Provable Defense Against Delusive Poisoning

Lue Tao[1,2] **Lei Feng**[3] **Jinfeng Yi**[4] **Sheng-Jun Huang**[1,2] **Songcan Chen**[1,2*]
[1]Nanjing University of Aeronautics and Astronautics
[2]MIIT key Laboratory of Pattern Analysis & Machine Intelligence
[3]Chongqing University
[4]JD AI Research

Abstract

Delusive poisoning is a special kind of attack to obstruct learning, where the learning performance could be significantly deteriorated by *only* manipulating (even slightly) the *features* of correctly labeled training examples. By formalizing this malicious attack as finding the worst-case distribution shift at training time within a specific ∞-Wasserstein ball, we show that minimizing adversarial risk on the *poison data* is equivalent to optimizing an upper bound of natural risk on the *original data*. This implies that adversarial training can be a principled defense method against delusive poisoning. To further understand the internal mechanism of the defense, we disclose that adversarial training can resist the training distribution shift 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 poisoning.

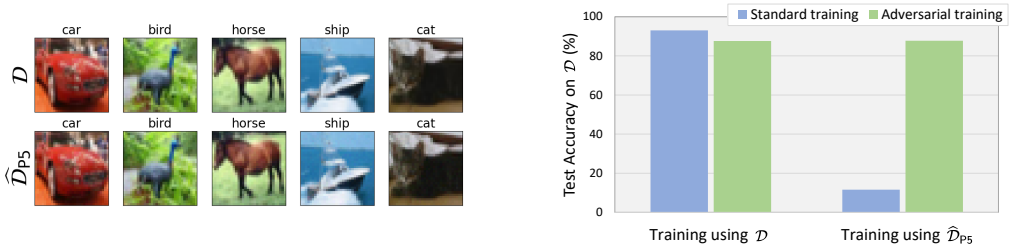
1 Introduction

Although modern machine learning (ML) models have achieved superior performance on many challenging tasks, their performance can be largely deteriorated when the training distribution and test distribution are *different*. For instance, standard models are prone to make mistakes on adversarial examples that are considered as worst-case data at *test* time [2, 44]. Compared with the above test distribution shift, a more threatening and easily overlooked threat is the *invisible* malicious distribution shift at *training* time, *i.e.*, delusive poisoning [31] that aims to maximize test error by only manipulating the features of correctly labeled training examples [5, 1].

In the era of big data, many practitioners collect 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 [5]. Although they denied any wrongdoings and clarified the false rumors, one can still imagine the disastrous consequences if that really happens in other security-critical applications (*e.g.*, autonomous driving or medical diagnosis). Furthermore, a recent survey of 28 organizations found that those industry practitioners are obviously more afraid of data poisoning than other threats from adversarial ML [24]. In a nutshell, delusive poisoning has become a realistic and horrible threat to practitioners.

Recently, [11] showed for the first time that delusive poisoning is feasible for deep networks, by proposing "training-time adversarial examples" that can significantly deteriorate model performance on clean test data (*i.e.*, natural accuracy). However, how to design learning algorithms that are robust to delusive attacks still remains an open question due to some crucial challenges [31, 11]. First of all, delusive poisoning cannot be avoided by standard data cleaning, since it does not require mislabeling, and the poisoned examples can maintain their malice even when they are labeled correctly by experts

*Correspondence to: Songcan Chen <s.chen@nuaa.edu.cn>.



(a) (b)

Figure 1: **Left**: Random samples from the CIFAR-10 [23] training set: the original training set $\mathcal{D}$; the poisond training set $\widehat{\mathcal{D}}_{\mathsf{P5}}$, generated using the P5 attack described in Section 5. **Right**: Natural accuracy 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 $\widehat{\mathcal{D}}_{\mathsf{P5}}$ yields poor generalization performance on $\mathcal{D}$, while adversarial training helps a lot. Details are deferred to Section 6.1.1.

(see Figure 1(a)). In addition, even if the poisoned examples can be distinguished using some detection techniques, it is wasteful 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 poisoned 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 poisoning without abandoning the poisoned examples?

In this work, we provide an affirmative answer to this question. We first formulate the task of delusive poisoning as finding the worst-case distribution shift at training time within a specific ∞-Wasserstein ball that prevents label changes (Section 3). By doing so, we find that minimizing the *adversarial risk* on the *poison data* is equivalent to optimizing an upper bound of natural risk on the *original data* (Section 4.1). This implies that *adversarial training* [14, 26] on the poison data can guarantee the natural accuracy of the clean data. Further, we disclose that adversarial training can resist training distribution shifts by preventing the learner from overly relying on non-robust features (that are predictive, yet brittle or incomprehensible to humans) in a simple and natural setting. Specifically, two opposite distribution shift directions are studied and adversarial training helps in both cases with slightly different mechanisms (Section 4.2). All these evidences suggest that adversarial training is a promising solution to defend against delusive poisoning.

Our findings significantly widen the scope of application of adversarial training, which was only considered as an effective defense method against test-time adversarial examples [8]. Note that adversarial training usually leads to a drop in natural accuracy [43, 46]. This makes it less practical in many real-world applications where adversarial attacks are rare and a high accuracy on clean test data is required. However, this study shows that adversarial training can also defend against a more threatening and invisible threat called delusive poisoning (see Figure 1(b)). We believe that it will be more widely used in practical applications in the future.

Moreover, we present five practical attacks to empirically validate the proposed defense (Section 5). 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 6).

2 Related Work

Adversarial training. Since the discovery of adversarial examples (*a.k.a.*, evasion attacks at test time) [2, 44], a number of defenses have been proposed to improve model robustness, with some of

the most reliable being adversarial training and its variants [14, 26, 36]. While model performance on adversarial examples is improved, adversarial training leads to a drop in natural accuracy in practice [46, 50]. Thus some modified adversarial training schemes turn to improve model accuracy on clean and shifted test examples, rather than on adversarial examples [48, 25]. Different from previous works, we aim to improve natural accuracy when the training data is shifted.

Data poisoning. Data poisoning attacks manipulate training data to cause models to fail during inference. Both targeted and indiscriminate attacks were extensively studied for classical models [31, 29, 3, 4, 47, 51]. However, for neural networks, most of the existing work focuses on targeted misclassification [38, 52, 18, 12], while there is little work on indiscriminate attacks [37]. Recently, [11] showed that indiscriminate attacks are workable for deep networks. We follow their settings where the training data is correctly labeled after delusive poisoning and propose a solution to defend against it. Unlike other sanitization-based defenses which may be overwhelmed by stronger attacks [22, 6], our defense against delusive attacks is principled and theoretically guaranteed. The work most similar to ours is that of [10]. They alleviate data poisoning for linear regression models by relaxing distributionally robust optimization (DRO) as a regularizer, while we handle delusive attacks for any model.

3 Preliminaries

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 $f : \mathcal{X} \to \mathcal{Y}$ minimizing a loss $\ell(f(\boldsymbol{x}), y)$. The term "natural accuracy" refers to model's accuracy evaluated on the original distribution $\mathcal{D}$. 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 $\mathcal{B}(\boldsymbol{x}, \epsilon, \Delta) = \mathcal{B}(\boldsymbol{x}, \epsilon)$. 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 $\boldsymbol{\Sigma}$. Throughout the paper, the adversary is allowed to perturb only the examples $\boldsymbol{x}$, not the labels y. Thus, similar to [41], 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 $(\boldsymbol{x}, y)$. Let $\mathcal{P}(\mathcal{Z})$ denote the set of all probability measures on $\mathcal{Z}$.

Natural risk. Given a data distribution $\mathcal{D}$ and a classifier f, *natural risk* is the expected risk on the distribution:

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

Standard training (ST) is minimizing the above natural risk.

Adversarial risk. Given a data distribution $\mathcal{D}$ and a classifier f, *adversarial risk* is the worst-case performance under pointwise perturbations within an ϵ-ball:

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

Adversarial training (AT) is a robust optimization problem that minimizes the above worst-case risk.

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:

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

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:

$$\mathcal{B}_{\mathrm{W}_p}(\mathcal{D}, \epsilon) = \left\{\mathcal{D}' \in \mathcal{P}(\mathcal{Z}) : \mathrm{W}_p\left(\mathcal{D}, \mathcal{D}'\right) \leq \epsilon\right\}. \tag{4}$$

Delusive poisoning. Delusive poisoning can manipulate the training data, as long as the training data is correctly labeled, to prevent a learner from training an accurate classifier. Given a data distribution $\mathcal{D}$, we formalize the task as finding the worst-case distribution shift bounded in the ∞-Wasserstein ball with radius ϵ at training time:

$$\begin{aligned} \max_{\widehat{\mathcal{D}} \in \mathcal{B}_{\mathrm{W}_\infty}(\mathcal{D}, \epsilon)} \quad & \mathop{\mathbb{E}}_{(\boldsymbol{x}, y) \sim \mathcal{D}} \left[\ell(f_{\widehat{\mathcal{D}}}(\boldsymbol{x}), y) \right], \\ \text{s.t.} \quad & f_{\widehat{\mathcal{D}}} = \arg\min_{f} \mathop{\mathbb{E}}_{(\boldsymbol{x}, y) \sim \widehat{\mathcal{D}}} \left[\ell\left(f\left(\boldsymbol{x}\right), y\right) \right]. \end{aligned} \tag{5}$$

In other words, Equation 5 is seeking a training distribution $\widehat{\mathcal{D}}$, so that the classifier trained on $\widehat{\mathcal{D}}$ has worst performance on the original distribution $\mathcal{D}$.

It is worth noting that using the ∞-Wasserstein distance to constrain delusive attacks has several advantages. Firstly, the cost function c used in Equation (3) prevents label changes during distribution shift since we only consider clean-label data poisoning. Secondly, our definition does not restrict the choice of the distance metric Δ of the input space, thus our theoretical analysis works with any delusive adversary, including the ℓ_∞-norm threat model considered in [11]. Finally, the ∞-Wasserstein ball is more aligned with adversarial risk than other uncertainty sets [42, 53].

4 Defense from the Perspective of Distributional Robustness

Adversarial examples are worst-case data at *test* time. Analogically, here delusive poisoning can be viewed as worst-case distribution shift at *training* time, and thus defensed from the perspective of distributional robustness. Next, we will justify the rationality of adversarial training, in the general case for any data distribution, as a principled defense method against delusive poisoning. 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 shows that adversarial training can resist the training distribution shift by preventing the learner from overly relying on the non-robust features.

4.1 Adversarial Training Certifies Natural Accuracy

In this subsection, we justify the rationality of adversarial training for any distribution and any delusive adversary bounded in the ∞-Wasserstein ball with radius ϵ. The following theorem proves that adversarial training on a delusive distribution is actually minimizing an upper bound of natural risk on the original distribution.

Theorem 4.1. *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. This theorem suggests that adversarial training is a principled defense method against delusive poisoning. Thus, when our training data are sampled from an untrusted distribution where delusive adversaries may exist, adversarial training can be applied to guarantee natural accuracy. Besides, it is also noteworthy that Theorem 4.1 highlights the importance of the budget $\mathcal{B}_{\mathrm{W}_\infty}(\widehat{\mathcal{D}}, \epsilon)$ for DRO. On the one hand, if the defender is overly pessimistic (*e.g.*, 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 (*e.g.*, the defender's budget is smaller or even equals to 0), then natural risk on the original distribution cannot be upper bounded by the adversarial risk. Our experiments in Section 6.1.3 also cover this case when the attacker's budget is under-descriptive.

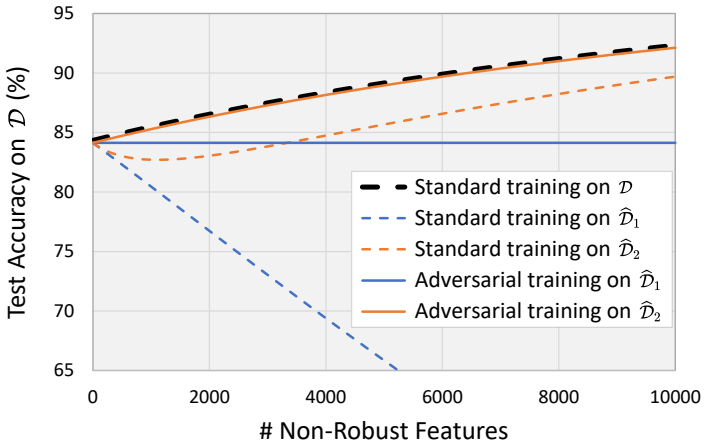


Figure 2: Comparison of the natural accuracy of five models trained on the mixture-Gaussian distributions as a function of the number of non-robust features. As a concrete example, we set $\sigma = 1$, $\eta = 0.01$ and varying d. We observe that, as the number of non-robust features increases, the natural 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. In both cases, adversarial training can mitigate the effect of distribution shift and thus improve test accuracy on the original distribution.

4.2 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 in the data. This shows that, similar to the situation in adversarial examples [46, 19], the classifier's reliance on non-robust features also allows delusive adversaries to take advantage of it, and adversarial training can resist training distribution shifts by preventing the learner from overly relying on the non-robust features.

As [19] 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 $\mathcal{D}$:

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

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 [46]. Typically, the number of non-robust features is far more than robust features (*i.e.*, $d \gg 1$). To restrict the capacity of delusive adversaries, here we chose the metric function $\Delta(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$. We assume that the adversarial budget ϵ satisfies $\epsilon \geq 2\eta$ and $\eta < 1/3$, so that an adversary: *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. Note that the Bayes optimal classifier (*i.e.*, minimization of the natural risk with 0-1 loss) for the original distribution $\mathcal{D}$ is $f_{\mathcal{D}}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\mu}^\top \boldsymbol{x})$, 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.

For the sake of illustration, we choose $\epsilon = 2\eta$ and consider two opposite distribution shift 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. All the following theoretical results can be easily adapted to any $\epsilon \geq 2\eta$ and any combinations of the two directions on non-robust features.

Formally, the original distribution $\mathcal{D}$ can be shifted to the delusive distribution $\widehat{\mathcal{D}}_1$: $\boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_1, \sigma^2 \boldsymbol{I})$, 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 direction for distribution shift is to strengthen the magnitude of non-robust features. This results in the delusive distribution $\widehat{\mathcal{D}}_2$: $\boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_2, \sigma^2 \boldsymbol{I})$, where $\widehat{\boldsymbol{\mu}}_2 = (1, 3\eta, \ldots, 3\eta)$ is the shifted mean vector. Then, the non-robust features will be over-utilized by the Bayes optimal classifier for $\widehat{\mathcal{D}}_2$, 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 $\mathrm{W}_\infty(\mathcal{D}, \widehat{\mathcal{D}}_2) \leq \epsilon$. They are also harmful. The following theorem directly compares the destructiveness of the attacks.

Theorem 4.2. *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 $\widehat{\mathcal{D}}_2$, 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 destructive since it always yields 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.

Adversarial training matters. Adversarial training with proper ϵ will mitigate the importance of non-robust features. The mechanism of adversarial training on $\widehat{\mathcal{D}}_1$ is similar to the case in [46], 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)[2] 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 can be mitigated. Hence, adversarial training helps in both cases. Compared with standard training on the delusive distributions, adversarial training can improve model performance on the original distribution. We make this formal in the following theorem.

Theorem 4.3. *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 $\widehat{\mathcal{D}}_2$, respectively. For any $0 < \eta < 1/3$, we have*

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

The proof is provided in Appendix C.3. This theorem indicates that robust models achieve lower natural risk on the original distribution. As we can see in the concrete example of Figure 2, when adversarially trained on $\widehat{\mathcal{D}}_1$, natural accuracy can be partially recovered and keeps unchanged as d increases. While on $\widehat{\mathcal{D}}_2$, natural accuracy is recovered better and keeps increasing as d increases. Beyond the theoretical analysis for this simple case, we observe that the phenomena in Theorem 4.2 and Theorem 4.3 also generalize well to our empirical experiments on real-world datasets in Section 6.1.

5 Practical Attacks for Testing Defense

To further demonstrate the destructiveness of delusive poisoning on real datasets and thus the necessity of adversarial training, in this section, we present five heuristic attacks by injecting non-robust features to finite-size training data.

In practice, we focus on the empirical distribution $\mathcal{D}_n$ over n data pairs $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ drawn i.i.d. from $\mathcal{D}$. To avoid the difficulty to search through the whole ∞-Wasserstein ball, one common choice is to consider the following set of empirical distributions [53]:

$$\mathcal{A}(\mathcal{D}_n, \epsilon) = \left\{ \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_n} \delta(\boldsymbol{x}', y) : \boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon) \right\}, \tag{7}$$

[2]Here we only employing linear classifiers, since considering non-linearity is highly nontrivial for minimizing the ℓ_∞ adversarial risk on the mixture-Gaussian distribution [9].

where $\delta(\boldsymbol{x}, y)$ is the dirac measure at $(\boldsymbol{x}, y)$. Note that the considered set $\mathcal{A}(\mathcal{D}_n, \epsilon) \subseteq \mathcal{B}_{\mathrm{W}_\infty}(\mathcal{D}_n, \epsilon)$, since each perturbed point $\boldsymbol{x}'$ is at most ϵ-away from $\boldsymbol{x}$.

The delusive attack `L2C` proposed in [11] is actually searching for worst-case distribution shift in $\mathcal{A}(\mathcal{D}_n, \epsilon)$ with the ℓ_∞ metric. However, `L2C` is directly optimizing the bi-level optimization problem (5), whose computational cost is very huge. Instead, we present five efficient attacks below. Inspired by "non-robust features suffice for classification" [19], we construct delusive distributions within $\mathcal{A}(\mathcal{D}_n, \epsilon)$ by injecting non-robust features correlated consistently with a specific label to each example.

Poison 1 (`P1`): The first construction is similar to that of the deterministic dataset in [19]. In our construction, the robust features are still correlated with their original labels. We modify each input-label pair $(\boldsymbol{x}, y)$ as follows. We select a target class t deterministically according to the source class (*e.g.*, using a fixed permutation of labels). Then we add a small adversarial perturbation to $\boldsymbol{x}$ in order to ensure it is classified as t by a standard model. Formally:

$$\boldsymbol{x}_{\mathsf{adv}} = \operatorname*{arg\,min}_{\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon)} \ell\left(f_{\mathcal{D}}(\boldsymbol{x}'), t\right), \tag{8}$$

where $f_{\mathcal{D}}$ is a standard classifier trained on the original distribution $\mathcal{D}$ (or its finite-sample counterpart $\mathcal{D}_n$). Finally, we assign the correct label y to the perturbed input. The resulting input-label pairs $(\boldsymbol{x}_{\mathsf{adv}}, y)$ make up the delusive dataset $\widehat{\mathcal{D}}_{\mathtt{P1}}$. This attack resembles the $\widehat{\mathcal{D}}_1$ for the mixture-Gaussian distribution in Section 4.2. It is noteworthy that this type of data poisoning was mentioned in the addendum of [28], but was not gotten further exploration.

Poison 2 (`P2`): This attack is motivated by recent studies on so-called "unadversarial examples" [35, 45]. We inject helpful non-robust features to inputs so that a standard model can easily do correct classification. Formally:

$$\boldsymbol{x}_{\mathsf{unadv}} = \operatorname*{arg\,min}_{\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x}, \epsilon)} \ell\left(f_{\mathcal{D}}(\boldsymbol{x}'), y\right). \tag{9}$$

The resulting input-label pairs $(\boldsymbol{x}_{\mathsf{unadv}}, y)$ make up the delusive dataset $\widehat{\mathcal{D}}_{\mathtt{P2}}$, where the helpful features are prevalent in all examples. However, in the original distribution $\mathcal{D}$, those artificial features may be relatively sparse. Thus the resulting learned classifier which overly relies on the artificial features may perform poorly on $\mathcal{D}$. This attack resembles the $\widehat{\mathcal{D}}_2$ for the mixture-Gaussian distribution in Section 4.2.

Poison 3 (`P3`): This attack is a variant of `P1`. To improve the transferability of the perturbations, we adopt the class-specific universal adversarial perturbation [27, 20]. Formally:

$$\boldsymbol{\xi}_t = \operatorname*{arg\,min}_{\boldsymbol{\xi} \in \mathcal{B}(\mathbf{0}, \epsilon)} \mathop{\mathbb{E}}_{(\boldsymbol{x}, y) \sim \mathcal{D}} \ell\left(f_{\mathcal{D}}(\boldsymbol{x} + \boldsymbol{\xi}), t\right), \tag{10}$$

where t is deterministically selected according to the source class y. The resulting input-label pairs $(\boldsymbol{x} + \boldsymbol{\xi}_t, y)$ make up the delusive dataset $\widehat{\mathcal{D}}_{\mathtt{P3}}$. Intuitively, if a specific feature repeatedly appears in all examples from the same class, the learned classifier will easily capture such features.

Poison 4 (`P4`): This attack is a variant of `P2`. Similar to `P3`, we adopt class-specific universal unadversarial perturbations and the resulting input-label pairs $(\boldsymbol{x} + \boldsymbol{\xi}_y, y)$ make up the delusive dataset $\widehat{\mathcal{D}}_{\mathtt{P4}}$.

Poison 5 (`P5`): This attack injects class-specific random perturbations to training data. We generate a random perturbation $\boldsymbol{r}_y \in \mathcal{B}(\mathbf{0}, \epsilon)$ for each class y (using uniform noise or Gaussian noise). Then the resulting input-label pairs $(\boldsymbol{x} + \boldsymbol{r}_y, y)$ make up the delusive dataset $\widehat{\mathcal{D}}_{\mathtt{P5}}$. Despite the simplicity of this poison, we find that it is surprisingly effective in some cases.

6 Experiments

To demonstrate the effectiveness and versatility of the proposed defense, we conduct experiments on CIFAR-10 [23], SVHN [30], a subset of ImageNet [33], and MNIST-CIFAR [39] datasets. Experimental details are provided in Appendix A.

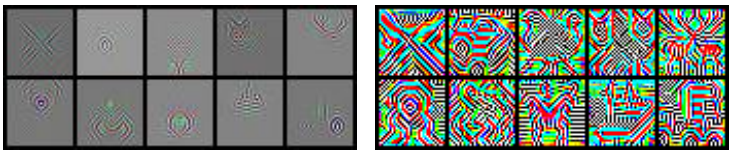


Figure 3: Universal perturbations for P3 and P4 on CIFAR-10. The threat models are the ℓ_2 ball with $\epsilon = 0.5$ (left) and the ℓ_∞ ball with $\epsilon = 0.032$ (right). Perturbations are rescaled for display. The resulting inputs are nearly indistinguishable from the originals to a human observer (see Appendix B Figure 12, 13, and 14).

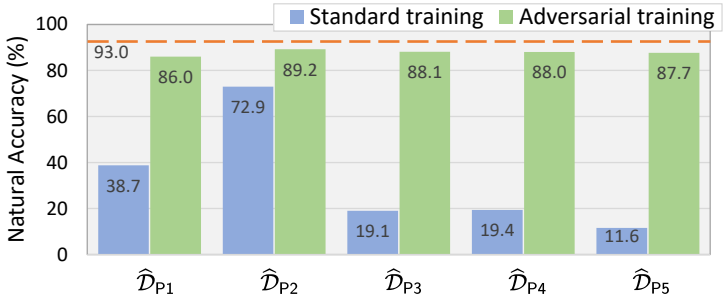


Figure 4: Natural accuracy of models trained on delusive datasets for CIFAR-10. The victim classifier is VGG-16. The threat model is the ℓ_2 ball with $\epsilon = 0.5$. The horizontal line indicates natural accuracy of a standard model trained on the original training set.

We first perform a set of experiments on supervised learning to provide a comprehensive understanding of delusive poisoning (Section 6.1). Then we demonstrate that the attacks can obstruct rotation-based self-supervised learning [13] and adversarial training also helps a lot in this case (Section 6.2). Finally, we show that adversarial training can be a promising method to overcome the simplicity bias on the MNIST-CIFAR dataset [39] when the ball $\mathcal{B}(\boldsymbol{x}, \epsilon)$ is chosen properly (Section 6.3).

6.1 Understanding Delusive Adversaries

6.1.1 Baseline Results

Here the typical ℓ_2-norm bounded threat model with $\epsilon = 0.5$ is considered. We use the attacks described in Section 5 to generate delusive datasets for CIFAR-10. To execute the attacks P1 $\sim$ P4, we pre-train a VGG-16 [40] as the standard model $f_{\mathcal{D}}$ using standard training on the original training set. Standard training and adversarial training are performed on the resulting delusive datasets ($\widehat{\mathcal{D}}_{\mathsf{P1}} \sim \widehat{\mathcal{D}}_{\mathsf{P5}}$). Standard data augmentation (*i.e.*, cropping, mirroring) is used. We evaluate natural accuracy on the clean test set of CIFAR-10 for the resulting models.

Results are summarized in Figure 4. We observe that natural accuracy of the standard models dramatically decrease when trained on the delusive datasets, especially on $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$ and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The most striking observation to emerge from the results is the effectiveness of P5. It seems that the model becomes exclusively relying on those small random patterns, even there are abundant natural features still in $\widehat{\mathcal{D}}_{\mathsf{P5}}$. Such behaviors resemble the conjunction learner[3] studied in the pioneering work [31]. They showed that such a learner is highly vulnerable to delusive attacks. Also, we note that such behaviors could be attributed to gradient starvation [32] or simplicity bias [39] 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 [16].

[3]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.

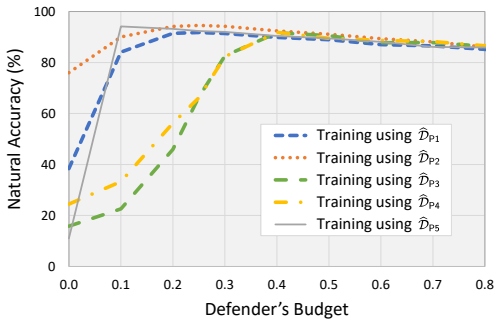


Figure 5: Natural accuracy as a function of the defender's budget on CIFAR-10. The victim classifier is ResNet-18. The poison data is generated within the ℓ_2 ball with $\epsilon = 0.5$. The budget ϵ for adversarial training is varying from 0 to 1.

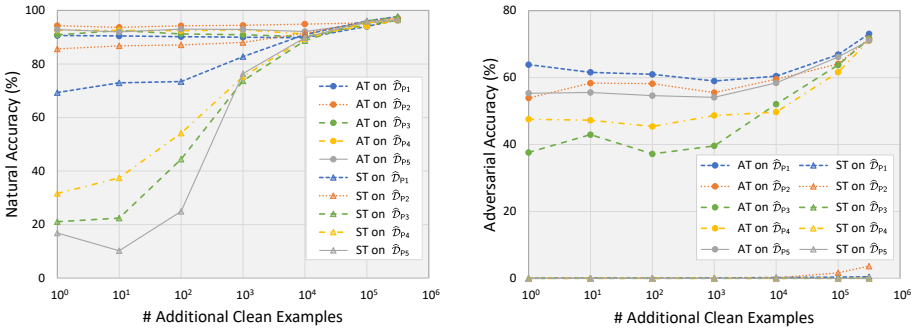


Figure 6: Natural accuracy and adversarial accuracy as a function of the number of additional clean examples on SVHN. The victim classifier is ResNet-18. The threat model is the ℓ_2 with $\epsilon = 0.5$.

Anyway, our results show that adversarial training can successfully exclude the delusive features within the ϵ-ball. Thus natural accuracy can be significantly improved by adversarial training in all cases, as shown in Figure 4. Moreover, we observe that P1 is more destructive than P2 even after adversarial training, which is consistent with our theoretical analysis of the hypothetical setting in Section 4.2.

6.1.2 Evaluation of Transferability

A more realistic setting is to poison different classifiers using the same delusive training examples. We consider various architectures including VGG-19, ResNet-18, ResNet-50, and DenseNet-121 as victim classifiers. The poison data is the same as in Section 6.1.1. Results are defered to Figure 9 in Appendix B. We observe that the attacks have good transferability across the victim classifiers and adversarial training can guarantee 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 mild. This means that different architectures have distinct simplicity biases. Due to space constraints, a detailed investigation is out of the scope of this work.

6.1.3 What if the Threat Model is Under-descriptive?

Our theoretical analysis in Section 4.1 highlights the importance of choosing a proper budget ϵ for adversarial training. We try to explore this situation where the threat model is under-descriptive by varying the defender's budget. Results 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 eliminate the attacks. Empirically, the optimal budget for P3 and P4 is about 0.4 and the optimal budget for P1 and

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

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

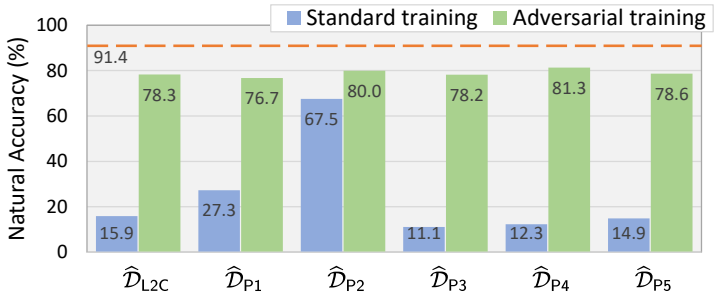


Figure 7: Natural accuracy of models trained on delusive datasets for CIFAR-10. The victim classifier is VGG-11. The threat model is the ℓ_∞ ball with $\epsilon = 0.032$. The horizontal line indicates natural accuracy of a standard model trained on the original training set.

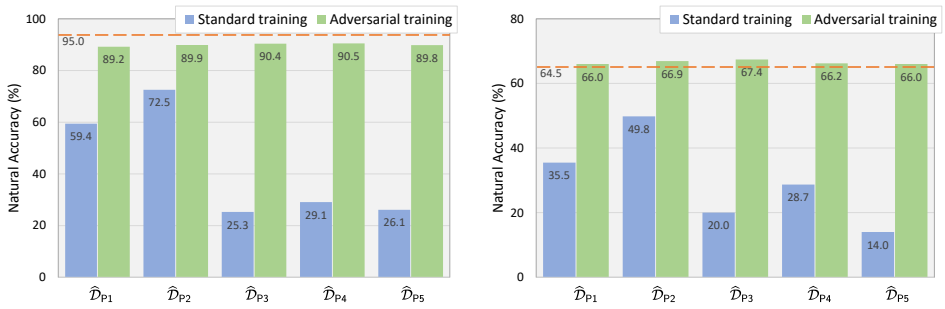
P2 is about 0.25. P5 is the easiest to defend—adversarial training with a small budget (about 0.1) can significantly mitigate the poisoning.

6.1.4 A Simple Countermeasure

In addition to adversarial training, a simple countermeasure is adding clean data to the training set. This will neutralize the poison data and bring it closer to the original distribution. We explore this countermeasure on SVHN because extensive extra training data is 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. Also, the performance of adversarial training can be improved with more data and is better than standard training in most cases. Overall, it is recommend that combining this simple countermeasure with adversarial training to further improve the natural accuracy. Besides the focus on natural accuracy in this work, another interesting measure is the model accuracy on adversarial examples. It shows that adversarial accuracy of the models can be improved with more data. In addition, we observe that different delusive attacks have different effects on the adversarial accuracy. A further study with more focus on adversarial accuracy is therefore suggested.

6.1.5 Comparison with L2C

We compare the heuristic attacks with L2C [11] 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 $\epsilon = 0.1$ is considered. The victim classifier is VGG-11 and ResNet-18 for CIFAR-10 and the two-class ImageNet, respectively. Table 1 shows 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 the attacks P3 and P4 are comparable with L2C and adversarial training can improve natural accuracy in all cases. Similar conclusions hold for the two-class ImageNet (see Appendix B



(a) Pretext task (b) Downstream task

Figure 8: Rotation-based self-supervised learning on CIFAR-10. The victim classifier is ResNet-18. The threat model is the ℓ_2 ball with $\epsilon = 0.5$. The horizontal line indicates natural accuracy of a standard model trained on the original training set.

Model	Test Accuracy on MNIST-CIFAR			MNIST-Randomized Accuracy		
	ST	AT [39]	AT (ours)	ST	AT [39]	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**

Table 2: Adversarial training on MNIST-CIFAR: The table above represents 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.

Figure 10).

6.2 Evaluation on Rotation-based Self-supervised Learning

To further show the versatility of the attacks and defense, we conduct experiments on rotation-based self-supervision [13], a process that learns representations by predicting rotation angles ($0°$, $90°$, $180°$, $270°$). The pretext task is trained on delusive datasets. 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 poorly performed for the downstream task. Surprisingly, the quality of the adversarially learned representations slightly outperforms that of standard models trained on the original training set. This is consistent with recent hypotheses stating that robust models transfer better [34]. In addition, our results suggest that the robustness of self-supervised learning methods [21, 7] against data poisoning is a promising direction for future research.

6.3 Overcoming Simplicity Bias

A recent work [39] 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 "simple" MNIST images $\boldsymbol{x}_m$ and the more complex CIFAR-10 images $\boldsymbol{x}_c$. 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 poisoning, the MNIST-CIFAR dataset can be regarded as a delusive distribution of the original CIFAR distribution. Thus adversarial training can mitigate this

training distribution shift, as Theorem 4.1 pointed out. However, [39] tried adversarial training on MNIST-CIFAR and failed. Contrary to their results, here we show 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 $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x}_m - \boldsymbol{x}'_m\|_\infty \leq 1\}$. The difference is the whole space of the MNIST features is considered as a non-robust region in our choice. Results are summarized in Table 2. We observe that our choice results in models that do not rely on the simple MNIST features, thus the simplicity bias in MNIST-CIFAR can be overcome by adversarial training.

7 Conclusion

In this work, we show that natural accuracy can be guaranteed by adversarial training without abandoning the poisoned examples generated by delusive adversaries, thanks to the fact that minimizing adversarial risk on the *poison data* is equivalent to optimizing an upper bound of natural risk on the *original data*. Both theoretical and empirical results vote for adversarial training when confronted with delusive poisoning. Significantly, our exploration broadens the applicability of adversarial training in practical applications whose primary goal is high accuracy on clean test data.

References

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

[2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases ((ECML-KDD))*, 2013.

[3] B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning (ACML)*, 2011.

[4] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, 2012.

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

[6] H. Chacon, S. Silva, and P. Rad. Deep learning poison data attack detection. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[8] F. Croce, M. Andriushchenko, V. Sehwag, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[9] E. Dobriban, H. Hassani, D. Hong, and A. Robey. Provable tradeoffs in adversarially robust classification. *arXiv preprint arXiv:2006.05161*, 2020.

[10] F. Farokhi. Regularization helps with mitigating poisoning attacks: Distributionally-robust machine learning using the wasserstein distance. *arXiv preprint arXiv:2001.10655*, 2020.

[11] J. Feng, Q.-Z. Cai, and Z.-H. Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

[13] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] W. R. Huang, J. Geiping, L. Fowl, G. Taylor, and T. Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[19] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[20] S. Jetley, N. Lord, and P. Torr. With friends like these, who needs adversaries? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

[22] P. W. Koh, J. Steinhardt, and P. Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

[23] A. Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[24] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia. Adversarial machine learning-industry perspectives. In *IEEE Security and Privacy Workshops (SPW)*, 2020.

[25] W.-A. Lin, C. P. Lau, A. Levine, R. Chellappa, and S. Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[29] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. In *Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[30] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[31] J. Newsome, B. Karp, and D. Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2006.

[32] M. Pezeshki, S.-O. Kaba, Y. Bengio, A. Courville, D. Precup, and G. Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of Computer Vision (IJCV)*, 2015.

[34] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? In *Advances in neural information processing systems (NeurIPS)*, 2020.

[35] H. Salman, A. Ilyas, L. Engstrom, S. Vemprala, A. Madry, and A. Kapoor. Unadversarial examples: Designing objects for robust vision. *arXiv preprint arXiv:2012.12235*, 2020.

[36] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[37] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. *arXiv preprint arXiv:2006.12557*, 2020.

[38] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[39] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

[41] A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations (ICLR)*, 2018.

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

[43] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *European Conference on Computer Vision (ECCV)*, 2018.

[44] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[45] L. Tao and S. Chen. With false friends like these, who can have self-knowledge? *arXiv preprint arXiv:2012.14738*, 2020.

[46] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[47] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning (ICML)*, 2015.

[48] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[49] H. Xu, X. Liu, Y. Li, and J. Tang. To be robust or to be fair: Towards fairness in adversarial training. *arXiv preprint arXiv:2010.06121*, 2020.

[50] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri. A closer look at accuracy vs. robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[51] M. Zhao, B. An, W. Gao, and T. Zhang. Efficient label contamination attacks against black-box learning models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[52] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning (ICML)*, 2019.

[53] S. Zhu, X. Zhang, and D. Evans. Learning adversarially robust representations via worst-case mutual information maximization. In *International Conference on Machine Learning (ICML)*, 2020.

A Experimental Setup

Experiments run with NVIDIA GeForce RTX 2080 Ti GPUs. We report the number with a single run of experiments. Our implementation is based on PyTorch and the code to reproduce our results will be publicly available.

A.1 Datasets and Models

Table 3 reports the parameters used for the datasets and models.

CIFAR-10[4]**.** This dataset consists of 60,000 32×32 colour images (50,000 images for training and 10,000 images for testing) in 10 classes ("airplane", "car", "bird", "cat", "deer", "dog", "frog", "horse", "ship", and "truck"). Early stopping is done with holding out 1000 examples from the training set. We use various architectures for this dataset, including VGG-11, VGG-16, VGG-19 [40], ResNet-18, ResNet-50 [15], and DenseNet-121 [17]. The initial learning rate is set to 0.1. For supervised learning on this dataset, we run 150 epochs on the training set, where we decay the learning rate by a factor 0.1 in the 100th and 125th epochs. For self-supervised learning, we run 70 epochs on the training set, where we decay the learning rate by a factor 0.1 in the 40th and 55th epochs.

SVHN[5]**.** This dataset consists of 630,420 32×32 colour images (73,257 images for training, 26,032 images for testing, and 531,131 additional images to use as extra training data) in 10 classes ("0", "1", "2", "3", "4", "5", "6", "7", "8", and "9"). Early stopping is done with holding out 1000 examples from the training set. We use the ResNet-18 architecture for this dataset. The initial learning rate is set to 0.01. We run 50 epochs on the training set, where we decay the learning rate by a factor 0.1 in the 30th and 40th epochs.

Two-class ImageNet[6]**.** Following [11], this dataset is a subset of ImageNet [33] that consists of 2,700 224×224 colour images (2,600 images for training and 100 images for testing) in 2 classes ("bulbul", "jellyfish"). Early stopping is done with holding out 380 examples from the training set. We use the ResNet-18 architecture for this dataset. The initial learning rate is set to 0.1. We run 100 epochs on the training set, where we decay the learning rate by a factor 0.1 in the 75th and 90th epochs.

MNIST-CIFAR[7]**.** Following [39], this dataset consists of 11,960 64×32 colour images (10,000 images for training and 1,960 images for testing) in 2 classes: images in class -1 and class 1 are vertical concatenations of MNIST digit zero & CIFAR-10 car and MNIST digit one & CIFAR-10 truck images, respectively. We use various architectures for this dataset, including VGG-16, ResNet-50, and DenseNet-121. The initial learning rate is set to 0.05. We run 100 epochs on the training set, where we decay the learning rate by a factor 0.2 in the 50th and 150th epochs and a factor 0.5 in the 100th epoch.

Parameter	CIFAR-10	SVHN	Two-class ImageNet	MNIST-CIFAR
# training examples	50,000	73,257	2,600	10,000
# test examples	10,000	26,032	100	1,960
# features	3,072	3,072	150,528	6,144
# classes	10	10	2	2
learning rate	0.1	0.01	0.1	0.05
momentum	0.9	0.9	0.9	0.9
weight decay	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$

Table 3: Experimental setup and parameters for the each dataset.

[4] https://www.cs.toronto.edu/~kriz/cifar.html
[5] http://ufldl.stanford.edu/housenumbers/
[6] https://github.com/kingfengji/DeepConfuse
[7] https://github.com/harshays/simplicitybiaspitfalls

A.2 Adversarial Training

We perform adversarial training to train robust classifiers following [26]. Specifically, we train against a projected gradient descent (PGD) adversary, starting from a random initial perturbation of the training data. We consider adversarial perturbations in ℓ_p norm where $p = \{2, \infty\}$. Unless otherwise specified, we use the values of ϵ provided in Table 4 to train our models. We use 7 steps of PGD with a step size of $\epsilon/5$.

For overcoming simplicity bias on MNIST-CIFAR, we modify the original ϵ-ball used in [39] (*i.e.*, $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty \leq 0.3\}$) to the whole space of the MNIST features $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x}_m - \boldsymbol{x}'_m\|_\infty \leq 1\}$, where $\boldsymbol{x}$ represents the vertical concatenation of a MNIST image $\boldsymbol{x}_m$ and a CIFAR-10 image $\boldsymbol{x}_c$.

Adversary	CIFAR-10	SVHN	Two-class ImageNet
ℓ_∞	0.032	-	0.1
ℓ_2	0.5	0.5	-

Table 4: Value of ϵ used for adversarial training of each dataset and ℓ_p norm.

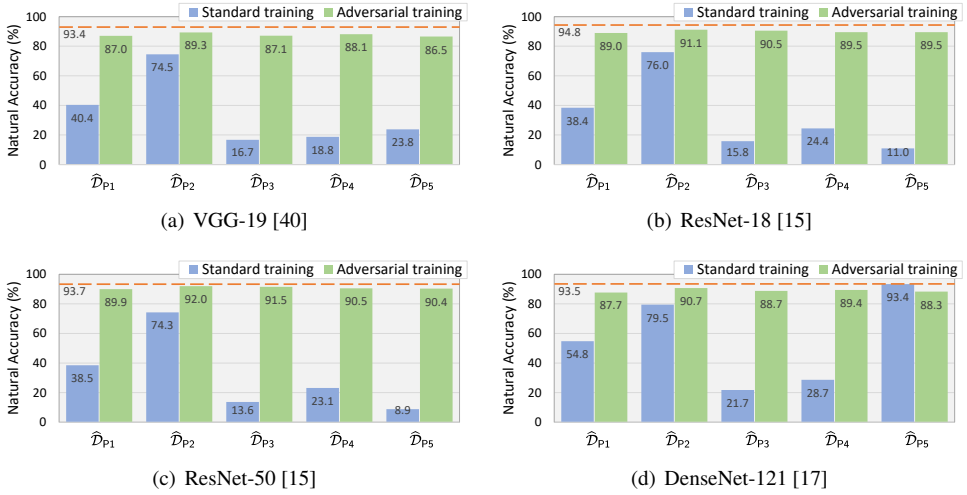
A.3 Delusive Poisoning

In Section 5, we present five heuristic attacks to construct delusive datasets that contain features relevant overly to the labels. Unless otherwise specified, we initialize the poisoned examples as a different randomly chosen sample from the original training set and the values of the attacker's ϵ are the same with adversarial training. To execute `P1` $\sim$ `P4`, we perform normalized gradient descent (ℓ_p-norm of gradient is fixed to be constant at each step). At each step we clip the input to in the $[0, 1]$ range so as to ensure that it is a valid image. To execute `P5`, noises are sampled from Gaussian distribution and then projected to the ball for ℓ_2-norm bounded perturbations; for ℓ_∞-norm bounded perturbations, noises are directly sampled from a uniform distribution. Details on the optimization procedure are shown in Table 5.

Parameter	`P1`	`P2`	`P3`	`P4`	`P5`
step size	$\epsilon/5$	$\epsilon/5$	$\epsilon/5$	$\epsilon/5$	ϵ
iterations	100	100	500	500	1

Table 5: Parameters used for optimization procedure to construct each delusive dataset in Section 5.

B Omitted Figures

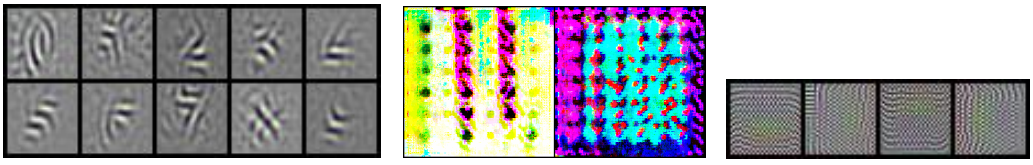


(a) VGG-19 [40] (b) ResNet-18 [15]

(c) ResNet-50 [15] (d) DenseNet-121 [17]

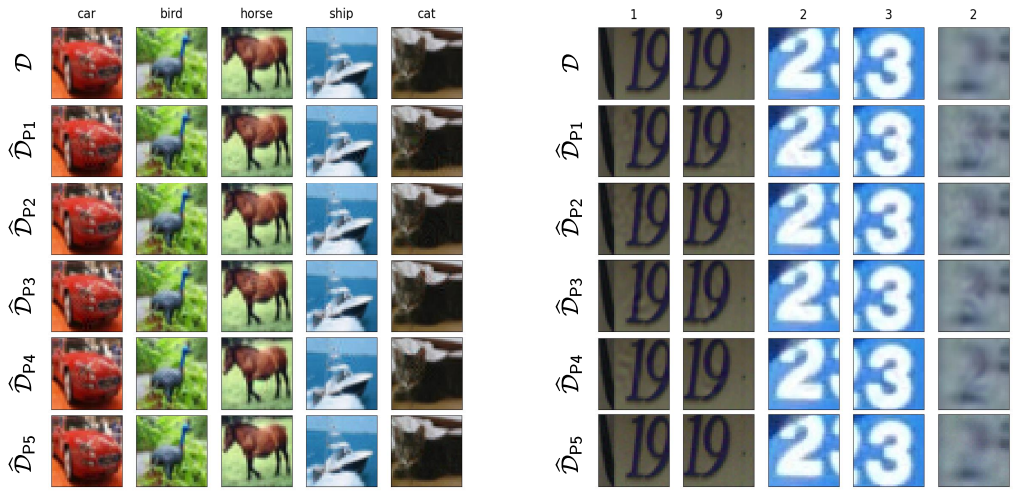
Figure 9: Natural accuracy of models trained on delusive datasets for CIFAR-10. The victim classifiers are VGG-19, ResNet-18, ResNet-50, and DenseNet-121. The threat model is ℓ_2 ball with $\epsilon = 0.5$. The horizontal orange line indicates natural accuracy of a standard model trained on the original training set.

Figure 10: Natural accuracy of models trained on delusive datasets for two-class ImageNet. The victim classifier is ResNet-18. The threat model is ℓ_∞ ball with $\epsilon = 0.1$. The horizontal orange line indicates natural accuracy of a standard model trained on the original training set.



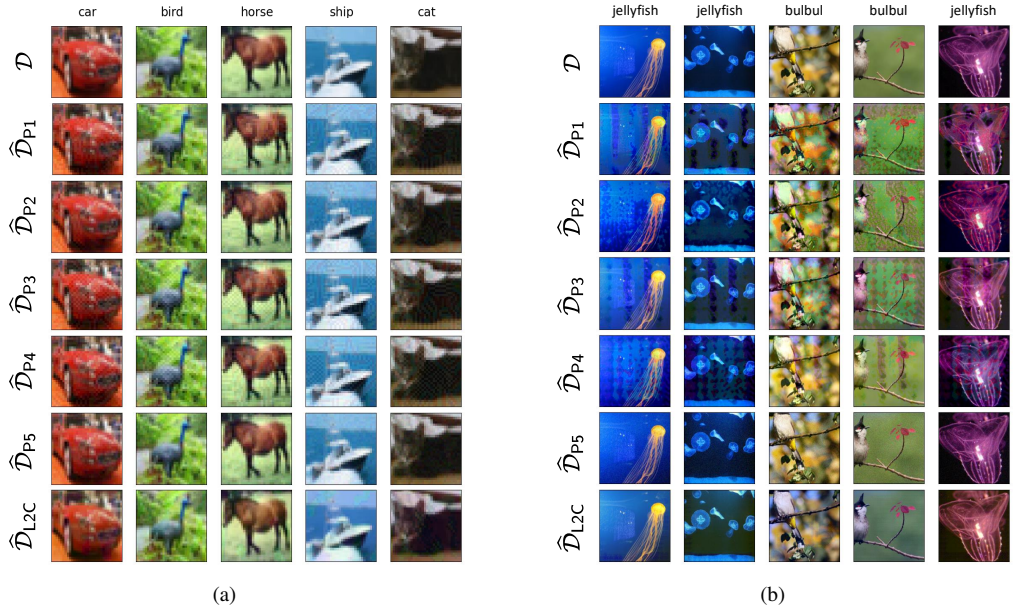
(a) (b) (c)

Figure 11: **Left**: Universal perturbations for P3 and P4 on SVHN. The threat model is the ℓ_2 ball with $\epsilon = 0.5$. **Middle**: Universal perturbations for P3 and P4 on the two-class ImageNet. The threat model is the ℓ_∞ ball with $\epsilon = 0.1$. **Right**: Universal perturbations for P3 and P4 on rotation-based self-supervision with CIFAR-10. The threat model is the ℓ_2 ball with $\epsilon = 0.5$. Perturbations are rescaled to lie in the $[0, 1]$ range for display.



(a) (b)

Figure 12: **Left**: Random samples from the CIFAR-10 training set: the original training $\mathcal{D}$; the poisoned training sets $\widehat{\mathcal{D}}_{\mathsf{P1}}$, $\widehat{\mathcal{D}}_{\mathsf{P2}}$, $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$, and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The threat model is the ℓ_2 ball with $\epsilon = 0.5$. **Right**: First five examples from the SVHN training set: the original training $\mathcal{D}$; the poisoned training sets $\widehat{\mathcal{D}}_{\mathsf{P1}}$, $\widehat{\mathcal{D}}_{\mathsf{P2}}$, $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$, and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The threat model is the ℓ_2 ball with $\epsilon = 0.5$.



(a) (b)

Figure 13: **Left**: Random samples from the CIFAR-10 training set: the original training $\mathcal{D}$; the poisoned training sets $\widehat{\mathcal{D}}_{\mathsf{P1}}$, $\widehat{\mathcal{D}}_{\mathsf{P2}}$, $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$, and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The threat model is the ℓ_∞ ball with $\epsilon = 0.032$. **Right**: First five examples from the two-class ImageNet training set: the original training $\mathcal{D}$; the poisoned training sets $\widehat{\mathcal{D}}_{\mathsf{P1}}$, $\widehat{\mathcal{D}}_{\mathsf{P2}}$, $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$, and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The threat model is the ℓ_∞ ball with $\epsilon = 0.1$.

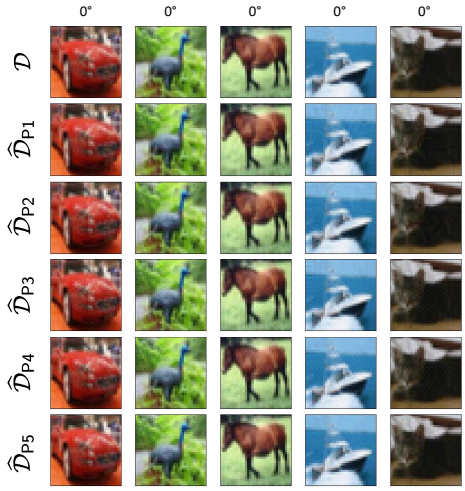


Figure 14: Random samples from the CIFAR-10 training set for rotation-based self-supervised learning: the original training $\mathcal{D}$; the poisoned training sets $\widehat{\mathcal{D}}_{\mathsf{P1}}$, $\widehat{\mathcal{D}}_{\mathsf{P2}}$, $\widehat{\mathcal{D}}_{\mathsf{P3}}$, $\widehat{\mathcal{D}}_{\mathsf{P4}}$, and $\widehat{\mathcal{D}}_{\mathsf{P5}}$. The threat model is the ℓ_2 ball with $\epsilon = 0.5$.

C Proofs

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

C.1 Proof of Theorem 4.1

The main tool in proving our key results is the following lemma, which characterizes the equivalence of adversarial risk and the distributionally robust optimization (DRO) bounded in an ∞-Wasserstein ball. This is proved by Proposition 3.1 in [42] and Lemma 3.3 in [53].

Lemma C.1. *Given a classifier $f : \mathcal{X} \to \mathcal{Y}$, for any data distribution $\mathcal{D}$, we have*

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}'\in\mathcal{B}_\epsilon(\boldsymbol{x},\epsilon)} \ell(f(\boldsymbol{x}'),y)\right] = \max_{\mathcal{D}'\in\mathcal{B}_{\mathrm{W}_\infty}(\widehat{\mathcal{D}},\epsilon)} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(f(\boldsymbol{x}),y)]$$

Theorem 4.1, restated below, shows that adversarial training on the poison data is optimizing an upper bound of natural risk on the original data.

Theorem 4.1. *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*

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

Proof. The first inequality comes from the symmetric of Wasserstein distance:

$$\mathrm{W}_\infty(\mathcal{D},\widehat{\mathcal{D}}) = \mathrm{W}_\infty(\widehat{\mathcal{D}},\mathcal{D}),$$

which means that the original distribution exists in the neighbor of $\widehat{\mathcal{D}}$:

$$\mathcal{D} \in \mathcal{B}_{\mathrm{W}_\infty}(\widehat{\mathcal{D}},\epsilon).$$

Thus the natural risk on the original distribution can be upper bounded by DRO on the delusive distribution.

For the last equality, we simply use the fact in Lemma C.1 that adversarial risk is equivalent to DRO defined with respect to the ∞-Wasserstein distance. This concludes the proof. □

C.2 Proof of Theorem 4.2

We first review the original mixture-Gaussian distribution $\mathcal{D}$, the corresponding delusive distribution $\widehat{\mathcal{D}}_1$, and $\widehat{\mathcal{D}}_2$.

The original mixture-Gaussian distribution $\mathcal{D}$:

$$y \overset{u\cdot a\cdot r}{\sim} \{-1,+1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y\cdot\boldsymbol{\mu},\sigma^2\boldsymbol{I}), \quad \text{where } \boldsymbol{\mu} = (1,\eta,\ldots,\eta) \in \mathbb{R}^{d+1}. \tag{11}$$

The first delusive mixture-Gaussian distribution $\widehat{\mathcal{D}}_1$:

$$y \overset{u\cdot a\cdot r}{\sim} \{-1,+1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y\cdot\widehat{\boldsymbol{\mu}}_1,\sigma^2\boldsymbol{I}), \quad \text{where } \widehat{\boldsymbol{\mu}}_1 = (1,-\eta,\ldots,-\eta) \in \mathbb{R}^{d+1}. \tag{12}$$

The second delusive mixture-Gaussian distribution $\widehat{\mathcal{D}}_2$:

$$y \overset{u\cdot a\cdot r}{\sim} \{-1,+1\}, \quad \boldsymbol{x} \sim \mathcal{N}(y\cdot\widehat{\boldsymbol{\mu}}_2,\sigma^2\boldsymbol{I}), \quad \text{where } \widehat{\boldsymbol{\mu}}_2 = (1,3\eta,\ldots,3\eta) \in \mathbb{R}^{d+1}. \tag{13}$$

Theorem 4.2, restated below, compares the effect of the delusive distributions on natural risk.

Theorem 4.2. *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 $\widehat{\mathcal{D}}_2$, respectively. For any $\eta > 0$, we have*

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

Proof. In the mixture-Gaussian distribution setting described above, the Bayes optimal classifier is linear. In particular, the expression for the classifier of $\mathcal{D}$ is

$$f_{\mathcal{D}}(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} \Pr_{y|\boldsymbol{x}}(y = c) = \operatorname{sign}(\boldsymbol{\mu}^\top \boldsymbol{x}).$$

Similarly, the Bayes optimal classifiers for $\widehat{\mathcal{D}}_1$ and $\widehat{\mathcal{D}}_2$ are respectively given by

$$f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_1^\top \boldsymbol{x}) \quad \text{and} \quad f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_2^\top \boldsymbol{x}).$$

Now we are ready to calculate the natural risk of each classifier. The natural risk of $f_{\mathcal{D}}(\boldsymbol{x})$ is

$$\begin{aligned}
\mathcal{R}_{\mathsf{nat}}(f_{\mathcal{D}}(\boldsymbol{x}), \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} [f_{\mathcal{D}}(\boldsymbol{x}) \neq y] \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\operatorname{sign}(\boldsymbol{\mu}^\top \boldsymbol{x}) \neq y\right] \\
&= \Pr\left[y \cdot \left(\mathcal{N}(y, \sigma^2) + \sum_{i=1}^{d} \eta \mathcal{N}(y\eta, \sigma^2)\right) < 0\right] \\
&= \Pr\left[\mathcal{N}(1, \sigma^2) + \sum_{i=1}^{d} \eta \mathcal{N}(\eta, \sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(1 + d\eta^2, (1 + d\eta^2)\sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(0, 1) > \frac{\sqrt{1 + d\eta^2}}{\sigma}\right].
\end{aligned}$$

The natural risk of $f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x})$ is

$$\begin{aligned}
\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x}), \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x}) \neq y\right] \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\operatorname{sign}(\widehat{\boldsymbol{\mu}}_1^\top \boldsymbol{x}) \neq y\right] \\
&= \Pr\left[\mathcal{N}(1, \sigma^2) - \sum_{i=1}^{d} \eta \mathcal{N}(\eta, \sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(1 - d\eta^2, (1 + d\eta^2)\sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(0, 1) > \frac{1 - d\eta^2}{\sigma\sqrt{1 + d\eta^2}}\right].
\end{aligned}$$

Similarly, the natural risk of $f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x})$ is

$$\begin{aligned}
\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x}), \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x}) \neq y\right] \\
&= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\operatorname{sign}(\widehat{\boldsymbol{\mu}}_2^\top \boldsymbol{x}) \neq y\right] \\
&= \Pr\left[\mathcal{N}(1, \sigma^2) + \sum_{i=1}^{d} 3\eta \mathcal{N}(\eta, \sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(1 + 3d\eta^2, (1 + 9d\eta^2)\sigma^2) < 0\right] \\
&= \Pr\left[\mathcal{N}(0, 1) > \frac{1 + 3d\eta^2}{\sigma\sqrt{1 + 9d\eta^2}}\right].
\end{aligned}$$

Since $\eta > 0$ and $d > 0$, we have $\sqrt{1 + d\eta^2} > \frac{1+3d\eta^2}{\sqrt{1+9d\eta^2}} > \frac{1-d\eta^2}{\sqrt{1+d\eta^2}}$. Therefore, we obtain

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

□

C.3 Proof of Theorem 4.3

We consider the problem of minimizing the adversarial risk on some delusive distribution $\widehat{\mathcal{D}}$ by using a linear classifier. Specifically, this can be formulated as:

$$\min_{\boldsymbol{w},b} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}} \left[\max_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} \mathbb{1}\left(\operatorname{sign}(\boldsymbol{w}^\top(\boldsymbol{x}+\boldsymbol{\xi})+b) \neq y\right)\right], \tag{14}$$

where $\mathbb{1}(\cdot)$ is the indicator function and $\epsilon = 2\eta$, the same budget used by the delusive adversary. Denote by $f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$ the linear classifier.

First, we show that the optimal linear ℓ_∞ robust classifier for $\widehat{\mathcal{D}}_1$ will rely solely on robust features, similar to the cases in Lemma D.5 of [46] and Lemma 2 of [49].

Lemma C.2. *Minimizing the adversarial risk of the loss (14) on the distribution $\widehat{\mathcal{D}}_1$ (12) results in a classifier that assigns 0 weight to features x_i for $i \ge 2$.*

Proof. The adversarial risk on the distribution $\widehat{\mathcal{D}}_1$ can be written as

$$\begin{aligned}
\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1) &= \Pr\left[\exists\, \|\boldsymbol{\xi}\|_\infty \le \epsilon, f(\boldsymbol{x}+\boldsymbol{\xi}) \neq y\right] \\
&= \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} (y \cdot f(\boldsymbol{x}+\boldsymbol{\xi})) < 0\right] \\
&= \Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} (f(\boldsymbol{x}+\boldsymbol{\xi})) > 0 \mid y = -1\right] \cdot \Pr[y=-1] \\
&\quad + \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} (f(\boldsymbol{x}+\boldsymbol{\xi})) < 0 \mid y = +1\right] \cdot \Pr[y=+1] \\
&= \underbrace{\Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} \left(w_1(\mathcal{N}(-1,\sigma^2)+\xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(\eta,\sigma^2)+\xi_i) + b\right) > 0\right]}_{\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(-1)})} \cdot \frac{1}{2} \\
&\quad + \underbrace{\Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \le \epsilon} \left(w_1(\mathcal{N}(1,\sigma^2)+\xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(-\eta,\sigma^2)+\xi_i) + b\right) < 0\right]}_{\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(+1)})} \cdot \frac{1}{2}.
\end{aligned}$$

Then we prove the lemma by contradiction. Consider any optimal solution $\boldsymbol{w}$ for which $w_i < 0$ for some $i \ge 2$, we have

$$\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(-1)}) = \Pr\left[\underbrace{\sum_{j\neq i} \max_{|\xi_j| \le \epsilon} \left(w_j(\mathcal{N}(-\widehat{\mu}_{1,j},\sigma^2)+\xi_j) + b\right)}_{\mathbb{A}} + \underbrace{\max_{|\xi_i| \le \epsilon} \left(w_i(\mathcal{N}(\eta,\sigma^2)+\xi_i)\right)}_{\mathbb{B}} > 0\right].$$

Because $w_i < 0$, $\mathbb{B}$ is maximized when $\xi_i = -\epsilon$. Then, the contribution of terms depending on w_i to $\mathbb{B}$ is a normally-distributed random variable with mean $\eta - \epsilon < 0$. Since the mean is negative, setting w_i to zero can only decrease the risk, contradicting the optimality of $\boldsymbol{w}$. Formally,

$$\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(-1)}) = \Pr\left[\mathbb{A} + w_i \mathcal{N}(\eta-\epsilon, \sigma^2) > 0\right] > \Pr\left[\mathbb{A} > 0\right].$$

We can also assume $w_i > 0$ and similar contradiction holds. Similar argument holds for $\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(+1)})$. Therefore, the adversarial risk is minimized when $w_i = 0$ for $i \ge 2$. □

Different from the case in Lemma C.2, below we show that the optimal linear ℓ_∞ robust classifier for $\widehat{\mathcal{D}}_2$ will rely on both robust and non-robust features.

Lemma C.3. *Minimizing the adversarial risk of the loss (14) on the distribution $\widehat{\mathcal{D}}_2$ (13) results in a classifier that assigns positive weights to features x_i for $i \geq 1$.*

Proof. The adversarial risk on the distribution $\widehat{\mathcal{D}}_1$ can be written as

$$\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2) = \underbrace{\Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(-1, \sigma^2) + \xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(-3\eta, \sigma^2) + \xi_i) + b\right) > 0\right]}_{\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2^{(-1)})} \cdot \frac{1}{2}$$
$$+ \underbrace{\Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(1, \sigma^2) + \xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(3\eta, \sigma^2) + \xi_i) + b\right) < 0\right]}_{\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2^{(+1)})} \cdot \frac{1}{2}.$$

Then we prove the lemma by contradiction. Consider any optimal solution $\boldsymbol{w}$ for which $w_i \leq 0$ for some $i \geq 1$, we have

$$\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2^{(-1)}) = \Pr\left[\underbrace{\sum_{j \neq i} \max_{|\xi_j| \leq \epsilon} \left(w_j(\mathcal{N}(-\widehat{\mu}_{2,j}, \sigma^2) + \xi_j) + b\right)}_{\mathbb{C}} + \underbrace{\max_{|\xi_i| \leq \epsilon} \left(w_i(\mathcal{N}(-\widehat{\mu}_{2,i}, \sigma^2) + \xi_i)\right)}_{\mathbb{D}} > 0\right].$$

Because $w_i \leq 0$, $\mathbb{D}$ is maximized when $\xi_i = -\epsilon$. Then, the contribution of terms depending on w_i to $\mathbb{D}$ is a normally-distributed random variable with mean $-\widehat{\mu}_{2,i} - \epsilon < 0$. Since the mean is negative, setting w_i to be positive can decrease the risk, contradicting the optimality of $\boldsymbol{w}$. Formally,

$$\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2^{(-1)}) = \Pr\left[\mathbb{C} + w_i \mathcal{N}(-\widehat{\mu}_{2,i} - \epsilon, \sigma^2) > 0\right] > \Pr\left[\mathbb{C} + p\mathcal{N}(-\widehat{\mu}_{2,i} - \epsilon, \sigma^2) > 0\right],$$

where $p > 0$ is any positive number. Similar contradiction holds for $\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1^{(+1)})$. Therefore, the optimal solution must assigns positive weights to all features. □

Now we are ready to derive the optimal linear robust classifiers.

Lemma C.4. *For the distribution $\widehat{\mathcal{D}}_1$ (12), the optimal linear ℓ_∞ robust classifier is*

$$f_{\widehat{\mathcal{D}}_1, \mathsf{rob}}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_{1,\mathsf{rob}}^\top \boldsymbol{x}), \quad \textit{where } \widehat{\boldsymbol{\mu}}_{1,\mathsf{rob}} = (1, 0, \ldots, 0).$$

Proof. By Lemma C.2, the robust classifier for the distribution $\widehat{\mathcal{D}}_1$ has zero weight on non-robust features (*i.e.*, $w_i = 0$ for $i \geq 2$). Also, the robust classifier will assign positive weight to the robust feature (*i.e.*, $w_1 > 0$). This is similar to the case in Lemma C.3 and we omit the proof here. Therefore, the adversarial risk on the distribution $\widehat{\mathcal{D}}_1$ can be simplified by solving the inner maximization problem first. Formally,

$$\begin{aligned}
\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_1) &= \Pr\left[\exists\, \|\boldsymbol{\xi}\|_\infty \leq \epsilon, f(\boldsymbol{x} + \boldsymbol{\xi}) \neq y\right] \\
&= \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} (y \cdot f(\boldsymbol{x} + \boldsymbol{\xi})) < 0\right] \\
&= \Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} (f(\boldsymbol{x} + \boldsymbol{\xi})) > 0 \mid y = -1\right] \cdot \Pr\left[y = -1\right] \\
&\quad + \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} (f(\boldsymbol{x} + \boldsymbol{\xi})) < 0 \mid y = +1\right] \cdot \Pr\left[y = +1\right] \\
&= \Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(-1, \sigma^2) + \xi_1) + b\right) > 0\right] \cdot \Pr\left[y = -1\right] \\
&\quad + \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(1, \sigma^2) + \xi_1) + b\right) < 0\right] \cdot \Pr\left[y = +1\right] \\
&= \Pr\left[w_1 \mathcal{N}(\epsilon - 1, \sigma^2) + b > 0\right] \cdot \Pr\left[y = -1\right] \\
&\quad + \Pr\left[w_1 \mathcal{N}(1 - \epsilon, \sigma^2) + b < 0\right] \cdot \Pr\left[y = +1\right],
\end{aligned}$$

which is equivalent to the natural risk on a mixture-Gaussian distribution $\widehat{\mathcal{D}}_1^*$: $\boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_1^*, \sigma^2 \boldsymbol{I})$, where $\widehat{\boldsymbol{\mu}}_1^* = (1-\epsilon, 0, \ldots, 0)$. The Bayes optimal classifier for $\widehat{\mathcal{D}}_1^*$ is $f_{\widehat{\mathcal{D}}_1^*}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_1^{*\top} \boldsymbol{x})$. Specifically,

$$\begin{aligned}
\mathcal{R}_{\mathsf{nat}}(f, \widehat{\mathcal{D}}_1^*) &= \Pr\left[f(\boldsymbol{x}) \neq y\right] \\
&= \Pr\left[y \cdot f(\boldsymbol{x}) < 0\right] \\
&= \Pr\left[w_1 \mathcal{N}(\epsilon - 1, \sigma^2) + b > 0\right] \cdot \Pr\left[y = -1\right] \\
&\qquad + \Pr\left[w_1 \mathcal{N}(1 - \epsilon, \sigma^2) + b < 0\right] \cdot \Pr\left[y = +1\right],
\end{aligned}$$

which can be minimized when $w_1 = 1 - \epsilon$ and $b = 0$. At the same time, $f_{\widehat{\mathcal{D}}_1^*}(\boldsymbol{x})$ is equivalent to $f_{\widehat{\mathcal{D}}_1, \mathsf{rob}}(\boldsymbol{x})$, since $\operatorname{sign}((1-\epsilon)x_1) = \operatorname{sign}(x_1)$. This concludes the proof of the lemma.

□

Lemma C.5. *For the distribution $\widehat{\mathcal{D}}_2$ (13), the optimal linear ℓ_∞ robust classifier is*

$$f_{\widehat{\mathcal{D}}_2, \mathsf{rob}}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_{2,\mathsf{rob}}^\top \boldsymbol{x}), \quad \textit{where } \widehat{\boldsymbol{\mu}}_{2,\mathsf{rob}} = (1 - 2\eta, \eta, \ldots, \eta).$$

Proof. By Lemma C.3, the robust classifier for the distribution $\widehat{\mathcal{D}}_2$ has positive weight on all features (*i.e.*, $w_i > 0$ for $i \geq 1$). Therefore, the adversarial risk on the distribution $\widehat{\mathcal{D}}_2$ can be simplified by solving the inner maximization problem first. Formally,

$$\begin{aligned}
\mathcal{R}_{\mathsf{adv}}(f, \widehat{\mathcal{D}}_2) &= \Pr\left[\exists\, \|\boldsymbol{\xi}\|_\infty \leq \epsilon, f(\boldsymbol{x} + \boldsymbol{\xi}) \neq y\right] \\
&= \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} (y \cdot f(\boldsymbol{x} + \boldsymbol{\xi})) < 0\right] \\
&= \Pr\left[\max_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(-1, \sigma^2) + \xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(-3\eta, \sigma^2) + \xi_i) + b\right) > 0\right] \cdot \Pr\left[y = -1\right] \\
&\qquad + \Pr\left[\min_{\|\boldsymbol{\xi}\|_\infty \leq \epsilon} \left(w_1(\mathcal{N}(1, \sigma^2) + \xi_1) + \sum_{i=2}^{d+1} w_i(\mathcal{N}(3\eta, \sigma^2) + \xi_i) + b\right) < 0\right] \cdot \Pr\left[y = +1\right] \\
&= \Pr\left[\max_{|\xi_1| \leq \epsilon} \big(w_1(\mathcal{N}(-1, \sigma^2) + \xi_1)\big) + \sum_{i=2}^{d+1} \max_{|\xi_i| \leq \epsilon} \big(w_i(\mathcal{N}(-3\eta, \sigma^2) + \xi_i)\big) + b > 0\right] \cdot \Pr\left[y = -1\right] \\
&\qquad + \Pr\left[\min_{|\xi_1| \leq \epsilon} \big(w_1(\mathcal{N}(1, \sigma^2) + \xi_1)\big) + \sum_{i=2}^{d+1} \min_{|\xi_i| \leq \epsilon} \big(w_i(\mathcal{N}(3\eta, \sigma^2) + \xi_i)\big) + b < 0\right] \cdot \Pr\left[y = +1\right] \\
&= \Pr\left[w_1 \mathcal{N}(\epsilon - 1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(\epsilon - 3\eta, \sigma^2) + b > 0\right] \cdot \Pr\left[y = -1\right] \\
&\qquad + \Pr\left[w_1 \mathcal{N}(1 - \epsilon, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(3\eta - \epsilon, \sigma^2) + b < 0\right] \cdot \Pr\left[y = +1\right] \\
&= \Pr\left[w_1 \mathcal{N}(2\eta - 1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(-\eta, \sigma^2) + b > 0\right] \cdot \Pr\left[y = -1\right] \\
&\qquad + \Pr\left[w_1 \mathcal{N}(1 - 2\eta, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(\eta, \sigma^2) + b < 0\right] \cdot \Pr\left[y = +1\right],
\end{aligned}$$

which is equivalent to the natural risk on a mixture-Gaussian distribution $\widehat{\mathcal{D}}_2^*$: $\boldsymbol{x} \sim \mathcal{N}(y \cdot \widehat{\boldsymbol{\mu}}_2^*, \sigma^2 \boldsymbol{I})$, where $\widehat{\boldsymbol{\mu}}_2^* = (1 - 2\eta, \eta, \ldots, \eta)$. The Bayes optimal classifier for $\widehat{\mathcal{D}}_2^*$ is $f_{\widehat{\mathcal{D}}_2^*}(\boldsymbol{x}) = \operatorname{sign}(\widehat{\boldsymbol{\mu}}_2^{*\top} \boldsymbol{x})$.

Specifically,

$$\begin{aligned}\mathcal{R}_{\mathsf{nat}}(f, \widehat{\mathcal{D}}_2^*) &= \Pr\left[f(\boldsymbol{x}) \neq y\right] \\ &= \Pr\left[y \cdot f(\boldsymbol{x}) < 0\right] \\ &= \Pr\left[w_1 \mathcal{N}(2\eta - 1, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(-\eta, \sigma^2) + b > 0\right] \cdot \Pr\left[y = -1\right] \\ &\quad + \Pr\left[w_1 \mathcal{N}(1 - 2\eta, \sigma^2) + \sum_{i=2}^{d+1} w_i \mathcal{N}(\eta, \sigma^2) + b < 0\right] \cdot \Pr\left[y = +1\right],\end{aligned}$$

which can be minimized when $w_1 = 1 - 2\eta$, $w_i = \eta$ for $i \geq 2$, and $b = 0$. Also, $f_{\widehat{\mathcal{D}}_2^*}(\boldsymbol{x})$ is equivalent to $f_{\widehat{\mathcal{D}}_2, \mathsf{rob}}(\boldsymbol{x})$. This concludes the proof of the lemma.

□

We have established that the optimal linear classifiers $f_{\widehat{\mathcal{D}}_1, \mathsf{rob}}$ and $f_{\widehat{\mathcal{D}}_2, \mathsf{rob}}$ in adversarial training. Now we are ready to compare their natural risks with standard classifiers. Theorem 4.3, restated below, indicates that adversarial training can mitigate the effect of delusive attacks.

Theorem 4.3. *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 $\widehat{\mathcal{D}}_2$, respectively. For any $0 < \eta < 1/3$, we have*

$$\begin{aligned}\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1, \mathsf{rob}}, \mathcal{D}), \\ \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2, \mathsf{rob}}, \mathcal{D}).\end{aligned}$$

Proof. The natural risk of $f_{\widehat{\mathcal{D}}_{1,\mathsf{rob}}}(\boldsymbol{x})$ is

$$\begin{aligned}\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_{1,\mathsf{rob}}}(\boldsymbol{x}), \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[f_{\widehat{\mathcal{D}}_{1,\mathsf{rob}}}(\boldsymbol{x}) \neq y\right] \\ &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\operatorname{sign}(\widehat{\boldsymbol{\mu}}_{1,\mathsf{rob}}^\top \boldsymbol{x}) \neq y\right] \\ &= \Pr\left[\mathcal{N}(1, \sigma^2) < 0\right] \\ &= \Pr\left[\mathcal{N}(0,1) > \frac{1}{\sigma}\right].\end{aligned}$$

Similarly, the natural risk of $f_{\widehat{\mathcal{D}}_{2,\mathsf{rob}}}(\boldsymbol{x})$ is

$$\begin{aligned}\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_{2,\mathsf{rob}}}(\boldsymbol{x}), \mathcal{D}) &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[f_{\widehat{\mathcal{D}}_{2,\mathsf{rob}}}(\boldsymbol{x}) \neq y\right] \\ &= \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\operatorname{sign}(\widehat{\boldsymbol{\mu}}_{2,\mathsf{rob}}^\top \boldsymbol{x}) \neq y\right] \\ &= \Pr\left[(1-2\eta)\mathcal{N}(1, \sigma^2) + \sum_{i=1}^{d} \eta \mathcal{N}(\eta, \sigma^2) < 0\right] \\ &= \Pr\left[\mathcal{N}(1 - 2\eta + d\eta^2, ((1-2\eta)^2 + d\eta^2)\sigma^2) < 0\right] \\ &= \Pr\left[\mathcal{N}(0,1) > \frac{1 - 2\eta + d\eta^2}{\sigma\sqrt{(1-2\eta)^2 + d\eta^2}}\right].\end{aligned}$$

Recall that the natural risk of the standard classifier $f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x})$ is

$$\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}(\boldsymbol{x}), \mathcal{D}) = \Pr\left[\mathcal{N}(0,1) > \frac{1 - d\eta^2}{\sigma\sqrt{1 + d\eta^2}}\right],$$

and the natural risk of the standard classifier $f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x})$ is

$$\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}(\boldsymbol{x}), \mathcal{D}) = \Pr\left[\mathcal{N}(0,1) > \frac{1+3d\eta^2}{\sigma\sqrt{1+9d\eta^2}}\right].$$

It is easy to see that $\frac{1-d\eta^2}{\sqrt{1+d\eta^2}} < 1$. Thus, we have

$$\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_1}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_{1,\mathsf{rob}}}, \mathcal{D}).$$

Also, $\frac{1+3d\eta^2}{\sqrt{1+9d\eta^2}} < \frac{1-2\eta+d\eta^2}{\sqrt{(1-2\eta)^2+d\eta^2}}$ is true when $0 < \eta < 1/3$ and $d > 0$. Therefore, we obtain

$$\mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_2}, \mathcal{D}) > \mathcal{R}_{\mathsf{nat}}(f_{\widehat{\mathcal{D}}_{2,\mathsf{rob}}}, \mathcal{D}).$$

This concludes the proof.

□