Can Adversarial Training Be Manipulated By Non-Robust Features?

Anonymous Author(s)
Affiliation
Address
email

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 attacks*, 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 ϵ provably fails to provide test robustness in a simple statistical setting, where the non-robust features of the training data can be reinforced by ϵ-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 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 ϵ-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 shown that adversarial training [39] is capable of mitigating this type of threat [64, 18]. In other words, even if the training data is manipulated by an adversary to maximize the test error, adversarially trained models can still achieve considerable accuracy on clean test data. 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:

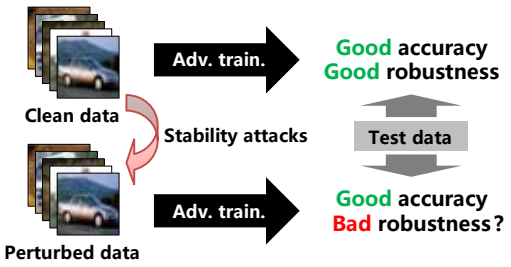


Figure 1: An illustration of stability attacks, where the training data is slightly perturbed to hinder adversarial training.

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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

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 attacks*, 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 ϵ will produce models that are *not* robust to ϵ-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 2ϵ 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 η 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 ϵ is insufficient under the threat of ϵ-bounded training-time perturbations. Our findings suggest that practitioners should consider a larger defense budget of no more than 2ϵ (practically, about $1.5\epsilon \sim 1.75\epsilon$) to achieve a better ϵ-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 $(\boldsymbol{x}, y)$ from an underlying distribution $\mathcal{D}$ over $\mathcal{X} \times [k]$. The goal is to learn a (robust) classifier $f : \mathcal{X} \to [k]$ that predicts a label y for a given input $\boldsymbol{x}$.

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:

$$\mathcal{R}_{\text{nat}}(f) \coloneqq \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathcal{L}(f(\boldsymbol{x}), y)\right]. \tag{1}$$

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 ϵ:

$$\mathcal{R}_{\text{adv}}(f) \coloneqq \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\Delta} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}), y)\right], \tag{2}$$

where we choose Δ to be the set of ϵ-bounded perturbations, i.e., $\Delta = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\| \le \epsilon\}$. 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 ϵ-robustness. It is worth noting that $\mathcal{R}_{\text{adv}}(f) \ge \mathcal{R}_{\text{nat}}(f)$ always holds for any f, and the equation holds when $\epsilon = 0$.

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

Threat model	Training-time perturbation	Test-time perturbation	Learning scheme	Test performance
None	✗	✗	NT	Good
Evasion attacks [5, 60, 23, 39]	✗	✓	NT AT	Bad Good
Delusive attacks [44, 16, 28, 64]	✓	✗	NT AT	Bad Good
Stability attacks (this paper)	✓	✓	AT (conventional) AT (our improved)	Bad Good

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, 56, 28, 79, 15, 64, 18, 19]. This malicious task can be formalized into the following bi-level optimization problem:

$$\begin{aligned} &\max_{\mathcal{P}\in\mathcal{S}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathcal{L}(f_{\mathcal{P}}(\boldsymbol{x}), y)\right] \\ &\text{s.t. } f_{\mathcal{P}} \in \arg\min_{f} \sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{T}} \left[\mathcal{L}(f(\boldsymbol{x}_i+\boldsymbol{p}_i), y_i)\right]. \end{aligned} \tag{3}$$

Here, the adversary aims to maximize the natural risk of the model $f_{\mathcal{P}}$ (that is trained on the manipulated training set) by applying the generated perturbations $\mathcal{P} = \{\boldsymbol{p}_i\}_{i=1}^n$ into the original training set $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The commonly used feasible region is $\mathcal{S} = \{\{\boldsymbol{p}_i\}_{i=1}^n \mid \|\boldsymbol{p}_i\| \le \epsilon\}$.

Generally, solving Equation (3) is computationally prohibitive for neural networks [64, 18]. Thus, various heuristic methods are explored to resolve this problem. For example, a representative method called *hypocritical perturbations* [63, 64, 28, 19] is crafted as follows:

$$\min_{\|\boldsymbol{p}_i\|\le\epsilon} \mathcal{L}(f_{\text{craft}}(\boldsymbol{x}_i+\boldsymbol{p}_i), y_i), \tag{4}$$

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

$$\max_{\|\boldsymbol{p}_i\|\le\epsilon} \mathcal{L}(f_{\text{craft}}(\boldsymbol{x}_i+\boldsymbol{p}_i), y_i). \tag{5}$$

Tao et al. [64] and Fowl et al. [18] independently 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:

$$\max_{\mathcal{P}\in\mathcal{S}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\Delta} \mathcal{L}(f_{\mathcal{P}}(\boldsymbol{x}+\boldsymbol{\delta}), y)\right], \tag{6}$$

where $f_{\mathcal{P}}$ 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.

Justification. This work focuses on adding bounded pertubrations 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 insights.

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 ϵ may not be sufficient to provide ϵ-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 $(\boldsymbol{x}, y)$ sampled from a Gaussian mixture distribution $\mathcal{D}$ as follows:

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

where η is much smaller than 1 (i.e., $0 < \eta \ll 1$). Hence, samples from $\mathcal{D}$ 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:

$$f_{\text{nat}}(\boldsymbol{x}) \coloneqq \text{sign}(\boldsymbol{w}_{\text{nat}}^\top \boldsymbol{x}), \text{ where } \boldsymbol{w}_{\text{nat}} \coloneqq [1, \eta, \ldots, \eta], \tag{8}$$

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:

$$f_{\text{rob}}(\boldsymbol{x}) \coloneqq \text{sign}(\boldsymbol{w}_{\text{rob}}^\top \boldsymbol{x}), \text{ where } \boldsymbol{w}_{\text{rob}} \coloneqq [1, 0, \ldots, 0], \tag{9}$$

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 ϵ 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_{nat} (8) and the robust classifier f_{rob} (9).

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

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

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* [42, 64, 18] and the *hypocritical perturbation* [63, 28, 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 ϵ 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 ϵ. The adversary may choose to shift each feature towards $-y$. Hence, the learner would see input-label pairs $(\boldsymbol{x}, y)$ sampled i.i.d. from a training distribution $\mathcal{T}_{\text{adv}}$ as follows:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad x_1 \sim \mathcal{N}((1-\epsilon)y, \sigma^2), \quad x_2, \ldots, x_{d+1} \overset{i.i.d}{\sim} \mathcal{N}((\eta-\epsilon)y, \sigma^2), \tag{10}$$

where each feature of the samples from $\mathcal{T}_{\text{adv}}$ is adversarially perturbed by a moderate ϵ. While these samples are deviate significantly from the original distribution $\mathcal{D}$ (7), adversarial training on them using a defense budget ϵ 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 $\mathcal{T}_{adv}$ (10) is moderate such that $\eta/2 \leq \epsilon < 1/2$. Then, the optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on $\mathcal{T}_{adv}$ with a defense budget ϵ is equivalent to the robust classifier (9).*

This theorem indicates that the adversarial perturbation is harmless: ϵ-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 $(\boldsymbol{x}, y)$ sampled i.i.d. from a training distribution $\mathcal{T}_{\text{hyp}}$ as follows[1]:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad x_1 \sim \mathcal{N}((1+\epsilon)y, \sigma^2), \quad x_2, \ldots, x_{d+1} \overset{i.i.d}{\sim} \mathcal{N}((\eta+\epsilon)y, \sigma^2), \tag{11}$$

where each feature of the samples from $\mathcal{T}_{\text{hyp}}$ is reinforced by a magnitude of ϵ. 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.

Theorem 2 (Hypocritical perturbation is harmful)**.** *The optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on the perturbed data $\mathcal{T}_{hyp}$ (11) with a defense budget ϵ 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 $\mathcal{T}_{\text{hyp}}$ (11) are actually hypocritical. Therefore, the adversary is highly motivated to hide such hypocritical features in the training data,

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

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 $\mathcal{T}_{\text{hyp}}$) actually has η-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 0.25ϵ-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 $\mathcal{T}_{\text{hyp}}$ (11). In this case, we show that a larger defense budget is necessary in the following theorem proved in Appendix C.4.

Theorem 3 ($\epsilon + \eta$ is necessary)**.** *The optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on the perturbed data $\mathcal{T}_{hyp}$ (11) 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 ϵ-bounded hypocritical perturbations, the learner needs a slightly larger defense budget $\epsilon + \eta$ to ensure ϵ-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 ϵ, training models to minimize the adversarial risk with a defense budget 2ϵ on the perturbed data is sufficient to ensure ϵ-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 2ϵ, 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 ℓ_∞ adversary with an *attack budget* $\epsilon_a = 8/255$ by following [28, 79, 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 previous studies [81, 69, 50]. By convention, the *defense budget* is equal to the attack budget, i.e., $\epsilon_d = 8/255$. 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) [79], and Adversarial Poisoning (whose crafting model is simply

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

Attack	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
None (clean)	82.17	56.63	50.63	50.35	49.37	46.99
DeepConfuse [16]	81.25	54.14	48.25	48.02	47.34	44.79
Unlearnable Examples [28]	83.67	57.51	50.74	50.31	49.81	47.25
NTGA [79]	82.99	55.71	49.17	48.82	47.96	45.36
Adversarial Poisoning [18]	**77.35**	53.93	49.95	49.76	48.35	46.13
Hypocritical Perturbation (ours)	88.07	**47.93**	**37.61**	**36.96**	**38.58**	**35.44**

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

Dataset	Attack	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
SVHN	None	93.95	71.83	57.15	56.02	54.93	50.50
	Adv.	**87.50**	**56.12**	46.71	46.32	45.70	42.48
	Hyp.	96.06	59.41	**38.17**	**37.29**	**40.54**	**35.43**
CIFAR-100	None	56.15	31.50	28.38	28.28	26.53	24.30
	Adv.	**52.14**	28.59	26.19	26.09	24.36	22.71
	Hyp.	62.22	**26.38**	**21.51**	**21.13**	**21.13**	**18.74**
Tiny-ImageNet	None	**49.34**	25.67	22.99	22.86	20.67	18.54
	Adv.	49.52	22.93	20.01	19.91	18.75	16.83
	Hyp.	55.92	**20.21**	**15.61**	**15.26**	**14.99**	**12.53**

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

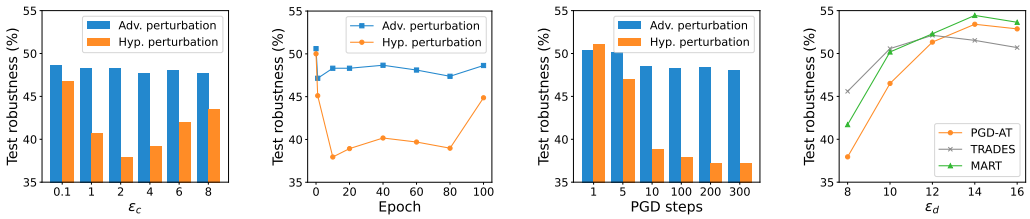
Attack	VGG-16	GoogLeNet	DenseNet-121	MobileNetV2	WideResNet-28-10
None	47.37 (77.15)	50.67 (83.03)	49.92 (80.08)	48.51 (80.83)	53.91 (85.81)
Adv.	44.70 (73.24)	47.72 (79.34)	48.00 (78.17)	45.90 (74.61)	51.01 (82.43)
Hyp.	**34.34** (87.20)	**37.03** (87.61)	**37.58** (88.04)	**35.58** (87.04)	**41.07**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

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, CW_∞ (ℓ_∞ 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 our implementation of stability attacks 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, which reliably degrades 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 [80], 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.



(a) Crafting budget (b) Crafting epoch (c) Number of PGD steps (d) Large defense budget

Figure 2: The ablation study experiments on CIFAR-10. Test robustness (%) is evaluated by PGD-20.

Table 5: Test robustness (%) of various adaptive defenses on the hypocritically perturbed CIFAR-10.

Defense	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
PGD-AT ($\epsilon_d = 8/255$)	88.07	47.93	37.61	36.96	38.58	35.44
+ Random Noise	87.62	47.46	38.35	37.90	39.07	36.25
+ Gaussian Smoothing	83.95	50.96	42.80	42.34	42.41	40.07
+ Cutout	**88.26**	49.23	39.77	39.25	40.38	37.61
+ AutoAugment	86.24	48.87	40.19	39.65	37.66	35.07
PGD-AT ($\epsilon_d = 14/255$)	80.00	56.86	52.92	52.83	**50.36**	**48.63**
TRADES ($\epsilon_d = 12/255$)	79.63	55.73	51.77	51.63	48.68	47.83
MART ($\epsilon_d = 14/255$)	77.29	**57.10**	**53.82**	**53.71**	49.03	47.67

Table 6: Test robustness (%) of PGD-AT by adjusting the amount of clean data included in the manipulated CIFAR-10. Test robustness (%) is evaluated by PGD-20. Values in parenthesis denote the accuracy on natural test data.

Attack\Clean proportion	0.1	0.2	0.4	0.6	0.8
None (clean subset)	30.65 (63.90)	37.99 (70.99)	44.95 (77.11)	47.17 (80.33)	49.78 (81.60)
Adversarial Perturbation	48.33 (77.71)	48.23 (76.94)	49.68 (78.54)	50.15 (82.46)	51.21 (82.03)
Hypocritical Perturbation	**41.51** (87.49)	**43.66** (88.30)	**46.98** (86.46)	**49.20** (85.29)	**50.56** (82.72)

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 [81] and MART [69] 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 attack, where the crafting budget $\epsilon_c = 2/255$ 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

effective attacks in this way. This is mainly because they adopted adversarial perturbations as poisons, which, as we observed, are inferior in degrading test robustness.

Ablation on defense budget. As discussed in Section 4, we are motivated to find the appropriate defense budget ϵ_d in the range $[\epsilon \sim 2\epsilon]$. Figure 2(d) shows that the optimal defense budgets against the proposed attack for PGD-AT, TRADES, and MART are $14/255$, $12/255$, and $14/255$, 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 pertubrations as stability attacks.

Effect on natural training. As a sanity check, we include the test accuracy of naturally trained models on CIFAR-10 in Appendix B Table 12. It shows that without adversarial training, the test robustness of the models becomes very poor. Thus, the goal of stability attacks is immediately achieved. On the other hand, We find that our HFs can degrade 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, 72, 81, 65, 46, 74, 84, 62]. Usually, adversarial training using a defense budget ϵ is expected to improve model robustness against ϵ-bounded adversarial examples. Thus, to break this defense, a direct way is to enlarge the typical ϵ-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 ϵ-ball.

Data poisoning. Data poisoning attacks, which manipulate training data to cause the resulting models to fail during inference [3], can be divided into *availability attacks* (to degrade overall test performance) [4, 75, 41, 47, 18] and *integrity attacks* (to cause specific misclassifications) [32, 10, 55, 85, 21, 54]. While the stability attacks considered 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 ϵ-robustness, while backdoor attacks mainly focus on embedding exploits (that can be invoked by pre-specified triggers) into naturally trained models [22, 51, 68]. Second, stability attacks only perturb the inputs slightly, while many works on backdoor attacks require mislabeling [25, 37, 45, 36]. Thus, backdoor defenses [7, 73, 35] 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 ϵ is insufficient under the threat. To resist it, we suggest that practitioners should consider a larger defense budget of no more than 2ϵ. Our theoretical analysis also explains why hypocritical perturbations work as stability attacks—they can reinforce the non-robust features and mislead the learner. Experiments demonstrate that hypocritical perturbations are harmful to conventional adversarial training on benchmark datasets, and enlarging the defense budget is essential for mitigating hypocritical pertubrations. Future work includes relaxing the assumption that the adversary perturbs all the entire training set and designing more effective stability attacks against adversarial training.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018.

[3] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[4] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *ICML*, 2012.

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML-PKDD*, 2013.

[6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. In *NeurIPS Deep Learning Symposium*, 2016.

[7] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP*, 2021.

[8] Nicholas Carlini. A critique of the deepsec platform for security analysis of deep learning models. *arXiv preprint arXiv:1905.07112*, 2019.

[9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017.

[10] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[11] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[12] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021. URL `https://openreview.net/forum?id=SSKZPJCt7B`.

[13] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *KDD*, 2004.

[14] Edgar Dobriban, Hamed Hassani, David Hong, and Alexander Robey. Provable tradeoffs in adversarially robust classification. *arXiv preprint arXiv:2006.05161*, 2020.

[15] Ivan Evtimov, Ian Covert, Aditya Kusupati, and Tadayoshi Kohno. Disrupting model training with adversarial shortcuts. In *ICML 2021 Workshop*, 2021.

[16] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *NeurIPS*, 2019.

[17] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv preprint arXiv:2103.02683*, 2021.

[18] Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojtek Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *NeurIPS*, 2021.

[19] Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=baUQQPwQiAg`.

[20] Yinghua Gao, Dongxian Wu, Jingfeng Zhang, Shu-Tao Xia, Gang Niu, and Masashi Sugiyama. Does adversarial robustness really imply backdoor vulnerability?, 2022. URL `https://openreview.net/forum?id=nG4DkcHDw_`.

[21] Jonas Geiping, Liam H Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *ICLR*, 2021.

[22] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

[23] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[24] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[25] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[28] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2021.

[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.

[30] Adel Javanmard and Mahdi Soltanolkotabi. Precise statistical analysis of classification accuracies for adversarial training. *arXiv preprint arXiv:2010.11213*, 2020.

[31] Junho Kim, Byung-Kwan Lee, and Yong Man Ro. Distilling robust and non-robust features in adversarial examples by information bottleneck. In *NeurIPS*, 2021.

[32] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[33] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[34] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[35] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021.

[36] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *CVPR*, 2021.

[37] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

[38] Zhuoran Liu, Zhengyu Zhao, Alex Kolmus, Tijn Berns, Twan van Laarhoven, Tom Heskes, and Martha Larson. Going grayscale: The road to understanding and improving unlearnable examples. *arXiv preprint arXiv:2111.13244*, 2021.

[39] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[40] Mohammad Mehrabi, Adel Javanmard, Ryan A Rossi, Anup Rao, and Tung Mai. Fundamental tradeoffs in distributionally adversarial training. In *ICML*, 2021.

[41] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

[42] Preetum Nakkiran. A discussion of 'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*, 2019. doi: 10.23915/distill.00019.5. https://distill.pub/2019/advex-bugs-discussion/response-5.

[43] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[44] James Newsome, Brad Karp, and Dawn Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, 2006.

[45] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

[46] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *ICLR*, 2021.

[47] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Accumulative poisoning attacks on real-time data. In *NeurIPS*, 2021.

[48] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. *arXiv preprint arXiv:2202.10103*, 2022.

[49] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020.

[50] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

[51] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI*, 2020.

[52] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[53] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

[54] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *ICML*, 2021.

[55] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*, 2018.

[56] Juncheng Shen, Xiaolei Zhu, and De Ma. Tensorclog: An imperceptible poisoning attack on deep neural network applications. *IEEE Access*, 2019.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[58] Jacob Springer, Melanie Mitchell, and Garrett Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. In *NeurIPS*, 2021.

[59] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018.

[60] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[61] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[62] Jihoon Tack, Sihyun Yu, Jongheon Jeong, Minseon Kim, Sung Ju Hwang, and Jinwoo Shin. Consistency regularization for adversarial robustness. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

[63] Lue Tao, Lei Feng, Jinfeng Yi, and Songcan Chen. With false friends like these, who can notice mistakes? *arXiv preprint arXiv:2012.14738*, 2020.

[64] Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *NeurIPS*, 2021.

[65] Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *NeurIPS*, 2019.

[66] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *ICML*, 2020.

[67] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[68] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[69] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

[70] Zhirui Wang, Yifei Wang, and Yisen Wang. Fooling adversarial training with induction noise, 2022. URL `https://openreview.net/forum?id=4o1xPXaS4X`.

[71] Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung Brandon Wu. On the trade-off between adversarial and backdoor robustness. In *NeurIPS*, 2020.

[72] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

[73] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021.

[74] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

[75] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *ICML*, 2015.

[76] 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.

[77] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *NeurIPS*, 2020.

[78] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Indiscriminate poisoning attacks are shortcuts. *arXiv preprint arXiv:2111.00898*, 2021.

[79] Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *ICML*, 2021.

[80] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[81] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[82] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

[83] Jingfeng Zhang, Xilie Xu, Bo Han, Tongliang Liu, Gang Niu, Lizhen Cui, and Masashi Sugiyama. Noilin: Do noisy labels always hurt adversarial training? *arXiv preprint arXiv:2105.14676*, 2021.

[84] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

[85] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *ICML*, 2019.

[86] Jianing Zhu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, Hongxia Yang, Mohan Kankanhalli, and Masashi Sugiyama. Understanding the interaction of adversarial training with noisy labels. *arXiv preprint arXiv:2102.03482*, 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]

A Additional Related Work

In this part, we discuss several independent (or concurrent) works that are related to this work.

Zhu et al. [86] study the effect of conventional adversarial training on differentiating noisy labels, while Zhang et al. [83] show that deliberately injected noisy labels may serve as a regularization that alleviates robust overfitting. Our results focus on the clean-label setting and provide evidence that conventional adversarial training can be hindered without modifying the labels.

Yu et al. [78] suggest explaining the success of availability attacks from the perspective of shortcuts. They further adopt pre-trained models to extract useful features for mitigating model reliance on the shortcuts. This direction is orthogonal to ours.

Liu et al. [38] improve the effectiveness of unlearnable examples [28] by generating grayscale perturbations and using data augmentations. They also conclude that conventional adversarial training will prevent a drop in accuracy measured both on clean images and adversarial images. Contrary to them, we show that, both theoretically and empirically, conventional adversarial training can be hindered by hypocritical perturbations, and we further analyze the necessity of enlarging the defense budget to resist stability attacks.

Gao et al. [20] revisit the trade-off between adversarial robustness and backdoor robustness [71]. They conclude that backdoor attacks are ineffective when the defense budget of adversarial training surpasses the trigger magnitude. In contrast, our results indicate that stability attacks are still harmful to adversarial training when the defense budget is not large enough. In a simple statistical setting, a defense budget $\epsilon + \eta$ is necessary (where η is a positive number). In the general case, a defense budget of 2ϵ is sufficient. In our experiments, a defense budget of about $1.5\epsilon \sim 1.75\epsilon$ provides the best empirical ϵ-robustness.

Wang et al. [70] argue that it is necessary to use robust features for compromising adversarial training. To this end, they adopt a relatively large attack budget $\epsilon_a = 32/255$ for crafting their poisons (they use one type of adversarial perturbations), and show that their poisons can decrease the performance of the models trained using smaller defense budgets (such as $\epsilon_d = 8/255$ and $\epsilon_d = 16/255$). In contrast, we focus on a more realistic setting that does not require a larger attack budget. We demonstrate that it is possible to hinder adversarial training when $\epsilon_a = \epsilon_d$. Furthermore, we provide both theoretical and empirical results showing how to adapt the defense to maintain robustness.

Fu et al. [19] explore how to protect data privacy against adversarial training. The main purpose of their poisons is to compromise adversarial training by requiring the perturbation budget of their poisons to be larger than that of adversarial training. In this way, they show that the natural accuracy of the adversarially trained models can be largely decreased, let alone robust accuracy. From this perspective, our work is complementary to theirs. We pursue to not increase the attack budget of stability attacks, keeping it as small as possible. We successfully demonstrate that stability attacks are still harmful to conventional adversarial training without enlarging the attack budget. This makes the threat of stability attacks more insidious than that of Fu et al. [19].

On the other hand, we find that our implementation of stability attacks using hypocritical perturbations has some similarities to the robust unlearnable examples in Fu et al. [19]. Specifically, although the robust unlearnable examples are generated via a complicated min-min-max optimization process [19], we notice that their noise generator can be viewed as an adversarially trained model. This implies that the robust error-minimizing (REM) noise [19] might be useful in demonstrating the feasibility of stability attacks. To verify this, we run the source code from the authors with default hyperparameters[2], and compare our crafted hypocritical perturbation with their generated noise under the setting of stability attacks. For a fair comparison, here we apply a very simple trick called EOT [2] in our method, since the trick is also used by REM [19]. The additional time cost of the EOT trick is very small and negligible.

Our experimental results, shown in Table 7, demonstrate that the robust error-minimizing noise is also effective as stability attacks, though it was originally proposed as a delusive attack. It is noteworthy that the robust accuracy is not evaluated in [19]. In this sense, the effectiveness of REM as an stability attack can be regarded as one of our novel findings.

Importantly, our method outperforms REM in terms of the robust accuracy against AutoAttack. Since AutoAttack is the most reliable evaluation metric of model robustness among the test-time attacks [11], this result indicates that our method is reliably more effective than REM in degrading model robustness. It is also noteworthy that our method is significantly more efficient than REM, as shown in Table 8. We note that the efficiency of our method is largely due to the fact that our crafting model is fast to train. Specifically, the time cost of training our crafting model is only 0.3 hours, while it takes 20.8 hours for REM. That is, our crafting model is nearly 70 times faster to train than that of REM. In short, our method is not only more effective, but also more efficient, than REM as a stability attack.

[2] `https://github.com/fshp971/robust-unlearnable-examples`

Table 7: Comparison with REM [19]: Test robustness (%) of PGD-AT using a defense budget $\epsilon_d = 8/255$ on CIFAR-10. We report mean and standard deviation over 3 random runs.

Attack	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
None (clean)	82.17 ± 0.71	56.63 ± 0.54	50.63 ± 0.56	50.35 ± 0.59	49.37 ± 0.57	46.99 ± 0.62
DeepConfuse [16]	81.25 ± 1.52	54.14 ± 0.63	48.25 ± 0.40	48.02 ± 0.40	47.34 ± 0.05	44.79 ± 0.36
Unlearnable Examples [28]	83.67 ± 0.86	57.51 ± 0.31	50.74 ± 0.37	50.31 ± 0.38	49.81 ± 0.24	47.25 ± 0.32
NTGA [79]	82.99 ± 0.40	55.71 ± 0.36	49.17 ± 0.27	48.82 ± 0.30	47.96 ± 0.16	45.36 ± 0.32
Adversarial Poisoning [18]	$\mathbf{77.35 \pm 0.43}$	53.93 ± 0.02	49.95 ± 0.11	49.76 ± 0.08	48.35 ± 0.04	46.13 ± 0.18
REM [19]	85.63 ± 1.05	$\mathbf{42.86 \pm 1.09}$	35.40 ± 0.04	35.11 ± 0.09	$\mathbf{35.24 \pm 0.33}$	33.09 ± 0.24
Hypocritical Perturbation (ours)	87.60 ± 0.45	45.00 ± 0.77	$\mathbf{34.89 \pm 0.36}$	$\mathbf{34.27 \pm 0.36}$	36.28 ± 0.38	$\mathbf{32.79 \pm 0.37}$

Table 8: Comparison with REM [19]: Time cost (min) of poisoning CIFAR-10.

Method	Training the crafting model	Perturbation generation	Total
REM [19]	1252.4	98.1	1350.5
Hypocritical Perturbation (ours)	**18.5**	**17.3**	**35.8**

B Omitted Tables

Table 9: Full table of Table 2: Test robustness (%) of PGD-AT using a defense budget $\epsilon_d = 8/255$ on CIFAR-10. We report mean and standard deviation over 3 random runs.

Attack	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
None (clean)	82.17 ± 0.71	56.63 ± 0.54	50.63 ± 0.56	50.35 ± 0.59	49.37 ± 0.57	46.99 ± 0.62
DeepConfuse [16]	81.25 ± 1.52	54.14 ± 0.63	48.25 ± 0.40	48.02 ± 0.40	47.34 ± 0.05	44.79 ± 0.36
Unlearnable Examples [28]	83.67 ± 0.86	57.51 ± 0.31	50.74 ± 0.37	50.31 ± 0.38	49.81 ± 0.24	47.25 ± 0.32
NTGA [79]	82.99 ± 0.40	55.71 ± 0.36	49.17 ± 0.27	48.82 ± 0.30	47.96 ± 0.16	45.36 ± 0.32
Adversarial Poisoning [18]	$\mathbf{77.35 \pm 0.43}$	53.93 ± 0.02	49.95 ± 0.11	49.76 ± 0.08	48.35 ± 0.04	46.13 ± 0.18
Hypocritical Perturbation (ours)	88.07 ± 1.10	$\mathbf{47.93 \pm 1.88}$	$\mathbf{37.61 \pm 0.77}$	$\mathbf{36.96 \pm 0.61}$	$\mathbf{38.58 \pm 1.15}$	$\mathbf{35.44 \pm 0.77}$

Table 10: Full table of Table 3: Test robustness (%) of PGD-AT using a defense budget $\epsilon_d = 8/255$ across different datasets. We report mean and standard deviation over 3 random runs.

Dataset	Attack	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
SVHN	None	93.95 ± 0.21	71.83 ± 1.10	57.15 ± 0.31	56.02 ± 0.33	54.93 ± 0.19	50.50 ± 0.44
	Adv.	$\mathbf{87.50 \pm 0.30}$	$\mathbf{56.12 \pm 0.33}$	46.71 ± 0.25	46.32 ± 0.26	45.70 ± 0.27	42.48 ± 0.21
	Hyp.	96.06 ± 0.01	59.41 ± 0.07	$\mathbf{38.17 \pm 0.19}$	$\mathbf{37.29 \pm 0.21}$	$\mathbf{40.54 \pm 0.27}$	$\mathbf{35.43 \pm 0.29}$
CIFAR-100	None	56.15 ± 0.17	31.50 ± 0.16	28.38 ± 0.39	28.28 ± 0.40	26.53 ± 0.27	24.30 ± 0.31
	Adv.	$\mathbf{52.14 \pm 0.34}$	28.59 ± 0.12	26.19 ± 0.11	26.09 ± 0.12	24.36 ± 0.09	22.71 ± 0.11
	Hyp.	62.22 ± 0.11	$\mathbf{26.38 \pm 0.11}$	$\mathbf{21.51 \pm 0.06}$	$\mathbf{21.13 \pm 0.02}$	$\mathbf{21.13 \pm 0.23}$	$\mathbf{18.74 \pm 0.10}$
Tiny-ImageNet	None	$\mathbf{49.34 \pm 2.61}$	25.67 ± 0.92	22.99 ± 0.37	22.86 ± 0.36	20.67 ± 0.69	18.54 ± 0.61
	Adv.	49.52 ± 0.19	22.93 ± 0.38	20.01 ± 0.24	19.91 ± 0.24	18.75 ± 0.19	16.83 ± 0.25
	Hyp.	55.92 ± 1.95	$\mathbf{20.21 \pm 0.84}$	$\mathbf{15.61 \pm 0.31}$	$\mathbf{15.26 \pm 0.26}$	$\mathbf{14.99 \pm 0.73}$	$\mathbf{12.53 \pm 0.57}$

Table 11: Full table of Table 5: Test robustness (%) of various adaptive defenses on the hypocritically perturbed CIFAR-10. We report mean and standard deviation over 3 random runs.

Defense	Natural	FGSM	PGD-20	PGD-100	CW_∞	AutoAttack
PGD-AT ($\epsilon_d = 8/255$)	88.07 ± 1.10	47.93 ± 1.88	37.61 ± 0.77	36.96 ± 0.61	38.58 ± 1.15	35.44 ± 0.77
+ Random Noise	87.62 ± 0.07	47.46 ± 0.08	38.35 ± 0.08	37.90 ± 0.07	39.07 ± 0.20	36.25 ± 0.14
+ Gaussian Smoothing	83.95 ± 0.27	50.96 ± 0.24	42.80 ± 0.40	42.34 ± 0.38	42.41 ± 0.19	40.07 ± 0.29
+ Cutout	$\mathbf{88.26 \pm 0.15}$	49.23 ± 0.42	39.77 ± 0.26	39.25 ± 0.25	40.38 ± 0.25	37.61 ± 0.35
+ AutoAugment	86.24 ± 1.14	48.87 ± 1.01	40.19 ± 0.67	39.65 ± 0.72	37.66 ± 0.88	35.07 ± 0.88
PGD-AT ($\epsilon_d = 14/255$)	80.00 ± 1.91	56.86 ± 1.42	52.92 ± 0.86	52.83 ± 0.86	$\mathbf{50.36 \pm 1.11}$	$\mathbf{48.63 \pm 0.93}$
TRADES ($\epsilon_d = 12/255$)	79.63 ± 0.06	55.73 ± 0.04	51.77 ± 0.15	51.63 ± 0.15	48.68 ± 0.06	47.83 ± 0.02
MART ($\epsilon_d = 14/255$)	77.29 ± 0.87	$\mathbf{57.10 \pm 0.57}$	$\mathbf{53.82 \pm 0.36}$	$\mathbf{53.71 \pm 0.34}$	49.03 ± 0.47	47.67 ± 0.51

Table 12: Test accuracy (%) of natural training on CIFAR-10. We report mean and standard deviation over 3 random runs.

Attack	Natural	PGD-20
None (clean)	94.23 ± 0.14	0.00 ± 0.00
DeepConfuse	17.22 ± 0.64	0.00 ± 0.00
Unlearnable Examples	22.72 ± 0.51	0.00 ± 0.00
NTGA	11.15 ± 0.27	0.00 ± 0.00
Adversarial Poisoning	$\mathbf{8.60 \pm 1.39}$	0.00 ± 0.00
Hypocritical Perturbation	75.92 ± 1.04	0.00 ± 0.00

C Proofs

In this section, we provide the proofs of our theoretical results in Section 3 and Section 4.

C.1 Proof of Proposition 1

Proposition 1 (restated). *Let $\epsilon = 2\eta$ and denote by $\mathcal{A}_{\text{adv}}(f)$ the adversarial accuracy, i.e., the probability of a classifier correctly predicting y on the data (7) under ℓ_∞ perturbations. Then, we have*

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

Proof. Recalling that in Equation (8), we have the natural classifier:

$$f_{\text{nat}}(\boldsymbol{x}) \coloneqq \text{sign}(\boldsymbol{w}_{\text{nat}}^\top \boldsymbol{x}), \text{ where } \boldsymbol{w}_{\text{nat}} \coloneqq [1, \eta, \ldots, \eta], \tag{12}$$

and in Equation (9), the robust classifier is defined as:

$$f_{\text{rob}}(\boldsymbol{x}) \coloneqq \text{sign}(\boldsymbol{w}_{\text{rob}}^\top \boldsymbol{x}), \text{ where } \boldsymbol{w}_{\text{rob}} \coloneqq [1, 0, \ldots, 0]. \tag{13}$$

Then, the adversarial accuracy of the natural classifier on the data $\mathcal{D}$ (7) is

$$\begin{aligned}
\mathcal{A}_{\text{adv}}(f_{\text{nat}}) &= 1 - \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\{\exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon, f_{\text{nat}}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= 1 - \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left\{\min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [y \cdot f_{\text{nat}}(\boldsymbol{x}+\boldsymbol{\delta})] < 0\right\} \\
&= 1 - \Pr\left\{\min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}\left[y \cdot \left(1 \cdot \left(\mathcal{N}(y,\sigma^2) + \delta_1\right) + \sum_{i=2}^{d+1} \eta \cdot \left(\mathcal{N}(y\eta,\sigma^2) + \delta_i\right)\right)\right] < 0\right\} \\
&\leq 1 - \Pr\left\{y \cdot \left(1 \cdot \left(\mathcal{N}(y,\sigma^2)\right) + \sum_{i=2}^{d+1} \eta \cdot \left(\mathcal{N}(y\eta,\sigma^2) - \epsilon\right)\right) < 0\right\} \\
&= 1 - \Pr\left\{\mathcal{N}(1,\sigma^2) + \eta \sum_{i=2}^{d+1} \mathcal{N}(\eta-\epsilon,\sigma^2) < 0\right\} \\
&= \Pr\left\{\mathcal{N}(1,\sigma^2) + \eta \sum_{i=2}^{d+1} \mathcal{N}(\eta-\epsilon,\sigma^2) > 0\right\} \\
&= \Pr\left\{\mathcal{N}(0,1) < \frac{1-d\eta^2}{\sigma\sqrt{1+d\eta^2}}\right\}.
\end{aligned} \tag{14}$$

Similarly, the adversarial accuracy of the robust classifier on the data $\mathcal{D}$ (7) is

$$\begin{aligned}
\mathcal{A}_{\text{adv}}(f_{\text{rob}}) &= 1 - \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \{\exists \|\boldsymbol{\delta}\|_\infty \le \epsilon, f_{\text{rob}}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= 1 - \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} [y \cdot f_{\text{rob}}(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \right\} \\
&= 1 - \Pr \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \left[y \cdot \left(1 \cdot \left(\mathcal{N}(y,\sigma^2) + \delta_1\right)\right)\right] < 0 \right\} \\
&= 1 - \Pr \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \left[\mathcal{N}(1,\sigma^2) + \delta_1 \right] < 0 \right\} \\
&= 1 - \Pr \left\{ \mathcal{N}(1,\sigma^2) - \epsilon < 0 \right\} \\
&= \Pr \left\{ \mathcal{N}(1-\epsilon,\sigma^2) > 0 \right\} \\
&= \Pr \left\{ \mathcal{N}(0,1) < \frac{1-2\eta}{\sigma} \right\}.
\end{aligned} \tag{15}$$

□

C.2 Proof of Theorem 1

The following theorems rely on the analytical solution of optimal linear ℓ_∞-robust classifier on mixture Gaussian distributions. Concretely, the optimization problem is to minimize the adversarial risk on a distribution $\widehat{\mathcal{D}}$ with a defense budget $\hat{\epsilon}$:

$$\min_f \mathcal{R}^{\hat{\epsilon}}_{\text{adv}}(f, \widehat{\mathcal{D}}), \quad \text{where} \quad \mathcal{R}^{\hat{\epsilon}}_{\text{adv}}(f, \widehat{\mathcal{D}}) := \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}} \left[\max_{\|\boldsymbol{\xi}\|_\infty \le \hat{\epsilon}} \mathbb{1}\left(\text{sign}(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\xi}) + b) \neq y \right) \right], \tag{16}$$

where $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$, and $\mathbb{1}(\cdot)$ denotes the indicator function.

We note that optimal linear robust classifiers have been obtained for certain data distributions in previous work [67, 29, 14, 30, 76, 64]. Here, our goal is to establish similar optimal linear robust classifiers for the classification tasks in our setting. We only employ linear classifiers, since it is highly nontrivial to consider non-linearity for adversarial training on mixture Gaussian distributions [14].

Lemma 1. *Assume that the adversarial perturbation in data $\mathcal{T}_{adv}$ (10) is moderate such that $\eta/2 \le \epsilon < 1/2$. Then, minimizing the adversarial risk (16) on the data $\mathcal{T}_{adv}$ with a defense budget ϵ can result in a classifier that assigns 0 weight to the features x_i for $i \ge 2$.*

Proof. We prove the lemma by contradiction.

The goal is to minimize the adversarial risk on the distribution $\mathcal{T}_{\text{adv}}$, which can be written as follows:

$$\begin{aligned}
\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}_{\text{adv}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \{\exists \|\boldsymbol{\delta}\|_\infty \le \epsilon, f(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} [y \cdot f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \right\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} [f(\boldsymbol{x}+\boldsymbol{\delta})] > 0 \mid y = -1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \{y = -1\} \\
&\quad + \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} [f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \mid y = +1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{adv}}} \{y = +1\} \\
&= \underbrace{\Pr \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \left[w_1(\mathcal{N}(\epsilon-1,\sigma^2) + \delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(\epsilon-\eta,\sigma^2) + \delta_i) + b \right] > 0 \right\}}_{\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}})} \cdot \frac{1}{2} \\
&\quad + \underbrace{\Pr \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \left[w_1(\mathcal{N}(1-\epsilon,\sigma^2) + \delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(\eta-\epsilon,\sigma^2) + \delta_i) + b \right] < 0 \right\}}_{\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(+1)}_{\text{adv}})} \cdot \frac{1}{2}
\end{aligned} \tag{17}$$

Consider an optimal solution $\boldsymbol{w}$ in which $w_i > 0$ for some $i \geq 2$. Then, we have

$$\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}}) = \Pr\left\{ \underbrace{\sum_{j \neq i} \max_{\|\delta_j\| \leq \epsilon} \left[w_j(\mathcal{N}(\epsilon - [\boldsymbol{w}_{\text{nat}}]_j, \sigma^2) + \delta_j) + b \right]}_{\mathbb{A}} + \underbrace{\max_{\|\delta_i\| \leq \epsilon} \left[w_i(\mathcal{N}(\epsilon - \eta, \sigma^2) + \delta_i) \right]}_{\mathbb{B}} > 0 \right\}, \tag{18}$$

where $\boldsymbol{w}_{\text{nat}} \coloneqq [1, \eta, \ldots, \eta]$ as in Equation (8). Since $w_i > 0$, $\mathbb{B}$ is maximized when $\delta_i = \epsilon$. Thus, the contribution of terms depending on w_i to $\mathbb{B}$ is a normally-distributed random variable with mean $2\epsilon - \eta$. Since $2\epsilon - \eta \geq 0$, setting w_i to zero can only decrease the risk. This contradicts the optimality of $\boldsymbol{w}$. Formally,

$$\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}}) = \Pr\left\{ \mathbb{A} + w_i \mathcal{N}(2\epsilon - \eta, \sigma^2) > 0 \right\} > \Pr\left\{ \mathbb{A} > 0 \right\}. \tag{19}$$

We can also assume $w_i < 0$ and similar contradiction holds. Therefore, minimizing the adversarial risk on $\mathcal{T}_{\text{adv}}$ leads to $w_i = 0$ for $i \geq 2$. □

Lemma 2. *Assume that the adversarial perturbation in data $\mathcal{T}_{adv}$ (10) is moderate such that $\eta/2 \leq \epsilon < 1/2$. Then, minimizing the adversarial risk (16) on the data $\mathcal{T}_{adv}$ with a defense budget ϵ results in a classifier that assigns a positive weight to the feature x_1.*

Proof. We prove the lemma by contradiction.

The goal is to minimize the adversarial risk on the distribution $\mathcal{T}_{\text{adv}}$, which has been written in Equation (17).

Consider an optimal solution $\boldsymbol{w}$ in which $w_1 \leq 0$. Then, we have

$$\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}}) = \Pr\left\{ \underbrace{\sum_{j=2}^{d+1} \max_{\|\delta_j\| \leq \epsilon} \left[w_j(\mathcal{N}(\epsilon - \eta, \sigma^2) + \delta_j) + b \right]}_{\mathbb{C}} + \underbrace{\max_{\|\delta_1\| \leq \epsilon} \left[w_1(\mathcal{N}(\epsilon - 1, \sigma^2) + \delta_1) \right]}_{\mathbb{D}} > 0 \right\}. \tag{20}$$

Since $w_1 \leq 0$, $\mathbb{D}$ is maximized when $\delta_1 = -\epsilon$. Thus, the contribution of the term depending on w_1 to $\mathbb{D}$ is a normally-distributed random variable with mean -1. Since the mean is negative, setting w_1 to be positive can decrease the risk. This contradicts the optimality of $\boldsymbol{w}$. Formally,

$$\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}}) = \Pr\left\{ \mathbb{C} + w_1 \mathcal{N}(-\eta, \sigma^2) > 0 \right\} > \Pr\left\{ \mathbb{C} + p\mathcal{N}(-\eta, \sigma^2) > 0 \right\}, \tag{21}$$

where $p > 0$ is any positive number. Therefore, minimizing the adversarial risk on $\mathcal{T}_{\text{adv}}$ leads to $w_1 > 0$. □

Theorem 1 (restated). *Assume that the adversarial perturbation in the training data $\mathcal{T}_{adv}$ (10) is moderate such that $\eta/2 \leq \epsilon < 1/2$. Then, the optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on $\mathcal{T}_{adv}$ with a defense budget ϵ is equivalent to the robust classifier (9).*

Proof. By Lemma 1 and Lemma 2, we have $w_1 > 0$ and $w_i = 0$ ($i \geq 2$) for an optimal linear ℓ_∞-robust classifier. Then, the adversarial risk on the distribution $\mathcal{T}_{\text{adv}}$ can be simplified by solving the inner maximization problem first. Formally,

$$\begin{aligned}
\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}_{\text{adv}}) &= \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \{ \exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon, f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y \} \\
&= \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [y \cdot f(\boldsymbol{x} + \boldsymbol{\delta})] < 0 \right\} \\
&= \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x} + \boldsymbol{\delta})] > 0 \mid y = -1 \right\} \cdot \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \{ y = -1 \} \\
&\quad + \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x} + \boldsymbol{\delta})] < 0 \mid y = +1 \right\} \cdot \Pr_{(\boldsymbol{x}, y) \sim \mathcal{T}_{\text{adv}}} \{ y = +1 \} \\
&= \Pr\left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(\epsilon - 1, \sigma^2) + \delta_1) + b \right] > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr\left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(1 - \epsilon, \sigma^2) + \delta_1) + b \right] < 0 \right\} \cdot \frac{1}{2} \\
&= \Pr\left\{ w_1 \mathcal{N}(2\epsilon - 1, \sigma^2) + b > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr\left\{ w_1 \mathcal{N}(1 - 2\epsilon, \sigma^2) + b < 0 \right\} \cdot \frac{1}{2},
\end{aligned} \tag{22}$$

which is equivalent to the natural risk on a mixture Gaussian distribution $\mathcal{D}_{\text{tmp}} : \boldsymbol{x} \sim \mathcal{N}(y \cdot \boldsymbol{\mu}_{\text{tmp}}, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{\mu}_{\text{tmp}} = (1 - 2\epsilon, 0, \ldots, 0)$. We note that the Bayes optimal classifier for $\mathcal{D}_{\text{tmp}}$ is $f_{\text{tmp}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{\mu}_{\text{tmp}}^\top \boldsymbol{x})$. Specifically, the natural risk

$$\begin{aligned}
\mathcal{R}^0_{\text{adv}}(f, \mathcal{D}_{\text{tmp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{tmp}}} \{f(\boldsymbol{x}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{tmp}}} \{y \cdot f(\boldsymbol{x}) < 0\} \\
&= \Pr\left\{w_1 \mathcal{N}(2\epsilon - 1, \sigma^2) + b > 0\right\} \cdot \frac{1}{2} \\
&\quad + \Pr\left\{w_1 \mathcal{N}(1 - 2\epsilon, \sigma^2) + b < 0\right\} \cdot \frac{1}{2},
\end{aligned} \tag{23}$$

which is minimized when $w_1 = 1 - 2\epsilon > 0$ and $b = 0$. That is, minimizing the adversarial risk $\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}_{\text{adv}})$ can lead to an optimal linear ℓ_∞-robust classifier $f_{\text{tmp}}(\boldsymbol{x})$. Meanwhile, $f_{\text{tmp}}(\boldsymbol{x})$ is equivalent to the robust classifier (9). This concludes the proof of the theorem. □

C.3 Proof of Theorem 2

Lemma 3. *Minimizing the adversarial risk (16) on the data $\mathcal{T}_{hyp}$ (11) with a defense budget ϵ results in a classifier that assigns positive weights to the features x_i for $i \geq 1$.*

Proof. We prove the lemma by contradiction.

The goal is to minimize the adversarial risk on the distribution $\mathcal{T}_{\text{hyp}}$, which can be written as follows:

$$\begin{aligned}
\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}_{\text{hyp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{hyp}} \{\exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon, f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{\min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [y \cdot f(\boldsymbol{x} + \boldsymbol{\delta})] < 0\right\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x} + \boldsymbol{\delta})] > 0 \mid y = -1\right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y = -1\} \\
&\quad + \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{\min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x} + \boldsymbol{\delta})] < 0 \mid y = +1\right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y = +1\} \\
&= \underbrace{\Pr\left\{\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(-1 - \epsilon, \sigma^2) + \delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(-\eta - \epsilon, \sigma^2) + \delta_i) + b\right] > 0\right\}}_{\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{hyp}})} \cdot \frac{1}{2} \\
&\quad + \underbrace{\Pr\left\{\min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(1 + \epsilon, \sigma^2) + \delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(\eta + \epsilon, \sigma^2) + \delta_i) + b\right] < 0\right\}}_{\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}^{(+1)}_{\text{hyp}})} \cdot \frac{1}{2}
\end{aligned} \tag{24}$$

Consider an optimal solution $\boldsymbol{w}$ in which $w_i \leq 0$ for some $i \geq 1$. Then, we have

$$\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{hyp}}) = \Pr\left\{\underbrace{\sum_{j \neq i} \max_{\|\delta_j\| \leq \epsilon} \left[w_j(\mathcal{N}(-[\boldsymbol{w}_{\text{nat}}]_j - \epsilon, \sigma^2) + \delta_j) + b\right]}_{\mathbb{G}} + \underbrace{\max_{\|\delta_i\| \leq \epsilon} \left[w_i(\mathcal{N}(-[\boldsymbol{w}_{\text{nat}}]_i - \epsilon, \sigma^2) + \delta_i)\right]}_{\mathbb{H}} > 0\right\}, \tag{25}$$

where $\boldsymbol{w}_{\text{nat}} := [1, \eta, \ldots, \eta]$ as in Equation (8). Since $w_i \leq 0$, $\mathbb{H}$ is maximized when $\delta_i = -\epsilon$. Thus, the contribution of terms depending on w_i to $\mathbb{H}$ is a normally-distributed random variable with mean $-[\boldsymbol{w}_{\text{nat}}]_i - 2\epsilon$. Since the mean is negative, setting w_i to be positive can decrease the risk. This contradicts the optimality of $\boldsymbol{w}$. Formally,

$$\mathcal{R}^\epsilon_{\text{adv}}(f, \mathcal{T}^{(-1)}_{\text{adv}}) = \Pr\left\{\mathbb{G} + w_i \mathcal{N}(-[\boldsymbol{w}_{\text{nat}}]_i - 2\epsilon, \sigma^2) > 0\right\} > \Pr\left\{\mathbb{G} + p\mathcal{N}(-[\boldsymbol{w}_{\text{nat}}]_i - 2\epsilon, \sigma^2) > 0\right\}, \tag{26}$$

where $p > 0$ is any positive number. Therefore, minimizing the adversarial risk on $\mathcal{T}_{\text{hyp}}$ leads to $w_i > 0$ for $i \geq 1$. □

Theorem 2 (restated). *The optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on the perturbed data $\mathcal{T}_{hyp}$ (11) with a defense budget ϵ is equivalent to the natural classifier (8).*

Proof. By Lemma 3, we have $w_i > 0$ for $i \geq 1$ for an optimal linear ℓ_∞-robust classifier. Then, we have

$$
\begin{aligned}
\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}_{\text{hyp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{\exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon, f(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [y \cdot f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \right\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x}+\boldsymbol{\delta})] > 0 \mid y = -1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y = -1\} \\
&\quad + \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} [f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \mid y = +1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y = +1\} \\
&= \Pr \left\{ \max_{\|\delta_1\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(-1-\epsilon, \sigma^2) + \delta_1) \right] + \sum_{i=2}^{d+1} \max_{\|\delta_i\|_\infty \leq \epsilon} [w_i(\mathcal{N}(-\eta-\epsilon) + \delta_i)] + b > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr \left\{ \min_{\|\delta_1\|_\infty \leq \epsilon} \left[w_1(\mathcal{N}(1+\epsilon, \sigma^2) + \delta_1) \right] + \sum_{i=2}^{d+1} \min_{\|\delta_i\|_\infty \leq \epsilon} [w_i(\mathcal{N}(\eta+\epsilon) + \delta_i)] + b < 0 \right\} \cdot \frac{1}{2} \\
&= \Pr \left\{ w_1 \mathcal{N}(-1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(-\eta, \sigma^2) + b > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr \left\{ w_1 \mathcal{N}(1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(\eta, \sigma^2) + b < 0 \right\} \cdot \frac{1}{2},
\end{aligned}
\tag{27}
$$

which is equivalent to the natural risk on the mixture Gaussian distribution $\mathcal{D} : \boldsymbol{x} \sim \mathcal{N}(y \cdot \boldsymbol{w}_{\text{nat}}, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{w}_{\text{nat}} = (1, \eta, \ldots, \eta)$. We note that the Bayes optimal classifier for $\mathcal{D}$ is $f_{\text{nat}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}_{\text{nat}}^\top \boldsymbol{x})$. Specifically, the natural risk

$$
\begin{aligned}
\mathcal{R}^{0}_{\text{adv}}(f, \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \{f(\boldsymbol{x}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \{y \cdot f(\boldsymbol{x}) < 0\} \\
&= \Pr \left\{ w_1 \mathcal{N}(-1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(-\eta, \sigma^2) + b > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr \left\{ w_1 \mathcal{N}(1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(\eta, \sigma^2) + b < 0 \right\} \cdot \frac{1}{2},
\end{aligned}
\tag{28}
$$

which is minimized when $w_1 = 1$, $w_i = \eta$ for $i \geq 2$, and $b = 0$. That is, minimizing the adversarial risk $\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}_{\text{hyp}})$ can lead to an optimal linear ℓ_∞-robust classifier $f_{\text{nat}}(\boldsymbol{x})$, which is equivalent to the natural classifier (8). This concludes the proof of the theorem.

□

C.4 Proof of Theorem 3

Lemma 4. *Minimizing the adversarial risk (16) on the data $\mathcal{T}_{hyp}$ (11) with a defense budget $\epsilon + \eta$ can result in a classifier that assigns 0 weight to the features x_i for $i \geq 2$.*

Proof. The goal is to minimize the adversarial risk on the distribution $\mathcal{T}_{\text{hyp}}$, which can be written as follows:

$$
\begin{aligned}
\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{\exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta, f(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [y \cdot f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \right\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [f(\boldsymbol{x}+\boldsymbol{\delta})] > 0 \mid y=-1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y=-1\} \\
&\quad + \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \mid y=+1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y=+1\} \\
&= \underbrace{\Pr \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} \left[w_1(\mathcal{N}(-1-\epsilon, \sigma^2)+\delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(-\eta-\epsilon, \sigma^2)+\delta_i) + b \right] > 0 \right\}}_{\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}^{(-1)})} \cdot \frac{1}{2} \\
&\quad + \underbrace{\Pr \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} \left[w_1(\mathcal{N}(1+\epsilon, \sigma^2)+\delta_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(\eta+\epsilon, \sigma^2)+\delta_i) + b \right] < 0 \right\}}_{\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}^{(+1)})} \cdot \frac{1}{2}
\end{aligned}
\tag{29}
$$

Consider an optimal solution $\boldsymbol{w}$ in which $w_i > 0$ for some $i \geq 2$. Then, we have

$$
\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}^{(-1)}) = \Pr \left\{ \underbrace{\sum_{j\neq i} \max_{\|\delta_j\| \leq \epsilon+\eta} \left[w_j(\mathcal{N}(-[\boldsymbol{w}_{\text{nat}}]_j - \epsilon, \sigma^2) + \delta_j) + b \right]}_{\mathbb{I}} + \underbrace{\max_{\|\delta_i\| \leq \epsilon+\eta} \left[w_i(\mathcal{N}(-\eta-\epsilon, \sigma^2)+\delta_i) \right]}_{\mathbb{J}} > 0 \right\},
\tag{30}
$$

where $\boldsymbol{w}_{\text{nat}} \coloneqq [1, \eta, \ldots, \eta]$. Since $w_i > 0$, $\mathbb{J}$ is maximized when $\delta_i = \epsilon + \eta$. Thus, the contribution of terms depending on w_i to $\mathbb{J}$ is a normally-distributed random variable with mean 0. Thus, setting w_i to zero will not increase the risk. Formally, we have

$$
\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}^{(-1)}) = \Pr \left\{ \mathbb{I} + w_i \mathcal{N}(0, \sigma^2) > 0 \right\} \geq \Pr \left\{ \mathbb{I} > 0 \right\}. \tag{31}
$$

We can also assume $w_i < 0$ and a similar argument holds. Similar arguments also hold for $\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}^{(+1)})$. Therefore, minimizing the adversarial risk on $\mathcal{T}_{\text{hyp}}$ can lead to $w_i = 0$ for $i \geq 2$. □

Theorem 3 (restated). *The optimal linear ℓ_∞-robust classifier obtained by minimizing the adversarial risk on the perturbed data $\mathcal{T}_{hyp}$ (11) 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.*

Proof. By Lemma 4, we have $w_i = 0$ ($i \geq 2$) for an optimal linear ℓ_∞-robust classifier. Also, the robust classifier will assign a positive weight to the first feature. This is similar to the case in Lemma 2 and we omit the proof here. Then, we have

$$
\begin{aligned}
\mathcal{R}_{\text{adv}}^{\epsilon+\eta}(f, \mathcal{T}_{\text{hyp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{\exists \|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta, f(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [y \cdot f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \right\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [f(\boldsymbol{x}+\boldsymbol{\delta})] > 0 \mid y=-1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y=-1\} \\
&\quad + \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} [f(\boldsymbol{x}+\boldsymbol{\delta})] < 0 \mid y=+1 \right\} \cdot \Pr_{(\boldsymbol{x},y)\sim\mathcal{T}_{\text{hyp}}} \{y=+1\} \\
&= \Pr \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} \left[w_1(\mathcal{N}(-1-\epsilon, \sigma^2)+\delta_1) + b \right] > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr \left\{ \min_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon+\eta} \left[w_1(\mathcal{N}(1+\epsilon, \sigma^2)+\delta_1) + b \right] < 0 \right\} \cdot \frac{1}{2} \\
&= \Pr \left\{ w_1 \mathcal{N}(-1-\eta, \sigma^2) + b > 0 \right\} \cdot \frac{1}{2} \\
&\quad + \Pr \left\{ w_1 \mathcal{N}(1-\eta, \sigma^2) + b < 0 \right\} \cdot \frac{1}{2},
\end{aligned}
\tag{32}
$$

which is equivalent to the natural risk on a mixture Gaussian distribution $\mathcal{D}_{\text{tmp}} : \boldsymbol{x} \sim \mathcal{N}(y \cdot \boldsymbol{\mu}_{\text{tmp}}, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{\mu}_{\text{tmp}} = (1-\eta, 0, \ldots, 0)$. We note that the Bayes optimal classifier for $\mathcal{D}_{\text{tmp}}$ is $f_{\text{tmp}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{\mu}_{\text{tmp}}^\top \boldsymbol{x})$. Specifically, the natural risk

$$\begin{aligned}
\mathcal{R}^0_{\text{adv}}(f, \mathcal{D}_{\text{tmp}}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{tmp}}} \{f(\boldsymbol{x}) \neq y\} \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{tmp}}} \{y \cdot f(\boldsymbol{x}) < 0\} \\
&= \Pr\left\{w_1 \mathcal{N}(-1-\eta, \sigma^2) + b > 0\right\} \cdot \frac{1}{2} \\
&\quad + \Pr\left\{w_1 \mathcal{N}(1-\eta, \sigma^2) + b < 0\right\} \cdot \frac{1}{2},
\end{aligned} \tag{33}$$

which is minimized when $w_1 = 1-\eta > 0$ and $b = 0$. That is, minimizing the adversarial risk $\mathcal{R}^{\epsilon+\eta}_{\text{adv}}(f, \mathcal{T}_{\text{hyp}})$ can lead to an optimal linear ℓ_∞-robust classifier $f_{\text{tmp}}(\boldsymbol{x})$, which is equivalent to the robust classifier (9).

Moreover, when the defense budget ϵ_d is less than $\epsilon + \eta$, the condition in Lemma 4 no longer holds. Instead, in this case, the robust classifier will assign positive weights to the features (i.e., $w_i > 0$ for $i \geq 1$). This is similar to the case in Lemma 3, and thus we omit the proof here. Consequently, this yields classifiers that still rely on all the non-robust features.

□

C.5 Proof of Theorem 4

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

Proof. For clarity, we rewrite the adversarial risk in (2) with a defense budget ϵ as follows:

$$\mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}) \coloneqq \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \left[\max_{\|\boldsymbol{\delta}\|\leq\epsilon} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}), y)\right], \tag{34}$$

where $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ denotes the empirical training data.

Consider any adversary with an attack budget ϵ, who can perturb $\boldsymbol{x}$ to $\boldsymbol{x} + \boldsymbol{p}$ such that $\|\boldsymbol{p}\| \leq \epsilon$. Then, the learner will receive a perturbed version of training data $\mathcal{T}' = \{(\boldsymbol{x}_i + \boldsymbol{p}_i, y_i)\}_{i=1}^n$.

For any perturbed data point $(\boldsymbol{x}_i + \boldsymbol{p}_i, y_i)$, we have

$$\begin{aligned}
\max_{\|\boldsymbol{\delta}\|\leq 2\epsilon} \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{p}_i + \boldsymbol{\delta}), y_i) &= \max_{\|\boldsymbol{\delta}\|\leq\epsilon, \|\boldsymbol{\xi}\|\leq\epsilon} \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{p}_i + \boldsymbol{\delta} + \boldsymbol{\xi}), y_i) \\
&\geq \max_{\|\boldsymbol{\delta}\|\leq\epsilon} \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{p}_i + \boldsymbol{\delta} - \boldsymbol{p}_i), y_i) \\
&= \max_{\|\boldsymbol{\delta}\|\leq\epsilon} \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{\delta}), y_i).
\end{aligned} \tag{35}$$

By summarizing the training points, we have

$$\mathcal{R}^{2\epsilon}_{\text{adv}}(f, \mathcal{T}') \geq \mathcal{R}^{\epsilon}_{\text{adv}}(f, \mathcal{T}). \tag{36}$$

That is, the adversarial risk with a defense budget 2ϵ on the perturbed data is an upper bound of the adversarial risk with a defense budget ϵ on the original data. Therefore, a defense budget 2ϵ is sufficient to ensure the learning of ϵ-robustness. □

D Experimental Settings

Adversary capability. We focus on the clean-label setting, where an adversary can only provide correctly labeled but misleading training data. In this setting, the main constraint is to craft perturbations as small as possible [16]. Thus, we consider an ℓ_∞ adversary with an *attack budget* $\epsilon_a = 8/255$ by following Huang et al. [28], Yuan and Wu [79], Tao et al. [64], Fowl et al. [18]. We note that this constraint is consistent with common research on test-time adversarial examples [1].

Crafting details. We conduct stability attacks by applying the hypocritical perturbation into the training set. Unless otherwise specified, we craft the perturbations by solving the error-minimizing objective (4) with 100 steps of PGD, where a step size of $0.8/255$ is used by following Fowl et al. [18]. Our crafting model is adversarially trained with a *crafting budget* $\epsilon_c = 0.25\epsilon_a$ for 10 epochs before generating perturbations. That is, setting $\epsilon_c = 2/255$ performs best, as shown in Figure 2(a).

Training details. We evaluate the effectiveness of the hypocritical perturbation on benchmark datasets including CIFAR-10/100 [33], SVHN [43], and Tiny-ImageNet [34]. 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 previous studies [81, 69, 50]. By convention, the *defense budget* is equal to the attack budget, i.e., $\epsilon_d = 8/255$. The networks are trained for 100 epochs using SGD with momentum 0.9, weight decay 5×10^{-4}, and an initial learning rate of 0.1 that is divided by 10 at the 75-th and 90-th epoch. Early stopping is done with holding out 1000 examples from the training set. Simple data augmentations such as random crop and horizontal flip are applied. The inner maximization problem during adversarial training is solved by 10-steps PGD (PGD-10) with step size $2/255$.

E Feature-level Analysis on CIFAR-10

In Section 3.1, we theoretically showed that the hypocritical perturbation can cause the poisoned model to rely more on non-robust features, thus the natural accuracy of the adversarially trained model is increased while the robust accuracy is decreased. In this part, we aim to provide empirical evidence on the role of non-robust features in the success of our poisoning method on a benchmark dataset. In particular, we will demonstrate that our hypocritical perturbation successfully makes the poisoned model learn more non-robust features.

To show this, by following Section 3.2 of Ilyas et al. [29], we construct a training set where the only features that are useful for classification are the non-robust features (that are extracted from the poisoned model). The standard accuracy of the classifier trained on the constructed dataset can reflect how many non-robust features are learned by the poisoned model (denoted as f). To accomplish this, we modify each input-label pair $(\boldsymbol{x}, y)$ as follows. We select a target class t uniformly at random among classes. Then, we add a small adversarial perturbation to $\boldsymbol{x}$ as follows:

$$\boldsymbol{x}_{\text{adv}} = \underset{\|\boldsymbol{x}'-\boldsymbol{x}\| \leq \epsilon}{\arg\min}\, \ell(f(\boldsymbol{x}'), t).$$

The resulting input-label pairs $(\boldsymbol{x}_{\text{adv}}, t)$ make up the new training set. Since the resulting inputs $\boldsymbol{x}_{\text{adv}}$ are nearly indistinguishable from the originals $\boldsymbol{x}$, the label t assigned to the modified input is simply incorrect to a human observer. Therefore, only the non-robust features in the training set are predictive, while the non-robust features are extracted from the poisoned model.

We compare the model poisoned by our hypocritical perturbation with the baseline model trained on clean data. These two models correspond to the second row and last row in Table 2, respectively. Using these two models, we construct two datasets in the above-mentioned manner, respectively. Then, two new predictors are trained on the two constructed datasets, respectively, and both predictors are evaluated on clean data. Training parameters follow exactly those adopted by Ilyas et al. [29]. Our numerical results are summarized in Table 13.

Table 13: The predictive ability of the non-robust features learned by the poisoned model.

Model for constructing the training set	Standard accuracy on the original test set (%)
The baseline model	27.46
The poisoned model	**56.77**

As shown in Table 13, the non-robust features learned by the poisoned model are much more predictive than the baseline. This indicates that the effect of our poisoning method on the non-robust features learned by the poisoned model is validated empirically.

F Broader Impact

The attack method in this work might be used by an agent in the real world to damage the robust availability of a machine-learning-based system. We discourage this malicious behavior by presenting the threat model of stability attacks to the community. We further propose an adaptive defense to mitigate this issue. The adaptive defense would help to build a more secure and robust machine learning system in the real world. At the same time, the adaptive defense introduces an additional time cost to search for an appropriate defense budget, which might have a negative impact on carbon emission reduction. Furthermore, society should not be overly optimistic about AI safety, since the current studies mostly focus on perturbations bounded by simple norms (e.g., ℓ_∞ norm in this paper). There might exist perturbations beyond the ℓ_p ball in the real world, and we are still far from complete model robustness.

G On the Trade-off between Accuracy and Robustness

An interesting implication of this work is that the hypocritical perturbation exploits the trade-off between standard generalization and adversarial robustness, a phenomenon that has been widely observed in existing works on adversarial training [67, 81, 14, 40, 59, 77].

Prior work mainly observed that adversarial training improves robust accuracy at the cost of natural accuracy *when the training data is clean*. An explanation for the phenomenon is that there are non-robust features in the original dataset, which are predictive yet brittle [67, 29]. Unlike prior work, the trade-off in this work occurs *when the training data is hypocritically perturbed*. Specifically, we make the following observations:

1. When trained on the hypocritically perturbed data, conventional adversarial training produces models with lower robust accuracy but higher natural accuracy (e.g., see Table 2, Table 3, and Table 4).
2. When trained on the hypocritically perturbed data, adversarial training with adaptive budget can improve robust accuracy while reducing natural accuracy (e.g., see Table 5).

These two observations align well with our theoretical analyses in Section 3 and Section 4, respectively. Concretely, our analyses suggest that the hypocritical perturbation works by reinforcing the non-robust features in the original data, so that the models adversarially trained on the manipulated data still rely on the non-robust features. In this way, the natural accuracy of the models increases because the non-robust features are predictive, while the robust accuracy decreases because the non-robust features are brittle. Furthermore, the effectiveness of the adaptive defense lies in the fact that the reinforced non-robust features can be neutralized by enlarging the defense budget of adversarial training. Thus, the adaptive defense improves robustness at the cost of accuracy.

Meanwhile, we note that it would be unsatisfactory that test robustness is improved at the cost of standard generalization. Several improvements have been proposed to alleviate this issue in the case where the training data is clean, such as RST [49], FAT [82], and SCORE [48]. Incorporating these advances would be helpful in resisting stability attacks, and we leave this as future work.

Finally, we remark that the focus of stability attacks is to degrade test robustness. For this reason, we do not impose additional restrictions on their impact on natural accuracy. Having that said, as a method of stability attacks, the hypocritical perturbation is observed to improve natural accuracy while reducing robust accuracy. We note that this makes stability attacks more insidious. For example, if a poisoned model exhibits higher natural accuracy, practitioners would be more easily enticed to deploy it in a real-world system. However, as its robust accuracy is actually undesirably low, the system is prone to losing its normal function when encountering test-time perturbations. In short, the negative impacts of stability attacks are serious, even with higher natural accuracy. Thus, it is imperative to design better defense methods to mitigate the threat of stability attacks.