Better Safe Than Sorry: Preventing Delusive Adversaries with Adversarial Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Delusive adversaries aim to substantially deteriorate the test accuracy of the learning model by *slightly* perturbing the *features* of correctly labeled training examples. By formalizing this malicious attack as finding the worst-case training data within a specific ∞-Wasserstein ball, we show that minimizing adversarial risk on the *perturbed data* is equivalent to optimizing an upper bound of natural risk on the *original data*. This implies that adversarial training can serve as a principled defense against delusive attacks. Thus, the test accuracy decreased by delusive adversaries can be largely recovered with adversarial training. To further understand the internal mechanism of the defense, we disclose that adversarial training can resist the delusive perturbations by preventing the learner from overly relying on non-robust features in a natural setting. Finally, we complement our theoretical findings with a set of experiments on popular benchmark datasets, which shows that the defense withstands six different practical attacks. Both theoretical and empirical results vote for adversarial training when confronted with delusive adversaries.

1 Introduction

Although machine learning (ML) models have achieved superior performance on many challenging tasks, their performance can be largely deteriorated when the training and test distributions are different. For instance, standard models are prone to make mistakes on the adversarial examples that are considered as worst-case data at *test* time [3, 32]. Compared with that, a more threatening and easily overlooked threat is the malicious perturbations at *training* time, i.e., the delusive attacks [22] that aim to maximize test error by slightly perturbing the correctly labeled training examples [1, 2].

In the era of big data, many practitioners collect training data from untrusted sources where delusive adversaries may exist. In particular, many companies are scraping large datasets from unknown users or public websites for commercial use. For example, Kaspersky Lab, a leading antivirus company, has been accused of poisoning competing products [2]. Although they denied any wrongdoings and clarified the false rumors, one can still imagine the disastrous consequences if that really happens in the security-critical applications. Furthermore, a recent survey of 28 organizations found that these industry practitioners are obviously more afraid of data poisoning than other threats from adversarial ML [16]. In a nutshell, delusive attacks has become a realistic and horrible threat to practitioners.

Recently, Feng et al. [7] showed for the first time that delusive attacks are feasible for deep networks, by proposing "training-time adversarial data" that can significantly deteriorate model performance on clean test data. However, how to design learning algorithms that are robust to delusive attacks still remains an open question due to several crucial challenges [22, 7]. First of all, delusive attacks cannot be avoided by standard data cleaning [14], since it does not require mislabeling, and the perturbed examples can maintain their malice even when they are correctly labeled by experts. In addition, even if the perturbed examples could be distinguished with some detection techniques, it is wasteful

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

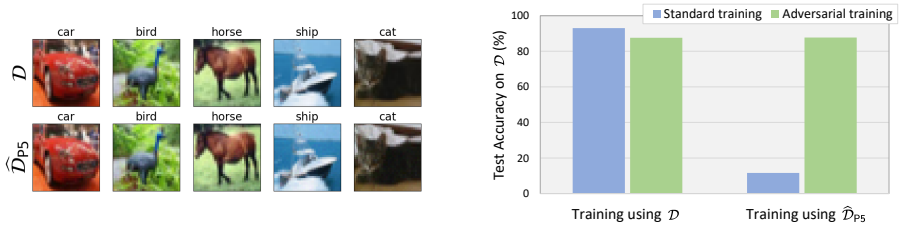


Figure 1: An illustration of delusive attacks and adversarial training. **Left**: Random samples from the CIFAR-10 [15] training set: the original training set $\mathcal{D}$; the perturbed training set $\widehat{\mathcal{D}}_{\mathsf{P5}}$, generated using the P5 attack described in Section 4. **Right**: Natural accuracy evaluated on the CIFAR-10 test set ($\mathcal{D}$) for models trained with: *i)* standard training on $\mathcal{D}$; *ii)* adversarial training on $\mathcal{D}$; *iii)* standard training on $\widehat{\mathcal{D}}_{\mathsf{P5}}$; *iv)* adversarial training on $\widehat{\mathcal{D}}_{\mathsf{P5}}$. Standard training on $\widehat{\mathcal{D}}_{\mathsf{P5}}$ incurs poor generalization performance on $\mathcal{D}$, while adversarial training helps a lot. Details are deferred to Section 5.1.1.

to filter out these correctly labeled examples, considering that deep models are data-hungry. In an extreme case where all examples in the training set are perturbed by a delusive adversary, there will leave no training examples after the filtering stage, thus the learning process can be still obstructed. Given these challenges, we aim to examine the following question in this study: *Is it possible to defend against delusive attacks without abandoning the perturbed examples?*

In this work, we provide an affirmative answer to this question. We first formulate the task of delusive attacks as finding the worst-case data at training time within a specific ∞-Wasserstein ball that prevents label changes (Section 2). By doing so, we find that minimizing the *adversarial risk* on the *perturbed data* is equivalent to optimizing an upper bound of natural risk on the *original data* (Section 3.1). This implies that *adversarial training* [9, 19] on the perturbed training examples can maximize the natural accuracy on the clean examples. Further, we disclose that adversarial training can resist the delusive perturbations by preventing the learner from overly relying on the non-robust features (that are predictive, yet brittle or incomprehensible to humans) in a simple and natural setting. Specifically, two opposite perturbation directions are studied, and adversarial training helps in both cases with slightly different mechanisms (Section 3.2). All these evidences suggest that adversarial training is a promising solution to defend against delusive attacks.

Importantly, our findings widen the scope of application of adversarial training, which was only considered as an effective defense method against test-time adversarial examples [5]. Note that adversarial training usually leads to a drop in natural accuracy [34]. This makes it less practical in many real-world applications where test-time attacks are rare and a high accuracy on clean test data is required [17]. However, this study shows that adversarial training can also defend against a more threatening and invisible threat called delusive adversaries (see Figure 1). We believe that adversarial training will be more widely used in practical applications in the future.

Finally, we present five practical attacks to empirically evaluate the proposed defense (Section 4). Extensive experimental results on various datasets (CIFAR-10, SVHN, and a subset of ImageNet) and tasks (supervised learning, self-supervised learning, and overcoming simplicity bias) demonstrate the effectiveness and versatility of adversarial training, which can significantly mitigate the destructiveness of various delusive attacks (Section 5).

2 Preliminaries

In this section, we introduce some notations and the main ideas we build upon: natural risk, adversarial risk, Wasserstein distance, and delusive adversaries. We discuss related work in Appendix E.

Notation. We consider a classification task with data $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ from a distribution $\mathcal{D}$. We seek to learn a model $f : \mathcal{X} \to \mathcal{Y}$ by minimizing a loss function $\ell(f(\boldsymbol{x}), y)$. Let $\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be some distance metric. Let $\mathcal{B}(\boldsymbol{x}, \epsilon, \Delta) = \{\boldsymbol{x}' \in \mathcal{X} : \Delta(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon\}$ be the ball around $\boldsymbol{x}$ with radius ϵ. When Δ is free of context, we simply write $\mathcal{B}(\boldsymbol{x}, \epsilon, \Delta) = \mathcal{B}(\boldsymbol{x}, \epsilon)$. Throughout the paper, the adversary is allowed to perturb only the inputs, not the labels. Thus, similar to Sinha et al. [29], we define the cost function $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \cup \{\infty\}$ by $c(\boldsymbol{z}, \boldsymbol{z}') = \Delta(\boldsymbol{x}, \boldsymbol{x}') + \infty \cdot \mathbf{1}\{y \neq y'\}$, where $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\mathcal{Z}$ is the set of possible values for $(\boldsymbol{x}, y)$. Denote by $\mathcal{P}(\mathcal{Z})$ the set of all probability

measures on $\mathcal{Z}$. For any $\boldsymbol{\mu} \in \mathbb{R}^d$ and positive definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$, denote by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the d-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Natural risk. The goal of standard training (ST) is to minimize the natural risk defined as

$$\mathcal{R}_{\mathsf{nat}}(f, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\ell(f(\boldsymbol{x}), y)\right]. \tag{1}$$

The term "natural accuracy" refers to the accuracy of a model evaluated on the unperturbed data.

Adversarial risk. The goal of adversarial training (AT) is to minimize the adversarial risk defined as

$$\mathcal{R}_{\mathsf{adv}}(f, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\max_{\boldsymbol{x}'\in\mathcal{B}(\boldsymbol{x},\epsilon)} \ell(f(\boldsymbol{x}'), y)], \tag{2}$$

which is a robust optimization problem that considers the worst-case performance under pointwise perturbations within an ϵ-ball. The main assumption here is that the inputs satisfying $\Delta(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$ preserve the label y of the original input $\boldsymbol{x}$.

Wasserstein distance. Wasserstein distance is a distance function defined between two probability distributions, which represents the cost of an optimal mass transportation plan. Given two data distributions $\mathcal{D}$ and $\mathcal{D}'$, the *p-th Wasserstein distance*, for any $p \geq 1$, is defined as:

$$\mathrm{W}_p(\mathcal{D}, \mathcal{D}') = (\inf_{\gamma\in\Pi(\mathcal{D},\mathcal{D}')} \int_{\mathcal{Z}\times\mathcal{Z}} c(\boldsymbol{z}, \boldsymbol{z}')^p d\gamma(\boldsymbol{z}, \boldsymbol{z}'))^{1/p}, \tag{3}$$

where $\Pi(\mathcal{D}, \mathcal{D}')$ is the collection of all probability measures on $\mathcal{Z} \times \mathcal{Z}$ with $\mathcal{D}$ and $\mathcal{D}'$ being the marginals of the first and second factor, respectively. The ∞-Wasserstein distance is defined as the limit of p-th Wasserstein distance, i.e., $\mathrm{W}_\infty(\mathcal{D}, \mathcal{D}') = \lim_{p\to\infty} \mathrm{W}_p(\mathcal{D}, \mathcal{D}')$. The *$p$-th Wasserstein ball* with respect to $\mathcal{D}$ and radius $\epsilon \geq 0$ is defined as: $\mathcal{B}_{\mathrm{W}_p}(\mathcal{D}, \epsilon) = \{\mathcal{D}' \in \mathcal{P}(\mathcal{Z}) : \mathrm{W}_p(\mathcal{D}, \mathcal{D}') \leq \epsilon\}$.

Staib and Jegelka [30] showed that adversarial training is actually equivalent to a distributionally robust optimization (DRO) problem which minimizes the worst-case distribution constrained by the ∞-Wasserstein distance. Next, we will use the distance to characterize delusive attacks.

Delusive adversaries. The *attacker* is capable of manipulating the training data, as long as the training data is correctly labeled, to prevent the *defender* from learning an accurate classifier [22]. Following Feng et al. [7], the game between the attacker and the defender proceeds as follows:

- n data points are drawn from $\mathcal{D}$ to produce a clean training dataset $\mathcal{D}_n$.
- The attacker perturbs each input $\boldsymbol{x}$ in $\mathcal{D}_n$ by adding small perturbations such that the perturbed input satisfies $\Delta(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$, where ϵ is a small constant that represents the attacker's budget. The perturbed inputs and their original labels constitute the perturbed dataset $\widehat{\mathcal{D}}_n$.
- The defender trains on the perturbed dataset $\widehat{\mathcal{D}}_n$ to produce a model, and incurs natural risk.

The defender's goal is to minimize the natural risk while the attacker's goal is to maximize it. Given a distribution $\mathcal{D}$, we then formulate the attacker's task as the following bi-level optimization problem:

$$\max_{\widehat{\mathcal{D}}\in\mathcal{B}_{\mathrm{W}_\infty}(\mathcal{D},\epsilon)} \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}}, \mathcal{D}), \quad \text{s.t. } f_{\widehat{\mathcal{D}}} = \arg\min_f \mathcal{R}_{\mathsf{nat}}(f, \widehat{\mathcal{D}}). \tag{4}$$

In other words, Eq. 4 is seeking a training distribution $\widehat{\mathcal{D}}$ bounded in the ∞-Wasserstein ball with radius ϵ, so that the model trained on $\widehat{\mathcal{D}}$ has the worst performance on the original distribution $\mathcal{D}$.

Remark 1. It is worth noting that using the ∞-Wasserstein distance to constrain delusive adversaries possesses several advantages. Firstly, the cost function c used in Eq. (3) prevents label changes after perturbation since we only consider clean-label attacks. Secondly, our formulation does not restrict the choice of the distance metric Δ of the input space, thus our theoretical analysis works with any metric, including the ℓ_∞ threat model considered in Feng et al. [7]. Finally, the ∞-Wasserstein ball is more aligned with adversarial risk than other uncertainty sets [30, 37].

Remark 2. Our formulation assumes an underlying distribution that represents the perturbed dataset. This assumption has been widely adopted by existing works [31, 33, 36]. The rationale behind the assumption is that generally, the defender treats the training dataset as an empirical distribution and trains the model on randomly shuffled examples (e.g., training deep networks via stochastic gradient descent). It is also easy to see that our formulation covers the formulation of Feng et al. [7]. On the other hand, this assumption has its limitations. For example, if the defender treats the training examples as sequential data [6], the attacker may utilize the dependence in the sequence to construct perturbations. This situation is beyond the scope of this work, and we leave it as future work.

3 Adversarial Training Beats Delusive Adversaries

In this section, we first justify the rationality of adversarial training as a principled defense method against delusive attacks in the general case for any data distribution. Further, to understand the internal mechanism of the defense, we explicitly explore the space that delusive adversaries can exploit in a simple and natural setting. This implies that adversarial training can resist the delusive perturbations by preventing the learner from overly relying on the non-robust features.

3.1 Adversarial Risk Bounds Natural Risk

Intuitively, the original training data is close to the data perturbed by delusive adversaries, so it will be found in the vicinity of the perturbed data. Thus, training models around the perturbed data can translate to good generalization to the original data. We make the intuition formal in the following theorem, which indicates that adversarial training on a delusive distribution is actually minimizing an upper bound of natural risk on the original distribution.

Theorem 1. *Given a classifier $f : \mathcal{X} \to \mathcal{Y}$, for any data distribution $\mathcal{D}$ and any delusive distribution $\widehat{\mathcal{D}}$ such that $\widehat{\mathcal{D}} \in \mathcal{B}_{\mathrm{W}_\infty}(\mathcal{D}, \epsilon)$, we have*

$$\mathcal{R}_{\mathsf{nat}}(f, \mathcal{D}) \leq \max_{\mathcal{D}' \in \mathcal{B}_{\mathrm{W}_\infty}(\widehat{\mathcal{D}}, \epsilon)} \mathcal{R}_{\mathsf{nat}}(f, \mathcal{D}') = \mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}).$$

The proof is provided in Appendix C.1. Theorem 1 suggests that adversarial training is a principled defense method against delusive attacks. Therefore, when our training data is collected from untrusted sources where delusive adversaries may exist, adversarial training can be applied to minimize the desired natural risk. Besides, Theorem 1 also highlights the importance of the budget $\mathcal{B}_{\mathrm{W}_\infty}(\widehat{\mathcal{D}}, \epsilon)$. On the one hand, if the defender is overly pessimistic (i.e., the defender's budget is larger than the attacker's budget), the tightness of the upper bound cannot be guaranteed. On the other hand, if the defender is overly optimistic (i.e., the defender's budget is relatively small or even equals to 0), the natural risk on the original distribution cannot be upper bounded by the adversarial risk. Our experiments in Section 5.1.3 cover this case when the attacker's budget is under-descriptive.

3.2 Internal Mechanism of the Defense

To further understand the internal mechanism of the defense, in this subsection, we consider a simple and natural setting that allows us to explicitly manipulate the non-robust features. It turns out that, similar to the situation in adversarial examples [34, 11], the model's reliance on non-robust features also allows delusive adversaries to take advantage of it, and adversarial training can resist delusive perturbations by preventing the learner from overly relying on the non-robust features.

As Ilyas et al. [11] has clarified that both robust and non-robust features in data constitute useful signals for standard classification, we are motivated to consider the following binary classification problem on a mixture of two Gaussian distributions $\mathcal{D}$:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y \cdot \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}), \tag{5}$$

where $\boldsymbol{\mu} = (1, \eta, \ldots, \eta) \in \mathbb{R}^{d+1}$ is the mean vector which consists of 1 robust feature with center 1 and d non-robust features with corresponding centers η, similar to the settings in Tsipras et al. [34]. Typically, the number of non-robust features is far more than robust features (i.e., $d \gg 1$). To restrict the capability of delusive adversaries, here we chose the metric function $\Delta(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$. We assume that the attacker's budget ϵ satisfies $2\eta \leq \epsilon \leq 2/3$, so that the attacker: *i)* can shift each non-robust feature towards becoming anti-correlated with the correct label; *ii)* cannot shift each non-robust feature to be strongly correlated with the correct label (as strong as the robust feature).

Delusive attack is easy. For the sake of illustration, here we choose $\epsilon = 2\eta$ and consider two opposite perturbation directions. One direction is that all non-robust features shift towards $-y$, the other is to shift towards y. These settings are chosen for mathematical convenience. The following analysis can be easily adapted to any $\epsilon \geq 2\eta$ and any combinations of the two directions on non-robust features.

Note that the Bayes optimal classifier (i.e., minimization of the natural risk with 0-1 loss) for the distribution $\mathcal{D}$ is $f_{\mathcal{D}}(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{\mu}^\top \boldsymbol{x})$, which relies on both robust feature and non-robust features. As a result, an ℓ_∞-bounded delusive adversary that is only allowed to perturb each non-robust feature

by a moderate ϵ can take advantage of the space of non-robust features. Formally, the original distribution $\mathcal{D}$ can be perturbed to the delusive distribution $\widehat{\mathcal{D}}_1$:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_1, \sigma^2 \boldsymbol{I}), \tag{6}$$

where $\widehat{\boldsymbol{\mu}}_1 = (1, -\eta, \ldots, -\eta)$ is the shifted mean vector. Now, every non-robust feature is correlated with $-y$, thus the Bayes optimal classifier for $\widehat{\mathcal{D}}_1$ will yield extremely poor performance on $\mathcal{D}$, for d large enough. Another interesting perturbation direction is to strengthen the magnitude of non-robust features. This leads to the delusive distribution $\widehat{\mathcal{D}}_2$:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_2, \sigma^2 \boldsymbol{I}), \tag{7}$$

where $\widehat{\boldsymbol{\mu}}_2 = (1, 3\eta, \ldots, 3\eta)$ is the shifted mean vector. Then, the Bayes optimal classifier for $\widehat{\mathcal{D}}_2$ will overly rely on the non-robust features, thus likewise yielding poor performance on $\mathcal{D}$.

The above two attacks are legal because the delusive distributions are close enough to the original distribution, that is, $\mathrm{W}_\infty(\mathcal{D}, \widehat{\mathcal{D}}_1) \leq \epsilon$ and $\mathrm{W}_\infty(\mathcal{D}, \widehat{\mathcal{D}}_2) \leq \epsilon$. Meanwhile, the attacks are also harmful. The following theorem directly compares the destructiveness of the attacks.

Theorem 2. *Let $f_{\mathcal{D}}$, $f_{\widehat{\mathcal{D}}_1}$, and $f_{\widehat{\mathcal{D}}_2}$ be the Bayes optimal classifiers for the mixture-Gaussian distributions $\mathcal{D}$, $\widehat{\mathcal{D}}_1$, and $\widehat{\mathcal{D}}_2$, defined in Eqs. 5, 6, and 7, respectively. For any $\eta > 0$, we have*

$$\mathcal{R}_{\mathsf{nat}}(f_{\mathcal{D}}, \mathcal{D}) < \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}, \mathcal{D}) < \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}, \mathcal{D}).$$

The proof is provided in Appendix C.2. This theorem indicates that both attacks will increase the natural risk of the Bayes optimal classifier. Moreover, $\widehat{\mathcal{D}}_1$ is more harmful because it always incurs higher natural risk than $\widehat{\mathcal{D}}_2$. The destructiveness depends on the dimension of non-robust features. For intuitive understanding, we plot the natural accuracy of the classifiers as a function of d in Figure 2. We observe that, as the number of non-robust features increases, the natural accuracy of the standard model $f_{\widehat{\mathcal{D}}_1}$ continues to decline, while the natural accuracy of $f_{\widehat{\mathcal{D}}_2}$ first decreases and then increases.

Adversarial training matters. Adversarial training with proper ϵ will mitigate the reliance on non-robust features. The mechanism of adversarial training on $\widehat{\mathcal{D}}_1$ is similar to the case in Tsipras et al. [34], while the mechanism on $\widehat{\mathcal{D}}_2$ will be slightly different, and there is no similar analysis before. Specifically, the optimal linear ℓ_∞ robust classifier (i.e., minimization of the adversarial risk with 0-1 loss) for $\widehat{\mathcal{D}}_1$ will rely solely on robust features. Different from the case in $\widehat{\mathcal{D}}_1$, the optimal linear ℓ_∞ robust classifier for $\widehat{\mathcal{D}}_2$ will rely on both robust and non-robust features, but the excessive reliance on non-robust features will be mitigated. Hence, adversarial training helps in both cases. Compared with standard training, adversarial training can achieve better natural accuracy under delusive attacks. We make this formal in the following theorem.

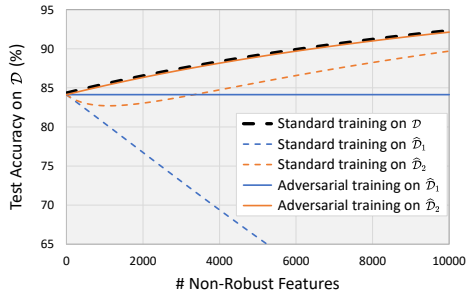


Figure 2: Comparison of the natural accuracy of five models trained on the mixture-Gaussian distributions as a function of the number of non-robust features. As a concrete example, here we set $\sigma = 1$, $\eta = 0.01$ and varying d.

Theorem 3. *Let $f_{\widehat{\mathcal{D}}_1,\mathsf{rob}}$ and $f_{\widehat{\mathcal{D}}_2,\mathsf{rob}}$ be the optimal linear ℓ_∞ robust classifiers for the delusive distributions $\widehat{\mathcal{D}}_1$ and $\widehat{\mathcal{D}}_2$, defined in Eqs. 6 and 7, respectively. For any $0 < \eta < 1/3$, we have*

$$\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1,\mathsf{rob}}, \mathcal{D}) \quad \textit{and} \quad \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2,\mathsf{rob}}, \mathcal{D}).$$

The proof is provided in Appendix C.3. This theorem indicates that the robust models achieve lower natural than standard models under delusive attacks. As we can see from Figure 2, after adversarial training on $\widehat{\mathcal{D}}_1$, natural accuracy can be largely recovered and keeps unchanged as d increases. While on $\widehat{\mathcal{D}}_2$, natural accuracy can be recovered better and keeps increasing as d increases. Beyond the theoretical analysis for this simple case, we also observe that the phenomena in Theorem 2 and Theorem 3 generalize well to our empirical experiments on real-world datasets in Section 5.1.

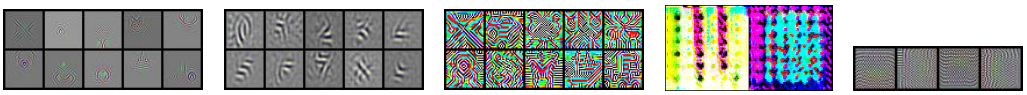
4 Practical Attacks for Testing Defense

Here we briefly describe five heuristic attacks. A detailed description is provided in Appendix D. The five attacks along with the L2C attack proposed by Feng et al. [7] will be used in next section for validating the destructiveness of delusive adversaries and thus the necessity of adversarial training.

In practice, we focus on the empirical distribution $\mathcal{D}_n$ over n data points drawn from $\mathcal{D}$. Inspired by "non-robust features suffice for classification" [11], we construct delusive perturbations by injecting non-robust features correlated consistently with a specific label. Given a standard model $f_{\mathcal{D}}$ trained on $\mathcal{D}_n$, the attacks perturb each input $\boldsymbol{x}$ (with label y) in $\mathcal{D}_n$ as follows:

- P1. It adds a small adversarial perturbation to $\boldsymbol{x}$ to ensure that it is misclassified as a target t by minimizing $\ell\left(f_{\mathcal{D}}(\boldsymbol{x}'), t\right)$ such that $\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon)$, where t is chosen deterministially based on y.
- P2. It adds a small helpful perturbation to $\boldsymbol{x}$ by minimizing $\ell\left(f_{\mathcal{D}}(\boldsymbol{x}'), y\right)$ such that $\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon)$, so that a standard model can easily do correct classification on $\boldsymbol{x}'$.
- P3. This attack is a variant of P1. It adds the class-specific universal adversarial perturbation $\boldsymbol{\xi}_t$ [20, 12] to $\boldsymbol{x}$. All inputs with the label y will be perturbed with the same perturbation $\boldsymbol{\xi}_t$.
- P4. This attack is a variant of P2. It adds the class-specific universal helpful perturbation $\boldsymbol{\xi}_y$ to $\boldsymbol{x}$. All inputs with the label y will be perturbed with the same perturbation $\boldsymbol{\xi}_y$.
- P5. This attack injects class-specific random perturbation $\boldsymbol{r}_y$ to each $\boldsymbol{x}$. All inputs with the label y will be perturbed with the same perturbation $\boldsymbol{r}_y$. Despite the simplicity of this attack, we find that it is surprisingly effective in some cases.

The perturbed inputs and their original labels constitute the perturbed datasets $\widehat{\mathcal{D}}_{\mathtt{P1}} \sim \widehat{\mathcal{D}}_{\mathtt{P5}}$.



(a) CIFAR-10 (ℓ_2) (b) SVHN (ℓ_2) (c) CIFAR-10 (ℓ_∞) (d) ImageNet (ℓ_∞) (e) SSL (ℓ_2)

Figure 3: Universal perturbations for the P3 and P4 attacks across different datasets and threat models. Perturbations are rescaled to lie in the $[0, 1]$ range for display. The resulting inputs are nearly indistinguishable from the originals to a human observer (see Appendix B Figures 10, 11, and 12).

5 Experiments

In order to demonstrate the effectiveness and versatility of the proposed defense, we conduct experiments on CIFAR-10 [15], SVHN [21], a subset of ImageNet [25], and MNIST-CIFAR [27] datasets. Experimental details are provided in Appendix A. Firstly, we perform a set of experiments on supervised learning to provide a comprehensive understanding of delusive adversaries (Section 5.1). Secondly, we demonstrate that the delusive attacks can also obstruct rotation-based self-supervised learning (SSL) [8] and adversarial training also helps a lot in this case (Section 5.2). Finally, we show that adversarial training can be a promising method to overcome the simplicity bias on the MNIST-CIFAR dataset [27] if the ϵ-ball is chosen properly (Section 5.3).

5.1 Understanding Delusive Adversaries

Here, we investigate delusive adversaries from 5 different perspectives: *i)* baseline results on CIFAR-10, *ii)* transferability of delusive perturbations to various architectures, *iii)* performance changes of various defender's budgets, *iv)* a simple countermeasure, and *v)* comparison with Feng et al. [7].

5.1.1 Baseline Results

We consider the typical ℓ_2 threat model with $\epsilon = 0.5$ for CIFAR-10 by following [11]. We use the attacks described in Section 4 to generate the delusive perturbations. To execute the attacks P1 $\sim$ P4, we pre-train a VGG-16 [28] as the standard model $f_{\mathcal{D}}$ using standard training on the original training set. We perform standard training and adversarial training on the delusive datasets $\widehat{\mathcal{D}}_{\mathtt{P1}} \sim \widehat{\mathcal{D}}_{\mathtt{P5}}$. Standard data augmentation (i.e., cropping, mirroring) is adopted. We evaluate the natural accuracy of the models on the clean test set of CIFAR-10.

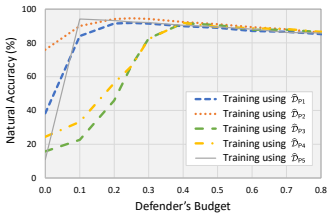


Figure 5: Natural accuracy as a function of the defender's budget on CIFAR-10.

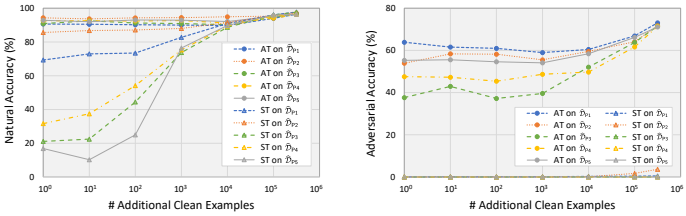


Figure 6: Natural accuracy (left) and adversarial accuracy (right) as a function of the number of additional clean examples on SVHN using ResNet-18 under ℓ_2 threat model.

Results are summarized in Figure 4. We observe that the natural accuracy of the standard models dramatically decreases when training on the delusive datasets, especially on $\widehat{\mathcal{D}}_{\mathtt{P3}}$, $\widehat{\mathcal{D}}_{\mathtt{P4}}$ and $\widehat{\mathcal{D}}_{\mathtt{P5}}$. The most striking observation to emerge from the results is the effectiveness of the P5 attack. It seems that the naturally trained model seems to rely exclusively on the small random patterns in this case, even though there are still abundant natural features in $\widehat{\mathcal{D}}_{\mathtt{P5}}$. Such behaviors resemble the conjunction learner[1] studied in the pioneering work [22]. They showed that such a learner is highly vulnerable to delusive attacks. Also, we note that such behaviors could be attributed to gradient starvation [24] and simplicity bias [27] phenomena of neural networks. These recent studies both show that neural networks trained by SGD preferentially capture a subset of features relevant for the task, despite the presence of other predictive features that fail to be discovered [10].

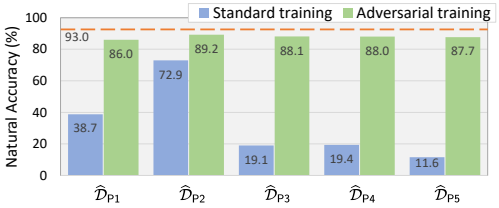


Figure 4: Natural accuracy on CIFAR-10 using VGG-16 under ℓ_2 threat model. The horizontal line indicates the natural accuracy of a standard model trained on the clean training set.

Anyway, our results demonstrate that adversarial training can successfully eliminate the delusive features within the ϵ-ball. As shown in Figure 4, natural accuracy can be significantly improved by adversarial training in all cases. Besides, we observe that P1 is more destructive than P2, which is consistent with our theoretical analysis of the hypothetical settings in Section 3.2.

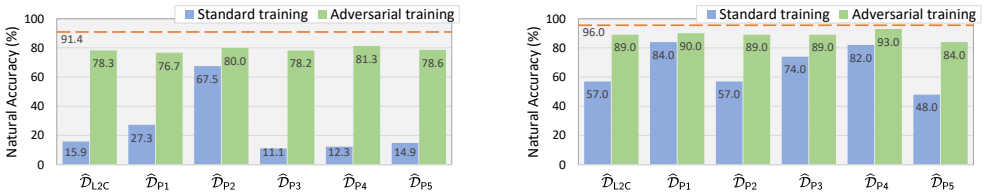
5.1.2 Evaluation of Transferability

A more realistic setting is to attack different classifiers using the same delusive perturbations. We consider various architectures including VGG-19, ResNet-18, ResNet-50, and DenseNet-121 as victim classifiers. The delusive datasets are the same as in Section 5.1.1. Results are deferred to Figure 9 in Appendix B. We observe that the attacks have good transferability across the architectures and adversarial training can improve natural accuracy in all cases. One exception is that the P5 attack is invalid for DenseNet-121. A possible explanation for this might be that the simplicity bias of DenseNet-121 on random patterns is minor. This means that different architectures may have distinct simplicity biases. Due to space constraints, a detailed investigation is out of the scope of this work.

5.1.3 What if the Threat Model Is Under-Descriptive?

Our theoretical analysis in Section 3.1 highlights the importance of choosing a proper budget ϵ for adversarial training. Here, we try to explore this situation where the threat model is under-descriptive by varying the defender's budget. Results on CIFAR-10 using ResNet-18 under ℓ_2 threat model are summarized in Figure 5. We observe that a budget that is too large will hurt performance, while a budget that is too small is not enough to mitigate the attacks. Empirically, the optimal budget for P3 and P4 is about 0.4 and the optimal budget for P1 and P2 is about 0.25. P5 is the easiest to defend—adversarial training with a small budget (about 0.1) can significantly mitigate its effect.

[1]The conjunction learner is to identify a subset of features that appears in every examples of a class, then classifies an example as the class if and only if it contains such features.



(a) CIFAR-10 (b) Two-class ImageNet

Figure 7: Natural accuracy on CIFAR-10 using VGG-11 (left) and two-class ImageNet using ResNet-18 (right) under ℓ_2 threat model. The horizontal orange line indicates the natural accuracy of a standard model trained on the clean training set.

5.1.4 A Simple Countermeasure

In addition to adversarial training, a simple countermeasure is adding clean data to the training set. This will neutralize the perturbed data and make it closer to the original distribution. We explore this countermeasure on SVHN since extensive extra training examples are available in this dataset. Results are summarized in Figure 6. We observe that the performance of standard training is improved with the increase of the number of additional clean examples, and the performance of adversarial training can also be improved with more data. Overall, it is recommend that combining this simple countermeasure with adversarial training to further improve natural accuracy. Besides the focus on natural accuracy in this work, another interesting measure is the model accuracy on adversarial examples. It turns out that adversarial accuracy of the models can also be improved with more data. We also observe that different delusive attacks have different effects on the adversarial accuracy. A further study with more focus on adversarial accuracy is therefore suggested.

5.1.5 Comparison with L2C

We compare the heuristic attacks with the `L2C` attack proposed by Feng et al. [7] and, show that adversarial training can mitigate all these attacks. Following their settings on CIFAR-10 and a two-class ImageNet, the ℓ_∞-norm bounded threat models with $\epsilon = 0.032$ and $\epsilon = 0.1$ are considered. The victim classifier is VGG-11 and ResNet-18 for CIFAR-10 and the two-class ImageNet, respectively. Table 1 reports the time cost for executing six attack methods on CIFAR-10. We find that the heuristic attacks are significantly faster than `L2C`, since the bi-level optimization process in `L2C` is extremely time-consuming. Figure 7 shows the performance of standard training and adversarial training on delusive datasets. The results indicate that most of the heuristic attacks are comparable with `L2C`, and adversarial training can improve natural accuracy in all cases.

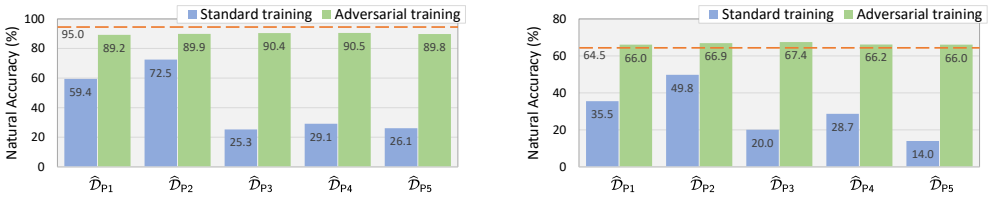
Table 1: Comparison of time cost. The `L2C` attack needs to train an autoencoder to generate perturbations. The `P1` $\sim$ `P4` attacks need to train a standard classifier to generate perturbations, and `P5` needs not.

Method	Time Cost (min)		
	Training	Generating	Total
`L2C`	7411.5	0.4	7411.9
`P1` / `P2`	25.9	12.6	38.5
`P3` / `P4`	25.9	4.6	30.5
`P5`	0.0	0.1	0.1

5.2 Evaluation on Rotation-based Self-supervised Learning

To further show the versatility of the attacks and defense, we conduct experiments on rotation-based self-supervised learning (SSL) [8], a process that learns representations by predicting rotation angles ($0°$, $90°$, $180°$, $270°$). SSL methods seem to be inherently resist to the poisoning attacks that require mislabeling, since they do not use human-annotated labels to learn representations. Here, we examine whether SSL can resist the delusive attacks. We use delusive attacks to perturb the training data for the pretext task. To evaluate the quality of the learned representations, the downstream task is trained on the clean data using logistic regression.

Results are summarized in Figure 8. We observe that the learning of the pretext task can be largely hijacked by the attacks. Thus the learned representations are poor for the downstream task. Again, adversarial training can significantly improve natural accuracy in all cases. An interesting observation from Figure 8(b) is that the quality of the adversarially learned representations is slightly better than that of standard models trained on the original training set. This is consistent with recent



(a) Pretext task (b) Downstream task

Figure 8: Rotation-based SSL on CIFAR-10 using ResNet-18 under ℓ_2 threat model. The horizontal line indicates the natural accuracy of a standard model trained on the clean training set.

Table 2: Adversarial training on MNIST-CIFAR: The table reports test accuracy on the MNIST-CIFAR test set and the MNIST-Randomized test set. Our customized adversarial training successfully overcomes SB, while others not. The MNIST-randomized accuracy indicates that our adversarially trained models achieve nontrivial performance when there are only CIFAR features exist in the inputs.

Model	Test Accuracy on MNIST-CIFAR			MNIST-Randomized Accuracy		
	ST	AT [27]	AT (ours)	ST	AT [27]	AT (ours)
VGG-16	99.9	100.0	91.3	49.1	51.6	**91.2**
ResNet-50	100.0	99.9	89.7	48.9	49.2	**88.6**
DenseNet-121	100.0	100.0	91.5	48.8	49.2	**90.8**

hypotheses stating that robust models transfer better [26, 18]. These results show the possibility of delusive attacks and defenses for SSL, and suggest that studying the robustness of the general SSL methods [13, 4] to data poisoning is a promising direction for future research.

5.3 Overcoming Simplicity Bias

A recent work [27] proposed the MNIST-CIFAR dataset to demonstrate the simplicity bias (SB) of using standard training to learn neural networks. Specifically, the MNIST-CIFAR images $\boldsymbol{x}$ are vertically concatenations of the "simple" MNIST images $\boldsymbol{x}_m$ and the more complex CIFAR-10 images $\boldsymbol{x}_c$ (i.e., $\boldsymbol{x} = [\boldsymbol{x}_m; \boldsymbol{x}_c]$). They found that standard models trained on MNIST-CIFAR will exclusively rely on the MNIST features and remain invariant to the CIFAR features. Thus randomizing the MNIST features drops the model accuracy to random guessing.

From the perspective of delusive adversaries, we can regard the MNIST-CIFAR dataset as a delusive version of the original CIFAR dataset. Thus, adversarial training can mitigate the delusive perturbations, as Theorem 1 pointed out. However, Shah et al. [27] tried adversarial training on MNIST-CIFAR and failed. Contrary to their results, here we demonstrate that adversarial training works in this case. The key factor is the choice of the threat model. They failed because they chose an improper ball $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty \leq 0.3\}$, while our choice is $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x}_m - \boldsymbol{x}'_m\|_\infty + \infty \cdot \|\boldsymbol{x}_c - \boldsymbol{x}'_c\|_\infty \leq 1\}$. Our choice forces the space of MNIST features to be a non-robust region during adversarial training, and prohibits the CIFAR features from being perturbed. Results are summarized in Table 2. We observe that our choice leads to models that do not rely on the simple MNIST features, thus adversarial training can eliminate the simplicity bias.

6 Conclusion and Future work

In this paper, we suggest applying adversarial training in practical applications, rather than standard training whose performance risks being substantially deteriorated by delusive adversaries. Both theoretical and empirical results vote for adversarial training when confronted with delusive adversaries. Nonetheless, there are limitations with our current study, and further investigations are expected in future work. Here, we briefly discuss some of them: *i)* Our implementation of adversarial training adopts the popular PGD-AT framework [19, 23], which is actually minimizing a lower bound of the true adversarial risk. However, finding ways to directly optimize the true adversarial risk may lead to better performance. *ii)* The success of our proposed defense relies on generalization, just like most machine learning algorithms. Thus, an analysis of the generalization error bound for this case will be useful. *iii)* Adversarial training may increase the disparity of accuracy between different groups of data [35]. This problem could be mitigated by fair robust learning [35].

References

[1] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

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

[3] 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 *Joint European conference on machine learning and knowledge discovery in databases ((ECML-KDD))*, 2013.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[5] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[6] Thomas G Dietterich. Machine learning for sequential data: A review. In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, 2002.

[7] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[11] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] Saumya Jetley, Nicholas Lord, and Philip Torr. With friends like these, who needs adversaries? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[13] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[14] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *CHI Conference on Human Factors in Computing Systems (CHI)*, 2011.

[15] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[16] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *IEEE Security and Privacy Workshops (SPW)*, 2020.

[17] Mathias Lechner, Ramin Hasani, Radu Grosu, Daniela Rus, and Thomas A Henzinger. Adversarial training is not ready for robot learning. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[18] Kaizhao Liang, Jacky Y Zhang, Oluwasanmi Koyejo, and Bo Li. Does adversarial transferability indicate knowledge transferability? *arXiv preprint arXiv:2006.14512*, 2020.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

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

[23] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021.

[24] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of Computer Vision (IJCV)*, 2015.

[26] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in neural information processing systems (NeurIPS)*, 2020.

[27] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[29] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations (ICLR)*, 2018.

[30] Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NeurIPS workshop on Machine Learning and Computer Security*, 2017.

[31] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[33] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in neural information processing systems (NeurIPS)*, 2018.

[34] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[35] Han Xu, Xiaorui Liu, Yaxin Li, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. *arXiv preprint arXiv:2010.06121*, 2020.

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

[37] Sicheng Zhu, Xiao Zhang, and David Evans. Learning adversarially robust representations via worst-case mutual information maximization. In *International Conference on Machine Learning (ICML)*, 2020.

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 and Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.

(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] See Appendix C.
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 Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported because it would be too computationally expensive.
   (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] See Appendix A.
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] See Appendix A.
   (b) Did you mention the license of the assets? [Yes] See Appendix 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]