Can Adversarial Training Be Manipulated By Non-Robust Features?

Lue Tao\(^1\)*  Lei Feng\(^2,3\)  Hongxin Wei\(^4\)
Jinfeng Yi\(^5\)  Sheng-Jun Huang\(^6\)  Songcan Chen\(^6\)†

\(^1\)National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
\(^2\)Chongqing University, Chongqing, China
\(^3\)RIKEN Center for Advanced Intelligence Project, Japan
\(^4\)Nanyang Technological University, Singapore
\(^5\)JD AI Research, Beijing, China
\(^6\)MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract

Adversarial training, originally designed to resist test-time adversarial examples, has shown to be promising in mitigating training-time availability attacks. This defense ability, however, is challenged in this paper. We identify a novel threat model named stability attack, which aims to hinder robust availability by slightly manipulating the training data. Under this threat, we show that adversarial training using a conventional defense budget \(\epsilon\) provably fails to provide test robustness in a simple statistical setting, where the non-robust features of the training data can be reinforced by \(\epsilon\)-bounded perturbation. Further, we analyze the necessity of enlarging the defense budget to counter stability attacks. Finally, comprehensive experiments demonstrate that stability attacks are harmful on benchmark datasets, and thus the adaptive defense is necessary to maintain robustness.[1]

1 Introduction

Robustness to input perturbations is crucial to machine learning deployment in various applications, such as spam filtering [13] and autonomous driving [6]. One of the most popular methods for improving test robustness is adversarial training [39, 1]. By augmenting the training data with \(\epsilon\)-bounded and on-the-fly crafted adversarial examples, adversarial training helps the learned model resist test-time perturbations [39].

On the other hand, machine learning systems are vulnerable to training-time availability attacks [3]. In particular, small perturbations applied into the training data (before training) suffice to degrade the overall test performance of naturally trained models [16] [28]. Fortunately, recent work has proven that adversarial training [39] is capable of mitigating this type of threat [64]. In other words, even if the training data is manipulated to maximize the test error, considerable accuracy on clean test data can still be achieved by adversarially trained models. However, previous work hardly inspects the test robustness of the models, which is what adversarial training was originally proposed for [23] [39]. This naturally raises the following question:

*Are the models adversarially trained on the manipulated data robust to test-time perturbations?*

\[^*\]Work done when Lue Tao was a master’s student at Nanjing University of Aeronautics and Astronautics.
\[^\dagger\]Corresponding author: Songcan Chen <s.chen@nuaa.edu.cn>.
\[^\dagger\]Our code is available at [https://github.com/TLMichael/Hypocritical-Perturbation](https://github.com/TLMichael/Hypocritical-Perturbation)

In this work, we show that conventional adversarial training may fail to provide test robustness when the training data is manipulated by an adversary, and thus an adaptive defense is necessary to resolve this issue. Our contributions are summarized as follows:

1. We introduce a novel threat model called stability attack, where an adversary aims to degrade the overall test robustness of adversarially trained models by slightly perturbing the training data. Figure 1 illustrates the threat of stability attacks.

2. We show that adversarial training using a conventional defense budget provably fails under stability attacks in a simple statistical setting. Specifically, a defense budget of $\epsilon$ will produce models that are not robust to $\epsilon$-bounded adversarial examples when the training data is hypocritically perturbed.

3. We unveil that the aforementioned vulnerability stems from the existence of the non-robust (predictive, yet brittle) features [29] in the original training data. When the non-robust features are reinforced by hypocritical perturbations, the conventional defense budget will be insufficient to offset the negative impact.

4. We further show that a defense budget of $2\epsilon$ is capable of resisting any stability attack for adversarial training, while the budget can be reduced to $\epsilon + \eta$ in a simple statistical setting, where $\eta$ is the magnitude of the non-robust features.

5. We demonstrate that stability attacks are harmful to conventional adversarial training on benchmark datasets. In addition, our empirical study suggests that enlarging the defense budget is essential for mitigating hypocritical perturbations.

To the best of our knowledge, this is the first work that studies the robustness of adversarial training against stability attacks. Both theoretical and empirical evidences show that the conventional defense budget $\epsilon$ is insufficient under the threat of $\epsilon$-bounded training-time perturbations. Our findings suggest that practitioners should consider a larger defense budget of no more than $2\epsilon$ (practically, about $1.5\epsilon \sim 1.75\epsilon$) to achieve a better $\epsilon$-robustness.

2 Threat Models

In this section, we formally introduce the threat model of stability attacks. We begin by revisiting the concepts of natural risk, adversarial risk, and delusive attacks. These concepts naturally give rise to our formulation of stability attacks.

2.1 Preliminaries

Setup. We consider a classification task with input-label pairs $(x, y)$ from an underlying distribution $\mathcal{D}$ over $\mathcal{X} \times [k]$. The goal is to learn a (robust) classifier $f : \mathcal{X} \rightarrow [k]$ that predicts a label $y$ for a given input $x$.

Natural training (NT). Most learning algorithms aim to maximize the generalization performance on unperturbed examples, i.e., natural accuracy. The goal is to minimize the natural risk defined as:

$$R_{nat}(f) := \mathbb{E}_{(x,y) \sim \mathcal{D}} [\mathcal{L}(f(x), y)].$$

(1)
Table 1: Comparisons between evasion attacks, delusive attacks, and stability attacks.

<table>
<thead>
<tr>
<th>Threat model</th>
<th>Training-time perturbation</th>
<th>Test-time perturbation</th>
<th>Learning scheme</th>
<th>Test performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>×</td>
<td>×</td>
<td>NT</td>
<td>Good</td>
</tr>
<tr>
<td>Evasion attacks</td>
<td>×</td>
<td>✓</td>
<td>NT</td>
<td>Bad</td>
</tr>
<tr>
<td>Delusive attacks</td>
<td>✓</td>
<td>×</td>
<td>NT</td>
<td>Bad</td>
</tr>
<tr>
<td>Stability attacks (this paper)</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>AT (conventional)</td>
<td>Bad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AT (our improved)</td>
<td>Good</td>
</tr>
</tbody>
</table>

Adversarial training (AT). Since the risk of adversarial examples (a.k.a. evasion attacks) was found to be unexpectedly high [5, 60], it has become increasingly important to defend the learner against the worst-case perturbations [23, 39]. In this context, the goal is to train a model that has low adversarial risk given a defense budget $\epsilon$:

$$R_{adv}(f) := \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \max_{\delta \in \Delta} \mathcal{L}(f(x + \delta), y) \right], \quad (2)$$

where we choose $\Delta$ to be the set of $\epsilon$-bounded perturbations, i.e., $\Delta = \{ \delta : \|\delta\| \leq \epsilon \}$. This choice is the most common one in the context of adversarial examples [67]. To simplify the notation, we refer to the robustness with respect to this set as $\epsilon$-robustness. It is worth noting that $R_{adv}(f) \geq R_{nat}(f)$ always holds for any $f$, and the equation holds when $\epsilon = 0$.

Delusive attacks. Delusive attacks, which belong to training-time availability attacks, aim to prevent the learner from producing an accurate model by manipulating the training data “imperceptibly” [44]. Concretely, the features of the training data can be perturbed, while the labels should remain correct [16, 42, 56, 28, 81, 15, 64, 18, 19]. This malicious task can be formalized into the following bi-level optimization problem:

$$\max_{P \in S} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \mathcal{L}(f_P(x), y) \right] \quad \text{s.t. } f_P \in \arg \min_f \sum_{(x_i, y_i) \in T} \left[ \mathcal{L}(f(x_i + p_i), y_i) \right]. \quad (3)$$

Here, the adversary aims to maximize the natural risk of the model $f_P$ (that is trained on the manipulated training set) by applying the generated perturbations $P = \{p_i\}_{i=1}^n$ into the original training set $T = \{(x_i, y_i)\}_{i=1}^n$. The commonly used feasible region is $S = \{ \{p_i\}_{i=1}^n : \|p_i\| \leq \epsilon \}$.

Generally, solving Equation (3) is computationally prohibitive for neural networks [64, 18]. Thus, various heuristic methods are proposed to achieve the goal. Among them, a representative method is the hypocritical perturbation [63, 64], crafted as follows:

$$\min_{\|p_i\| \leq \epsilon} \mathcal{L}(f_{craft}(x_i + p_i), y_i), \quad (4)$$

where $f_{craft}$ is called the crafting model, pre-trained before generating poisons. Tao et al. [64] simply adopted a naturally trained classifier as the crafting model, while Huang et al. [28] proposed a min-min bi-level optimization process to pre-train the crafting model. Fu et al. [19] further built their crafting model via a min-min-max three-level optimization process, and generated their poisons by replacing Equation (4) with a min-max bi-level objective.

Another representative method of delusive attacks is the adversarial perturbation, crafted by solving

$$\max_{\|p_i\| \leq \epsilon} \mathcal{L}(f_{craft}(x_i + p_i), y_i). \quad (5)$$

Tao et al. [64] and Fowl et al. [18] both found that applying the adversarial perturbation to the training data is very effective at compromising naturally trained models. However, adversarial training has proven to be promising in defending against various delusive attacks [64].
2.2 Stability Attacks

In contrast to delusive attacks that aim at increasing the natural risk, stability attacks attempt to maximize the adversarial risk of the learner by slightly perturbing the training data:

$$\max_{P \in S} \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in \Delta} \mathcal{L}(f_P(x + \delta), y) \right],$$  \hspace{1cm} (6)

where $f_P$ denotes the victim model, which is naturally or adversarially trained on the perturbed data. In other words, stability attacks seek to hinder the robust availability of the training data. Table 1 shows the comparisons among different threat models.

The goal of stability attacks can be immediately achieved for naturally trained models, since they have already incurred high adversarial risk, even if the training data is clean [60]. To ease the problem of high adversarial risk, adversarial training has been widely used to improve model’s adversarial robustness [24, 12]. Hence, the main goal of stability attacks becomes to compromise the test robustness of adversarially trained models.

Note that Equation (6) is a multi-level optimization problem that is not easy to solve, our next question is how to conduct effective stability attacks against adversarial training. In the following sections, we introduce an effective stability attack method and analyze the cost of resisting it.

**Remark 1.** This work focuses on adding bounded perturbations as small as possible. We mostly assume that the adversary has full control of training data (instead of changing a few) by following previous works [16, 28, 64, 17–19]. This is a realistic assumption [16, 18]. For instance, in some applications an organization may agree to release some internal data for peer assessment, while preventing competitors from easily building a model with high test robustness; this can be achieved by perturbing the entire dataset via stability attacks before releasing. Moreover, this assumption enables a worst-case analysis of the robustness of adversarial training, which may facilitate important theoretical implications.

3 How to Manipulate Adversarial Training

Previous work suggests that adversarial training could defend against both evasion attacks and delusive attacks [39, 64]. However, in this paper, we show that adversarial training using a conventional defense budget $\epsilon$ may not be sufficient to provide $\epsilon$-robustness when confronted with stability attacks. In particular, we present a simple theoretical model where the conventional defense scheme provably fails when the training data is hypocritically perturbed.

The binary classification task. The data model is largely based on the setting proposed by Tsipras et al. [67], which draws a distinction between robust features and non-robust features. Specifically, it consists of input-label pairs $(x, y)$ sampled from a Gaussian mixture distribution $D$ as follows:

$$y \overset{u.a.r.}{\sim} \{-1, +1\}, \quad x_1 \sim \mathcal{N}(y, \sigma^2), \quad x_2, \ldots, x_{d+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\eta y, \sigma^2),$$  \hspace{1cm} (7)

where $\eta$ is much smaller than 1 (i.e., $0 < \eta \ll 1$). Hence, samples from $D$ consist of a robust feature ($x_1$) that is strongly correlated with the label, and $d$ non-robust features ($x_2, \ldots, x_{d+1}$) that are very weakly correlated with it. Typically, an adversary can manipulate a large number of non-robust features - e.g. $d = \Theta(1/\eta^2)$ will suffice.

Before introducing the way to hinder robust availability, we briefly illustrate the success of adversarial training when the training data is unperturbed.

Natural and robust classifiers. For standard classification, we consider a natural classifier:

$$f_{\text{nat}}(x) := \text{sign}(w_{\text{nat}}^\top x), \quad \text{where} \quad w_{\text{nat}} := [1, \eta, \ldots, \eta],$$  \hspace{1cm} (8)

which is a minimizer of the natural risk (1) with 0-1 loss on the data (7), i.e., the Bayes optimal classifier. However, in the adversarial setting, this natural classifier is quite brittle. Thus, it is imperative to obtain a robust classifier:

$$f_{\text{rob}}(x) := \text{sign}(w_{\text{rob}}^\top x), \quad \text{where} \quad w_{\text{rob}} := [1, 0, \ldots, 0],$$  \hspace{1cm} (9)

which relies only on the robust feature $x_1$. 

Illustration of adversarial accuracy. In the adversarial setting, an adversary that is only allowed to perturb each feature by a moderate $\epsilon$ can effectively subvert the natural classifier Tsipras et al. $[67]$. In particular, if $\epsilon = 2\eta$, an adversary can essentially force each non-robust feature to be anti-correlated with the correct label. The following proposition, proved in Appendix C.1, gives the adversarial accuracies of the natural classifier $f_{\text{nat}}$ $[8]$ and the robust classifier $f_{\text{rob}}$ $[9]$. 

**Proposition 1.** Let $\epsilon = 2\eta$ and denote by $A_{\text{adv}}(f)$ the adversarial accuracy, i.e., the probability of a classifier correctly predicting $y$ on the data $[7]$ under $\ell_\infty$ perturbations. Then, we have

$$A_{\text{adv}}(f_{\text{nat}}) \leq \Pr \left\{ \mathcal{N}(0, 1) < \frac{1-d\eta^2}{\sigma \sqrt{1+d\eta^2}} \right\}, \quad A_{\text{adv}}(f_{\text{rob}}) = \Pr \left\{ \mathcal{N}(0, 1) < \frac{1-2\eta}{\sigma} \right\}.$$ 

Proposition 1 implies that the adversarial accuracy of the natural classifier is $< 50\%$ when $d \geq 1/\eta^2$. Even worse, when $\sigma \leq 1/3$ and $d \geq 3/\eta^2$, the adversarial accuracy of the natural classifier $[8]$ is always lower than $1\%$. In contrast, the robust classifier $[9]$ yields a much higher adversarial accuracy (always $> 50\%$); when $\sigma \leq (1-2\eta)/3$, its adversarial accuracy will be higher than $99\%$.

### 3.1 Hypocritical Features Are Harmful

The results above reveal the advantages of robust classifiers over natural classifiers. Note that such a robust classifier can be obtained by adversarial training on the original data $[7]$. However, this defense effect may not hold when the adversary is allowed to perturb the training data.

We show this by analyzing two representative perturbations: the **adversarial perturbation** $[64, 18]$ and the **hypocritical perturbation** $[63, 64]$. When applied into the training data, both perturbations are effective as delusive attacks for naturally trained models. In the following, we show that the former is harmless: adversarial training using a defense budget $\epsilon$ on the adversarially perturbed data can still provide test robustness. In contrast, the latter is harmful: we find that the same defense budget can only produce non-robust classifiers when the training data is hypocritically perturbed.

**A harmless case.** Consider an adversary who is capable of perturbing the training data by an attack budget $\epsilon$. The adversary may choose to shift each feature towards $-y$. Hence, the learner would see input-label pairs $(x, y)$ sampled i.i.d. from a training distribution $T_{\text{adv}}$ as follows:

$$y \sim \{1, -1\}, \quad x_1 \sim \mathcal{N}((1-\epsilon)y, \sigma^2), \quad x_2, \ldots, x_{d+1} \sim \mathcal{N}((\eta-\epsilon)y, \sigma^2),$$

where each feature of the samples from $T_{\text{adv}}$ is adversarially perturbed by a moderate $\epsilon$. While these samples are deviate significantly from the original distribution $D([7])$, adversarial training on them using a defense budget $\epsilon$ is still able to neutralize the non-robust features. Formally, in Appendix C.2 we prove the following theorem.

**Theorem 1** (Adversarial perturbation is harmless). Assume that the adversarial perturbation in the training data $T_{\text{adv}}([10])$ is moderate such that $\eta/2 \leq \epsilon < 1/2$. Then, the optimal linear $\ell_\infty$-robust classifier obtained by minimizing the adversarial risk on $T_{\text{adv}}$ with a defense budget $\epsilon$ is equivalent to the robust classifier $[9]$.

This theorem indicates that the adversarial perturbation is harmless: $\epsilon$-robustness can still be obtained by adversarial training on such perturbed training data.

**A harmful case.** However, this defense effect can be completely broken by the hypocritical perturbation. That is, the adversary can instead shift each feature towards $y$. Hence, the learner would see input-label pairs $(x, y)$ sampled i.i.d. from a training distribution $T_{\text{hyp}}$ as follows:

$$y \sim \{1, -1\}, \quad x_1 \sim \mathcal{N}((1+\epsilon)y, \sigma^2), \quad x_2, \ldots, x_{d+1} \sim \mathcal{N}((\eta+\epsilon)y, \sigma^2),$$

where each feature of the samples from $T_{\text{hyp}}$ is reinforced by a magnitude of $\epsilon$. While these samples become more separable, adversarial training on them using the same defense budget will fail to neutralize the hypocritically perturbed features. Consequently, the resulting classifiers will inevitably have low adversarial accuracy. We make this formal in the following theorem proved in Appendix C.3.

---

$^2$To see how this relates to the hypocritical perturbation $[4]$, let us consider the logistic loss $L(f(x), y) = \log(1 + \exp(-yf(x)))$, and use the natural classifier $[8]$ as the crafting model. Then, the problem $[4]$ has a closed-form solution $p^* = ye \cdot \text{sign}(w_{\text{nat}}) = [ye, \ldots, ye]$. Applying $p^*$ to each $x$ yields the distribution $T_{\text{hyp}}$. 

5
Theorem 2 (Hypocritical perturbation is harmful). The optimal linear $\ell_\infty$-robust classifier obtained by minimizing the adversarial risk on the perturbed data $T_{hyp}$ with a defense budget $\epsilon$ is equivalent to the natural classifier (8).

This theorem implies that the conventional defense scheme can only produce non-robust classifiers, whose adversarial accuracy is as low as that of the natural classifier (8). That is saying, if $\epsilon = 2\eta$, $\sigma \leq 1/3$ and $d \geq 3/\eta^2$, the classifiers cannot get adversarial accuracy better than 1%.

Implications. As it turns out, the seemingly beneficial features in $T_{hyp}$ are actually hypocritical. Therefore, the adversary is highly motivated to hide such hypocritical features in the training data, intending to cajole an innocent learner into relying on the non-robust features. Intriguingly, we notice that the natural classifier (8) (i.e., the crafting model used to derive the distribution $T_{hyp}$) actually has $\eta$-robustness. This is essentially because the non-robust features in the data (7) can resist small-magnitude perturbations by design. This motivates us to use “slightly robust” classifiers as the crafting model in practice. Indeed, our experimental results show that training the crafting model with $0.25\epsilon$-robustness performs the best for conducting stability attacks. This is different from the previous works [64, 18] that use naturally trained models as the crafting model for poisoning.

4 The Necessity of Large Defense Budget

We have shown that the hypocritical perturbation is harmful to the conventional adversarial training scheme. Fortunately, it is possible to strengthen the defense by using a larger defense budget, while the crux of the matter is how large the budget is needed.

We find that the minimum value of the defense budget for a successful defense depends on the specific data distribution. Let us first consider the hypocritical data in $T_{hyp}$. In this case, we show that a larger defense budget is necessary in the following theorem proved in Appendix C.

Theorem 3 ($\epsilon + \eta$ is necessary). The optimal linear $\ell_\infty$-robust classifier obtained by minimizing the adversarial risk on the perturbed data $T_{hyp}$ with a defense budget $\epsilon + \eta$ is equivalent to the robust classifier (9). Moreover, any defense budget lower than $\epsilon + \eta$ will yield classifiers that still rely on all the non-robust features.

This theorem implies that, in the case of the mixture Gaussian distribution under the threat of $\epsilon$-bounded hypocritical perturbations, the learner needs a slightly larger defense budget $\epsilon + \eta$ to ensure $\epsilon$-robustness.

While it is challenging to analyze the minimum value of the defense budget in the general case, the following theorem provides an upper bound of the budget.

Theorem 4 (General case). For any data distribution and any adversary with an attack budget $\epsilon$, training models to minimize the adversarial risk with a defense budget $2\epsilon$ on the perturbed data is sufficient to ensure $\epsilon$-robustness.

The proof of Theorem 4 is deferred in Appendix C.5. It implies that a defense budget twice to the attack budget should be safe enough under the threat of stability attacks. Theorem 3 also suggests that the minimum budget might be much smaller than $2\epsilon$, and it depends on the specific attack methods and data distributions. In the following section, we empirically search for an appropriate defense budget on real-world datasets.

5 Experiments

In this section, we conduct comprehensive experiments to demonstrate the effectiveness of the hypocritical perturbation as stability attacks on popular benchmark datasets and the necessity of an adaptive defense for better robustness.

We conduct stability attacks by applying hypocritical perturbations into the training set. We focus on an $\ell_\infty$ adversary with an attack budget $\epsilon_a = 8/255$ by following [28, 81, 64, 18]. Our crafting model is adversarially trained with a crafting budget $\epsilon_c = 2/255$ for 10 epochs before generating perturbations. Unless otherwise specified, we use ResNet-18 [26] as the default architecture for both the crafting model and the learning model. For adversarial training, we mainly follow the settings in

6
Table 2: Test robustness (%) of PGD-AT using a defense budget $\varepsilon_d = 8/255$ on CIFAR-10.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Natural</th>
<th>FGSM</th>
<th>PGD-20</th>
<th>PGD-100</th>
<th>$\ell_\infty$</th>
<th>AutoAttack</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (clean)</td>
<td>82.17</td>
<td>56.63</td>
<td>50.63</td>
<td>50.35</td>
<td>49.37</td>
<td>46.99</td>
</tr>
<tr>
<td>DeepConfuse [16]</td>
<td>81.25</td>
<td>54.14</td>
<td>48.25</td>
<td>48.02</td>
<td>47.34</td>
<td>44.79</td>
</tr>
<tr>
<td>Unlearnable Examples [28]</td>
<td>83.67</td>
<td>57.51</td>
<td>50.74</td>
<td>50.31</td>
<td>49.81</td>
<td>47.25</td>
</tr>
<tr>
<td>NTGA [81]</td>
<td>82.99</td>
<td>55.71</td>
<td>49.17</td>
<td>48.82</td>
<td>47.96</td>
<td>45.36</td>
</tr>
<tr>
<td>Adversarial Poisoning [18]</td>
<td>77.35</td>
<td>53.93</td>
<td>49.95</td>
<td>49.76</td>
<td>48.35</td>
<td>46.13</td>
</tr>
<tr>
<td>Hypocritical Perturbation (ours)</td>
<td>88.07</td>
<td>47.93</td>
<td>37.61</td>
<td>36.96</td>
<td>38.58</td>
<td>35.44</td>
</tr>
</tbody>
</table>

Table 3: Test robustness (%) of PGD-AT using a defense budget $\varepsilon_d = 8/255$ across different datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attack</th>
<th>Natural</th>
<th>FGSM</th>
<th>PGD-20</th>
<th>PGD-100</th>
<th>$\ell_\infty$</th>
<th>AutoAttack</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN</td>
<td>None</td>
<td>93.95</td>
<td>71.83</td>
<td>57.15</td>
<td>56.02</td>
<td>54.93</td>
<td>50.50</td>
</tr>
<tr>
<td></td>
<td>Adv.</td>
<td>87.50</td>
<td>56.12</td>
<td>46.71</td>
<td>46.32</td>
<td>45.70</td>
<td>42.48</td>
</tr>
<tr>
<td></td>
<td>Hyp.</td>
<td>96.06</td>
<td>59.41</td>
<td>38.17</td>
<td>37.29</td>
<td>40.54</td>
<td>35.43</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>None</td>
<td>56.15</td>
<td>31.50</td>
<td>28.38</td>
<td>28.28</td>
<td>26.53</td>
<td>24.30</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>None</td>
<td>49.34</td>
<td>25.67</td>
<td>22.99</td>
<td>22.86</td>
<td>20.67</td>
<td>18.54</td>
</tr>
<tr>
<td></td>
<td>Adv.</td>
<td>49.52</td>
<td>22.93</td>
<td>20.01</td>
<td>19.91</td>
<td>18.75</td>
<td>16.83</td>
</tr>
<tr>
<td></td>
<td>Hyp.</td>
<td>55.92</td>
<td>20.21</td>
<td>15.61</td>
<td>15.26</td>
<td>14.99</td>
<td>12.53</td>
</tr>
</tbody>
</table>

Table 4: Test robustness (%) of PGD-AT using a defense budget $\varepsilon_d = 8/255$ on CIFAR-10 across different architectures. Test robustness is evaluated by PGD-20. Values in parenthesis denote the accuracy on natural test data.

<table>
<thead>
<tr>
<th>Attack</th>
<th>VGG-16</th>
<th>GoogLeNet</th>
<th>DenseNet-121</th>
<th>MobileNetV2</th>
<th>WideResNet-28-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>47.37 (77.15)</td>
<td>50.67 (83.03)</td>
<td>49.92 (80.08)</td>
<td>48.51 (80.83)</td>
<td>53.91 (85.81)</td>
</tr>
<tr>
<td>Adv.</td>
<td>44.70 (73.24)</td>
<td>47.72 (79.34)</td>
<td>48.00 (78.17)</td>
<td>45.90 (74.61)</td>
<td>51.01 (82.43)</td>
</tr>
<tr>
<td>Hyp.</td>
<td>34.34 (87.20)</td>
<td>37.03 (87.61)</td>
<td>37.58 (88.04)</td>
<td>35.58 (87.04)</td>
<td>41.07 (89.14)</td>
</tr>
</tbody>
</table>

Previous studies [83, 70, 50]. By convention, the defense budget is equal to the attack budget, i.e., $\varepsilon_d = 8/255$. More details on experimental settings are provided in Appendix D.

5.1 Benchmarking (Non-)Robustness

**Attack evaluation.** We compare our crafted hypocritical perturbation to existing methods, which were originally proposed as delusive attacks, including DeepConfuse (which builds an adversarial auto-encoder to generate their perturbations) [16], Unlearnable Examples (which use a min-min bi-level optimization process to pre-train their crafting model) [28], NTGA (which adopts neural tangent kernels as its crafting model) [81], and Adversarial Poisoning (whose crafting model is simply a naturally trained classifier) [18]. It is noteworthy that none of these previous works evaluated the test robustness of their poisoned models.

Results using ResNet-18 on CIFAR-10 are summarized in Table 2. “Natural” denotes the accuracy on natural test data. Various test-time adversarial attacks are used to evaluate test robustness, including FGSM, PGD-20/100, $\ell_\infty$ (version of CW loss [9] optimized by PGD-100), and AutoAttack (a reliable evaluation metric via an ensemble of diverse attacks [11]). We observe that the hypocritical perturbation widely outperforms previous training-time perturbations in degrading the test robustness of PGD-AT [39]. This demonstrates that stability attacks are indeed harmful to the conventional defense scheme. We note that our method increases the natural accuracy. This is reasonable, since our analysis in Section 3.1 has implied that the hypocritical perturbation can increase model reliance on the non-robust features, which are predictive but brittle [29].

Moreover, we evaluate the hypocritical perturbation on other benchmark datasets including SVHN, CIFAR-100, and Tiny-ImageNet. Both the crafting model and the victim model use the ResNet-18
architecture. Results are summarized in Table 3. “Hyp.” denotes the hypocritical perturbation generated by our crafting model. As a comparison, we also evaluate the adversarial perturbation generated using the same crafting model (denoted as “Adv.”). Again, the results show that the hypocritical perturbations are more threatening than the adversarial perturbations to standard adversarial training. This phenomenon is consistent with our analytical results in Section 3.1.

Besides, we find that the hypocritical perturbation can transfer well from ResNet-18 to other architectures, successfully degrading the test robustness of a wide variety of popular architectures including VGG-16 [57], GoogLeNet [61], DenseNet-121 [27], MobileNetV2 [52], and WideResNet-28-10 [82], as shown in Table 4. Note that this is a completely black-box setting where the attacker has no knowledge of the victim model’s initialization, architecture, learning rate scheduler, etc.

Adaptive defense. To prevent the harm of stability attacks, our analysis in Section 4 suggests that a larger defense budget would be helpful. We find that this is indeed the case on CIFAR-10. As shown in Table 5, a large defense budget $\epsilon_d = 14/255$ for PGD-AT performs significantly better than the conventional defense budget $\epsilon_d = 8/255$. We also combine several data augmentations with PGD-AT as defenses by following Fowl et al. [18]. The results show that they are beneficial, while their improvements are inferior to PGD-AT with $\epsilon_d = 14/255$. In addition, we adopt other adversarial training variants including TRADES [83] and MART [70] to defend against the hypocritical perturbation, and find that they achieve comparable defense effects with large defense budgets.

Finally, we note that the adaptive defense has several limitations: i) robust accuracy is improved at the cost of natural accuracy; ii) finding an appropriate defense budget is time-consuming for adversarial training; iii) adversarial training with large budgets may lead to learning obstacles such as inherent large sample complexity [53]. We leave the detailed study of these questions as future work.
5.2 Ablation Studies

In this part, we conduct a set of experiments to provide an empirical understanding of the proposed attack. We train ResNet-18 using PGD-AT on CIFAR-10 by following the same settings described in Appendix D unless otherwise specified.

Analysis on the crafting method. Different from previous work, we use “slightly robust” classifiers as our crafting model. Figure 2(a) shows that this technique greatly improves the potency of the hypocritical perturbation, where the crafting budget \( \epsilon_c = \frac{2}{255} \) performs best in degrading test robustness. We also observe that training the crafting model for 10~80 epochs works well in Figure 2(b) and that optimizing the crafted perturbations over 100 steps performs well in Figure 2(c). Finally, we note that Fowl et al. [18] also tried to use adversarially trained models as the crafting model, but they failed to produce an effective attack in this way. This is mainly because they adopted adversarial perturbations as poisons, which, as we observed, are inferior in degrading test performance.

Ablation on defense budget. As discussed in Section 4, we are motivated to find the appropriate defense budget \( \epsilon_d \) in the range \( [\epsilon \sim 2\epsilon] \). Figure 2(d) shows that the optimal defense budgets against the hypocritical perturbation are 14/255, 12/255, and 14/255 for PGD-AT, TRADES, and MART, respectively. We also observe that all these adversarial training variants are inferior when using the conventional defense budget 8/255.

Less data. We follow Fowl et al. [18] to test the effectiveness of attacks by varying the proportion of clean data and perturbed data. Attacks are then considered effective if they cannot significantly increase performance over training on the clean subset alone. As shown in Table 6, the proposed attack often degrades the test robustness below what one would achieve using full clean dataset. More importantly, the hypocritical perturbations are consistently more harmful than the adversarial perturbations. This again verifies the superiority of hypocritical perturbations as stability attacks.

Effect on natural training. As a sanity check, we include the test accuracy of naturally trained models on CIFAR-10 in Table 7. It shows that without adversarial training, the test robustness of the models becomes very poor—all models only have 0% accuracy under PGD-20 attack. Thus, the goal of stability attacks is immediately achieved. On the other hand, We find that our method degrades the test accuracy from 94.23% to 75.92%, though this is not the main focus of this work. We also observe that Adversarial Poisoning [18] is the most effective method in degrading the test accuracy of naturally trained models. This observation is consistent with Fowl et al. [18].

6 Related Work

Adversarial training. The presence of non-robust features has been demonstrated on popular benchmark datasets [29, 31], which naturally leads to model vulnerability to adversarial examples [67, 58]. To improve test robustness against adversarial examples, adversarial training methods have been developed [23, 39, 73, 83, 65, 46, 76, 62, 69]. Usually, adversarial training using a defense budget \( \epsilon \) is expected to improve model robustness against \( \epsilon \)-bounded adversarial examples. Thus, to break this defense, a direct way is to enlarge the typical \( \epsilon \)-ball used to constrain the attack; however, this may risk changing the true label [8, 66]. In this work, we aim to show that it is possible to achieve this by slightly perturbing the training data without enlarging the \( \epsilon \)-ball.

Data poisoning. Data poisoning attacks, which manipulate training data to cause the resulting models to fail during inference [31], can be divided into availability attacks (to degrade overall test performance) [4, 27, 41, 47, 18] and integrity attacks (to cause specific misclassifications) [32, 10, 55, 87, 21, 54]. While the stability attacks studied in this work may be reminiscent of backdoor attacks [10], we note that they share several key differences. First, stability attacks aim to hinder adversarial training with well-defined \( \epsilon \)-robustness, while backdoor attacks mainly focus on
embedding malicious behaviors (that can be invoked by pre-specified triggers) into naturally trained models [22, 51, 68]. Second, stability attacks only perturb the inputs slightly, while most works on backdoor attacks require mislabeling [25, 37, 45, 36, 74]. Thus, backdoor defenses [7, 75, 65] might not be directly applied to resist stability attacks.

Additional related works are discussed in Appendix A.

7 Conclusion

In this work, we establish a framework to study the robustness of adversarial training against stability attacks. We unveil the threat of stability attacks—small hypocritical perturbations applied into the training data suffice to hinder conventional adversarial training. The conventional defense budget $\epsilon$ is insufficient under the threat. To resist it, we suggest a larger defense budget of no more than $2\epsilon$. Our theoretical analysis explains why hypocritical perturbations are effective as stability attacks—they can mislead the learner by reinforcing the non-robust features. Experiments demonstrate that hypocritical perturbations are harmful to conventional adversarial training on benchmark datasets, and enlarging the defense budget is essential for mitigating hypocritical perturbations. Future work includes relaxing the assumption that the adversary perturbs the entire training set and designing more effective stability attacks against adversarial training.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 61732006, 62076124, 62076128, 62106028), the National Key R&D Program of China (2020AAA0107000). Lei Feng was also supported by Chongqing Overseas Chinese Entrepreneurship, Innovation Support Program and CAAI-Huawei MindSpore Open Fund. The authors wish to thank the anonymous reviewers for their helpful comments and suggestions.

References


[29] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In NeurIPS, 2019.


[78] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In ICML, 2021.


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 2.2 and Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix F.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix B.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All experiments are run on a single NVIDIA GeForce RTX 3090 GPU.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]