

TOWARDS ROBUST DATASET LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

We focus on addressing the problem of learning a robust dataset such that *any* classifier *naturally* trained on the dataset is adversarially robust. Such a dataset benefits the downstream tasks as natural training is much faster than adversarial training, and demonstrates that the desired property of robustness is transferable between models and data. In this work, we propose a principled tri-level optimization to formulate the robust dataset learning problem. We show that, under an abstraction model that characterizes robust *vs.* non-robust features, the proposed method provably learns a robust dataset. Extensive experiments on benchmark datasets demonstrate the effectiveness of our new algorithm with different network initializations and architectures.

1 INTRODUCTION

Deep learning models are vulnerable to adversarial examples (Szegedy et al., 2014; Biggio et al., 2013): an adversary can arbitrarily manipulate the prediction results of deep neural networks with slight perturbations to the data. Many defense approaches, including heuristic defenses (Papernot et al., 2016; Xie et al., 2017; Kannan et al., 2018; Liao et al., 2018; Carmon et al., 2019; Mustafa et al., 2019; Zhang et al., 2019; 2020b; Wu et al., 2020; Dong et al., 2020; Tramer et al., 2020) and certified defenses (Raghunathan et al., 2018; Wong & Kolter, 2018; Wong et al., 2018; Singh et al., 2018; Xiao et al., 2018; Gowal et al., 2018; Lecuyer et al., 2019; Croce et al., 2019; Li et al., 2019; Cohen et al., 2019; Zhang et al., 2020a; Xu et al., 2020; Zhai et al., 2020; Balunović & Vechev, 2020; Zhang et al., 2021), have been developed to protect deep learning models from these threats. The focus of this paper is on integrating the property of adversarial robustness into a dataset, such that a robust model (against small perturbation to the original test data) can be obtained through *natural training* on the learned dataset.

There are several reasons of studying this problem. 1) Discovered by Ilyas et al. (2019) in their seminal work, the desirable property of adversarial robustness is transferable between models and data. We propose a *principled* approach of robust feature extraction with theoretical guarantees and improved empirical performance. 2) Expensive computational cost of most existing defenses hinders their applicability to the scenarios of limited computational resources. Although the task of learning robust dataset might itself be time-consuming, once the *one-time* task has been outsourced to Machine Learning as a Service (MLaaS), one can benefit from the robust dataset for fast training of their own customized robust models, as natural training only requires light computational cost. 3) Distributing a robust dataset is more flexible than distributing a robust model. This is because loading a robust model requires extensive compatibility among deep learning framework (e.g., PyTorch, TensorFlow, MXNet, Keras, etc.), network architecture, and checkpoint. On the other hand, distributing a robust dataset allows everyone to train a network with their preferred architecture and deep learning framework for downstream tasks. Moreover, a robust dataset can be small, e.g., of size only 10% of the original dataset, making it easy to transmit.

However, there are only few works on robust dataset learning. A related but orthogonal research topic is dataset distillation (Wang et al., 2018; Cazenavette et al., 2022), which aims at reducing the scale of dataset by distilling knowledge from a large dataset to a small dataset. Despite dataset distillation has become a popular research topic in machine learning with various applications (Bohdal et al., 2020; Nguyen et al., 2020; Sucholutsky & Schonlau, 2021; Zhao et al., 2021; Zhao & Bilen, 2021; Nguyen et al., 2021; Cazenavette et al., 2022), how to learn a *robust* dataset is less explored. To our best knowledge, the only attempt on building a robust dataset is by Ilyas et al. (2019) on robust feature extraction. Thus, more theoretical and empirical results are desired for an in-depth understanding of robust dataset learning.

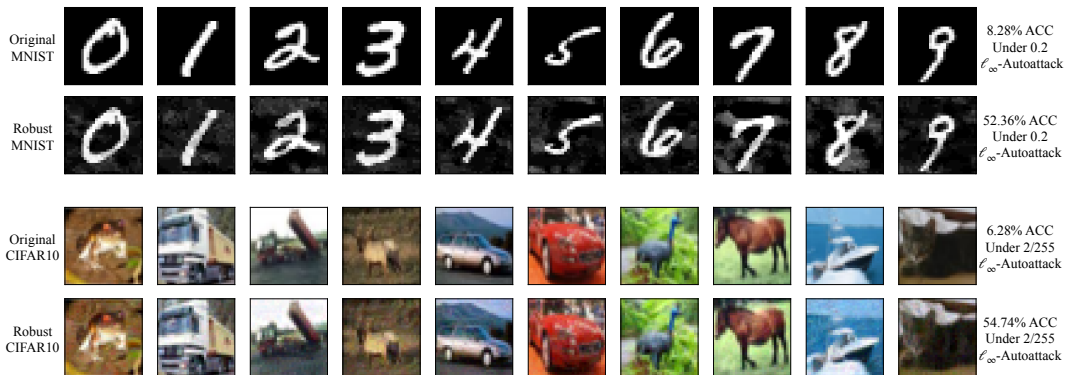


Figure 1: Illustration of original and robust images. For MNIST and CIFAR10, the first row represents the original images, while the second row represents the robust dataset generated by our algorithm. The rightmost column shows the robust accuracy.

The idea behind our robust dataset learning is to represent a classifier as a function of a dataset, so that one can treat the dataset as a learnable parameter of the classifier. Throughout the paper, we name such a classifier the *data-parameterized classifier*. We formulate robust dataset learning as a min-max, tri-level optimization problem. We theoretically show the efficiency of our algorithm, and empirically verify our robust dataset learning algorithm on MNIST, CIFAR10, and TinyImageNet datasets.

Summary of contributions. Our work tackles the problem of robust dataset learning and advances the area in the following ways.

- Algorithmically, we formulate robust dataset learning via a tri-level optimization problem, where we parameterize the model weights by data, find adversarial examples of the model, and optimize adversarial loss over the data. This learning objective encourages the algorithm to maximize both clean and robust accuracy of the data-parameterized classifier.
- Theoretically, we investigate this tri-level optimization problem under an abstraction model that characterizes robust vs. non-robust features (Tsipras et al., 2019), where the objective is to find a dataset that minimizes robust error on the data-parameterized classifier. We show that while the classifier naturally trained on clean dataset is non-robust, our data-parameterized classifier (trained on the robust dataset) is provably robust.
- Experimentally, we evaluate the clean and robust accuracy of our algorithm on MNIST, CIFAR10, and TinyImageNet. We consider baselines for robust dataset learning, which use datasets generated through adversarial attacks or robust feature extraction (Ilyas et al., 2019). We show that our algorithm outperforms the baselines by a large margin. For example, on the CIFAR10 dataset with 0.25 ℓ_2 threat model, our method achieves robust accuracy as high as 59.52% under AutoAttack, beating the state-of-the-art 48.20% in the same setting by a large margin.

2 RELATED WORKS

Dataset distillation. The ultimate goal of dataset distillation is to reduce training dataset size by distilling knowledge from the data. Wang et al. (2018) proposed the first dataset distillation algorithm, which expressed the model parameters with the distilled images and optimized the images using gradient descent method. The subsequent works significantly improved the results by various strategies, such as learning soft labels (Sucholutsky & Schonlau, 2021), strengthening learning signal through gradient matching (Zhao et al., 2021), adopting differentiable Siamese augmentation (Zhao & Bilen, 2021), optimizing with the neural tangent kernel under infinite-width limit (Nguyen et al., 2021), and matching training trajectories (Cazenavette et al., 2022). Dataset distillation has been applied to many machine learning fields, including federated learning (Sucholutsky & Schon-

lau, 2020; Zhou et al., 2020), privacy-preserving ML (Li et al., 2020), and neural architecture search Zhao & Bilen (2021).

Robust feature extraction. Ilyas et al. (2019) studied the existence and pervasiveness of adversarial examples. They theoretically demonstrated that adversarial examples are related to the presence of non-robust features, which are in fact highly predictive to neural networks, but brittle and imperceptible to humans. They proposed an empirical algorithm to separate the robust and non-robust features in the data to verify their theoretical results. A by-product of their algorithm is to obtain a robust model through natural training on the robust features. However, they did not provide a principled approach for robust dataset learning.

3 LEARNING ROBUST DATASET: A PRINCIPLED APPROACH

In this section, we propose a principled optimization formulation for robust dataset learning.

Problem settings. There are two essential settings for the robust dataset learning problem: 1) construct a dataset; 2) obtain a robust model with only natural training on the constructed dataset.

Goal. Given a original dataset X^{nat} , The optimization objective is to find an optimal robust dataset X^{rob} such that the neural network that is naturally trained on X^{rob} is robust against adversarial perturbation to the clean test data, where the test data follows the same distribution as X^{nat} .

Optimization. The idea behind our method is to represent a classifier as a function of a dataset and to find the optimal dataset such that the classifier is robust against adversarial perturbations. With this idea, we formulate robust dataset learning as a tri-level optimization problem. Denote by $X^{\text{nat}} := \{(x_i, y_i)\}_{i=1}^n$ the original training data pairs, and by $X^{\text{rob}} := \{(x_i^{\text{rob}}, y_i)\}_{i=1}^n$ the robust dataset to be learned. Notice we only optimize the data points $\{x_i^{\text{rob}}\}_{i=1}^n$ and keep the labels $\{y_i\}_{i=1}^n$ unchanged. Step 1). For a given loss \mathcal{L} , in the first level we create a data-parameterized classifier $f_{\theta(X^{\text{rob}})}$ through minimizing the loss on X^{rob} , which is initialized by X^{nat} and updated by gradient descent; Step 2). In the second level, we calculate the adversarial samples $X^{\text{adv}} := \{(x_i^{\text{adv}}, y_i)\}_{i=1}^n$ of X^{nat} by attacking $f_{\theta(X^{\text{rob}})}$; Step 3). In the third level, we search for the optimal X^{rob} which minimizes the loss of $f_{\theta(X^{\text{rob}})}$ on X^{adv} . With the above steps, our optimization problem is given by

$$\min_{X^{\text{rob}}} \frac{1}{n} \sum_{i=1}^n \max_{x_i^{\text{adv}} \in \mathcal{B}(x_i, \epsilon)} \mathcal{L}(f_{\theta(X^{\text{rob}})}(x_i^{\text{adv}}), y_i), \text{ s.t. } \theta(X^{\text{rob}}) = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i), \quad (1)$$

where f_{θ} stands for a neural network parameterized by θ , and \mathcal{L} is a loss function, e.g., the cross entropy loss or hinge loss.

Efficient algorithm. However, Eq. 1 might be hard to be solved as it is a tri-level optimization problem. The main difficulty is to find the closed form of the parameterized weight $\theta(\{x_i^{\text{rob}}\}_{i=1}^n)$ which minimizes the loss w.r.t. $\{(x_i^{\text{rob}}, y_i)\}_{i=1}^n$. We consider the second-order Taylor expansion of the loss $\mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i)$ at a predefined $\theta = \theta_0$ (e.g., a random set of parameters or the parameters of a naturally trained neural network) for a small $t > 0$:

$$\mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i) \approx \mathcal{L}(f_{\theta_0}(x_i^{\text{rob}}), y_i) + \langle \theta - \theta_0, \nabla_{\theta} \mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i)|_{\theta=\theta_0} \rangle + \frac{1}{2t} \|\theta - \theta_0\|_2^2,$$

which implies that

$$\theta(\{x_i^{\text{rob}}\}_{i=1}^n) = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i) \approx \theta_0 - t \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} \mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i)|_{\theta=\theta_0}.$$

Therefore, we have the following approximation of Eq. 1:

$$\min_{X^{\text{rob}}} \frac{1}{n} \sum_{i=1}^n \max_{x_i^{\text{adv}} \in \mathcal{B}(x_i, \epsilon)} \mathcal{L}(f_{\theta(X^{\text{rob}})}(x_i^{\text{adv}}), y_i), \text{ s.t. } \theta(X^{\text{rob}}) = \theta_0 - t \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} \mathcal{L}(f_{\theta}(x_i^{\text{rob}}), y_i)|_{\theta=\theta_0}. \quad (2)$$

To solve the inner maximization problem, we apply PGD-attack (Madry et al., 2017) via repeatedly using

$$x_i^{\text{adv}} \leftarrow \operatorname{Clip}_{\mathcal{B}(x_i^{\text{nat}}, \epsilon)}(x_i^{\text{adv}} + \alpha \operatorname{sign}(\nabla_{x_i^{\text{adv}}} \mathcal{L}(f_{\theta}(x_i^{\text{adv}}), y_i))),$$

Algorithm 1: Robust dataset learning.

Input: original training set X^{nat} ; number of training epochs T ; classifier f (with weight θ), initialized by θ_0 ; learning rate γ of classifier; learning rate β of robust dataset; PGD steps for generating adversarial example s ; PGD steps size α ; PGD attack budgets ϵ . initialize classifier weights θ with θ_0 , initialize X^{rob} with X^{nat} ;

for $1:T$ **do**

for mini-batches $(b^{\text{nat}}, y^{\text{batch}}) \subseteq X^{\text{nat}}$ and $(b^{\text{rob}}, y^{\text{batch}}) \subseteq X^{\text{rob}}$ **do**

step 1. update classifier with robust data:

$\theta \leftarrow \theta - \gamma \frac{1}{|b^{\text{rob}}|} \sum_{(x,y) \in (b^{\text{rob}}, y^{\text{batch}})} \nabla_{\theta} \mathcal{L}(f_{\theta}(x), y)$;

step 2. calculate adversarial examples from original training set via PGD attack:

 for each $(x^{\text{nat}}, y) \in (b^{\text{nat}}, y^{\text{batch}})$, initialize x^{adv} with x^{nat} ;

for $1:s$ **do**

 generate perturbation through FSGM and clip it within a ball centered at x^{nat} with radius ϵ : $x^{\text{adv}} \leftarrow \text{Clip}_{B(x^{\text{nat}}, \epsilon)}(x^{\text{adv}} + \alpha \text{sign}(\nabla_{x^{\text{adv}}} \mathcal{L}(f_{\theta}(x^{\text{adv}}), y)))$;

end

step 3. for each $x^{\text{rob}} \in b^{\text{rob}}$, update robust data by minimizing robust error:

$x^{\text{rob}} \leftarrow x^{\text{rob}} - \beta \text{sign}(\frac{1}{|b^{\text{rob}}|} \sum_{(x^{\text{adv}}, y)} \nabla_{x^{\text{rob}}} \mathcal{L}(f_{\theta}(x^{\text{adv}}), y))$;

end

end

return robust dataset X^{rob} .

where ϵ is the attack budget, $B(x_i^{\text{nat}}, \epsilon)$ stands for a ball centered at x_i^{nat} with radius ϵ , and α is the step size of the PGD-attack. We use gradient descent to optimize the robust dataset:

$$x_i^{\text{rob}} \leftarrow x_i^{\text{rob}} - \beta \text{sign}\left(\frac{1}{n} \sum_{i=1}^n \nabla_{x_i^{\text{rob}}} \mathcal{L}(f_{\theta}(\{x_i^{\text{rob}}\}_{i=1}^n)(x_i^{\text{adv}}, y_i)\right),$$

where β is the learning rate for the robust data. Notice that since we focus on the image classification tasks, we apply fast sign gradient method to modify the data. This method can be replaced by other gradient descent methods for a specific task. The details of our robust learning algorithm is shown in Algorithm 1.

Difference between our work and Ilyas et al. (2019). Ilyas et al. (2019) formulated the robust feature extraction problem by finding $\{(x_i^{\text{rob}}, y_i)\}_{i=1}^n$ that minimizes $\|g(x_i^{\text{rob}}) - g(x_i)\|_2$, where g is a robust pre-trained representation model trained by adversarial training, and a proper initialization of x_i^{rob} is involved to avoid converging to a trivial solution x_i . As we will see in Sections 4 and 5, our new formulation Eq. 1 not only provably extracts the robust feature under a simple abstraction model, but also enjoys an improved empirical performance on many experiment settings compared with Ilyas et al. (2019).

4 LEARNING ROBUST DATASET: A GLIMPSE ON AN ABSTRACTION MODEL

In this section, we present a fairly simple theoretical model to analyze the above-mentioned robust dataset learning problem. Our analysis is structured as follows: In Sec. 4.1, we introduce the problem settings of the data distribution, classifier, and the optimization objective of the robust dataset learning problem; In Sec. 4.2, we prove that the optimal classifier trained on clean dataset can be non-robust; In Sec. 4.3, we demonstrate that our optimization objective leads to a robust dataset. All proof details can be found in Appendix A.

4.1 PROBLEM SETTINGS

Setting. We consider the data distribution commonly used in prior works (Tsipras et al., 2019; Ilyas et al., 2019), where the instance $\mathbf{x} = (x_1, \dots, x_{d+1}) \sim \mathcal{D}$ and the label y follow:

$$y \sim \text{Uniform}\{-1, 1\}, \quad x_1 \sim \begin{cases} y, & \text{with probability } p; \\ -y, & \text{with probability } 1 - p, \end{cases} \quad x_i \sim \mathcal{N}(\mu y, 1), \quad i = 2, 3, \dots, d+1, \quad (3)$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean $\mu = o(1)$ and variance σ^2 . It consists of strongly-correlated (to the label) feature x_1 and weakly-correlated features x_2, \dots, x_{d+1} (as μ is selected small enough). The strongly-correlated feature is robust against $\Theta(1)$ perturbations but the weakly-correlated features are not. Thus, a naturally trained classifier on this distribution is non-robust as it puts heavy weights on non-robust features. Besides, it is easy to achieve high natural accuracy on this data distribution. For example, we can set $\mu = \Theta(1/\sqrt{d})$ such that a simple classifier, e.g., $\text{sign}(x_2)$, can achieve at least 99% natural accuracy. The probability p quantifies the correlation between the feature x_1 and the label y . We can set p to be moderately large, i.e., $p = 0.97$. We note that the same data distribution was used to demonstrate the trade-off between robustness and accuracy (Tsipras et al., 2019) and to provide a clean abstraction of robust and non-robust features (Ilyas et al., 2019). In this paper, we use the same distribution to show a separation between natural training on the clean dataset and on the robust dataset returned by our algorithm.

We model the natural training by a soft SVM loss (a.k.a. the hinge loss) of a linear classifier:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}) := \mathbb{E}_{(\mathbf{x} \sim \mathcal{D}, y)} [\max\{0, 1 - y\mathbf{w}^T \mathbf{x}\}] + \lambda \|\mathbf{w}\|_2^2, \quad (4)$$

where $\mathbf{w} = (w_1, \dots, w_{d+1})$ is the weight vector and $\lambda > 0$ is a regularization parameter. For a given distribution \mathcal{D}' , the data-parameterized classifier (w.r.t. \mathcal{D}') is given by $\text{sign}(\mathbf{w}_{\mathcal{D}'}^T \mathbf{x})$, where $\mathbf{w}_{\mathcal{D}'} := \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}')$ is the optimal weight of the SVM w.r.t. \mathcal{D}' .

Goal. Our goal is to create a robust data distribution¹ \mathcal{D}' , such that $\mathbf{w}_{\mathcal{D}'}$ is robust. We formulate this problem via our proposed tri-level optimization framework Eq. 1, where the algorithm is supposed to find \mathcal{D}' that minimizes the adversarial loss w.r.t. the weight $\mathbf{w}_{\mathcal{D}'}$ and the worst-case perturbation δ . In particular, we define our robust dataset learning problem as

$$\min_{\mathcal{D}'} \max_{\|\delta\|_\infty \leq \epsilon} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T (\mathbf{x} + \delta)\}] + \lambda \|\mathbf{w}_{\mathcal{D}'}\|_2^2, \quad \text{s.t. } \mathbf{w}_{\mathcal{D}'} := \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}'), \quad (5)$$

where the inner maximization $\max_{\|\delta\|_\infty \leq \epsilon} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T (\mathbf{x} + \delta)\}] + \lambda \|\mathbf{w}_{\mathcal{D}'}\|_2^2$ is the adversarial loss that applies ℓ_∞ perturbation with budget ϵ to attack $\mathbf{w}_{\mathcal{D}'}$, and the outer minimization optimizes the adversarial loss w.r.t. \mathcal{D}' . Intuitively, the optimal solution of this min-max problem implies a robust data distribution.

4.2 NATURAL TRAINING ON THE CLEAN DATASET IS NON-ROBUST

As a comparison, we begin by showing that the optimal classifier of Eq. 4 is non-robust.

We consider the robustness of classifiers under ℓ_∞ adversarial perturbations with attack budget ϵ , which means that an adversary can modify each feature by at most a value of $\pm\epsilon$. Note that the first feature of \mathbf{x} (Eq. 3) is strongly correlated with the label, and the rest d features are only weakly correlated with the label. We prove that both strongly and weakly-correlated features contribute to the prediction in the optimal classifier, while the effect of the weakly-correlated features dominates the strongly-correlated one, i.e., the weight of SVM on the weakly-correlated features is larger than the weight on the strongly-correlated feature. Under ℓ_∞ -perturbations with $\epsilon = \Theta(1/\sqrt{d})$, the positive effect of weakly-correlated features will be overridden by the perturbation, i.e., the weakly-correlated features hurt the prediction under attacks. For example, if $\epsilon = 2\mu$, the weakly-correlated features will be shifted to be anti-correlated features by the adversary, i.e., $(\mathcal{N}(\mu y, 1) \rightarrow \mathcal{N}(-\mu y, 1))$. As the weakly-correlated features dominate the prediction of the optimal classifier, the classifier will predict opposite labels under such perturbations. In the following theorem, we formally state the above discussion.

We start with calculating the strongest ℓ_∞ adversarial perturbations of the SVM classifier.

Lemma 1. *For arbitrary \mathbf{w} , the optimal δ of the maximization problem*

$$\max_{\|\delta\|_\infty \leq \epsilon} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}^T (\mathbf{x} + \delta)\}] + \lambda \|\mathbf{w}\|_2^2$$

is given by $\delta = -\epsilon \text{sign}(y\mathbf{w})$.

¹We study the popular error in this section, where the problem of robust dataset learning reduces to the problem of robust data distribution learning.

This lemma provides a closed form of strongest ℓ_∞ adversarial perturbations of the SVM classifier. It's easy to verify this lemma via Holder's inequality $-y\mathbf{w}^T\boldsymbol{\delta} \leq \|\mathbf{w}\|_1\|\boldsymbol{\delta}\|_\infty = \epsilon\|\mathbf{w}\|_1$. Thus $\mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1-y\mathbf{w}^T(\mathbf{x}+\boldsymbol{\delta})\}] \leq \mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1-y\mathbf{w}^T\mathbf{x}+\epsilon\|\mathbf{w}\|_1\}]$. Taking $\boldsymbol{\delta} = -\epsilon \text{sign}(y\mathbf{w})$, we can reach this maximum.

Theorem 1. *If $\mu \geq \frac{4}{\sqrt{d}}$ and $p \leq 0.975$, the optimal classifier $\mathbf{w}^* = (w_1^*, \dots, w_{d+1}^*)$ of Eq. 4 achieves more than 99% natural accuracy but less than 1% robust accuracy with ℓ_∞ -perturbation of size $\epsilon \geq 2\mu$.*

This theorem states that the optimal classifier naturally trained on \mathbf{x} has low accuracy under $\Theta(1/\sqrt{d})$ ℓ_∞ -attacks, which indicates achieving robustness on this dataset is non-trivial.

4.3 NATURAL TRAINING ON THE DATASET OF OUR ALGORITHM IS ROBUST

In this part, we will show that the optimal dataset of our min-max optimization Eq. 5 can lead to a robust classifier against $\Theta(1)$ ℓ_∞ -perturbations.

According to Lemma 1, the inner maximization problem has a closed form solution

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T(\mathbf{x} + \boldsymbol{\delta})\}] = \mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}'}\|_1\}].$$

Thus, we only need to solve the minimization problem

$$\min_{\mathcal{D}'} \mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}'}\|_1\}] + \lambda\|\mathbf{w}_{\mathcal{D}'}\|_2^2. \quad (6)$$

However, it is computationally intractable to find the optimal distribution \mathcal{D}' directly, as we cannot represent $\mathbf{w}_{\mathcal{D}'}$ with \mathcal{D}' explicitly. Instead, we try to find the necessary and sufficient conditions of $\mathbf{w}_{\mathcal{D}'}$ that minimizes Eq. 6. We show the existence of distributions such that the related $\mathbf{w}_{\mathcal{D}'}$ satisfies the conditions.

Theorem 2. *Let*

$$\mathcal{D}^* = \underset{\mathcal{D}'}{\text{argmin}} \mathbb{E}_{(\mathbf{x},y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}'}\|_1\}] + \lambda\|\mathbf{w}_{\mathcal{D}'}\|_2^2.$$

If $1 > \epsilon \geq \mu$, then $\mathbf{w}_{\mathcal{D}^} := (w_{\mathcal{D}^*}^{(1)}, \dots, w_{\mathcal{D}^*}^{(d+1)})$ must satisfy $w_{\mathcal{D}^*}^{(1)} > 0$ and $w_{\mathcal{D}^*}^{(2)} = w_{\mathcal{D}^*}^{(3)} = \dots = w_{\mathcal{D}^*}^{(d+1)} = 0$.*

This theorem states a necessary condition of the optimal dataset from Eq. 5. The robust dataset should suffice that the data-parameterized classifier (naturally trained on the dataset) is independent of the weak-correlated features x_2, \dots, x_{d+1} . Thus, the data-parameterized classifier should be more robust than vanilla classifier trained on original dataset.

Theorem 3. *The optimal SVM w.r.t. the robust dataset \mathcal{D}^* has clean and robust accuracy $\geq p$ under ℓ_∞ -perturbation (with budget less than 1) on the original dataset. An optimal solution of \mathcal{D}^* is given by*

$$y \sim \text{Uniform}\{-1, 1\}, \quad x_1 \sim \begin{cases} y, & \text{with probability } p; \\ -y, & \text{with probability } 1-p, \end{cases} \quad x_i = 1, \quad i = 2, 3, \dots, d+1.$$

According to Theorem 2, the weight of the optimal SVM learned from \mathcal{D}^* should satisfy $w_{\mathcal{D}^*}^{(2)} = w_{\mathcal{D}^*}^{(3)} = \dots = w_{\mathcal{D}^*}^{(d+1)} = 0$. Thus the clean and robust accuracy are only related to the first feature of \mathbf{x} . It is easy to see the natural accuracy is equal to the probability that $\text{sign}(x_1) = y$, which is p . Besides, when the perturbation budget $\epsilon < 1$, the adversary does not change the sign of x_1 . Thus the robust accuracy is also p .

Compared to the original distribution \mathcal{D} (Eq. 3), the robust distribution \mathcal{D}^* keeps the strongly-correlated feature x_1 unchanged and modifies the weakly-correlated features x_2, \dots, x_{d+1} to uncorrelated features (a constant). In this way the optimal SVM trained on the robust distribution will not assign weights on the uncorrelated features, because they do not contribute to the predictions. Thus, the resulting classifier is relatively robust, as it depends only on the strongly-correlated feature.

Table 1: Experimental results of robust dataset learning on MNIST, CIFAR10 and TinyImageNet, where we naturally train classifiers on the datasets created by different methods. Numbers with * refer to the experimental results reported by the original work.

	Robust acc (%) / Natural acc (%)	0.1 (ℓ_∞)	0.2 (ℓ_∞)	1.0 (ℓ_2)	2.0 (ℓ_2)
MNIST	Natural dataset	71.73/98.10	8.28/98.10	79.14/98.10	21.28/98.10
	Adv. data of natural classifier	84.17/97.41	19.40/94.71	86.81/97.85	26.58/95.80
	Adv. data of robust classifier	76.70/97.93	11.79/97.50	79.67/97.58	24.21/97.83
	Robust dataset (ours)	93.53/98.69	52.36/97.29	91.60/98.76	48.40/98.25
	Robust acc (%) / Natural acc (%)	2/255 (ℓ_∞)	4/255 (ℓ_∞)	0.25 (ℓ_2)	0.5 (ℓ_2)
CIFAR10	Natural dataset	6.28/93.23	0.02/93.23	9.85/93.23	0.04/93.23
	Adv. data of natural classifier	48.21/84.66	17.86/80.48	47.05/81.83	21.33/81.14
	Adv. data of robust classifier	10.51/86.06	0.21/85.83	11.68/88.70	0.39/86.64
	Ilyas et al. (2019)	36.36/77.53	14.56/78.61	48.20*/85.40*	21.85*/85.40*
	Robust dataset (ours)	54.74/87.19	26.79/85.55	59.52/86.59	27.35/85.10
	Robust acc (%) / Natural acc (%)	2/255 (ℓ_∞)	4/255 (ℓ_∞)	0.25 (ℓ_2)	0.5 (ℓ_2)
TinyImageNet	Natural dataset	0.34/70.94	0.16/70.94	4.55/70.94	0.52/70.94
	Adv. data of natural classifier	12.96/65.22	5.10/65.13	29.97/65.98	9.36/64.14
	Robust dataset (ours)	25.43/60.02	18.42/60.36	39.55/61.13	25.48/60.92

Extension to general data distributions. We now show that our theorems hold for a more general distribution. Consider the case where the instance x and the label y follow the distribution below:

$$y \sim \text{Uniform}\{-1, 1\}, \quad x_1 \sim \begin{cases} y, & \text{with probability } p; \\ -y, & \text{with probability } 1 - p, \end{cases} \quad x_i \sim \mathcal{D}_i, \quad i = 2, 3, \dots, d + 1,$$

where \mathcal{D}_i are symmetric distributions with mean $\mu_i \leq 1$. We prove that the parameterized SVM with the optimal robust dataset \mathcal{D}^* achieves at least p clean and robust accuracy under ℓ_∞ -perturbation (with budget less than 1). The details can be found in Sec. A.4.

5 EXPERIMENTS

In this section, we conduct comprehensive experiments to demonstrate the effectiveness of our algorithm on MNIST (LeCun, 1998), CIFAR10 (Krizhevsky et al., 2009), and TinyImageNet (Deng et al., 2009).

5.1 ROBUSTNESS

In this part, we compare the performance of our (robust) data-parameterized model with models obtained from several baseline methods under ℓ_2 and ℓ_∞ attacks. We use the state-of-the-art attack method—Autoattack (Croce & Hein, 2020) for evaluating the adversarial robustness of models.

Baseline. Ilyas et al. (2019) is the only work related to robust dataset learning. We include this work as one of the baseline for CIFAR10.² Besides, motivated by adversarial training, we create two other baselines. In adversarial training, we utilize adversarial examples to improve robustness, so we take the adversarial data generated from both natural (see Adv. data of natural classifier in Table 1) and robust pre-trained classifiers (see Adv. data of robust classifier in Table 1) as two baseline robust datasets. In order to make a fair comparison, we require all robust datasets to have the same size.

Pre-train settings. In data pre-processing phase, we randomly cropped the image to 28×28 for MNIST, 32×32 for CIFAR10, and 64×64 for TinyImageNet with 4 pixels padding. Then we apply random horizontal flip to the images and normalize them with mean 0.1307 and variance 0.3081. During training, we use SGD (Bottou, 2010) with learning rate 0.01, momentum 0.9, weight decay $5e-4$, and cosine learning rate decay to fine tune the pre-train models. For the robust pre-train model we train TRADES (Zhang et al., 2019) with $0.2 \ell_\infty$ perturbations for MNIST and $4/255 \ell_\infty$ perturbations for CIFAR 10.

²Ilyas et al. (2019) released a robust dataset on CIFAR10 but did not release the code for generating these images. See <https://github.com/MadryLab/constructed-datasets>. In this work, we use the numbers reported in their work for comparison.

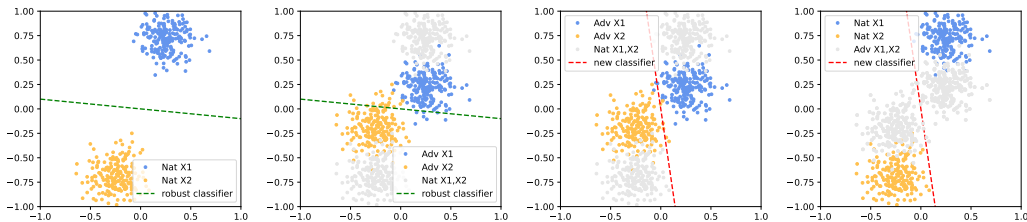


Figure 2: The process (from left to right) on how natural training on adversarial examples of a robust classifier leads to a non-robust model. The green line is the robust classifier used to generate adversarial examples, and the red line is the new classifier naturally trained on the adversarial examples of a robust classifier. “Nat X1” and “Nat X2” stand for natural data of class 1 and 2, respectively, and “Adv X1” and “Adv X2” are their adversarial counterparts. From the 4th plot we can see that the new classifier is non-robust.

Evaluation. During the evaluation, we fine-tune a natural pre-trained classifier with the given (robust) dataset and evaluate the model with adversarial attacks. For MNIST and CIFAR-10, we use Autoattack (Croce & Hein, 2020) with various budgets to evaluate the robustness of the models on the test set. Since Autoattack is computationally expensive on TinyImageNet, we evaluate our algorithm with the same set of budgets using PGD-10 attack instead.³

Experiment setup. We use a CNN which has two convolutional layers, followed by two fully-connected layers for MNIST. We apply ResNet-18 for CIFAR10 and TinyImageNet. The output size of the last layer is the number of classes of each dataset. In our robust dataset learning algorithm (Algorithm 1), we set θ_0 to be the weights of the naturally pre-trained classifier. During training, we use PGD-20 to generate adversarial samples for MNIST and PGD-10 for the other two datasets. The step size of PGD attack is selected as $\epsilon/10$. Besides, we set the learning rate of the classifier to $\gamma = 0.01$ and the learning rate of the robust dataset to $\beta = 0.5/255$. For the baseline methods, the datasets are generated using PGD attacks on natural and robust pre-train models (See Adv. data of natural and robust classifier in Table 1). To generate robust pre-train models, we train TRADES (Zhang et al., 2019) using the corresponding budgets. During evaluation phase, we use SGD with learning rate 0.01, momentum 0.9, and weight decay $5e-4$, to fine tune the same pre-trained model on all datasets.

Result analysis. Table 1 illustrates the experimental results on the three datasets. Compared to the baseline methods, the classifier naturally trained on our robust dataset achieves nearly 10% increase on the robust accuracy on all tasks and attacks. We also notice that the classifier trained on the adversarial examples of robust classifier suffers from poor robust accuracy. We provide a simple example (Fig. 2) to show that natural training on adversarial examples of a robust classifier may lead to a non-robust model.

Why do we not compare with adversarial training? There are two reasons: 1) while the output of adversarial training is a classifier, in the robust dataset learning task, the input of our evaluation benchmark is a dataset on which we would naturally train a classifier. Therefore, adversarial training does not fit our evaluation benchmark. Instead, we modify adversarial training as another baseline “adversarial data of robust classifier” in Table 1. 2) We remark that our work is not aiming to set a new SOTA benchmark for adversarial defense, but rather to design a time-efficient method that benefits scenarios with limited computational resources. For example, learning a robust CIFAR10 dataset takes around 2 hours on a NVIDIA RTX A5000 GPU; fine tuning a pre-trained classifier on our robust dataset takes at most 10 minutes. However, adversarial training, e.g., PGD Adversarial Training (Madry et al., 2017), TRADES (Zhang et al., 2019), takes more than one day on the same GPU.

5.2 ABLATION STUDY

In this part, we conduct ablation experiments to study the effect of dataset size and the transferability of our robust dataset to different initialization and architectures. The robust dataset learning settings and evaluation methods are the same as in Sec. 5.1.

³Additional results of TinyImageNet using Autoattack can be found in Appendix B.

Table 2: Experiments with different size of robust dataset on MNIST, CIFAR10, and TinyImageNet, where we naturally train classifiers on the datasets created by different methods.

Robust acc (%) / Natural acc (%)	Threat model	Natural dataset	Robust dataset (ours)		
			10% size	20% size	100% size
MNIST	0.2 (ℓ_∞)	8.28/98.10	25.65/96.11	41.87/96.83	52.36/97.29
	2.0 (ℓ_2)	21.28/98.10	34.47/96.75	40.21/97.59	48.40/98.25
CIFAR10	2/255 (ℓ_∞)	6.28/93.23	37.69/83.26	43.87/86.10	54.74/87.19
	0.25 (ℓ_2)	9.85/93.23	42.61/81.50	45.54/82.80	59.52/86.59
TinyImageNet	2/255 (ℓ_∞)	0.34/70.94	15.64/60.58	17.75/60.35	25.43/60.02
	0.25 (ℓ_2)	4.55/70.94	24.38/63.42	26.71/62.71	39.55/61.13

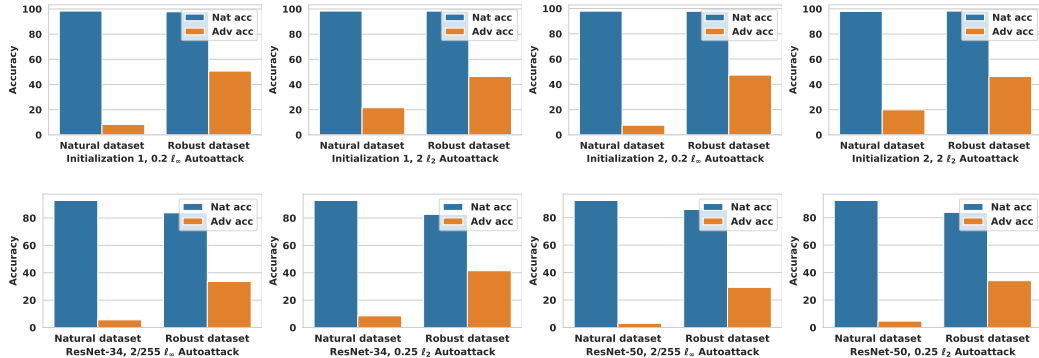


Figure 3: Experiments of transferability of our robust dataset. **Top:** Different initialization of the same CNN model on MNIST. We fine tune two pre-trained CNN models (initialization 1 and initialization 2) on the robust MNIST dataset generated by a third initialization. **Bottom:** Different architectures on CIFAR10. We fine tune one ResNet-34 and one ResNet-50 models on the robust CIFAR10 generated by ResNet-18.

Dataset size. We study the effect of different robust dataset size, where the robust dataset size is constrained to only 10% and 20% of the original dataset. We evaluate the robustness of classifiers under several adversarial budgets for MNIST, CIFAR10 and TinyImageNet. The results are shown in Table 2. We find that even if trained with 10% of the size of the original dataset, the resulting classifier can still achieve 42.61% robust accuracy on CIFAR10 and 24.38% robust accuracy on TinyImageNet.

Different network initializations. We evaluate the transferability of our robust dataset by using different network initializations. In our experiments, we apply the same CNN with different initializations to evaluate the classifier trained with MNIST robust dataset under 0.2 ℓ_∞ and 2.0 ℓ_2 Autoattacks. In Fig. 3 Top, we see that our robust dataset effectively improves the robustness of the classifier with different network initializations.

Different network architectures. We also investigate the case where the naturally trained network during testing has a different architecture from that we use to learn the robust dataset. In the experiments, we use our robust dataset to fine-tune ResNet-34 and ResNet-50 pre-trained models. We evaluate the adversarial robustness under 2/255 ℓ_∞ and 0.25 ℓ_2 Autoattack. In Fig. 3 Bottom, we see that our robust dataset enjoys descent transferability across different network architectures.

6 CONCLUSION

In this work, we propose a principle, tri-level optimization algorithm to solve the robust dataset learning problem. We theoretically prove the guarantee of our algorithm on an abstraction model, and empirically verify its effectiveness and efficiency on three popular image classification datasets. Our proposed algorithm provides a principled way to integrate the property of adversarial robustness into a dataset. The evaluation results of the method imply various real-world applications under scenarios with limited computational resources.

ETHICS STATEMENT

We did not see obvious negative ethical impacts in our work. Our work provides a time-efficient method to improve adversarial robustness of neural networks, which has potential positive effect on the safety of AI.

REPRODUCIBILITY

In Sec. 5, we include the details of our model structures, training hyperparameters, and evaluation methods. We also share our code in supplementary materials.

REFERENCES

- Mislav Balunović and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *8th International Conference on Learning Representations (ICLR 2020)(virtual)*. International Conference on Learning Representations, 2020.
- Battista Biggio, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Nedim Srndic, 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*, pp. 387–402, 2013.
- Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020.
- Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT’2010*, pp. 177–186. Springer, 2010.
- Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.
- George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4750–4759, 2022.
- Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *ICML*, 2019.
- Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.
- Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2057–2066. PMLR, 2019.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020.
- Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.
- Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

- Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial Logit Pairing. In *arXiv:1803.06373*, 2018.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.
- Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pp. 9464–9474, 2019.
- Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Soft-label anonymous gastric x-ray image distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 305–309. IEEE, 2020.
- Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1778–1787, 2018.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3385–3394, 2019.
- Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020.
- Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.
- Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. In *arXiv:1511.04508*, 2016.
- Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.
- Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018.
- Ilija Sucholutsky and Matthias Schonlau. Secdd: Efficient and secure method for remotely training neural networks. *arXiv preprint arXiv:2009.09155*, 2020.
- Ilija Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *International Joint Conference on Neural Networks*, pp. 1–8, 2021.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICML*, 2014.
- Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.
- Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.
- Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.
- Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.
- Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems*, pp. 2958–2969, 2020.
- Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.
- Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.
- Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kaikhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.
- Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.
- Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In *International Conference on Machine Learning*, pp. 12368–12379. PMLR, 2021.
- Dinghui Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. *Advances in Neural Information Processing Systems*, 33:2316–2326, 2020a.
- Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.
- Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Li zhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, 2020b.
- Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pp. 12674–12685. PMLR, 2021.
- Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 1(2):3, 2021.
- Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

A ADDITIONAL PROOFS

Lemma 2. *The optimal solution $\mathbf{w}^* = (w_1^*, \dots, w_{d+1}^*)$ of our optimization problem Eq. 4 must satisfy $w_2 = \dots = w_{d+1}$ and $\text{sign}(w_i) \geq 0, i \in [d+1]$.*

Proof. We prove this lemma by contradiction, assume w.l.o.g. the optimal solution $\mathbf{w}^* = (w_1^*, \dots, w_{d+1}^*)$ satisfying $w_2^* \neq w_3^*$, we can let $\mathbf{w}' = (w_1^*, w_3^*, w_2^*, w_4^*, \dots, w_{d+1}^*)$. In this case, we have $y\mathbf{w}^{*T}\mathbf{x} = y\mathbf{w}'^T\mathbf{x}$ because both $w_2^*x_2 + w_3^*x_3$ and $w_2^*x_3 + w_3^*x_2$ follow $\mathcal{N}((w_2^* + w_3^*)\mu y, w_2^{*2} + w_3^{*2})$. So we have $\mathcal{L}(\mathbf{w}'; \mathcal{D}) = \mathcal{L}(\mathbf{w}^*; \mathcal{D})$. Moreover, since the margin term $\mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}^T\mathbf{x}\}]$ is convex in \mathbf{w} , by Jensen's inequality, averaging \mathbf{w}^* and \mathbf{w}' will not increase the value of that margin term. On the other hand, $\|\frac{\mathbf{w}^* + \mathbf{w}'}{2}\|_2^2 < \|\mathbf{w}^*\|_2^2$ as $2(\frac{w_2^* + w_3^*}{2})^2 < w_2^{*2} + w_3^{*2}$ when $w_2^* \neq w_3^*$. Thus $\mathcal{L}(\frac{\mathbf{w}^* + \mathbf{w}'}{2}; \mathcal{D}) < \mathcal{L}(\mathbf{w}^*; \mathcal{D})$, which yields contradiction. Analogously, if there exists i , such that $\text{sign}(w_i^*) < 0$, let $\mathbf{w}' = (w_1^*, \dots, -w_i^*, \dots, w_{d+1}^*)$, we have $\|\mathbf{w}'\|_2^2 = \|\mathbf{w}^*\|_2^2$ and $\mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}'^T\mathbf{x}\}] \leq \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}^{*T}\mathbf{x}\}]$, which yields another contradiction. \square

Lemma 3. *If $\mu \geq \frac{4}{\sqrt{d}}$ and $p \leq 0.975$, the optimal solution $\mathbf{w}^* = (w_1^*, \dots, w_{d+1}^*)$ of our optimization problem Eq. 4 must satisfy $w_1^* < \sqrt{d}w_2^*$.*

Proof. Assume for the sake of contradiction that $w_1^* \geq \sqrt{d}w_2^*$. By Lemma 2 we have $w_2^* = \dots = w_{d+1}^*$. Assume w.l.o.g. $\|\mathbf{w}^*\|_2 = 1$, we have $w_2^* \leq \frac{1}{\sqrt{2d}}$. Then, with probability at least $1 - p$, the first feature predicts the wrong label and without enough weight, the remaining features cannot compensate for it. Concretely,

$$\begin{aligned} \mathbb{E}[\max(0, 1 - y\mathbf{w}^{*T}\mathbf{x})] &\geq (1 - p)\mathbb{E}[\max(0, 1 + w_1^* - w_2^* \sum_{i=2}^{d+1} \mathcal{N}(\mu, 1))] \\ &\geq (1 - p)\mathbb{E}[\max(0, 1 + \sqrt{d}w_2^* - w_2^* \mathcal{N}(d\mu, d))] \\ &\geq (1 - p)\mathbb{E}[\max(0, 1 + 1/\sqrt{2} - \mathcal{N}(\sqrt{\frac{d}{2}}\mu, \frac{1}{2}))]. \end{aligned} \quad (7)$$

We will now show that a solution $\mathbf{w}' = (w'_1, \dots, w'_{d+1})$ that assigns zero weight on the first feature ($w'_2 = \dots = w'_{d+1} = \frac{1}{\sqrt{d}}$ and $w'_1 = 0$), achieves a better loss.

$$\mathbb{E}[\max(0, 1 - y\mathbf{w}'^T\mathbf{x})] = \mathbb{E}[\max(0, 1 - \mathcal{N}(\sqrt{d}\mu, 1))]. \quad (8)$$

By the optimality of \mathbf{w}^* we must have

$$\mathbb{E}[\max(0, 1 - \mathcal{N}(\sqrt{d}\mu, 1))] \geq (1 - p)\mathbb{E}[\max(0, 1 + 1/\sqrt{2} - \mathcal{N}(\sqrt{\frac{d}{2}}\mu, \frac{1}{2}))],$$

which yields $p \geq 1 - \frac{\mathbb{E}[\max(0, 1 - \mathcal{N}(\sqrt{d}\mu, 1))]}{\mathbb{E}[\max(0, 1 + 1/\sqrt{2} - \mathcal{N}(\sqrt{\frac{d}{2}}\mu, \frac{1}{2}))]} \geq 1 - \frac{\mathbb{E}[\max(0, 1 - \mathcal{N}(4, 1))]}{\mathbb{E}[\max(0, 1 + 1/\sqrt{2} - \mathcal{N}(2\sqrt{2}, \frac{1}{2}))]} > 0.975$, which contradicts to the condition that $p \leq 0.975$. \square

A.1 PROOF OF THEOREM 1

Proof. Part 1. We show that the optimal classifier can achieve high natural accuracy. By Lemma 2 and Lemma 3 we have $w_2^* = \dots = w_{d+1}^*$, $\text{sign}(w_i^*) \geq 0, i \in [d+1]$, and $w_1^* \leq \sqrt{d}w_2^*$. Consider $\mathbf{w}^{*T}\mathbf{x} = w_1^*x_1 + w_2^* \sum_{i=2}^{d+1} \mathcal{N}(\mu y, 1) = w_2^* \mathcal{N}(\frac{w_1^*}{w_2^*}x_1 + d\mu y, d)$, because $\epsilon \geq 2\mu$ and $\mu \geq \frac{4}{\sqrt{d}}$, the

probability that \mathbf{x} is correctly classified is

$$\begin{aligned}
\Pr(\text{sign}(\mathbf{w}^{*T} \mathbf{x}) = \text{sign}(y)) &= p \Pr(\mathcal{N}(\frac{w_1^*}{w_2^*} + d\mu, d) > 0) \\
&\quad + (1-p) \Pr(\mathcal{N}(-\frac{w_1^*}{w_2^*} + d\mu, d) > 0) \\
&\geq p \Pr(\mathcal{N}(\mu d, d) > 0) + (1-p) \Pr(\mathcal{N}(\mu - \sqrt{d}, d) > 0) \quad (9) \\
&\geq p \Pr(\mathcal{N}(4\sqrt{d}, d) > 0) + (1-p) \Pr(\mathcal{N}(3\sqrt{d}, d) > 0) \\
&= p \Pr(\mathcal{N}(4, 1) > 0) + (1-p) \Pr(\mathcal{N}(3, 1) > 0) \\
&\geq \Pr(\mathcal{N}(3, 1) > 0) \\
&\geq 0.9986.
\end{aligned}$$

Thus the natural accuracy of the optimal classifier \mathbf{w}^* is greater than 99%.

Part 2. We show that the optimal classifier achieve low robust accuracy. Firstly, according to Lemma 1, the perturbed distribution $\mathbf{x} + \delta$ is given by

$$y \sim \{-1, 1\}, \quad x_1 \sim \begin{cases} y(1 - \epsilon), & \text{with probability } p; \\ -y(1 + \epsilon), & \text{with probability } 1 - p, \end{cases} \quad x_i \sim \mathcal{N}((\mu - \epsilon)y, 1), \quad i \geq 2. \quad (10)$$

By Lemma 2 and Lemma 3, we have $w_2^* = \dots = w_{d+1}^*$, $\text{sign}(w_i^*) \geq 0, i \in [d+1]$, and $w_1^* \leq \sqrt{d}w_2^*$. Consider $\mathbf{w}^{*T}(\mathbf{x} + \delta) = w_1^*x_1 + w_2^* \sum_{i=2}^{d+1} \mathcal{N}((\mu - \epsilon)y, 1) = w_2^* \mathcal{N}(\frac{w_1^*}{w_2^*}x_1 + d(\mu - \epsilon)y, d)$. Because $\epsilon \geq 2\mu$ and $\mu \geq \frac{4}{\sqrt{d}}$, the probability that $\mathbf{x} + \delta$ is correctly classified is

$$\begin{aligned}
\Pr(\text{sign}(\mathbf{w}^{*T} \mathbf{x}) = \text{sign}(y)) &= p \Pr(\mathcal{N}(\frac{w_1^*}{w_2^*} + d(\mu - \epsilon), d) > 0) \\
&\quad + (1-p) \Pr(\mathcal{N}(-\frac{w_1^*}{w_2^*} + d(\mu - \epsilon), d) > 0) \\
&\leq p \Pr(\mathcal{N}(\sqrt{d} - \mu d, d) > 0) + (1-p) \Pr(\mathcal{N}(-\mu d, d) > 0) \quad (11) \\
&\leq p \Pr(\mathcal{N}(-3\sqrt{d}, d) > 0) + (1-p) \Pr(\mathcal{N}(-4\sqrt{d}, d) > 0) \\
&= p \Pr(\mathcal{N}(-3, 1) > 0) + (1-p) \Pr(\mathcal{N}(-4, 1) > 0) \\
&\leq \Pr(\mathcal{N}(-3, 1) > 0) \\
&\leq 0.00135.
\end{aligned}$$

Thus the robust accuracy of the optimal classifier \mathbf{w}^* is less than 0.2%. \square

A.2 PROOF OF THEOREM 2

Proof. The optimal \mathcal{D}^* should make $1 - y\mathbf{w}_{\mathcal{D}^*}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}^*}\|_1$ and $\|\mathbf{w}_{\mathcal{D}^*}\|_2^2$ as small as possible.

$$1 - y\mathbf{w}_{\mathcal{D}^*}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}^*}\|_1 = 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon |w_{\mathcal{D}^*}^{(1)}| + \sum_{i=2}^{d+1} (\epsilon |w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i), \quad (12)$$

as $x_i \sim \mathcal{N}(\mu y, 1), i \geq 2$, we have $\sum_{i=2}^{d+1} (\epsilon |w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i) \sim \mathcal{N}(\sum_{i=2}^{d+1} (\epsilon |w_{\mathcal{D}^*}^{(i)}| - \mu y w_{\mathcal{D}^*}^{(i)}), \sum_{i=2}^{d+1} w_{\mathcal{D}^*}^{(i)2})$.

Denote by $\mathcal{L}_{\mathcal{D}'} := \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}'}\|_1\}]$. Assume there exist $w_{\mathcal{D}^*}^{(i)} \neq 0, i \geq 2$, we will show there exists \mathcal{D}_0 such that $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$ and

$$\mathcal{L}_{\mathcal{D}_0} + \lambda \|\mathbf{w}_{\mathcal{D}_0}\|_2^2 < \mathcal{L}_{\mathcal{D}^*} + \lambda \|\mathbf{w}_{\mathcal{D}^*}\|_2^2. \quad (13)$$

Step 1: We will show if $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$,

$$\mathcal{L}_{\mathcal{D}_0} + \lambda \|\mathbf{w}_{\mathcal{D}_0}\|_2^2 < \mathcal{L}_{\mathcal{D}^*} + \lambda \|\mathbf{w}_{\mathcal{D}^*}\|_2^2.$$

Firstly, it is easy to observe that $\lambda \|\mathbf{w}_{\mathcal{D}_0}\|_2^2 < \lambda \|\mathbf{w}_{\mathcal{D}^*}\|_2^2$. Then we focus on the term $\mathcal{L}_{\mathcal{D}_0}$ and $\mathcal{L}_{\mathcal{D}^*}$.

Denote by $A = 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon|w_{\mathcal{D}^*}^{(1)}|$, $\mu' = \sum_{i=2}^{d+1}(\epsilon|w_{\mathcal{D}^*}^{(i)}| - \mu yw_{\mathcal{D}^*}^{(i)})$, $\sigma'^2 = \sum_{i=2}^{d+1} w_{\mathcal{D}^*}^{(i)2}$, and $z = \sum_{i=2}^{d+1}(\epsilon|w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i) \sim \mathcal{N}(\mu', \sigma'^2)$. Then by Eq. 23 we have

$$1 - y\mathbf{w}_{\mathcal{D}^*}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}^*}\|_1 = A + z,$$

and thus we can simplify $\mathcal{L}_{\mathcal{D}_0}$, $\mathcal{L}_{\mathcal{D}^*}$ as below:

$$\mathcal{L}_{\mathcal{D}_0} = \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}_0}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}_0}\|_1\}] = \mathbb{E}_{x_1, y}[\max\{0, 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon|w_{\mathcal{D}^*}^{(1)}|\}] = \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] \quad (14)$$

$$\mathcal{L}_{\mathcal{D}^*} = \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}^*}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}^*}\|_1\}] = \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}] \quad (15)$$

Consider $\mathcal{L}_{\mathcal{D}^*} = \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}]$,

$$\begin{aligned} \mathcal{L}_{\mathcal{D}^*} &= \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}] \geq \mathbb{E}_{z, x_1, y}[(A + z)\mathbb{I}_{A+z \geq 0}\mathbb{I}_{A \geq 0}] \\ &= \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[(A + z)\mathbb{I}_{z \geq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z \geq -A}] + \mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \end{aligned} \quad (16)$$

Now we consider $\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]$, as $z \sim \mathcal{N}(\mu', \sigma'^2)$, we have $\frac{z - \mu'}{\sigma'} \sim \mathcal{N}(0, 1)$ and

$$\begin{aligned} \mathbb{E}_z[z\mathbb{I}_{z \geq -A}] &= \mathbb{E}_z[z\mathbb{I}_{\mu' \geq z \geq -A}] + \mathbb{E}_z[z\mathbb{I}_{2\mu' + A \geq z \geq \mu'}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &= \mathbb{E}_{s \sim \mathcal{N}(0, 1)}[(\sigma' s + \mu')\mathbb{I}_{0 \geq s \geq -\frac{A + \mu'}{\sigma'}}] + \mathbb{E}_{s \sim \mathcal{N}(0, 1)}[(\sigma' s + \mu')\mathbb{I}_{\frac{A + \mu'}{\sigma'} \geq s \geq 0}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &= 2\mu'\mathbb{E}_{s \sim \mathcal{N}(0, 1)}[\mathbb{I}_{0 \geq s \geq -\frac{A + \mu'}{\sigma'}}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \end{aligned} \quad (17)$$

since $\epsilon > \mu$, we have

$$\mu' = \sum_{i=2}^{d+1}(\epsilon|w_{\mathcal{D}^*}^{(i)}| - \mu yw_{\mathcal{D}^*}^{(i)}) \geq \sum_{i=2}^{d+1}(\epsilon - \mu)|w_{\mathcal{D}^*}^{(i)}| > 0$$

Thus

$$\begin{aligned} \mathbb{E}_z[z\mathbb{I}_{z \geq -A}] &= 2\mu'\mathbb{E}_{s \sim \mathcal{N}(0, 1)}[\mathbb{I}_{0 \geq s \geq -\frac{A + \mu'}{\sigma'}}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &> \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &> (2\mu' + A)\mathbb{E}_z[\mathbb{I}_{z \geq 2\mu' + A}] \\ &> A\mathbb{E}_z[\mathbb{I}_{z \geq 2\mu' + A}] \\ &= A\mathbb{E}_z[\mathbb{I}_{z \leq -A}] \end{aligned} \quad (18)$$

Plug Eq. 18 into Eq. 16 we have

$$\begin{aligned} \mathcal{L}_{\mathcal{D}^*} &= \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}] \geq \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \\ &> \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z \leq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] \\ &= \mathcal{L}_{\mathcal{D}_0}. \end{aligned} \quad (19)$$

Step 2: We will show the existence of distribution \mathcal{D}_0 such that $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$, we can set $x_2 = x_3 = \dots = x_{d+1} = 1$ in \mathcal{D}_0 and $x_1 = \frac{1}{w_{\mathcal{D}^*}^{(1)}}y$ such that $w_{\mathcal{D}_0}^{(1)} = w_{\mathcal{D}^*}^{(1)}$ and $w_{\mathcal{D}_0}^{(i)} = 0, i \geq 2$.

Combining Step 1 and 2 yields contradiction. \square

A.3 PROOF OF THEOREM 3

Proof. According to Theorem 2, the weight of the optimal SVM learned from \mathcal{D}^* should satisfy $w_{\mathcal{D}^*}^{(2)} = w_{\mathcal{D}^*}^{(3)} = \dots = w_{\mathcal{D}^*}^{(d+1)} = 0$. Thus the clean and robust accuracy are only related to the first feature of \mathbf{x} . It is easy to see the natural accuracy is equal to the probability that $\text{sign}(x_1) = y$, which is p . Besides, when the perturbation budget $\epsilon < 1$, the adversary does not change the sign of x_1 . Thus the robust accuracy is also p . \square

A.4 ADVANCED THEORETICAL ANALYSIS ON A GENERAL DATASET

Consider dataset distribution $(\mathbf{x}, y) \in \mathbb{R}^{(d+1) \times 1}$ follow the distribution below:

$$y \sim \{-1, 1\}, \quad x_1 \sim \begin{cases} y, & \text{with prob } p; \\ -y, & \text{with prob } 1 - p, \end{cases} \quad x_i \sim \mathcal{D}_i, \quad i \geq 2, \quad (20)$$

where \mathcal{D}_i are symmetric distributions with mean $\mu_i \leq 1$.

Lemma 4. *The sum of independent symmetric distributions is also symmetric.*

Proof. Based on the fact (which is easy to prove) that a random variable is symmetric if and only if its characteristic function is real-valued. The characteristic function of the sum of independent symmetric distributions is given by the multiplication of the characteristic function of independent symmetric distributions, which is also real-valued. Thus the sum of independent symmetric distributions is also symmetric. \square

Following the settings in the above section we have the lemma below

Lemma 5. *Our minmax optimization problem is*

$$\min_{\mathcal{D}'} \max_{\|\delta\|_\infty \leq \epsilon} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T(\mathbf{x} + \delta)\}] = \min_{\mathcal{D}'} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}'}\|_1\}] \quad (21)$$

Let

$$\mathcal{D}^* = \operatorname{argmin}_{\mathcal{D}'} \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}'}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}'}\|_1\}], \quad (22)$$

if $1 > \epsilon \geq \max_{i \geq 2} \mu_i$, then $\mathbf{w}_{\mathcal{D}^*} := (w_{\mathcal{D}^*}^{(1)}, \dots, w_{\mathcal{D}^*}^{(d+1)})$ must satisfy $w_{\mathcal{D}^*}^{(2)} = w_{\mathcal{D}^*}^{(3)} = \dots = w_{\mathcal{D}^*}^{(d+1)} = 0$

Proof. The optimal \mathcal{D}^* should make $1 - y\mathbf{w}_{\mathcal{D}^*}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}^*}\|_1$ as small as possible.

$$1 - y\mathbf{w}_{\mathcal{D}^*}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}^*}\|_1 = 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon|w_{\mathcal{D}^*}^{(1)}| + \sum_{i=2}^{d+1} (\epsilon|w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i), \quad (23)$$

as $x_i \sim \mathcal{N}(\mu y, 1)$, $i \geq 2$, we have $\sum_{i=2}^{d+1} (\epsilon|w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i) \sim \mathcal{N}(\sum_{i=2}^{d+1} (\epsilon|w_{\mathcal{D}^*}^{(i)}| - \mu y w_{\mathcal{D}^*}^{(i)}), \sum_{i=2}^{d+1} w_{\mathcal{D}^*}^{(i)2})$.

Assume there exist $w_{\mathcal{D}^*}^{(i)} \neq 0$, $i \geq 2$, we will show there exists \mathcal{D}_0 such that $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$ and

$$\mathcal{L}_{\mathcal{D}_0} := \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}_0}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}_0}\|_1\}] < \mathbb{E}_{(\mathbf{x}, y)} [\max\{0, 1 - y\mathbf{w}_{\mathcal{D}^*}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}^*}\|_1\}] =: \mathcal{L}_{\mathcal{D}^*}$$

Step 1: We will show if $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$,

$$\mathcal{L}_{\mathcal{D}_0} < \mathcal{L}_{\mathcal{D}^*}.$$

Denote by $A = 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon|w_{\mathcal{D}^*}^{(1)}|$, $\mu' = \sum_{i=2}^{d+1} (\epsilon|w_{\mathcal{D}^*}^{(i)}| - \mu y w_{\mathcal{D}^*}^{(i)})$, and $z = \sum_{i=2}^{d+1} (\epsilon|w_{\mathcal{D}^*}^{(i)}| - yw_{\mathcal{D}^*}^{(i)}x_i) \sim \mathcal{S}$, by Lemma 4 we know \mathcal{S} is symmetric with mean μ' . Then by Eq. 23 we have

$$1 - y\mathbf{w}_{\mathcal{D}^*}^T\mathbf{x} + \epsilon\|\mathbf{w}_{\mathcal{D}^*}\|_1 = A + z,$$

and thus we can simplify $\mathcal{L}_{\mathcal{D}_0}$, $\mathcal{L}_{\mathcal{D}^*}$ as below:

$$\mathcal{L}_{\mathcal{D}_0} = \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}_0}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}_0}\|_1\}] = \mathbb{E}_{x_1, y}[\max\{0, 1 - yw_{\mathcal{D}^*}^{(1)}x_1 + \epsilon |w_{\mathcal{D}^*}^{(1)}|\}] = \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}], \quad (24)$$

$$\mathcal{L}_{\mathcal{D}^*} = \mathbb{E}_{(\mathbf{x}, y)}[\max\{0, 1 - y\mathbf{w}_{\mathcal{D}^*}^T \mathbf{x} + \epsilon \|\mathbf{w}_{\mathcal{D}^*}\|_1\}] = \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}]. \quad (25)$$

Consider $\mathcal{L}_{\mathcal{D}^*} = \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}]$,

$$\begin{aligned} \mathcal{L}_{\mathcal{D}^*} &= \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}] \geq \mathbb{E}_{z, x_1, y}[(A + z)\mathbb{I}_{A+z \geq 0}\mathbb{I}_{A \geq 0}] \\ &= \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[(A + z)\mathbb{I}_{z \geq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z \geq -A}] + \mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \end{aligned} \quad (26)$$

Now we consider $\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]$, as $z \sim \mathcal{S}$ and \mathcal{S} is symmetric with μ' . We have $\mathcal{S} - \mu'$ is symmetric with 0 and

$$\begin{aligned} \mathbb{E}_z[z\mathbb{I}_{z \geq -A}] &= \mathbb{E}_z[z\mathbb{I}_{\mu' \geq z \geq -A}] + \mathbb{E}_z[z\mathbb{I}_{2\mu' + A \geq z \geq \mu'}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &= \mathbb{E}_{s \sim \mathcal{S} - \mu'}[(s + \mu')\mathbb{I}_{0 \geq s \geq -A - \mu'}] + \mathbb{E}_{s \sim \mathcal{S} - \mu'}[(s + \mu')\mathbb{I}_{A + \mu' \geq s \geq 0}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &= 2\mu'\mathbb{E}_{s \sim \mathcal{S} - \mu'}[\mathbb{I}_{0 \geq s \geq -A - \mu'}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &= 2\mu'\mathbb{E}_z[\mathbb{I}_{\mu' \geq z \geq -A}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}]. \end{aligned} \quad (27)$$

Since $\epsilon > \mu$, we have

$$\mu' = \sum_{i=2}^{d+1} (\epsilon |w_{\mathcal{D}^*}^{(i)}| - \mu y w_{\mathcal{D}^*}^{(i)}) \geq \sum_{i=2}^{d+1} (\epsilon - \mu) |w_{\mathcal{D}^*}^{(i)}| > 0.$$

Thus

$$\begin{aligned} \mathbb{E}_z[z\mathbb{I}_{z \geq -A}] &= 2\mu'\mathbb{E}_z[\mathbb{I}_{\mu' \geq z \geq -A}] + \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &> \mathbb{E}_z[z\mathbb{I}_{z \geq 2\mu' + A}] \\ &> (2\mu' + A)\mathbb{E}_z[\mathbb{I}_{z \geq 2\mu' + A}] \\ &> A\mathbb{E}_z[\mathbb{I}_{z \geq 2\mu' + A}] \\ &= A\mathbb{E}_z[\mathbb{I}_{z \leq -A}]. \end{aligned} \quad (28)$$

Plugging Eq. 28 into Eq. 26, we have

$$\begin{aligned} \mathcal{L}_{\mathcal{D}^*} &= \mathbb{E}_{\mathbf{x}, y}[(A + z)\mathbb{I}_{A+z \geq 0}] \geq \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[\mathbb{I}_{A \geq 0}\mathbb{E}_z[z\mathbb{I}_{z \geq -A}]] \\ &> \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] - \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z < -A}]] + \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}\mathbb{E}_z[\mathbb{I}_{z \leq -A}]] \\ &= \mathbb{E}_{x_1, y}[A\mathbb{I}_{A \geq 0}] \\ &= \mathcal{L}_{\mathcal{D}_0}. \end{aligned} \quad (29)$$

Step 2: We will show the existence of distribution \mathcal{D}_0 such that $\mathbf{w}_{\mathcal{D}_0} := (w_{\mathcal{D}^*}^{(1)}, 0, \dots, 0)$, we can set $x_2 = x_3 = \dots = x_{d+1} = 1$ in \mathcal{D}_0 and $x_1 = \frac{1}{w_{\mathcal{D}^*}^{(1)}}y$ such that $w_{\mathcal{D}_0}^{(1)} = w_{\mathcal{D}^*}^{(1)}$ and $w_{\mathcal{D}_0}^{(i)} = 0, i \geq 2$.

Combining Step 1 and 2 yields contradiction. \square

Lemma 6. *The optimal SVM learned from \mathcal{D}^* have at least p clean and robust accuracy (with ℓ_∞ budget less than 1) on the original dataset.*

Proof. According to Lemma 5, the weight of the optimal SVM learned from \mathcal{D}^* should satisfy $w_{\mathcal{D}^*}^{(2)} = w_{\mathcal{D}^*}^{(3)} = \dots = w_{\mathcal{D}^*}^{(d+1)} = 0$. Thus the clean and robust accuracy are only related to the first feature of \mathbf{x} . It is easy to see the natural accuracy is equal to the probability that $\text{sign}(x_1) = y$, which is p . Besides, when the perturbation budget $\epsilon < 1$, the adversary does not change the sign of x_1 . Thus the robust accuracy is also p . \square

B ADDITIONAL EXPERIMENT RESULTS

Table 3: TinyImagenet results (Under AutoAttack)

Robust acc / Natural acc	2/255 (ℓ_∞)	0.25 (ℓ_2)
Natural dataset	0.01/70.94	1.6/70.94
Adv. data of natural classifier	7.35/61.66	25.06/65.29
Robust dataset (ours)	15.71/61.60	34.72/62.91