. Given an input sample x and a ReLU network f trained for multi-category classification based on a softmax function, let $z'_i \in \mathbb{R}, 1 \leq i \leq c$ denote the network output of the i-th confident category, *i.e.,* $z'_1 > z'_2 > \cdots > z'_c$. Then, the probability for the most confident category is given as follows.

$$\begin{aligned} p_1 &= \frac{\exp(z'_1)}{\sum_{i=1}^{c} \exp(z'_i)} \\ &= \frac{1}{\sum_{i=1}^{c} \exp(z'_i - z'_1)}. \end{aligned} \tag{44}$$

When the second-best category is much stronger than other categories, we have $\forall i > 2, \exp(z'_i - z'_1) \ll \exp(z'_2 - z'_1) < \exp(z'_1 - z'_1) = 1$. In this way, Eq. (44) can be re-written as

$$p_1 = \frac{1}{\sum_{i=1}^{c} \exp(z'_i - z'_1)} \approx \frac{1}{\exp(z'_2 - z'_1) + 1} = \frac{1}{1 + \exp(-(z'_1 - z'_2))}. \tag{45}$$

Let $z = z'_1 - z'_2 \in \mathbb{R}$, and we have $f(x) = p_1 \approx \frac{1}{1+\exp(-z)}$. In this way, attacks on the multi-category classification can be approximated by attacks on the binary classification between the best and the second-best categories.

Hence, Assumption 2 is proven. □

E PROOF OF LEMMA 1 IN MAIN PAPER

In this section, we prove Lemma 1 in Section 2.2 of the main paper.

Lemma 1 in main paper. *Let us focus on the cross-entropy loss $L(f(x), y)$. If 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. If the classification is based on a sigmoid operation, the scalar $H_z \geq g_z^2 \geq 0$, as long as the attacking has not finished (still $z(x) \cdot y > 0, y \in \{-1, +1\}$). Here, $g_z = \frac{\partial}{\partial z} L(f(x), y) \in \mathbb{R}$.*

Proof. Let us first consider the classification based on a softmax operation. Given an input sample x and a ReLU network f, the output of the network can be written as $z(x) = f(x) \in \mathbb{R}^c$. In this case, let $p_i = \exp(z_i) / \sum_{k=1}^{c} \exp(z_k)$ denote the probability that the network f classifies the input sample x as the i-th category, where $z_i \in \mathbb{R}$ is referred to as the network output of the i-th category. Then, the cross-entropy loss can be represented as $L(f(x), y) = -\sum_{i=1}^{c} y_i \log(p_i)$, where $y_i \in \{0, 1\}$ denotes the label. Here, let i denote the ground-truth label for the input sample x, *i.e.,* $y_i = 1$, and $\forall k \neq i, y_k = 0$. In this way, the gradient of the loss $L(f(x), y)$ *w.r.t* the network output $z(x) \in \mathbb{R}^c$ is given as

$$g_z = \frac{\partial L(f(x), y)}{\partial z(x)} = -\frac{y_i}{p_i} \cdot \frac{\partial p_i}{\partial z(x)} = -\frac{1}{p_i} \cdot \frac{\partial p_i}{\partial z(x)}. \tag{46}$$

Let us first focus on the network output z_i *w.r.t.* the ground-truth category i. In this scenario, we have

$$\begin{aligned} \frac{\partial p_i}{\partial z_i} &= \frac{\exp(z_i)(\sum_{k=1}^{c} \exp(z_k)) - \exp(z_i)\exp(z_i)}{(\sum_{k=1}^{c} \exp(z_k))^2} \\ &= \frac{\exp(z_i)}{\sum_{k=1}^{c} \exp(z_k)} \cdot (1 - \frac{\exp(z_i)}{\sum_{k=1}^{c} \exp(z_k)}) \\ &= p_i(1 - p_i) = p_i(y_i - p_i). \quad // \quad y_i = 1 \end{aligned} \tag{47}$$

As for $z_k, k \neq i$, we have

$$\begin{aligned} \frac{\partial p_i}{\partial z_k} &= \frac{-\exp(z_i)\exp(z_k)}{(\sum_{k'=1}^{c} \exp(z_{k'}))^2} \\ &= -\frac{\exp(z_i)}{\sum_{k'=1}^{c} \exp(z_{k'})} \cdot \frac{\exp(z_k)}{\sum_{k'=1}^{c} \exp(z_{k'})} \\ &= -p_i p_k = p_i(y_k - p_k). \quad // \quad y_k = 0 \end{aligned} \tag{48}$$

Combining Eq. (46), Eq. (47), and Eq. (48), we have

$$g_z = \mathbf{p} - \mathbf{y}, \tag{49}$$

where $\mathbf{p} = [p_1, p_2, \cdots, p_c] \in \mathbb{R}^c$, and $\mathbf{y} = [y_1, y_2, \cdots, y_c] \in \mathbb{R}^c$.

In this way, based on Eq. (49), the Hessian matrix $H_z \overset{\text{def}}{=} \frac{\partial^2}{\partial z \partial z^T} L(f(x), y)$ of the loss *w.r.t* the network output $z(x)$ can be written as

$$\begin{aligned} H_z &= \frac{\partial^2 L(f(x), y)}{\partial z \partial z^T} = \frac{\partial g_z}{\partial z(x)} \\ &= \frac{\partial(\mathbf{p} - \mathbf{y})}{\partial z(x)} = \frac{\partial \mathbf{p}}{\partial z(x)}. \end{aligned} \tag{50}$$

According to Eq. (47) and Eq. (48), we have $\frac{\partial p_i}{\partial z_i} = p_i(1 - p_i) = p_i - p_i^2$, and $\forall k \neq i$, $\frac{\partial p_i}{\partial z_k} = -p_i p_k$. Then, the Hessian matrix H_z can be re-written as

$$H_z = \frac{\partial p}{\partial z(x)} = \text{diag}([p_1, p_2, \cdots, p_c]) - \mathbf{p}\mathbf{p}^T. \tag{51}$$

In order to prove the Hessian matrix H_z is positive semi-define, we need to verify that all eigenvalues of the Hessian matrix H_z are non-negative. To this end, we use Gershgorin Circle theorem to estimate the bound of eigenvalues. Specifically, Eq. (51) shows that for the k-th row of the Hessian matrix

H_z, the k-th diagonal element of the Hessian matrix H_z is $p_i(1-p_i)$, and the sum of absolute values of non-diagonal elements in the k-th row is $\sum_{k'=1,k'\neq k}^{c} |p_k p_{k'}| = p_k(1-p_k)$. In this way, according to Gershgorin Circle theorem, each eigenvalue λ of the Hessian matrix H_z satisfies $0 \leq \lambda \leq \max_k 2p_k(1-p_k)$. In other words, all eigenvalues of H_z are non-negative. Hence, the Hessian matrix H_z is proven to be positive semi-definite.

Moreover, let us focus on the classification based on a sigmoid operation. In this case, the network output $z(x) \in \mathbb{R}$ is a scalar, and the cross-entropy loss can be represented as $L(f(x), y) = -\log \frac{\exp(z(x)\cdot y)}{1+\exp(z(x)\cdot y)}$, where $y \in \{-1, +1\}$. Then, the gradient of the loss $L(f(x), y)$ *w.r.t* the network output $z(x) \in \mathbb{R}$ is given as

$$\begin{aligned} g_z &= \frac{\partial L(f(x), y)}{\partial z(x)} \\ &= -\frac{1+\exp(z(x)\cdot y)}{\exp(z(x)\cdot y)} \cdot \frac{\exp(z(x)\cdot y)}{(1+\exp(z(x)\cdot y))^2} \cdot y \\ &= -\frac{y}{1+\exp(z(x)\cdot y)} \in \mathbb{R}. \end{aligned} \tag{52}$$

Based on Eq. (52), $H_z \overset{\text{def}}{=} \frac{\partial^2}{\partial z \partial z^T} L(f(x), y) \in \mathbb{R}$ of the loss *w.r.t* the network output $z(x)$ can be written as

$$\begin{aligned} H_z &= \frac{\partial g_z}{\partial z(x)} \\ &= -y \cdot -\frac{y\exp(z(x)\cdot y)}{(1+\exp(z(x)\cdot y))^2} \\ &= \frac{y^2\exp(z(x)\cdot y)}{(1+\exp(z(x)\cdot y))^2} \geq 0. \end{aligned} \tag{53}$$

Combining Eq. (52) and Eq. (53), we have

$$\frac{H_z}{g_z^2} = \frac{y^2\exp(z(x)\cdot y)}{(1+\exp(z(x)\cdot y))^2} \cdot (-\frac{1+\exp(z(x)\cdot y)}{y})^2 = \exp(z(x)\cdot y) \tag{54}$$

If the attacking has not finished yet, *i.e.,* $z(x)\cdot y > 0$, then we have $\exp(z(x)\cdot y) > 1$, thereby $H_z > g_z^2$. Based on Eq. (52), we obtain $g_z^2 = y^2/(1+\exp(z\cdot y))^2 \in \mathbb{R} > 0$, thereby $H_z > g_z^2 > 0$.

Thus, Lemma 1 in main paper is proven. □

F PROOF OF THEOREM 3

In this section, we prove Theorem 3 in Section 2.2 of the main paper, which explains training effects of the adversarial perturbation $\hat{\delta}$ in Theorem 2 on adversarial training.

Specifically, if we use vanilla training to fine-tune the network on the original input sample x for a single step, then the gradient of the loss *w.r.t.* the weight W is given as $g_W = \frac{\partial}{\partial W} L(f(x), y)$. In comparison, if we train the network on the adversarial example $x + \hat{\delta}$ for a single step, then we will get the gradient $g_W^{(\text{adv})} = \frac{\partial}{\partial W} L(f(x + \hat{\delta}), y)$. In this way, $\Delta g_W = g_W^{(\text{adv})} - g_W$ denotes additional effects of adversarial training on the gradient.

$$\begin{aligned} \Delta g_W = g_W^{(\text{adv})} - g_W &= \frac{\partial}{\partial W} L(f(x + \hat{\delta}), y) - \frac{\partial}{\partial W} L(f(x), y) \\ &= x(\bar{H}_h \Delta h)^T + \hat{\delta}(g_h + \bar{H}_h \Delta h)^T. \end{aligned} \tag{55}$$

$\Delta h = W^T \hat{\delta}$ denotes the change of the intermediate-layer feature h caused by the perturbation $\hat{\delta}$, where $W^T = W_j^T \Sigma_{j-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$. For simplicity, we analyze the equivalent weight W for all the first j linear layers, but actually W has similar behavior as W_j, without hurting the generality of the analysis. It is because W can be considered as $W = W_j^T A$, where $A = \Sigma_{j-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$. In this way, the output feature $h = W_j^T x' + b_j$ of the j-th layer can be taken as $h = W^T x + b'$, where x' can be roughly considered as $x' \approx Ax$. Hence, using W for analysis will not significantly hurt the generality of our theorems. $g_h = \frac{\partial}{\partial h} L(f(x), y)$ indicates the gradient of the loss *w.r.t.* the feature h. The matrix $\bar{H}_h = \tilde{g}_h \bar{H}_z \tilde{g}_h^T$, where $\tilde{g}_h = \frac{\partial}{\partial h} z(x)$ indicates the gradient of the network output $z(x)$ *w.r.t.* the feature h. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x + \delta^{(t)}), y)$.

Proof. According to the chain rule, the gradient of the weight W can be written as $g_w = (\frac{\partial L(f(x),y)}{\partial W^T})^T = (\frac{\partial L(f(x),y)}{\partial h^T} \frac{\partial h}{\partial W^T})^T$. Without loss of generality, let us first consider the i-th dimension of h, *i.e.* $h_i = W_i^T x \in \mathbb{R}$, which is only related to the i-th row of W^T. Thus, the gradient of the loss *w.r.t.* $W_i^T \in \mathbb{R}^{1 \times n}$ is given as

$$\frac{\partial L(f(x), y)}{\partial W_i^T} = \frac{\partial L(f(x), y)}{\partial h_i} \frac{\partial h_i}{\partial W_i^T} = \frac{\partial L(f(x), y)}{\partial h_i} x^T. \tag{56}$$

In this way, combining all dimensions of h, we have

$$\begin{aligned} \frac{\partial L(f(x), y)}{\partial W^T} &= [\frac{\partial L(f(x), y)}{\partial W_1^T}, \frac{\partial L(f(x), y)}{\partial W_2^T}, \cdots, \frac{\partial L(f(x), y)}{\partial W_D^T}]^T \\ &= \frac{\partial L(f(x), y)}{\partial h} x^T. \end{aligned} \tag{57}$$

In other words, the gradient g_w of the loss *w.r.t* the weight W can be represented as

$$\begin{aligned} g_W = (\frac{\partial L(f(x), y)}{\partial W^T})^T &= (\frac{\partial L(f(x), y)}{\partial h} x^T)^T \\ &= x \frac{\partial L(f(x), y)}{\partial h^T} = x g_h^T. \end{aligned} \tag{58}$$

According to Eq. (58), the gradient $g_W^{(\text{adv})} = \frac{\partial}{\partial W} L(f(x + \hat{\delta}), y)$ can be re-written as follows, where $g_{h+\Delta h} = \frac{\partial}{\partial h + \Delta h} L(f(x + \hat{\delta}), y)$.

$$g_W^{(\text{adv})} = (x + \hat{\delta})(g_{h+\Delta h})^T. \tag{59}$$

Similar to Eq. (19), the gradient of $g_{h+\Delta h}$ can be re-written as follows.

$$g_{h+\Delta h} \approx g_h + \bar{H}_h \Delta h + \sum_{t=1}^{m} \tilde{R}_2(W^T \Delta x^{(t)}). \tag{60}$$

The matrix $\bar{H}_h = \tilde{g}_h \bar{H}_z \tilde{g}_h^T$ is used to approximate the gradient $g_{h+\Delta h}$, where $\tilde{g}_h = \frac{\partial}{\partial h} z(x)$ indicates the gradient of the network output $z(x)$ *w.r.t.* the feature h. The matrix $\bar{H}_z =$

$\frac{1}{\sum_{t=1}^{m-1}\|\Delta x^{(t)}\|}\sum_{t=1}^{m-1}\|\Delta x^{(t)}\|H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x+\delta^{(t)}), y)$. $\tilde{R}_2(W^T\Delta x^{(t)}) = \frac{\partial}{\partial h}\tilde{R}_2(W^T\Delta x^{(t)})$, where $R_2(W^T\Delta x^{(t)})$ denotes the terms higher than the second order in the Taylor expansion.

Substituting Eq. (60) back to Eq. (59), the gradient $g_W^{(\text{adv})}$ can be represented as

$$g_W^{(\text{adv})} = (x+\hat{\delta})\bigg(g_h + \bar{H}_h\Delta h + \sum_{t=1}^{m}\tilde{R}_2(W^T\Delta x^{(t)})\bigg)^T. \tag{61}$$

Thus, the additional effects of adversarial training on the gradient can be written as follows.

$$\begin{aligned}\Delta g_W &= g_W^{(\text{adv})} - g_W \\ &= x(\bar{H}_h\Delta h)^T + \hat{\delta}(g_h + \bar{H}_h\Delta h)^T + (x+\hat{\delta})\bigg(\sum_{t=1}^{m}\tilde{R}_2(W^T\Delta x^{(t)})\bigg)^T \\ &\approx x(\bar{H}_h\Delta h)^T + \hat{\delta}(g_h + \bar{H}_h\Delta h)^T.\end{aligned} \tag{62}$$

According to Lemma 3 in Appendix, each dimension in the term $\sum_{t=1}^{m} R_2(\Delta x^{(t)})$ is the order of $O(1/m)$. In this way, the complexity of each dimension in the residual term $(x+\hat{\delta})\big(\sum_{t=1}^{m}\tilde{R}_2(W^T\Delta x^{(t)})\big)^T$ is the order of $O(1/m)$. Considering the step number m is infinite, $m \to +\infty$, the effects of the residual term $(x+\hat{\delta})\big(\sum_{t=1}^{m} R_2(W^T\Delta x^{(t)})\big)^T$ in Eq. (62) can be ignored, without affecting the subsequent proofs.

□

Assumption 3 (in Appendix)**.** *Given a ReLU network f, let $W^T = W_j^T\Sigma_{j-1}\cdots\Sigma_2 W_2^T\Sigma_1 W_1^T \in \mathbb{R}^{D\times n}$. Because each column of W^TW is a high-dimensional vector, we can roughly consider that any pair of columns in W^TW is linearly dependent. Thus, W^TW is a full rank matrix, and there exists $(W^TW)^{-1}$.*

Lemma 4 (in Appendix)**.** *Based on Assumption 2 in the main paper, let us focus on the binary classification based on a sigmoid function. Then, the Hessian matrix $H_h^{(t)} = \frac{\partial^2}{\partial h \partial h^T} L(f(x+\delta^{(t)}), y)$ and $H_x^{(t)} = \frac{\partial^2}{\partial x \partial x^T} L(f(x+\delta^{(t)}), y)$ can be represented as $H_h^{(t)} = H_z^{(t)}\tilde{g}_h\tilde{g}_h^T$ and $H_x^{(t)} = H_z^{(t)}\tilde{g}_x\tilde{g}_x^T = WH_h^{(t)}W^T$, respectively. Here, $\tilde{g}_h = \frac{\partial}{\partial h}z(x)$ indicates the gradient of the network output $z(x)$ w.r.t. the feature h, and $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x+\delta^{(t)}), y) \in \mathbb{R}$.*

Proof.

$$\begin{aligned}H_h^{(t)} &= \frac{\partial^2 L(f(x+\delta^{(t)}), y)}{\partial h \partial h^T} = \frac{\partial(\frac{\partial L(f(x+\delta^{(t)}),y)}{\partial z(x)}\frac{\partial z(x)}{\partial h^T})^T}{\partial h^T} \\ &= \frac{(\frac{\partial z(x)}{\partial h^T})^T}{\partial h^T}\cdot\frac{\partial L(f(x+\delta^{(t)}), y)}{\partial z(x)} + (\frac{\partial z(x)}{\partial h^T})^T\cdot\frac{\partial(\frac{\partial L(f(x+\delta^{(t)}),y)}{\partial z(x)})}{\partial h^T} \\ &= (\frac{\partial z(x)}{\partial h^T})^T\frac{\partial^2 L(f(x+\delta^{(t)}), y)}{\partial z(x)\partial z(x)}\frac{\partial z(x)}{\partial h^T} \\ &= H_z^{(t)}\tilde{g}_h\tilde{g}_h^T. \quad // \quad z\in\mathbb{R}, H_z^{(t)}\in\mathbb{R}, \text{according to Assumption 2 in the main paper}\end{aligned} \tag{63}$$

Similarly, we have

$$\begin{aligned} H_x^{(t)} &= \frac{\partial^2 L(f(x+\delta^{(t)}),y)}{\partial x \partial x^T} = \frac{\partial (\frac{\partial L(f(x+\delta^{(t)}),y)}{\partial z(x)} \frac{\partial z(x)}{\partial x^T})^T}{\partial x^T} \\ &= \frac{(\frac{\partial z(x)}{\partial x^T})^T}{\partial x^T} \cdot \frac{\partial L(f(x+\delta^{(t)}),y)}{\partial z(x)} + (\frac{\partial z(x)}{\partial x^T})^T \cdot \frac{\partial (\frac{\partial L(f(x+\delta^{(t)}),y)}{\partial z(x)})}{\partial x^T} \\ &= (\frac{\partial z(x)}{\partial x^T})^T \frac{\partial^2 L(f(x+\delta^{(t)}),y)}{\partial z(x) \partial z(x)} \frac{\partial z(x)}{\partial x^T} \\ &= \tilde{W} H_z^{(t)} (\tilde{W})^T \\ &= H_z^{(t)} \tilde{g}_x \tilde{g}_x^T. \quad // \quad z \in \mathbb{R}, H_z^{(t)} \in \mathbb{R}, \text{according to Assumption 2 in the main paper} \end{aligned} \tag{64}$$

Furthermore, we use the chain rule to re-write the gradient $\tilde{g}_x$ of the network output $z(x)$ *w.r.t* the input sample x.

$$\begin{aligned} \tilde{g}_x = (\frac{\partial z(x)}{\partial x^T})^T &= (\frac{\partial z(x)}{\partial h^T} \frac{\partial h}{\partial x^T})^T \\ &= (\tilde{g}_h^T W^T)^T = W \tilde{g}_h. \end{aligned} \tag{65}$$

In this way, substituting Eq. (65) back to Eq. (64), we get

$$H_x^{(t)} = H_z^{(t)} \tilde{g}_x \tilde{g}_x^T = H_z^{(t)} W \tilde{g}_h (W \tilde{g}_h)^T = W H_h^{(t)} W^T. \tag{66}$$

Thus, Lemma 4 in Appendix is proven. □

Lemma 5 (in Appendix)**.** *Based on Assumption 1, when the loss function is formulated as the cross-entropy loss, the Hessian matrix $H_x^{(t)}$ is positive semi-definite, which is proven in (Yao et al., 2018). Moreover, based on Lemma 1, the matrix $\bar{H}_z$ is positive semi-definite, so the matrix $\bar{H}_x$ is positive semi-definite, as well.*

Proof. Let us first focus on the positive semi-definiteness of the Hessian matrix $H_x^{(t)}$. According to Lemma 1, the Hessian matrix $H_z^{(t)}$ is positive semi-definite (proven in Section E). Then, for any vector $a \in \mathbb{R}^n$, we have

$$\begin{aligned} a^T \ H_x^{(t)} \ a &= a^T \tilde{W} \ H_z^{(t)} \ (\tilde{W})^T a \quad // \quad \text{According to Eq. (64).} \\ &= (\tilde{W}^T a)^T \ H_z^{(t)} \ (\tilde{W}^T a) \\ &\geq 0. \end{aligned} \tag{67}$$

Moreover, it is because the matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1} \|\Delta x^{(t)}\|} \sum_{t=1}^{m-1} \|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)}$, where each Hessian matrix $H_z^{(t)}$ is positive semi-definite. Thus, the matrix $\bar{H}_z$ is also positive semi-definite.

In this way, the positive semi-definiteness of the matrix $\bar{H}_x$ is proven as follows, where $a \in \mathbb{R}^n$ is an arbitrary vector.

$$\begin{aligned} a^T \ \bar{H}_x \ a &= a^T \tilde{W} \ \bar{H}_z \ (\tilde{W})^T a \quad // \quad \text{According to the definition of the matrix } \bar{H}_x. \\ &= (\tilde{W}^T a)^T \ \bar{H}_z \ (\tilde{W}^T a) \\ &\geq 0. \end{aligned} \tag{68}$$

Thus, Lemma 5 is proven.

□

Lemma 6 (in Appendix). *Let $\tilde{g}_x = \frac{\partial}{\partial x} z(x)$ denote the gradient of the network output z w.r.t the input sample x, and $\mathcal{A} = \beta \bar{H}_z \|\tilde{g}_x\|^2 \in \mathbb{R}$. Then, we have*

$$\bar{H}_x \ \Delta g_W = (e^{\mathcal{A}} - 1)\bar{H}_x x g_h^T + \frac{1}{\bar{H}_z \|\tilde{g}_x\|^2}(e^{2\mathcal{A}} - e^{\mathcal{A}})\bar{H}_x g_x g_h^T. \tag{69}$$

Proof. To prove Lemma 6 in Appendix, we multiply $\bar{H}_x$ on both sides of Eq. (55).

$$\bar{H}_x \ \Delta g_W = \bar{H}_x \ (g_W^{(\text{adv})} - g_W) = \bar{H}_x \ x(H_h \Delta h)^T + \bar{H}_x \ \hat{\delta}(g_h + H_h \Delta h)^T. \tag{70}$$

Let us first focus on the first term $\bar{H}_x x(\bar{H}_h \Delta h)^T$ in Eq. (70). According to Eq. (37) and Lemma 4 in Appendix, we can write Δh as follows.

$$\begin{aligned} \Delta h = W^T \hat{\delta} &\approx \alpha W^T [I + (I + \alpha \bar{H}_x) + \cdots + (I + \alpha \bar{H}_x)^{m-1}] g_x \\ &= \alpha W^T [I + (I + \alpha W \bar{H}_h W^T) + \cdots + (I + \alpha W \bar{H}_h W^T)^{m-1}] g_x \quad // \quad \text{according to Eq. (66)} \\ &= \alpha [W^T + W^T (I + \alpha W \bar{H}_h W^T) + \cdots + W^T (I + \alpha W \bar{H}_h W^T)^{m-1}] g_x. \end{aligned} \tag{71}$$

As discussed in Section B.2, each dimension in the residual term $\hat{\rho}$ is the order of $O(1/m)$. Since the step number m is infinite, $m \to +\infty$, the effects of the residual term$\hat{\rho}$ is small enough to be ignored, without hurting the trustworthiness of the subsequent proof. Thus, we ignore the residual term $\hat{\rho}$ in Eq. (37).

Furthermore, to simplify Δh, we apply the mathematical induction to prove that $\forall t, 1 \leq t \leq m, W^T (I + \alpha W \bar{H}_h W^T)^t = (I + \alpha W^T W \bar{H}_h)^t W^T$.

Base case: When $t = 1$, $W^T (I + \alpha W \bar{H}_h W^T) = (W^T + \alpha W^T W \bar{H}_h W^T) = (I + \alpha W^T W \bar{H}_h) W^T$.

Inductive step: For $t > 1$, assuming $W^T (I + \alpha W \bar{H}_h W^T)^{t-1} = (I + \alpha W^T W \bar{H}_h)^{t-1} W^T$, we have

$$\begin{aligned} W^T (I + \alpha W \bar{H}_h W^T)^t &= W^T (I + \alpha W \bar{H}_h W^T)^{t-1} (I + \alpha W \bar{H}_h W^T) \\ &= (I + \alpha W^T W \bar{H}_h)^{t-1} W^T (I + \alpha W \bar{H}_h W^T) \\ &= (I + \alpha W^T W \bar{H}_h)^{t-1} (I + \alpha W^T W \bar{H}_h) W^T \\ &= (I + \alpha W^T W \bar{H}_h)^t W^T \end{aligned} \tag{72}$$

Conclusion: Since both the base case and the inductive step have been proven to be true, we obtain $W^T (I + \alpha W \bar{H}_h W^T)^t = (I + \alpha W^T W \bar{H}_h)^t W^T$.

In this way, we combine Eq. (71) and Eq. (72). The change of the intermediate-layer feature h caused by the perturbation $\hat{\delta}$ can be represented as

$$\Delta h = \alpha [I + (I + \alpha W^T W \bar{H}_h) + \cdots + (I + \alpha W^T W \bar{H}_h)^{m-1}] W^T g_x. \tag{73}$$

Multiply $(I + \alpha W^T W \bar{H}_h)$ on the both sides of Eq. (73), and we get

$$(I + \alpha W^T W \bar{H}_h) \Delta h = \alpha [(I + \alpha W^T W \bar{H}_h) + \cdots + (I + \alpha W^T W \bar{H}_h)^m] W^T g_x. \tag{74}$$

Then, the difference between Eq. (74) and Eq. (73) is

$$\begin{aligned} (I + \alpha W^T W \bar{H}_h) \Delta h - \Delta h &= \alpha [(I + \alpha W^T W \bar{H}_h)^m - I] W^T g_x \\ \Rightarrow \alpha W^T W \bar{H}_h \Delta h &= \alpha [(I + \alpha W^T W \bar{H}_h)^m - I] W^T g_x \\ \Rightarrow W^T W \bar{H}_h \Delta h &= [(I + \alpha W^T W \bar{H}_h)^m - I] W^T g_x. \end{aligned} \tag{75}$$

Therefore, based on Eq. (75), we have

$$\begin{aligned} (\bar{H}_h \Delta h)^T W^T W &= (W^T W \bar{H}_h \Delta h)^T \\ &= ([(I + \alpha W^T W \bar{H}_h)^m - I] W^T g_x)^T \\ &= g_x^T W [(I + \alpha W^T W \bar{H}_h)^m - I]^T \\ &= g_h^T W^T W [(I + \alpha W^T W \bar{H}_h)^m - I]^T \\ &= g_h^T W^T W (I + \alpha W^T W \bar{H}_h)^m - g_h^T W^T W. \end{aligned} \tag{76}$$

Furthermore, the term $g_h^T W^T W(I+\alpha W^T W \bar{H}_h)^m$ in Eq. (76) can be re-written as

$$\begin{aligned} g_h^T W^T W(I+\alpha \bar{H}_h W^T W)^m &= g_h^T W^T W(I+\alpha \bar{H}_h W^T W)(I+\alpha \bar{H}_h W^T W)^{m-1} \\ &= g_h^T (W^T W + \alpha W^T W \bar{H}_h W^T W)(I+\alpha \bar{H}_h W^T W)^{m-1} \\ &= g_h^T (I+\alpha W^T W \bar{H}_h) W^T W (I+\alpha \bar{H}_h W^T W)^{m-1} \\ &= g_h^T (I+\alpha W^T W \bar{H}_h) W^T W (I+\alpha \bar{H}_h W^T W)(I+\alpha \bar{H}_h W^T W)^{m-2} \\ &= g_h^T (I+\alpha W^T W \bar{H}_h)(W^T W + \alpha W^T W \bar{H}_h W^T W)(I+\alpha \bar{H}_h W^T W)^{m-2} \\ &= g_h^T (I+\alpha W^T W \bar{H}_h)^2 W^T W (I+\alpha \bar{H}_h W^T W)^{m-2} \\ &\cdots \\ &= g_h^T (I+\alpha W^T W \bar{H}_h)^m W^T W. \end{aligned} \tag{77}$$

Based on Lemma 4 in Appendix, the term $g_h^T (I+\alpha W^T W \bar{H}_h)^m$ in Eq. (77) can be simplified as

$$\begin{aligned} g_h^T (I+\alpha W^T W \bar{H}_h)^m &= g_h^T (I+\alpha W^T W \bar{H}_h)(I+\alpha W^T W \bar{H}_h)^{m-1} \\ &= g_h^T (I+\alpha W^T W \bar{H}_z \tilde{g}_h \tilde{g}_h^T)(I+\alpha W^T W \bar{H}_h)^{m-1} \\ &= (g_h^T + \alpha\ \bar{H}_z \tilde{g}_h^T W^T W \tilde{g}_h\ g_h^T)(I+\alpha W^T W \bar{H}_h)^{m-1} \\ &= (1+\alpha \mathcal{B}) g_h^T (I+\alpha W^T W \bar{H}_h)^{m-1} \quad // \quad \mathcal{B} = \bar{H}_z \tilde{g}_h^T W^T W \tilde{g}_h \in \mathbb{R} \\ &= (1+\alpha \mathcal{B}) g_h^T (I+\alpha W^T W \bar{H}_h)(I+\alpha W^T W \bar{H}_h)^{m-2} \\ &= (1+\alpha \mathcal{B}) g_h^T (I+\alpha W^T W \bar{H}_z \tilde{g}_h \tilde{g}_h^T)(I+\alpha W^T W \bar{H}_h)^{m-2} \\ &= (1+\alpha \mathcal{B})(g_h^T + \alpha\ \bar{H}_z \tilde{g}_h^T W^T W \tilde{g}_h\ g_h^T)(I+\alpha W^T W \bar{H}_h)^{m-2} \\ &= (1+\alpha \mathcal{B})^2 g_h^T (I+\alpha W^T W \bar{H}_h)^{m-2} \\ &\cdots \\ &= (1+\alpha \mathcal{B})^m g_h^T. \end{aligned} \tag{78}$$

In this way, combining Eq. (78) and Eq. (77), we get

$$\begin{aligned} g_h^T W^T W(I+\alpha \bar{H}_h W^T W)^m &= g_h^T (I+\alpha W^T W \bar{H}_h)^m W^T W \\ &= (1+\alpha \mathcal{B})^m g_h^T W^T W. \end{aligned} \tag{79}$$

Substitute Eq. (79) back to Eq. (76), and we get

$$\begin{aligned} (\bar{H}_h \Delta h)^T W^T W &= g_h^T W^T W (I+\alpha W^T W \bar{H}_h)^m - g_h^T W^T W \\ &= (1+\alpha \mathcal{B})^m g_h^T W^T W - g_h^T W^T W \\ &= [(1+\alpha \mathcal{B})^m - 1] g_h^T W^T W, \end{aligned} \tag{80}$$

where $\mathcal{B} = \bar{H}_z \tilde{g}_h^T W^T W \tilde{g}_h \in \mathbb{R}$.

According to Assumption 3 in Appendix, there exists $(W^T W)^{-1}$. Hence, multiplying $(W^T W)^{-1}$ on both sides of Eq. (80), we get

$$(\bar{H}_h \Delta h)^T = [(1+\alpha \mathcal{B})^m - 1] g_h^T. \tag{81}$$

Since the adversarial perturbation $\hat{\delta}$ is crafted via the infinite-step attack with the infinitesimal step size, *i.e.,* $m \to +\infty$, we have

$$\lim_{m \to +\infty} (1+\alpha \mathcal{B})^m = e^{\alpha m \mathcal{B}} = e^{\beta \mathcal{B}}. \tag{82}$$

Hence, combining Eq. (65) and Eq. (82), we get

$$\begin{aligned}\lim_{m\to+\infty}(1+\alpha\mathcal{B})^m &= e^{\beta\mathcal{B}} \\ &= e^{\beta\bar{H}_z\tilde{g}_h^T W^T W\tilde{g}_h} \\ &= e^{\beta\bar{H}_z\|\tilde{g}_x\|^2} = e^{\mathcal{A}},\end{aligned} \tag{83}$$

where $\mathcal{A} = e^{\beta\bar{H}_z\|\tilde{g}_x\|^2} \in \mathbb{R}$.

Multiply $\bar{H}_x x$ to both side of Eq. (81), and then the first term $\bar{H}_x x(\bar{H}_h\Delta h)^T$ in Eq. (70) can be written as

$$\begin{aligned}\bar{H}_x x(\bar{H}_h\Delta h)^T &= \lim_{m\to+\infty}[(1+\alpha\mathcal{B})^m - 1]\bar{H}_x x g_h^T \\ &= (e^{\mathcal{A}} - 1)\bar{H}_x x g_h^T.\end{aligned} \tag{84}$$

Then, let us focus on the second term $\bar{H}_x\hat{\delta}(g_h + \bar{H}_h\Delta h)^T$ in Eq. (70). Based on Eq. (37) and Lemma 4 in Appendix, the second term $\bar{H}_x\hat{\delta}(g_h + \bar{H}_h\Delta h)^T$ can be re-written as follows.

$$\begin{aligned}&\bar{H}_x\hat{\delta}(g_h + \bar{H}_h\Delta h)^T \\ &= \bar{H}_x\alpha[I + (I + \alpha W\bar{H}_h W^T) + \cdots + (I + \alpha W\bar{H}_h W^T)^{m-1}]g_x(g_h + \bar{H}_h\Delta h)^T \\ &= \bar{H}_x\alpha[I + (I + \alpha W\bar{H}_h W^T) + \cdots + (I + \alpha W\bar{H}_h W^T)^{m-1}]W g_h(g_h + \bar{H}_h\Delta h)^T.\end{aligned} \tag{85}$$

As discussed in Section B.2, each dimension of the residual term $\hat{\rho}$ is the order of $O(1/m)$. Since the step number m is infinite, $m \to +\infty$, the effects of the residual term $\hat{\rho}$ is small enough to be ignored, without hurting the trustworthiness of the subsequent proof. Thus, we ignore the residual term $\hat{\rho}$ in Eq. (37).

For simplicity, let $S = I + (I + \alpha W\bar{H}_h W^T) + \cdots + (I + \alpha W\bar{H}_h W^T)^{m-1}$. Then, multiply $(I + \alpha W\bar{H}_h W^T)$ to both sides of S, and we get

$$\begin{aligned}&(I + \alpha W\bar{H}_h W^T)S = (I + \alpha W\bar{H}_h W^T) + \cdots + (I + \alpha W\bar{H}_h W^T)^m \\ \Rightarrow\ &(I + \alpha W\bar{H}_h W^T)S - S = (I + \alpha W\bar{H}_h W^T)^m - I \\ &\Rightarrow \bar{H}_x\alpha S = (I + \alpha W\bar{H}_h W^T)^m - I. \quad // \quad \text{according to Eq. (66)}\end{aligned} \tag{86}$$

Substituting Eq. (86) back to Eq. (85), we have

$$\bar{H}_x\hat{\delta}(g_h + \bar{H}_h\Delta h)^T = [(I + \alpha W\bar{H}_h W^T)^m - I]W g_h(g_h + \bar{H}_h\Delta h)^T. \tag{87}$$

To simplify Eq. (87), let us first consider the term $(I + \alpha W\bar{H}_h W^T)^m - I$. Specifically, we apply the mathematical induction to derive the term $(I + \alpha W\bar{H}_h W^T)^m - I$, and get $\forall t, 1 \le t \le m, (I + \alpha W\bar{H}_h W^T)^t - I = \frac{1}{\mathcal{B}}[(1 + \alpha\mathcal{B})^t - 1]W\bar{H}_h W^T$, where $\mathcal{B} = \bar{H}_z\tilde{g}_h^T W^T W\tilde{g}_h \in \mathbb{R}$.

Base case: When $t = 1$,

$$\begin{aligned}(I + \alpha W\bar{H}_h W^T)^1 - I &= \alpha W\bar{H}_h W^T \\ &= \frac{1}{\mathcal{B}}[(1 + \alpha\mathcal{B})^1 - 1]W\bar{H}_h W^T.\end{aligned} \tag{88}$$

Inductive step: For $t > 1$, assuming $(I + \alpha W\bar{H}_h W^T)^{t-1} - I = \frac{1}{\mathcal{B}}[(1 + \alpha\mathcal{B})^{t-1} - 1]W\bar{H}_h W^T$, we get

$$\begin{aligned}(I + \alpha W\bar{H}_h W^T)^t - I &= (I + \alpha W\bar{H}_h W^T)^{t-1}(I + \alpha W\bar{H}_h W^T) - I \\ &= (I + \alpha W\bar{H}_h W^T)^{t-1} + (I + \alpha W\bar{H}_h W^T)^{t-1}\alpha W\bar{H}_h W^T - I \\ &= \frac{1}{\mathcal{B}}[(1 + \alpha\mathcal{B})^{t-1} - 1]W\bar{H}_h W^T \\ &\quad + (I + \alpha W\bar{H}_h W^T)^{t-1}\alpha W\bar{H}_h W^T.\end{aligned} \tag{89}$$

Since $(I+\alpha W\bar{H}_h W^T)^{t-1} - I = \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_h W^T$, we obtain $(I+\alpha W\bar{H}_h W^T)^{t-1} = I + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_h W^T$. In this way, based on Lemma 4 in Appendix, Eq. (89) can be further simplified as

$$
\begin{aligned}
&(I+\alpha W\bar{H}_h W^T)^t - I \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1\Big]W\bar{H}_h W^T \\
&\quad + \alpha\Big[I + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_h W^T\Big]W\bar{H}_h W^T \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1\Big]W\bar{H}_h W^T \\
&\quad + \alpha\Big[W\bar{H}_h W^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_h W^T W\bar{H}_h W^T\Big] \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1\Big]W\bar{H}_h W^T \\
&\quad + \alpha\Big[W\bar{H}_h W^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_z\tilde{g}_h\tilde{g}_h^T W^T W\bar{H}_z\tilde{g}_h\tilde{g}_h^T W^T\Big] \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1\Big]W\bar{H}_h W^T \\
&\quad + \alpha\Big[W\bar{H}_h W^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^{t-1}-1]\,\mathcal{B}\,W\bar{H}_z\tilde{g}_h\tilde{g}_h^T W^T\Big] \quad // \quad \mathcal{B} = \bar{H}_z\tilde{g}_h^T W^T W\tilde{g}_h \in \mathbb{R} \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1\Big]W\bar{H}_h W^T + \alpha\Big[W\bar{H}_h W^T + [(1+\alpha\mathcal{B})^{t-1}-1]W\bar{H}_h W^T\Big] \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t-1}-1+\alpha\mathcal{B}(1+\alpha\mathcal{B})^{t-1}\Big]W\bar{H}_h W^T \\
&= \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t}-1\Big]W\bar{H}_h W^T.
\end{aligned}
\tag{90}
$$

Conclusion: Since both the base case and the inductive step have been proven, we have

$$
(I+\alpha W\bar{H}_h W^T)^t - I = \frac{1}{\mathcal{B}}\Big[(1+\alpha\mathcal{B})^{t}-1\Big]W\bar{H}_h W^T, \tag{91}
$$

where $\mathcal{B} = \bar{H}_z\tilde{g}_h^T W^T W\tilde{g}_h \in \mathbb{R}$.

Substituting Eq. (91) back to Eq. (87), we have

$$
\begin{aligned}
\bar{H}_x\hat{\delta}(g_h + H_h\Delta h)^T &= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m - 1]W\bar{H}_h W^T W g_h(g_h + \bar{H}_h\Delta h)^T \\
&= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m - 1]\bar{H}_x W g_h(g_h + \bar{H}_h\Delta h)^T \\
&= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m - 1]\bar{H}_x g_x(g_h + \bar{H}_h\Delta h)^T \quad // \quad \text{According to Eq. (65)} \\
&= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m - 1]\bar{H}_x g_x g_h^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m - 1]\bar{H}_x g_x(\bar{H}_h\Delta h)^T.
\end{aligned}
\tag{92}
$$

Based on Eq. (81), the term $\bar{H}_x g_x(\bar{H}_h\Delta h)^T$ can be represented as

$$
\bar{H}_x g_x(\bar{H}_h\Delta h)^T = [(1+\alpha\mathcal{B})^m - 1]\bar{H}_x g_x g_h^T. \tag{93}
$$

Combining Eq. (92) and Eq. (93), we have

$$\begin{aligned}\bar{H}_x\hat{\delta}(g_h+\bar{H}_h\Delta h)^T &= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m-1]\bar{H}_x g_x g_h^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m-1]\bar{H}_x g_x(\bar{H}_h\Delta h)^T \\ &= \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m-1]\bar{H}_x g_x g_h^T + \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^m-1]^2\bar{H}_x g_x g_h^T \\ &= \frac{1}{\mathcal{B}}(1+\alpha\mathcal{B})^m[(1+\alpha\mathcal{B})^m-1]\bar{H}_x g_x g_h^T.\end{aligned} \tag{94}$$

Based on Eq. (83), the second term $\bar{H}_x\hat{\delta}(g_h+\bar{H}_h\Delta h)^T$ in Eq. (70) can be written as follows, when the adversarial perturbation $\hat{\delta}$ is generated via the infinite-step attack, $m\to+\infty$. Here, $\mathcal{A}=e^{\beta\bar{H}_z\|\tilde{g}_x\|^2}\in\mathbb{R}$, and $\mathcal{B}=\bar{H}_z\tilde{g}_h^T W^T W\tilde{g}_h\in\mathbb{R}$.

$$\begin{aligned}\bar{H}_x\hat{\delta}(g_h+H_h\Delta h)^T &= \lim_{m\to+\infty}\frac{1}{\mathcal{B}}(1+\alpha\mathcal{B})^m[(1+\alpha\mathcal{B})^m-1]\bar{H}_x g_x g_h^T \\ &= \frac{1}{\mathcal{B}}(e^{2\beta\mathcal{B}}-e^{\beta\mathcal{B}})\bar{H}_x g_x g_h^T \\ &= \frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\beta\bar{H}_z\|\tilde{g}_x\|^2}-e^{\beta\bar{H}_z\|\tilde{g}_x\|^2})\bar{H}_x g_x g_h^T \\ &= \frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\mathcal{A}}-e^{\mathcal{A}})\bar{H}_x g_x g_h^T.\end{aligned} \tag{95}$$

In this way, combining Eq. (84) and Eq. (95), Eq. (70) can be represented as

$$\begin{aligned}\bar{H}_x\ \Delta g_W &= \bar{H}_x\ x(\bar{H}_h\Delta h)^T + \bar{H}_x\ \hat{\delta}(g_h+\bar{H}_h\Delta h)^T \\ &= (e^{\mathcal{A}}-1)\bar{H}_x x g_h^T + \frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\mathcal{A}}-e^{\mathcal{A}})\bar{H}_x g_x g_h^T.\end{aligned} \tag{96}$$

Thus, Lemma 6 in Appendix is proven. □

F.1 Proof of Theorem 3

Theorem 3. *Based on Assumptions 1 and 2, 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 $\tilde{g}_x=\frac{\partial z(x)}{\partial x}$ is formulated as follows. $\Delta\tilde{g}_x=-\eta\Delta g_W\tilde{g}_h$ represents the additional effects of adversarial training on changing $\tilde{g}_x$, because adversarial training makes an additional change $-\eta\Delta g_W$ on W[7]. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x$ measures the significance of such additional changes along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T\Delta\tilde{g}_x=-\eta\tilde{g}_x^T\Delta g_W\tilde{g}_h=(e^{\mathcal{A}}-1)\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)}-\frac{\eta g_z^2\ \|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}}-e^{\mathcal{A}}), \tag{97}$$

where $\tilde{g}_h=\frac{\partial z(x)}{\partial h}$, $\mathcal{A}=\beta\bar{H}_z\|\tilde{g}_x\|^2\in\mathbb{R}$, and η denotes the learning rate to update the weight. Considering the footnote[7], $\Delta\tilde{g}_x^{(ori)}=-\eta g_W\tilde{g}_h$ measures the effects of vanilla training on changing $\tilde{g}_x$ in the current back-propagation.

Proof. Based on Lemma 4 in Appendix and Lemma 6 in Appendix, we have

$$\begin{aligned}&\bar{H}_x\Delta g_W=(e^{\mathcal{A}}-1)\bar{H}_x x g_h^T+\frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\mathcal{A}}-e^{\mathcal{A}})\bar{H}_x g_x g_h^T \\ \Rightarrow\ &\bar{H}_z\tilde{g}_x\tilde{g}_x^T\Delta g_W=(e^{\mathcal{A}}-1)\bar{H}_z\tilde{g}_x\tilde{g}_x^T x g_h^T+\frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\mathcal{A}}-e^{\mathcal{A}})\bar{H}_z\tilde{g}_x\tilde{g}_x^T g_x g_h^T \\ \Rightarrow\ &\tilde{g}_x\tilde{g}_x^T\Delta g_W=(e^{\mathcal{A}}-1)\tilde{g}_x\tilde{g}_x^T x g_h^T+\frac{1}{\bar{H}_z\|\tilde{g}_x\|^2}(e^{2\mathcal{A}}-e^{\mathcal{A}})\tilde{g}_x\tilde{g}_x^T g_x g_h^T.\quad //\quad \bar{H}_z\in\mathbb{R}\end{aligned} \tag{98}$$

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

Multiply $\tilde{g}_x^T$ and $\tilde{g}_h$ on both sides of Eq. (98), and we get

$$\begin{aligned}
\tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T \Delta g_W \tilde{g}_h &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T x g_h^T \tilde{g}_h + \frac{1}{\bar{H}_z \|\tilde{g}_x\|^2}(e^{2\mathcal{A}} - e^{\mathcal{A}})\tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T g_x g_h^T \tilde{g}_h \\
\Rightarrow \tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T \Delta g_W \tilde{g}_h &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T g_W \tilde{g}_h + \frac{g_z^2}{\bar{H}_z \|\tilde{g}_x\|^2}(e^{2\mathcal{A}} - e^{\mathcal{A}})\tilde{g}_x^T \tilde{g}_x \tilde{g}_x^T \tilde{g}_x \tilde{g}_h^T \tilde{g}_h \\
\Rightarrow \tilde{g}_x^T \Delta g_W \tilde{g}_h &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T g_W \tilde{g}_h + \frac{g_z^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}})\tilde{g}_h^T \tilde{g}_h \\
\Rightarrow \tilde{g}_x^T \Delta g_W \tilde{g}_h &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T g_W \tilde{g}_h + \frac{g_z^2 \|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}})
\end{aligned} \tag{99}$$

Let $\Delta\tilde{g}_x = -\eta \Delta g_W \tilde{g}_h$ represent the additional effects of adversarial training on changing $\tilde{g}_x$, because adversarial training makes an additional change $-\eta\Delta g_W$ on W[7]. Let $\Delta\tilde{g}_x^{(\text{ori})} = -\eta g_W \tilde{g}_h$ reflect the effects of vanilla training on changing $\tilde{g}_x$ in the current back-propagation, considering the footnote[7]. In this way, Eq. (99) can be re-written as

$$\begin{aligned}
\tilde{g}_x^T(-\eta)\Delta g_W \tilde{g}_h &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T(-\eta) g_W \tilde{g}_h - \frac{\eta g_z^2 \|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}) \\
\Rightarrow \tilde{g}_x^T \Delta\tilde{g}_x &= (e^{\mathcal{A}} - 1)\tilde{g}_x^T \Delta\tilde{g}_x^{(\text{ori})} - \frac{\eta g_z^2 \|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}).
\end{aligned} \tag{100}$$

Thus, Theorem 3 is proven. □

G PROOF OF THEOREM 4

In this section, we prove Theorem 4 in Section 2.2 of the main paper, which explains training effects of the adversarial perturbation $\hat{\delta}$ in Theorem 2 on adversarial training.

Theorem 4. *Based on Assumptions 1 and 2, 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 perturbations $\hat{\delta}$ in Theorem 2. Considering the footnote[7], $\Delta\tilde{g}_x^{(adv)} = -\eta g_W^{(adv)}\tilde{g}_h$ reflects effects of adversarial training on changing the gradient $\tilde{g}_x$. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x^{(adv)}$ represents the significance of such effects along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T\Delta\tilde{g}_x^{(adv)} = -\eta\tilde{g}_x^T g_W^{(adv)}\tilde{g}_h = e^{\mathcal{A}}\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - \frac{\eta g_z^2(e^{2\mathcal{A}} - e^{\mathcal{A}})}{\bar{H}_z}\|\tilde{g}_h\|^2. \tag{101}$$

Proof. Based on Eq. (55), $\Delta g_W = g_W^{(\text{adv})} - g_W$, we add $\tilde{g}_x^T g_W\tilde{g}_h$ on both sides of Eq. (99).

$$\begin{aligned}
&\tilde{g}_x^T\Delta g_W\tilde{g}_h + \tilde{g}_x^T g_W\tilde{g}_h = (e^{\mathcal{A}} - 1)\tilde{g}_x^T g_W\tilde{g}_h + \tilde{g}_x^T g_W\tilde{g}_h + \frac{g_z^2\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}) \\
&\Rightarrow \tilde{g}_x^T(\Delta g_W + g_W)\tilde{g}_h = e^{\mathcal{A}}\tilde{g}_x^T g_W\tilde{g}_h + \frac{g_z^2\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}) \\
&\Rightarrow \tilde{g}_x^T g_W^{(\text{adv})}\tilde{g}_h = e^{\mathcal{A}}\tilde{g}_x^T g_W\tilde{g}_h + \frac{g_z^2\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}).
\end{aligned} \tag{102}$$

Let $\Delta\tilde{g}_x^{(\text{adv})} = -\eta g_W^{(\text{adv})}\tilde{g}_h$ represent effects of adversarial training on changing the gradient $\tilde{g}_x$. Then, Eq. (102) can be simplified as

$$\begin{aligned}
&\tilde{g}_x^T(-\eta)g_W^{(\text{adv})}\tilde{g}_h = e^{\mathcal{A}}\tilde{g}_x^T(-\eta)g_W\tilde{g}_h - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}) \\
&\Rightarrow \tilde{g}_x^T\Delta\tilde{g}_x^{(\text{adv})} = e^{\mathcal{A}}\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{ori})} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}(e^{2\mathcal{A}} - e^{\mathcal{A}}).
\end{aligned} \tag{103}$$

Thus, Theorem 4 is proven. □

H PROOF OF THEOREM 5

In this section, we prove Theorem 5 in Section 2.2 of the main paper, which approximately explains adversarial training based on perturbations of the ℓ_2 attack and the ℓ_∞ attack.

Specifically, if we use vanilla training to fine-tune the network on the original input sample x for a single step, then the gradient of the loss *w.r.t.* the weight W is given as $g_W = \frac{\partial}{\partial W} L(f(x), y)$. In comparison, if we train the network on the adversarial example $x+\hat{\delta}^{(\text{norm})}$ for a single step, then we will get the gradient $g_W^{(\text{adv,norm})} = \frac{\partial}{\partial W} L(f(x+\hat{\delta}^{(\text{norm})}), y)$. In this way, $\Delta g_W^{(\text{norm})} = g_W^{(\text{adv,norm})} - g_W$ represents additional effects on the gradient brought by adversarial training, when we use the normalized perturbation $\hat{\delta}^{(\text{norm})}$ in Remark 1 (related to the ℓ_2 attack and the ℓ_∞ attack).

$$\begin{aligned}\Delta g_W^{(\text{norm})} &= g_W^{(\text{adv,norm})} - g_W = \frac{\partial}{\partial W} L(f(x+\hat{\delta}^{(\text{norm})}), y) - \frac{\partial}{\partial W} L(f(x), y). \\ &= x(\bar{H}_h \Delta h^{(\text{norm})})^T + \hat{\delta}^{(\text{norm})}(g_h + \bar{H}_h \Delta h^{(\text{norm})})^T \\ &= x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h \Delta h)^T + \frac{C\cdot\hat{\delta}}{\|\hat{\delta}\|}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h \Delta h)^T,\end{aligned} \tag{104}$$

where $\Delta h^{(\text{norm})} = W^T \hat{\delta}^{(\text{norm})} = \frac{C}{\|\hat{\delta}\|} W^T \hat{\delta} = \frac{C}{\|\hat{\delta}\|}\Delta h$ denotes the change of the intermediate-layer feature h caused by the perturbation $\hat{\delta}^{(\text{norm})}$. Here, $W^T = W_j^T \Sigma_{j-1} \cdots \Sigma_2 W_2^T \Sigma_1 W_1^T$. Note that, for simplicity, we analyze the equivalent weight W for all the first j linear layers, but W actually has similar behavior as W_j, without hurting the generality of the analysis.

Proof. According to Eq. (58), $g_W = x g_h^T$, the gradient $g_W^{(\text{adv,norm})} = \frac{\partial}{\partial W} L(f(x+\hat{\delta}^{(\text{norm})}), y)$ can be re-written as follows, where $g_{h+\Delta h^{(\text{norm})}} = \frac{\partial}{\partial h + \Delta h^{(\text{norm})}} L(f(x+\hat{\delta}^{(\text{norm})}), y)$.

$$g_W^{(\text{adv,norm})} = (x+\hat{\delta}^{(\text{norm})})(g_{h+\Delta h^{(\text{norm})}})^T. \tag{105}$$

Similar to Eq. (19), the gradient of $g_{h+\Delta h^{(\text{norm})}}$ can be re-written as follows.

$$g_{h+\Delta h^{(\text{norm})}} \approx g_h + \bar{H}_h \Delta h^{(\text{norm})} + \sum_{t=1}^{m} \tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T \Delta x^{(t)}). \tag{106}$$

The matrix $\bar{H}_h = \tilde{g}_h \bar{H}_z \tilde{g}_h^T$ is used to approximate the gradient $g_{h+\Delta h^{(\text{norm})}}$, where $\tilde{g}_h = \frac{\partial}{\partial h} z(x)$ indicates the gradient of the network output $z(x)$ *w.r.t.* the feature h. The matrix $\bar{H}_z = \frac{1}{\sum_{t=1}^{m-1}\|\Delta x^{(t)}\|}\sum_{t=1}^{m-1}\|\Delta x^{(t)}\| H_z^{(t)}$ is a weighted sum of the Hessian matrix $H_z^{(t)} = \frac{\partial^2}{\partial z \partial z^T} L(f(x+\delta^{(t)}), y)$. $\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)}) = \frac{\partial}{\partial h}\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})$, where $R_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})$ denotes the terms higher than the second order in the Taylor expansion.

Substituting Eq. (106) back to Eq. (105), the gradient $g_W^{(\text{adv,norm})}$ can be represented as

$$g_W^{(\text{adv,norm})} = (x+\hat{\delta}^{(\text{norm})})\bigg(g_h + \bar{H}_h \Delta h^{(\text{norm})} + \sum_{t=1}^{m}\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})\bigg)^T. \tag{107}$$

Thus, the additional effects of adversarial training on the gradient can be written as follows.

$$\begin{aligned}\Delta g_W^{(\text{norm})} &= g_W^{(\text{adv,norm})} - g_W \\ &= x(\bar{H}_h \Delta h^{(\text{norm})})^T + \hat{\delta}^{(\text{norm})}(g_h + \bar{H}_h \Delta h^{(\text{norm})})^T + (x+\hat{\delta}^{(\text{norm})})\bigg(\sum_{t=1}^{m}\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})\bigg)^T \\ &\approx x(\bar{H}_h \Delta h^{(\text{norm})})^T + \hat{\delta}^{(\text{norm})}(g_h + \bar{H}_h \Delta h^{(\text{norm})})^T \\ &= x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h \Delta h)^T + \frac{C\cdot\hat{\delta}}{\|\hat{\delta}\|}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h \Delta h)^T.\end{aligned} \tag{108}$$

According to Lemma 3 in Appendix, each dimension in the term $\sum_{t=1}^{m} R_2(\Delta x^{(t)})$ is the order of $O(1/m)$. In this way, each dimension in the residual term $(x+\hat{\delta}^{(\text{norm})})\big(\sum_{t=1}^{m}\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})\big)^T$ is the order of $O(\frac{1}{m})$. Considering the step number m is infinite, $m\to+\infty$, the effects of the

residual term $(x+\hat{\delta}^{\textbf{(norm)}})\big(\sum_{t=1}^{m}\tilde{R}_2(\frac{C}{\|\hat{\delta}\|}\cdot W^T\Delta x^{(t)})\big)^T$ **in Eq. (108) can be ignored, without affecting the subsequent proofs.**

□

Lemma 7 (in Appendix). *Let $\tilde{g}_x=\frac{\partial}{\partial x}z(x)$ denote the gradient of the network output z w.r.t the input sample x, and $\mathcal{A}=\beta\bar{H}_z\|\tilde{g}_x\|^2\in\mathbb{R}$. Then, we have*

$$\bar{H}_x\ \Delta g_W^{(norm)}=\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bar{H}_x x g_h^T+\frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\big[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\big]\bar{H}_x g_x g_h^T. \tag{109}$$

Proof. To prove Lemma 7 in Appendix, we multiply H_x on both sides of Eq. (104).

$$\begin{aligned}\bar{H}_x\ \Delta g_W^{(\text{norm})}&=\bar{H}_x\ (g_W^{(\text{adv,norm})}-g_W)\\&=\bar{H}_x\ x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T+\bar{H}_x\ \frac{C\cdot\hat{\delta}}{\|\hat{\delta}\|}(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T.\end{aligned} \tag{110}$$

Let us first focus on the first term $\bar{H}_x\ x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T$ in Eq. (110). Based on Eq. (84), $\bar{H}_x x(\bar{H}_h\Delta h)^T=(e^{\mathcal{A}}-1)\bar{H}_x x g_h^T$, we have

$$\bar{H}_x x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T=\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bar{H}_x x g_h^T. \tag{111}$$

Then, let us focus on the second term $\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T$ in Eq. (110). Based on Eq. (37) and Lemma 4 in Appendix, the second term $\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T$ can be re-written as follows.

$$\begin{aligned}&\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T\\&=\frac{C}{\|\hat{\delta}\|}\bar{H}_x\alpha[I+(I+\alpha W\bar{H}_h W^T)+\cdots+(I+\alpha W\bar{H}_h W^T)^{m-1}]g_x(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T\\&=\frac{C}{\|\hat{\delta}\|}\bar{H}_x\alpha[I+(I+\alpha W\bar{H}_h W^T)+\cdots+(I+\alpha W\bar{H}_h W^T)^{m-1}]W g_h(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T.\end{aligned} \tag{112}$$

As discussed in Section B.2, each dimension of the residual term $\hat{\rho}$ in Eq. (37) is the order of $O(1/m)$. Since the step number m is infinite, $m\to+\infty$, the effect of the residual term $\hat{\rho}$ of Eq. (37) is small enough to be ignored, without hurting the trustworthiness of the subsequent proof. Thus, we ignore the residual term $\hat{\rho}$ in Eq. (37).

For simplicity, let $S=I+(I+\alpha W\bar{H}_h W^T)+\cdots+(I+\alpha W\bar{H}_h W^T)^{m-1}$. According to Eq. (86), we have proven $\bar{H}_x\alpha S=(I+\alpha W\bar{H}_h W^T)^m-I$. In this way, Eq. (112) can be further simplified as

$$\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T=\frac{C}{\|\hat{\delta}\|}\big[(I+\alpha W\bar{H}_h W^T)^m-I\big]W g_h(g_h+\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T. \tag{113}$$

Moreover, we have proven $(I+\alpha W\bar{H}_h W^T)^m - I = \frac{1}{\mathcal{B}}[(1+\alpha\mathcal{B})^t - 1]W\bar{H}_h W^T$ in Eq. (91), where $\mathcal{B} = \bar{H}_z \tilde{g}_h^T W^T W \tilde{g}_h \in \mathbb{R}$. In this way, we get

$$\begin{aligned}
&\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]W\bar{H}_h W^T W g_h(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x W g_h(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \quad // \quad \text{According to Eq. (65)} \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x g_h^T \\
&\quad + \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T.
\end{aligned} \tag{114}$$

Based on Eq. (81), the term $\bar{H}_x g_x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T$ can be represented as

$$\bar{H}_x g_x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T = \frac{C}{\|\hat{\delta}\|}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x g_h^T. \tag{115}$$

Combining Eq. (115) and Eq. (114), we have

$$\begin{aligned}
\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T &= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x g_h^T \\
&\quad + \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\bar{H}_x g_x g_h^T \\
&\quad + \frac{C^2}{\|\hat{\delta}\|^2\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]^2\bar{H}_x g_x g_h^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\Big[1 + \frac{C}{\|\hat{\delta}\|}\big[(1+\alpha\mathcal{B})^m - 1\big]\Big]\bar{H}_x g_x g_h^T.
\end{aligned} \tag{116}$$

It is because in Eq. (83), we have proven $\lim_{m\to+\infty}(1+\alpha\mathcal{B})^m = e^{\mathcal{A}}$, where $\mathcal{A} = e^{\beta\bar{H}_z\|\tilde{g}_x\|^2} \in \mathbb{R}$. Then, the second term $\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T$ in Eq. (110) can be further written as follows, when the adversarial perturbation $\hat{\delta}$ is generated via the infinite-step attack, $m \to +\infty$.

$$\begin{aligned}
&\frac{C}{\|\hat{\delta}\|}\bar{H}_x\hat{\delta}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \lim_{m\to+\infty}\frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}\Big[(1+\alpha\mathcal{B})^m - 1\Big]\Big[1 + \frac{C}{\|\hat{\delta}\|}\big[(1+\alpha\mathcal{B})^m - 1\big]\Big]\bar{H}_x g_x g_h^T \\
&= \frac{C}{\|\hat{\delta}\|\cdot\mathcal{B}}(e^{\mathcal{A}} - 1)\Big[1 + \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}} - 1)\Big]\bar{H}_x g_x g_h^T \\
&= \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}} - 1)\Big[1 + \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}} - 1)\Big]\bar{H}_x g_x g_h^T.
\end{aligned} \tag{117}$$

In this way, combining Eq. (111) and Eq. (117), Eq. (110) can be represented as

$$\begin{aligned}
\bar{H}_x\ \Delta g_W^{(\text{norm})} &= \bar{H}_x\ x(\frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T + \bar{H}_x\ \frac{C\cdot\hat{\delta}}{\|\hat{\delta}\|}(g_h + \frac{C}{\|\hat{\delta}\|}\bar{H}_h\Delta h)^T \\
&= \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}} - 1)\bar{H}_x x g_h^T + \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}} - 1)\Big[1 + \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}} - 1)\Big]\bar{H}_x g_x g_h^T.
\end{aligned} \tag{118}$$

Thus, Lemma 7 in Appendix is proven. □

H.1 PROOF OF THEOREM 5

Theorem 5. *Based on Assumptions 1 and 2, 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 normalized perturbations $\hat{\delta}^{(norm)}$ in Remark 1. Considering the footnote[8], $\Delta\tilde{g}_x^{(norm)} = -\eta\Delta g_W^{(norm)}\tilde{g}_h = -\eta(g_W^{(adv,\ norm)} - g_W)\tilde{g}_h$ represents additional effects of adversarial training on changing $\tilde{g}_x$. In this way, $\tilde{g}_x^T\Delta\tilde{g}_x^{(norm)} = -\eta\tilde{g}_x^T\Delta g_W^{(norm)}\tilde{g}_h$ reflects the significance of such additional effects along the direction of the gradient $\tilde{g}_x$.*

$$\tilde{g}_x^T\Delta\tilde{g}_x^{(norm)} = C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|})\tilde{g}_x^T\Delta\tilde{g}_x^{(ori)} - C\cdot\frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}\bigg(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|} + C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} - \frac{1}{\|\hat{\delta}\|})^2\bigg). \quad (119)$$

Proof. Based on Lemma 4 in Appendix and Lemma 7 in Appendix, we have

$$\begin{aligned}
\bar{H}_x\ \Delta g_W^{(\text{norm})} &= \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bar{H}_x x g_h^T + \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\bigg[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bigg]\bar{H}_x g_x g_h^T \\
\Rightarrow\ \bar{H}_z\tilde{g}_x\tilde{g}_x^T\Delta g_W^{(\text{norm})} &= \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bar{H}_z\tilde{g}_x\tilde{g}_x^T x g_h^T \\
&\quad + \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\bigg[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bigg]\bar{H}_z\tilde{g}_x\tilde{g}_x^T g_x g_h^T \\
\Rightarrow\ \tilde{g}_x\tilde{g}_x^T\Delta g_W^{(\text{norm})} &= \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\tilde{g}_x\tilde{g}_x^T x g_h^T \\
&\quad + \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\bigg[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\bigg]\tilde{g}_x\tilde{g}_x^T g_x g_h^T. \quad // \quad \bar{H}_z\in\mathbb{R}
\end{aligned} \quad (120)$$

Multiply $\tilde{g}_x^T$ and $\tilde{g}_h$ on both sides of Eq. (120), and we get

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

$$\tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T\Delta g_W^{(\text{norm})}\tilde{g}_h = \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T x g_h^T\tilde{g}_h + \frac{C}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\Big[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\Big]\tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T g_x g_h^T\tilde{g}_h$$

$$\Rightarrow \tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T\Delta g_W^{(\text{norm})}\tilde{g}_h = \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T g_W\tilde{g}_h + \frac{Cg_z^2}{\|\hat{\delta}\|\bar{H}_z\|\tilde{g}_x\|^2}(e^{\mathcal{A}}-1)\Big[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\Big]\tilde{g}_x^T\tilde{g}_x\tilde{g}_x^T\tilde{g}_x\tilde{g}_h^T\tilde{g}_h$$

$$\Rightarrow \tilde{g}_x^T\Delta g_W^{(\text{norm})}\tilde{g}_h = \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\tilde{g}_x^T g_W\tilde{g}_h + \frac{Cg_z^2}{\|\hat{\delta}\|\bar{H}_z}(e^{\mathcal{A}}-1)\Big[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\Big]\tilde{g}_h^T\tilde{g}_h \tag{121}$$

$$\Rightarrow \tilde{g}_x^T\Delta g_W^{(\text{norm})}\tilde{g}_h = \frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\tilde{g}_x^T g_W\tilde{g}_h + \frac{Cg_z^2\|\tilde{g}_h\|^2}{\|\hat{\delta}\|\bar{H}_z}(e^{\mathcal{A}}-1)\Big[1+\frac{C}{\|\hat{\delta}\|}(e^{\mathcal{A}}-1)\Big]$$

$$\Rightarrow \tilde{g}_x^T\Delta g_W^{(\text{norm})}\tilde{g}_h = C\cdot\big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}\big)\tilde{g}_x^T g_W\tilde{g}_h + C\cdot\frac{g_z^2\|\tilde{g}_h\|^2}{\bar{H}_z}\Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}+C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|})^2\Big).$$

Let $\Delta\tilde{g}_x^{(\text{norm})} = -\eta\Delta g_W^{(\text{norm})}\tilde{g}_h$ represent the additional effects of adversarial training on changing $\tilde{g}_x$, considering the footnote[1]. In this way, Eq. (121) can be re-written as

$$\begin{aligned}\tilde{g}_x^T(-\eta)\Delta g_W^{(\text{norm})}\tilde{g}_h &= C\cdot\big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}\big)\tilde{g}_x^T(-\eta)g_W\tilde{g}_h - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}\Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}+C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|})^2\Big)\\ \Rightarrow \tilde{g}_x^T\Delta\tilde{g}_x^{(\text{norm})} &= C\cdot\big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}\big)\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{ori})} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}\Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}+C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|})^2\Big).\end{aligned} \tag{122}$$

Thus, Theorem 5 is proven. □

H.2 Proof for the strength of the training effect $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{norm})}$ in Theorem 5

Given a relatively strong attack, Theorem 2 shows $\|\hat{\delta}\| \to \exp(\beta\|\tilde{g}_x\|^2 g_z^2)/\|g_x\|$. In this way, we can ignore the term $1/\|\hat{\delta}\| \to 0$ in Eq. (12), and prove that the strength of the training effect $\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{norm})}$ is mainly determined by the term $\exp(\mathcal{A})/\|\hat{\delta}\| \approx \|g_x\|\cdot\exp(\beta\|\tilde{g}_x\|^2(\bar{H}_z - g_z^2))$. The proof is as follows.

Proof. Given a relatively strong attack, we can ignore the term $1/\|\hat{\delta}\| \to 0$ in Eq. (12), because a a relatively strong adversarial strength β usually makes $\|\hat{\delta}\| \to \exp(\beta\|\tilde{g}_x\|^2 g_z^2)/\|g_x\|$ with an exponential strength. In this way, Eq. (12) can be re-written as

$$\begin{aligned}\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{norm})} &= C\cdot\big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}\big)\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{ori})} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}\Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|}+C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}-\frac{1}{\|\hat{\delta}\|})^2\Big)\\ &\approx C\cdot\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}\tilde{g}_x^T\Delta\tilde{g}_x^{(\text{ori})} - \frac{\eta g_z^2\,\|\tilde{g}_h\|^2}{\bar{H}_z}\Big(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}+C\cdot(\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|})^2\Big)\\ &= \frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}\left[C\cdot\tilde{g}_x^T\Delta\tilde{g}_x^{\text{ori}} - \frac{\eta g_z^2\|\tilde{g}_h\|^2}{\bar{H}_z}\left(1+C\cdot\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}\right)\right].\end{aligned} \tag{123}$$

Thus, $\tilde{g}_x^T \Delta \tilde{g}_x^{(\text{norm})}$ is determined by the term $\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}$. Since the attack is relatively strong, we have $\|\hat{\delta}\| \approx \exp(\beta \|\tilde{g}_x\|^2 g_z^2)/\|g_x\|$. In this case, the term $\frac{e^{\mathcal{A}}}{\|\hat{\delta}\|}$ can be represented as

$$\begin{aligned} \frac{e^{\mathcal{A}}}{\|\hat{\delta}\|} &\approx \frac{\|g_x\| \exp(\mathcal{A})}{\exp(\beta \|\tilde{g}_x\|^2 g_z^2)} \\ &= \|g_x\| \exp\left[\beta \|\tilde{g}_x\|^2 (\bar{H}_z - g_z^2)\right]. \end{aligned} \tag{124}$$

Hence, we can consider the strength of the training effect $\tilde{g}_x^T \Delta \tilde{g}_x^{(\text{norm})}$ is mainly determined by the term $\exp(\mathcal{A})/\|\hat{\delta}\| \approx \|g_x\| \cdot \exp(\beta \|\tilde{g}_x\|^2 (\bar{H}_z - g_z^2))$. □

I MORE DISCUSSIONS ABOUT RELATED WORK

In fact, Section 3 has discussed the relationship between our theorems and previous findings of adversarial training. Here, we further discuss previous related works, although these works did not all focus on explaining adversarial training. Nevertheless, if this paper is accepted, we will move this section to the main paper.

Some previous studies (Liu et al., 2020; Kanai et al., 2021; Wu et al., 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 et al. (2016) demonstrated that label leaking hindered adversarial training. Tsipras et al. (2019) had proven that compared to vanilla training, adversarial training relied on robust 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 et al., 2018; Tramèr et al., 2018) led to a false sense of security in defenses against adversarial examples. Ilyas et al. (2019) had proven that adversarial examples were attributed to the presence of highly predictive but non-robust features. Some works (Sinha et al., 2017; Zhang & Wang, 2019b; Miyato et al., 2018) demonstrated adversarial examples generated in the supervised way usually corrupted the underlying data structure, which hindered adversarial training (QIAN et al., 2022).

Crucially, it has been discovered that adversarial training usually has a more significant overfitting problem than vanilla training (Rice et al., 2020). Liu et al. (2021) had proven that the overfitting in adversarial training was caused by the model's attempt to fit hard adversarial examples. Chen et al. (2020) considered that the model overfitted the attacks generated in the early stage of adversarial training, and failed to generalize to the attacks in the late stage. Stutz et al. (2020) demonstrated that the overfitting in adversarial training was a result of enforcing high-confidence predictions on adversarial examples. Schmidt et al. (2018) and Zhai et al. (2019) considered that the significantly high adversarial data complexity made adversarial training difficult to achieve good generalization capacity. Rice et al. (2020) used early stopping to reduce overfitting in adversarial training.

Unlike previous studies, this paper analyzes the dynamics of adversarial perturbations, and theoretically explains the difficulty of adversarial training, based on the derived analytic solution. More crucially, our proof can also provide a theoretical explanation for previous findings/understandings of adversarial training (Liu et al., 2020; Kanai et al., 2021; Wu et al., 2020; Yamada et al., 2021; Athalye et al., 2018; Tsipras et al., 2019; Ilyas et al., 2019; Liu et al., 2021; Chen et al., 2020; Rice et al., 2020) in Section 3.

J EXPERIMENTAL VERIFICATION 1 OF THEOREM 2

To verify the correctness of Theorem 2, we conducted experiments to generate adversarial perturbations on four types of ReLU networks, and examined whether the derived analytic solution well fitted the real perturbation measured in practice. Specifically, we calculated the metric $\kappa = \mathbb{E}_x[\|\delta^* - \hat{\delta}\|]/\mathbb{E}_x[\|\delta^*\|]$ to evaluate the error between the derived analytic solution $\hat{\delta}$ in Theorem 2 and the real perturbation δ^* measured in experiments. Here, we followed the same scenario in Wu et al. (2020) to generate adversarial perturbations δ^* via gradient ascent. Specifically, we set the step size $\alpha = 0.005$ to approximately represent the infinite-step attack, *i.e.,* setting $m = 200$. The attacking stopped when the ℓ_2-norm of adversarial perturbations reached the constraint $\epsilon = 128/255$ for fair comparison.

To this end, we learned four types of ReLU networks, including MLPs, CNNs, MLPs with skip connections (namely ResMLP), and CNNs with skip connections (namely ResCNN) on the MNIST dataset (LeCun et al., 1998) via adversarial training. Here, we followed settings in (Ren et al., 2022) to construct five different MLPs, which consisted of $1, 2, 3, 4, 5$ fully-connected (FC) layers, respectively. Each FC layer contained 200 neurons. We also built five different CNNs, which consisted of $1, 2, 3, 4, 5$ convolutional layers, respectively, with a FC layer on the top. Each convolutional layer contained 32 filters. Additionally, we added a skip-connection to each block of a FC layer and a ReLU layer in the above MLPs to construct different ResMLPs. We also added a skip connection to each block consisting of a convolutional layer and a ReLU layer in the above CNNs to build different ResCNNs.

To generate adversarial perturbations, we constructed four baseline attacks. In the first baseline, we set the loss function to the MSE loss, and controlled the gating states of each ReLU layer in each step of the adversarial attack to be the same as those corresponding to the original input sample x. In this way, this baseline attack ignored the residual term $\hat{\boldsymbol{\rho}}$ in Theorem 2, and neglected changes of gating states in Assumption 1, thereby being termed **attack w/o $\hat{\boldsymbol{\rho}}$ w/o $\Delta\Sigma$**. For the second baseline attack, we did not fix the gating states of each ReLU layer, thereby being termed **attack w/o $\hat{\boldsymbol{\rho}}$** For the third baseline attack, we controlled the gating states of ReLU layer, and set the loss function to the cross-entropy loss without ignoring the residual term $\hat{\boldsymbol{\rho}}$, thereby being named as **attack w/o $\Delta\Sigma$**. For the fourth baseline attack, we both set the loss function to the cross-entropy loss and did not fix the gating states, thereby being named as **attack**. Then, for each baseline attack, we averaged the error κ over 40 randomly-selected training samples.

Table 4 reports errors κ computed in four different experimental settings, which were small. Such a phenomenon indicated that the theoretically derived perturbations $\hat{\delta}$ well fitted the real one, which successfully verified Theorem 2. In other words, the residual term $\hat{\boldsymbol{\rho}}$ could be ignored, without hurting the trustworthiness of analyzing the adversarial perturbation $\hat{\delta}$.

Table 4: The error κ between the derived analytic solution $\hat{\delta}$ in Theorem 2 and the real perturbation based on different ReLU networks. The error κ based on each network was small, which successfully verified Theorem 2.

Attacking methods	1-layer MLP	2-layer MLP	3-layer MLP	4-layer MLP	5-layer MLP	3-layer ResMLP	4-layer ResMLP	5-layer ResMLP
Attack w/o $\hat{\boldsymbol{\rho}}$ w/o $\Delta\Sigma$	7.9×10^{-4}	1.2×10^{-4}	1.5×10^{-5}	3.5×10^{-6}	6.6×10^{-7}	1.3×10^{-4}	1.5×10^{-4}	1.5×10^{-4}
Attack w/o $\hat{\boldsymbol{\rho}}$	7.9×10^{-4}	4.3×10^{-2}	6.5×10^{-2}	2.8×10^{-2}	2.3×10^{-2}	4.7×10^{-2}	9.0×10^{-2}	7.8×10^{-2}
Attack w/o $\Delta\Sigma$	3.3×10^{-4}	3.6×10^{-5}	5.1×10^{-6}	1.1×10^{-6}	1.9×10^{-7}	4.2×10^{-5}	3.9×10^{-5}	4.2×10^{-5}
Attack	3.3×10^{-4}	2.5×10^{-2}	2.9×10^{-2}	1.4×10^{-2}	2.3×10^{-2}	3.5×10^{-2}	5.9×10^{-2}	6.0×10^{-2}
Attacking methods	1-layer CNN	2-layer CNN	3-layer CNN	4-layer CNN	5-layer CNN	3-layer ResCNN	4-layer ResCNN	5-layer ResCNN
Attack w/o $\hat{\boldsymbol{\rho}}$ w/o $\Delta\Sigma$	1.2×10^{-6}	4.5×10^{-6}	3.4×10^{-7}	5.1×10^{-8}	4.7×10^{-8}	1.3×10^{-5}	1.5×10^{-5}	3.7×10^{-5}
Attack w/o $\hat{\boldsymbol{\rho}}$	1.5×10^{-1}	1.8×10^{-2}	3.0×10^{-2}	2.8×10^{-2}	1.0×10^{-2}	9.6×10^{-2}	8.3×10^{-2}	5.5×10^{-2}
Attack w/o $\Delta\Sigma$	4.1×10^{-7}	1.5×10^{-6}	1.0×10^{-7}	1.7×10^{-8}	1.4×10^{-8}	4.0×10^{-6}	4.8×10^{-6}	1.2×10^{-5}
Attack	9.3×10^{-2}	1.6×10^{-2}	2.6×10^{-2}	2.3×10^{-2}	1.0×10^{-2}	7.5×10^{-2}	8.4×10^{-2}	4.7×10^{-2}

K EXPERIMENTAL VERIFICATION 1 OF THEOREM 3

To verify the correctness of Theorem 3, we conducted experiments to examine whether the derived training effect well fitted the real effect, based on sixteen adversarially trained ReLU networks in Section J. Specifically, we calculated the metric $\kappa = \mathbb{E}_x[\|\phi^* - \hat{\phi}\|/\|\phi^*\|]$ to evaluate the fitting between the theoretical derivation $\hat{\phi}$ computed using the right side of Eq. (10) and $\phi^* = \tilde{g}_x^T \Delta \tilde{g}_x$ measured in experiments, where ϕ^* was computed using real measurements of $\tilde{g}_x, \eta, g_W^{(\text{adv})}, g_W$, and $\tilde{g}_h$ on each ReLU network.

To generate adversarial perturbations, we set the loss function to the MSE loss. Here, we randomly selected 40 training samples to generate adversarial perturbations. Specifically, we followed the same scenario in Wu et al. (2020) to generate adversarial perturbations δ^* via gradient ascent. We set the step size $\alpha = 0.005$ to approximately represent the infinite-step attack, *i.e.,* setting $m = 200$. The attacking stopped when the ℓ_2-norm of adversarial perturbations reached the constraint $\epsilon = 128/255$ for fair comparison. Considering Theorem 3 was based on the assumption of consistent gating states in Assumption 1, we measured an additional effect ϕ' in experiments by manually forcing gating states of each ReLU layer in the process of generating adversarial perturbations to be the same as gating states for the input sample without being perturbed. To this end, we calculated a new error $\kappa' = \mathbb{E}_x[\|\phi' - \hat{\phi}\|/\|\phi'\|]$. Such a setting well fitted Assumption 1. Table 5 reports errors κ and κ' computed in two different experimental settings, which both verified the correctness of Theorem 3. Particularly, the change of gating states was unpredictable, which brought significant instability in the computation of ϕ^* on a few adversarial examples, *e.g.,* causing dividing 0. Thus, we used 90% samples corresponding to the smallest errors between $\hat{\phi}$ and ϕ^* to calculate the metric κ. Experimental results show that the derived training effect ϕ^* still well explained real effects on most adversarial examples.

Table 5: Experimental verification of Theorem 3 on different adversarially trained ReLU networks. Both the error κ and the error κ' are small, which verifies Theorem 3.

	1-layer MLP	2-layer MLP	3-layer MLP	4-layer MLP	5-layer MLP	3-layer ResMLP	4-layer ResMLP	5-layer ResMLP
κ	3.4×10^{-3}	3.5×10^{-2}	2.0×10^{-1}	1.7×10^{-1}	1.5×10^{-1}	6.1×10^{-2}	2.8×10^{-1}	5.8×10^{-2}
κ'	3.4×10^{-3}	3.3×10^{-4}	3.9×10^{-5}	8.8×10^{-6}	1.5×10^{-6}	3.7×10^{-4}	4.5×10^{-4}	4.3×10^{-4}
	1-layer CNN	2-layer CNN	3-layer CNN	4-layer CNN	5-layer CNN	3-layer ResCNN	4-layer ResCNN	5-layer ResCNN
κ	1.6×10^{-2}	1.5×10^{-2}	1.3×10^{-1}	2.0×10^{-1}	1.1×10^{-1}	7.4×10^{-3}	1.6×10^{-1}	4.0×10^{-2}
κ'	2.9×10^{-6}	1.1×10^{-5}	8.5×10^{-7}	1.3×10^{-7}	1.2×10^{-7}	3.4×10^{-5}	3.9×10^{-5}	9.0×10^{-5}

L DETAILED DISCUSSIONS OF EXPERIMENTAL SETTINGS

• In "*experimental verification 2 of Theorem 2*", we used SGD with learning rate 0.01, and set the batch size to 128 to train VGG-11 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), and ResNet-18 (He et al., 2016) on MNIST dataset, respectively. In this way, for each network architecture, we used the model that was trained for 50 epochs to generate adversarial perturbations. Specifically, we crafted adversarial perturbations $\hat{\delta}$ in Theorem 2 by the gradient $g_{x+\hat{\delta}^{(t)}} = \frac{\partial}{\partial x} L(f(x+\hat{\delta}^{(t)}), y)$ for 500 steps with the step size $\alpha = \frac{1}{100}\epsilon = 0.02$. We further generated adversarial perturbations of the ℓ_2 attack by $g^{(\ell_2)}_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y) / \| \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y) \|$ for 200 steps with the step size $\alpha = \frac{1}{100}\epsilon = 0.02$. Besides, we also crafted adversarial perturbations of the ℓ_∞ attack by applying $g^{(\ell_\infty)}_{x+\delta^{(t)}} = \text{sign}(\frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y))$ for 20 steps with the step size $\alpha = \frac{1}{100}\epsilon = 0.02$. Note that the goal of this experiment was to verify whether the norm of the gradient $\|g_{x+\hat{\delta}}\|$, and the norm of the adversarial perturbation $\|\hat{\delta}\|$ increased with the step number m in an approximately exponential manner. Hence, we ignored the constraint $\|\hat{\delta}\|_p < \epsilon$ of adversarial perturbations, in order to prevent the analysis from being affected by the constraint $\|\hat{\delta}\|_p < \epsilon$. Additionally, in this experiment, we randomly selected 100 training samples in the MNIST dataset for evaluation. Moreover, in subfigure (a) of Fig. 1, we controlled the gating states of each ReLU layer in each step of the adversarial attack to be the same as those corresponding to the original input sample x, in order to remove side effects brought by the chaotic gating states. Whereas, in subfigures (b-e) of Fig. 1, we did not control the gating states of each ReLU layer, which was a more common setting in adversarial attack.

• In "*experimental verification 2 of Theorem 3*", we trained VGG-11 and AlexNet on MNIST dataset against a PGD adversary with 20 steps of the step size $\frac{1}{10}\epsilon = 0.2$. We learned the above networks using SGD with learning rate 0.01.Then, for each network architecture, we used the model that was trained for 50 epochs to generate adversarial perturbations. Specifically, the adversarial perturbation $\hat{\delta}$ for evaluation was generated via the gradient $g_{x+\hat{\delta}^{(t)}} = \frac{\partial}{\partial x} L(f(x+\hat{\delta}^{(t)}), y)$ for 100 steps with the step size $\alpha = \frac{1}{100}\epsilon = 0.02$. Here, we still neglected the constraint of adversarial perturbations. Additionally, in this experiment, we randomly selected 100 training samples in the MNIST dataset for evaluation.

• In "*experimental verification 3 of Theorem 3*", we trained VGG-11 and AlexNet on MNIST dataset against a PGD adversary with 20 steps of the step size $\frac{1}{10}\epsilon = 0.2$. We learned the above networks using SGD with learning rate 0.01.Then, for each network architecture, we used the model that was trained for 50 epochs to generate adversarial perturbations. Specifically, the adversarial perturbation $\hat{\delta}$ for evaluation were generated via the gradient $g_{x+\hat{\delta}^{(t)}} = \frac{\partial}{\partial x} L(f(x+\hat{\delta}^{(t)}), y)$ for $100, 150$ and 200 steps with the step size $\alpha = \frac{1}{100}\epsilon = 0.02$, respectively. Here, we ignored the constraint of adversarial perturbations. Additionally, in this experiment, we randomly selected 100 training samples in the MNIST dataset for evaluation.

• In "*experimental verification of Theorem 6*", we used SGD with learning rate 0.01, and set the batch size to 128 to train VGG-11 and AlexNet on MNIST dataset, respectively. Given an input sample x, we generated adversarial example $x+\delta$ of the ℓ_2 attack by $g^{(\ell_2)}_{x+\delta^{(t)}} = \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y) / \| \frac{\partial}{\partial x} L(f(x+\delta^{(t)}), y) \|$ for 20 steps with the step size $\alpha = \frac{1}{10}\epsilon = 0.2$. To verify Theorem 6, we used the original input sample x and the corresponding adversarial example $x+\delta$ to update the weight W_j in the j-th layer by the same length $\|\Delta W_j\| = \|\Delta W_j^{(\text{adv})}\| = 0.001$. Additionally, in this experiment, we randomly selected 100 training samples in the MNIST dataset for evaluation.

M MORE RESULTS FOR EXPERIMENTAL VERIFICATION 3 OF THEOREM 3

In this section, we conducted additional experiments for "*experimental verification of Theorem 3*." Different from "*experimental verification of Theorem 3*" in Section 2.2 of the main paper, here, we used the model that was trained for 20 epochs to verify the conclusion that the optimization direction of adversarial training was dominated by a few input samples with large $\mathcal{A} = \beta \bar{H}_z \|\tilde{g}_x\|^2$ values.

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. Then, based on the adversarially trained networks in "*experimental verification 2 of Theorem 3*" in Section 2.2, 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 $\Delta \bar{g}_W = \mathbb{E}_{x+\delta}[\Delta g_W]$ over different adversarial examples.

Fig. 5 demonstrated a similar phenomenon to Fig. 3 in Section 2.2 of the main paper. That is, the direction of the average effect $\Delta \bar{g}_W$ was similar to (dominated by) training effects of a few input samples with large $\hat{\mathcal{A}}$ values (the real $\mathcal{A}$ calculated in experiments). Thus, the trustworthiness of Theorem 3 was verified.

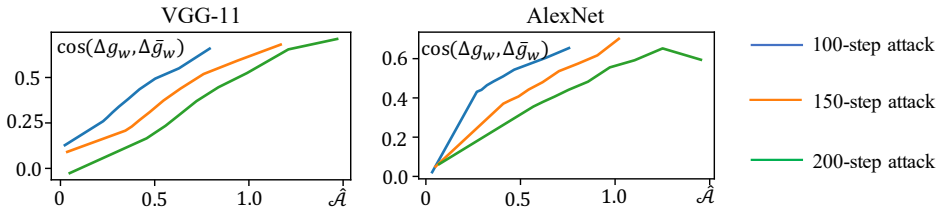

Figure 5: Average cosine similarity $\mathbb{E}_x[\cos(\Delta g_W|_x, \Delta \bar{g}_W)]$ between $\Delta \bar{g}_W$ and each sample x with a specific $\hat{\mathcal{A}}$ value. $\Delta \bar{g}_W$ is similar to the direction of Δg_W *w.r.t.* samples with large $\hat{\mathcal{A}}$ values.

N EVIDENCE FOR THAT ADVERSARIAL TRAINING TENDS TO OSCILLATE IN THE DIRECTIONS OF A FEW UNCONFIDENT SAMPLES

In this section, we conducted new experiments to verify the conclusion that adversarial training was more likely to oscillate in the direction of a few unconfident samples.

Specifically, we constructed a synthetic dataset with 5000 samples, 90% of which were confident samples and 10% of which were unconfident samples. We followed settings in (Wu et al., 2020) to train a 5-layer MLP on this synthetic dataset against a PGD adversary. To verify that adversarial training was more likely to oscillate in the directions of a few unconfident samples, we checked whether the training curve of unconfident samples was more likely to oscillate than the training curve of confident samples.

Fig. 6 shows the training loss *w.r.t.* the confident sample and the training loss *w.r.t.* the unconfident sample, respectively. We discovered that compared to confident samples, training curves of different unconfident samples exhibited differently. Such a phenomenon demonstrated that adversarial training was more likely to oscillate in the directions of a few unconfident samples, which successfully verified our conclusion.

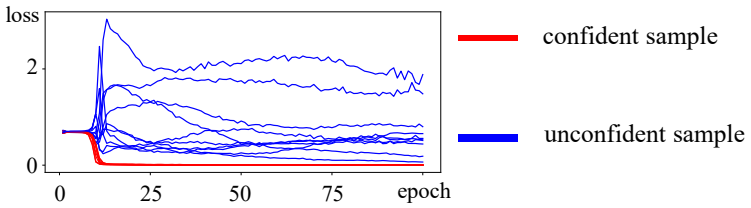

Figure 6: The training loss *w.r.t.* the confident sample and the training loss *w.r.t.* the unconfident sample. Adversarial training was more likely to oscillate in the directions of a few unconfident samples