Perturbation Type Categorization for Multiple Adversarial Perturbation Robustness

Pratyush Maini[1] **Xinyun Chen**[2] **Bo Li**[3] **Dawn Song**[2]

[1]Carnegie Mellon University
[2]University of California, Berkeley
[3]University of Illinois at Urbana-Champaign

Abstract

Recent works in adversarial robustness have proposed defenses to improve the robustness of a single model against the union of multiple perturbation types. However, these methods still suffer significant trade-offs compared to the ones specifically trained to be robust against a single perturbation type. In this work, we introduce the problem of categorizing adversarial examples based on their perturbation types. We first theoretically show on a toy task that adversarial examples of different perturbation types constitute different distributions—making it possible to distinguish them. We support these arguments with experimental validation on multiple ℓ_p attacks and common corruptions. Instead of training a single classifier, we propose PROTECTOR, a two-stage pipeline that first categorizes the perturbation type of the input, and then makes the final prediction using the classifier specifically trained against the predicted perturbation type. We theoretically show that at test time the adversary faces a natural trade-off between fooling the perturbation classifier and the succeeding classifier optimized with perturbation-specific adversarial training. This makes it challenging for an adversary to plant strong attacks against the whole pipeline. Experiments on MNIST and CIFAR-10 show that PROTECTOR outperforms prior adversarial training-based defenses by over 5% when tested against the union of $\ell_1, \ell_2, \ell_\infty$ attacks. Additionally, our method extends to a more diverse attack suite, also showing large robustness gains against multiple ℓ_p, spatial and recolor attacks.

1 INTRODUCTION

Machine learning models have been shown to be vulnerable to different types of adversarial examples—inputs with a small magnitude of perturbation added to mislead the classifier's prediction [Szegedy et al., 2013]. Consequently, many defenses have been proposed to improve their robustness, a majority of which focus on achieving robustness against a specific perturbation type [Goodfellow et al., 2015, Madry et al., 2018, Kurakin et al., 2017, Tramèr et al., 2018, Dong et al., 2018, Zhang et al., 2019, Carmon et al., 2019]. However, as ML models get adopted in real-world applications, it becomes important for the defenses to be robust against different types of perturbations given the flexibility of practical attackers. In addition, prior work showed that when models are trained to be robust against one perturbation type, the robustness is typically not preserved against attacks of a different type [Schott et al., 2018, Kang et al., 2019].

Motivated by the need for robustness against diverse perturbation types, recent works have attempted to train models that are robust against multiple perturbation types [Tramèr and Boneh, 2019, Maini et al., 2020, Laidlaw et al., 2021]. These works consider perturbations restricted by their ℓ_p norms ($p \in \{1, 2, \infty\}$) or spatial and color transformations. The proposed methods improve the overall robustness against multiple perturbation types. However, when evaluating the robustness against each individual perturbation type, the robustness of models trained by these methods is still considerably worse than those trained on a single perturbation type. Given these empirical observations, in this work we aim to answer: *Are different types of perturbations separable? Can we categorize them to improve robustness to multiple adversarial perturbations?*

To address these questions and explore the properties of different perturbation types, we introduce the problem of *categorizing adversarial examples* based on their perturbation types. We present theoretical analysis on a toy task to show that when we add different types of perturbations to benign samples of a given ground-truth class, their new dis-

Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022), PMLR 180:1317–1327.

tributions are distinct and separable. We experimentally validate our theoretical results on both (mathematically) well-defined perturbation regions such as ℓ_p balls, as well as various common corruptions [Hendrycks and Dietterich, 2019]. We find that deep networks are able to categorize different perturbation types with high accuracy ($> 95\%$). Further, our perturbation classifier shows high generalization accuracy ($\sim 90\%$) to *unseen* common corruptions, i.e., correctly predicting their categories (weather, noise, blur, or digital) without training on them. While in this work we focus on improving worst-case adversarial robustness, applications of categorizing perturbation types extend beyond it—such as detecting *systematic* distribution shifts (e.g. presence of snow for self-driving cars [Michaelis et al., 2020]). Further, using a perturbation classifier as the discriminator may improve the effectiveness and variety of adversarial examples produced by generative models [Wong and Kolter, 2021, Xiao et al., 2018a, Song et al., 2018].

Based on our theoretical analysis, we propose PROTECTOR, a two-stage pipeline that performs *Perturbation Type Categorization to Improve Robustness* against multiple perturbations. First, the top-level perturbation classifier predicts the perturbation type of the input. Then, among the second-level predictors, PROTECTOR selects the one that is the most robust to the predicted perturbation type to make the final prediction. We theoretically show that there exists a natural tension between attacking the perturbation classifier and the second-level predictors. Specifically, strong attacks against the second-level predictors make it easier for the perturbation classifier to predict the adversarial perturbation type; on the other hand, fooling the perturbation classifier requires planting weaker (or less representative) attacks against the second-level predictors. As a result, even an *imperfect* perturbation classifier significantly improves the model's overall robustness to multiple perturbation types. We also supplement our theoretical statements on the toy task with experimental validation in the exact same setting.

Empirically[1], we first show that the perturbation classifier generalizes well on classifying a wide range of adversarial perturbations. Then we compare PROTECTOR with recent defenses against multiple attack types on MNIST and CIFAR-10. Even though we do not utilize adversarial training [Goodfellow et al., 2015] to train the perturbation classifier, an ensemble of diverse perturbation classifiers along with adding small noise to inputs help make PROTECTOR robust against adaptive attacks. Specifically, we combine predictions of perturbation classifiers that classify adversarial examples in their image and Fourier domains [Yin et al., 2019a]. This further increases the tension between attacking top-level and second-level components by reducing the space of successful adversarial attacks. PROTECTOR outperforms prior approaches by over 5% against the union of ℓ_1, ℓ_2 and ℓ_∞ attacks. From the suite of 15 different attacks tested, the average improvement over all the attacks w.r.t. the state-of-art baseline defense is $\sim 15\%$ on both MNIST and CIFAR-10. Training a model to be robust against multiple attacks typically imposes a significant tradeoff against the accuracy on benign samples, but PROTECTOR attains $\sim 7\%$ greater benign test accuracy on CIFAR-10 as compared to recent works [Laidlaw et al., 2021, Maini et al., 2020]. We further demonstrate how our defense naturally extends beyond ℓ_p perturbation types, where we assess the robustness of our model against the union of ℓ_∞, ℓ_2, spatial [Wong et al., 2019, Xiao et al., 2018b] and recolor [Bhattad et al., 2020, Laidlaw and Feizi, 2019] attacks on CIFAR-10. Our defense exceeds the robustness of recent work [Laidlaw et al., 2021] by over 13% against all attacks. In addition, PROTECTOR provides the flexibility to plug in and integrate new defenses against individual perturbation types into the existing framework as second-level predictors, thus the defense performance of PROTECTOR can be continuously improved with the development of more advanced defenses against single perturbation types.

2 RELATED WORK

Adversarial examples. Among the different types of adversarial attacks studied in prior work [Szegedy et al., 2013, Goodfellow et al., 2015, Madry et al., 2018, Hendrycks et al., 2019, Bhattad et al., 2020], the majority constrain the perturbation within a small ℓ_p region around the original input. To improve model robustness in the presence of such adversaries, most existing defenses utilize adversarial training [Goodfellow et al., 2015], which augments the training dataset with adversarial examples. Till date, different variants of adversarial training algorithms remain the most successful defenses against adversarial attacks [Carmon et al., 2019, Zhang et al., 2019, Wong et al., 2020, Rice et al., 2020, Wang et al., 2020]. Other types of defenses include input transformation [Guo et al., 2018, Buckman et al., 2018] and network distillation [Papernot et al., 2016], but were rendered ineffective under stronger adversaries [He et al., 2017, Carlini and Wagner, 2017, Athalye et al., 2018, Tramer et al., 2020].

Defenses against multiple perturbation types. Some recent works have focused on defending against a union of norm bounded ℓ_p attacks. Schott et al. [2018], Kang et al. [2019] showed that models that were trained for a given ℓ_p-norm bounded attack are not robust against attacks in a different ℓ_q region. Schott et al. [2018] proposed the use of multiple variational autoencoders to achieve robustness to multiple ℓ_p attacks on MNIST. Tramèr and Boneh [2019] used simple aggregations of multiple adversaries to achieve non-trivial robust accuracy against $\ell_1, \ell_2, \ell_\infty$ attacks. Maini et al. [2020] proposed MSD that takes gradient steps in the union of multiple ℓ_p regions to improve multiple perturba-

[1] Code for reproducing our experiments can be found at https://github.com/sunblaze-ucb/adversarial-protector.

tion robustness. Most recently, Laidlaw et al. [2021] proposed a defense against unseen perturbations using perceptual adversarial training. They evaluate their work against ℓ_∞, ℓ_2, spatial, recolor adversaries.

Detection of adversarial examples. Multiple prior works have focused on detecting adversarial examples [Feinman et al., 2017, Lee et al., 2018, Ma et al., 2018, Cennamo et al., 2020, Fidel et al., 2019, Yin et al., 2019b]. However, most of these methods were rendered ineffective in the presence of adaptive adversaries [Carlini and Wagner, 2017, Tramer et al., 2020]. In comparison, our work focuses on a more challenging problem of categorizing perturbation types. To this end, Yin et al. [2019a] proposed the examination of Fourier transforms of adversarial examples to determine the adversarial attack and corruption types.

3 SEPARABILITY OF PERTURBATION TYPES

In this section, we formally illustrate the setup of perturbation categorization. In Theorem 1, we show the existence of a classifier that can separate adversarial examples belonging to different perturbation types. We focus on ℓ_p attacks (that can be fully specified mathematically) on a simplified binary classification task for the convenience of theoretical analysis. However, PROTECTOR can also improve the empirical robustness of models trained on common image classification benchmarks against both ℓ_p and non-ℓ_p attacks. We will discuss the empirical examination in Section 6.

3.1 PROBLEM SETTING

Data distribution. We consider a distribution $\mathcal{D}$ of inputs sampled from the union of two multi-variate Gaussian distributions such that the input-label pairs (x, y) can be described as:

$$\begin{gathered} y \overset{u.a.r}{\sim} \{-1,+1\}, \\ x_0 \sim \mathcal{N}(y\alpha, \sigma^2), \quad x_1, \ldots, x_d \overset{i.i.d}{\sim} \mathcal{N}(y\eta, \sigma^2), \end{gathered} \tag{1}$$

where $x = [x_0, x_1, \ldots, x_d] \in \mathcal{R}^{d+1}$ and $\eta = \frac{\alpha}{\sqrt{d}}$. This setting demonstrates the distinction between a feature x_0 that is strongly correlated with the label, and d weakly correlated features that are (independently) normally distributed with the mean $y\eta$ and the variance σ^2. In our work, we assume that $\frac{\alpha}{\sigma} > 10$ (x_0 is strongly correlated) and $d > 100$ (remaining d features are weakly correlated, but together represent a strongly correlated feature). This setting was adapted from Ilyas et al. [2019], and more discussion can be found in Appendix A.

Perturbation types. We focus our theoretical discussion on adversaries constrained within a fixed ℓ_p region of radius ϵ_p around the original input, for $\ell_p \in \mathcal{S} = \{\ell_1, \ell_\infty\}$. Such adversaries are frequently studied in existing work for finding the optimal first-order perturbation for different attack types. Let $\ell(\cdot, \cdot)$ be the cross-entropy loss, and $\Delta_{\mathcal{S}} = \bigcup_{\ell_p \in \mathcal{S}} \Delta_{\ell_p, \epsilon}$ for the ℓ_p threat model, $\Delta_{\ell_p, \epsilon_p}$, of radius ϵ_p. Then, for a model f_θ, the optimal perturbation δ^* is given by:

$$\delta^* = \arg\max_{\delta \in \Delta_{\mathcal{S}}} \ell(f_\theta(x + \delta), y). \tag{2}$$

3.2 SEPARABILITY OF ℓ_p PERTURBATIONS

Consider a classifier M trained with the objective of correctly classifying inputs $x \in \mathcal{D}$. The goal of the adversary is to fool M by finding the optimal perturbation $\delta_{\mathcal{A}}\ \forall \mathcal{A} \in S$. The theorem below shows that the distributions of adversarial inputs within different ℓ_p regions can be separated with a high accuracy.

Theorem 1 (Separability of perturbation types)**.** *Given a binary Gaussian classifier M trained on $\mathcal{D}$, consider $\mathcal{D}_p^y$ to be the distribution of optimal adversarial inputs (for a class y) against M, within ℓ_p regions of radius ϵ_p, where $\epsilon_1 = \alpha$, $\epsilon_\infty = \alpha/\sqrt{d}$. Distributions $\mathcal{D}_p^y$ ($p \in \{1, \infty\}$) can be accurately separated by a binary Gaussian classifier C_{adv} with a misclassification probability $P_e \leq 10^{-24}$.*

The proof sketch is as follows. We first calculate the optimal weights of a binary Gaussian classifier M trained on $\mathcal{D}$. Accordingly, for any input $x \in \mathcal{D}$, we find the optimal adversarial perturbation $\delta_{\mathcal{A}}\ \forall \mathcal{A} \in \{\ell_1, \ell_\infty\}$ against M. We discuss how these perturbed inputs $x + \delta_{\mathcal{A}}$ also follow a normal distribution, with shifted means. Finally, for data points of a given label, we show that C_{adv} is able to predict the correct perturbation type with a very low error. We present the formal proof in Appendix B.

4 PROTECTOR: PERTURBATION TYPE CATEGORIZATION FOR ROBUSTNESS

We illustrate the PROTECTOR pipeline in Figure 1. PROTECTOR performs the classification task as a two-stage process. Given an input, PROTECTOR first utilizes a *perturbation classifier C_{adv}* to predict its perturbation type. Then, based on the predicted type, PROTECTOR uses the corresponding second-level predictor $M_{\mathcal{A}}$ to provide the final prediction, where $M_{\mathcal{A}}$ is specially trained to be robust against the attack $\mathcal{A} \in \mathcal{S}$. Formally, let f_θ be the PROTECTOR model, then:

$$f_\theta(x) = M_{\mathcal{A}}(x); \quad s.t. \quad \mathcal{A} = \operatorname{argmax} C_{adv}(x). \tag{3}$$

4.1 ADVERSARIAL TRADE-OFF

In Section 3.2, we showed that the optimal perturbations of different attack types belong to different data distributions, and can be separated by a simple classifier. However, in the

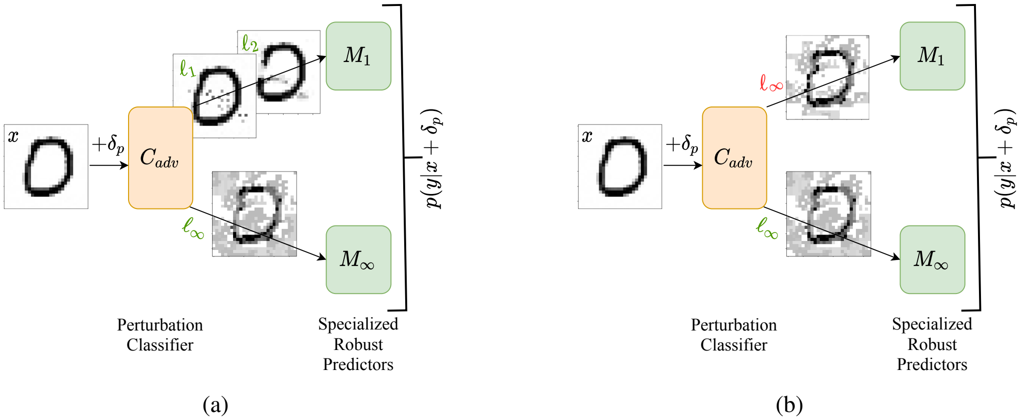


Figure 1: An overview of PROTECTOR. (a) The perturbation classifier C_{adv} categorizes representative attacks of different types. (b) An illustration of the trade-off in Theorem 2. An adversarial example fooling C_{adv} (the ℓ_∞ sample marked in red) becomes weaker to attack the second-level $M_{\mathcal{A}}$ models. Stronger or more representative attacks (marked green) are correctly categorized.

white-box setting, the adversary has knowledge of both the perturbation classifier (C_{adv}) and specialized robust models ($M_{\mathcal{A}}$). This allows it to adapt the attack to fool the entire pipeline instead of individual models alone. To validate the robustness of PROTECTOR, we provide a theoretical justification in Theorem 2, showing that PROTECTOR naturally offers a trade-off between fooling C_{adv} and the individual models $M_{\mathcal{A}}$. This makes it difficult for adversaries to stage successful attacks against PROTECTOR.

Note that there are some overlapping regions among different perturbation constraints. For example, every adversary could set $\delta_p = 0$ as a valid perturbation, in which case C_{adv} can not correctly classify all attacks. However, such perturbations are not useful to the adversary, because any $M_{\mathcal{A}}$ can correctly classify unperturbed inputs with a high probability. In the following theorem, we examine the robustness of PROTECTOR in the presence of such strong dynamic adversaries.

Theorem 2 (Adversarial trade-off)**.** *Given a data distribution $\mathcal{D}$, adversarially trained models M_{ℓ_p,ϵ_p}, and an attack classifier C_{adv} that distinguishes perturbations of different ℓ_p attack types for $p \in \{1, \infty\}$; the probability of a successful attack by the strongest adversary over the* PROTECTOR *pipeline is $P_e < 0.01$ for $\epsilon_1 = \alpha + 2\sigma$ and $\epsilon_\infty = \frac{\alpha+2\sigma}{\sqrt{d}}$.*

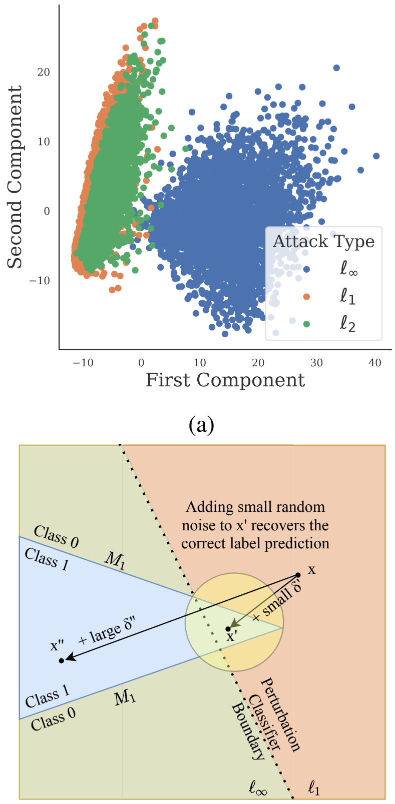
Here, the *worst-case adversary* refers to an adaptive adversary that has full knowledge of the defense strategy. In Appendix C.2, we discuss how $\epsilon_1, \epsilon_\infty$ are set so that the ℓ_1 and ℓ_∞ adversaries can fool $M_{\ell_\infty,\epsilon_\infty}$ and M_{ℓ_1,ϵ_1} models respectively with a high success rate. To prove Theorem 2, we first show that when trained on $\mathcal{D}$, an adversarially robust model $M_{\mathcal{A}}$ can achieve robust accuracy $> 99\%$ against the attack type it was trained for, and $< 2\%$ against an alternate attack. By "alternate" we mean that for an ℓ_q attack, the prediction is made by the M_{ℓ_p,ϵ_p} model. Then, we analyze the modified distributions of the inputs perturbed by different ℓ_p attacks. Based on this, we construct a simple decision rule for the perturbation classifier C_{adv}. Finally, we compute the perturbation induced by the worst-case adversary. We show that there exists a trade-off between fooling the C_{adv} (to allow the alternate M_{ℓ_p,ϵ_p} model to make the final prediction for an ℓ_q attack $\forall p, q \in \{1, \infty\}; p \neq q$), and fooling the alternate M_{ℓ_p,ϵ_p} model itself. We provide an illustration of the trade-off in Figure 1b, and a formal proof and *experimental validation* on the toy task in Appendix C.

5 TRAINING AND INFERENCE

We now extend PROTECTOR to deep neural networks trained on common image classification benchmarks. Following prior work on defending against multiple perturbation types, we evaluate on MNIST [LeCun et al., 2010] and CIFAR-10 [Krizhevsky, 2012] datasets. Here, we present the training details, the formulation of an ensemble of perturbation classifiers, and adaptive white-box attacks against PROTECTOR.

5.1 DATASET CREATION

To train our perturbation classifier C_{adv}, we create a dataset that includes adversarial examples of different perturbation types. We perform adversarial attacks against each of the individual $M_{\mathcal{A}}$ models used in PROTECTOR to curate the training and test sets. In the case of ℓ_p examples, we use the PGD attack [Madry et al., 2018], and for spatial [Xiao et al., 2018b] and recolor [Laidlaw and Feizi, 2019] attacks, we use their original attack formulation. The time for creating the dataset against each $M_{\mathcal{A}}$ is the same as running a single epoch of adversarial training. Since most recent works typically train their models for ∼200 epochs, the dataset creation time is insignificant when compared with the cost of training an $M_{\mathcal{A}}$ model.



(a)

(b)

Figure 2: (a) PCA for different adversarial perturbations on MNIST. (b) Illustration of the effect of random noise on generating adversarial examples. The notion of small, large perturbations is only used to illustrate the scenario in Figure 2b, and neither perturbation region subsumes the other.

Combining perturbation types. When training PROTECTOR to be robust against a set $\mathcal{S}$ of multiple (k) attacks, we combine certain perturbation types under the same label to improve the overall robustness. This is beneficial when: (a) a specialized model $M_\mathcal{A}$ also shows a high degree of robustness to a different attack $\mathcal{B} \in \mathcal{S}$, s.t. $\mathcal{A} \neq \mathcal{B}$; (b) two different attack types $\mathcal{A}, \mathcal{B} \in \mathcal{S}$ have similar characteristics. For instance, in case of ℓ_p attacks, we perform binary classification between $\mathcal{A} = \{\{\ell_1, \ell_2\}, \ell_\infty\}$. We hypothesize that compared to ℓ_∞ adversarial examples, ℓ_1 and ℓ_2 adversarial examples show similar characteristics. To provide an intuitive illustration, we randomly sample 10K adversarial examples generated with PGD attacks on MNIST, and present their Principal Component Analysis (PCA) in Figure 2a. We observe that the first two principal components for ℓ_1 and ℓ_2 adversarial examples are largely overlapping, while those for ℓ_∞ are clearly from a different distribution.[2] For the MNIST dataset, we use the $M_{\ell_2}, M_{\ell_\infty}$ models in PROTECTOR, and we use $M_{\ell_1}, M_{\ell_\infty}$ models for CIFAR-10. The choice is made based on the robustness of $\{M_{\ell_2}, M_{\ell_1}\}$ models against $\{\ell_1, \ell_2\}$ attacks respectively, as will be depicted in Table 2. Similarly, when defending against the union of ℓ_p and non-ℓ_p perturbation types on CIFAR-10, we classify $\mathcal{A} = \{\{\ell_\infty, \ell_2, \text{ReColor}\}, \text{StAdv}\}$ attacks based on the robustness of each $M_\mathcal{A}$ against every attack $\mathcal{B} \in \mathcal{S}$. We report the robustness of PROTECTOR with varying number of second-level predictors in Appendix J.3.

5.2 TRAINING

Past works [Maini et al., 2020, Tramèr and Boneh, 2019] on robustness to multiple attack types require intensive hyperparameter tuning to *balance* different attack types when one attack is stronger than others. We find that a similar phenomenon plagues the adversarial training (AT) of C_{adv}. Therefore, we train C_{adv} over a static dataset, which is fast and stable. Specifically, using a single GTX 1080Ti GPU, C_{adv} can be trained within 5 and 30 minutes on MNIST and CIFAR-10 respectively (given that we already have access to perturbation-specific robust models). On the other hand, training state-of-the-art models robust to a single perturbation type requires up to 2 days to train on the same amount of GPU power, and existing defenses against multiple (k) perturbation types take k times as long as the training time for robustness against a single perturbation type. Instead, even when the individual $M_\mathcal{A}$ are unavailable, we can train the k models in parallel to improve training speed.

A key advantage of PROTECTOR's design is that it can build upon existing defenses against individual perturbation types. Specifically, we leverage the adversarially trained models developed in prior work [Zhang et al., 2019, Carmon et al., 2019] as $M_\mathcal{A}$ models in our pipeline. The architecture of C_{adv} is also similar to a single $M_\mathcal{A}$ model. See Appendix D for more details.

5.3 INFERENCE PROCEDURE

Ensemble of diverse perturbation classifiers. While C_{adv} learns the ability to distinguish between different attack types, it is not immune to the presence of adaptive adversaries that try to fool C_{adv} and the $M_\mathcal{A}$ models together. To improve model robustness against such adversaries, we attempt to increase the trade-off in PROTECTOR that was described in Section 4.1. We use an ensemble (average of prediction logits) of two perturbation classifiers that classify adversarial examples in different domains – via the Fourier and image domains.[3] Owing to this diversity, the classification landscape of each C_{adv} is different. Intuitively, the trade-off between fooling the two stages of PROTECTOR confines the adversary in a very small region for generating successful adversarial attacks when using an ensemble

[2] The visualization only serves as motivation. It does not suggest that ℓ_1, ℓ_2 examples are not separable.

[3] Adversaries can still back-propagate through the Fourier transformation steps.

of perturbation classifiers. In Appendix G, we show how the adversarial examples can be visually separated in the Fourier domain [Yin et al., 2019a] and discuss further implementation details of the ensemble.

Constraining the adversary using random noise. While past work has [Hu et al., 2019] suggested that adding random noise does not help defend against adversarial inputs, it is the *unique* exhibition of the trade-off described in Theorem 2 that adversarial attacks against PROTECTOR, on the contrary, are likely to fail when added with random noise. Intuitively, the trade-off between fooling the two stages of PROTECTOR confines the adversary in a very small region for crafting successful attacks.

Consider the illustrative example in Figure 2b. The input $(x, y = 0)$ is subjected to an ℓ_∞ attack. Assume that the $M_{\ell_\infty, \epsilon_\infty}$ model is a perfect classifier for adversarial examples within a fixed ϵ_∞ region. The dotted line shows the decision boundary for C_{adv}, which correctly classifies inputs subjected to ℓ_∞ perturbations δ'' as ℓ_∞ attacks (green), but misclassifies samples with smaller perturbations. When the adversary adds a large perturbation δ'', the prediction of M_{ℓ_1} for the resulted input x'' becomes wrong, but the perturbation classifier also categorizes it as an M_{ℓ_∞} attack, thus the final prediction of PROTECTOR is still correct since it will be produced by $M_{\infty, \epsilon_\infty}$ model instead. On the other hand, when the adversary adds a small perturbation δ' to fool the perturbation classifier, adding a small amount of random noise can recover the correct prediction with a high probability. Note that every point on the boundary of the noise region (yellow circle) is correctly classified by the pipeline. In this way, adding random noise exploits an adversarial trade-off for PROTECTOR to achieve a high accuracy against adversarial examples, in the absence of adversarial training. In our implementation, we sample random noise $z \sim \mathcal{N}(0, I)$, and add $\hat{z} = \epsilon_2 \cdot z/|z|_2$ to the model input.

5.4 ADAPTIVE ATTACKS AGAINST PROTECTOR

Gradient propagation. Since the final prediction in Equation 3 only depends on a single $M_\mathcal{A}$ model, the pipeline does not allow gradient flow across the two levels. This can make it difficult for gradient-based adversaries to attack PROTECTOR. Therefore, we utilize a combination of predictions from each individual $M_\mathcal{A}$ model by modifying $f_\theta(x)$ in Equation 3 as follows:

$$\begin{aligned} c &= \operatorname{softmax}(C_{adv}(x)); \\ f_\theta(x) &= \sum_{\mathcal{A} \in \mathcal{S}} c_\mathcal{A} \cdot M_\mathcal{A}(x), \end{aligned} \tag{4}$$

where $c_\mathcal{A}$ denotes the probability of the input x being classified as the perturbation type $\mathcal{A}$ by C_{adv}. Equation 4 is only used for the purpose of generating adversarial examples and performing gradient-based attack optimization. For consistency, we still use Equation 3 to compute the model prediction at inference (final forward-propagation). We do not see any significant performance advantages of either choice during inference, and briefly report a comparison in Appendix I.1.

Separately attacking C_{adv} and $M_\mathcal{A}$. We also experiment with other strategies of aggregating the predictions of different components, e.g., tuning the loss to balance direct attacks on C_{adv} and each $M_\mathcal{A}$ model. We find that this attack formulation performs worse than attacking the entire pipeline with Equation 4. We provide a discussion on this attack in Appendix I.

6 EXPERIMENTS

In this section, we present our results on MNIST and CIFAR-10 datasets, both for the perturbation classifier C_{adv} alone, and for the entire PROTECTOR pipeline.

6.1 PERTURBATION CATEGORIZATION BY C_{adv}

Categorizing ℓ_p perturbations. First, we justify our choice of ϵ_p radii by empirically quantifying the overlapping regions of different types of adversarial attacks. We observe that the empirical overlap is exactly 0% in all cases on both MNIST and CIFAR-10, and we present the full analysis in Appendix H.1. We then evaluate the categorization performance of C_{adv} on a dataset of adversarial examples which are generated against the six models we use as the baseline defenses in our experiments. Note that C_{adv} is only trained on adversarial examples against the two $M_\mathcal{A}$ models that are part of PROTECTOR.

Next, we evaluate the test set generalization across the various datasets created. We observe that C_{adv} transfers well across the board. First, C_{adv} generalizes to adversarial examples against new models, i.e., it preserves a high accuracy, even if the adversarial examples are generated against models that are unseen during training. Further, C_{adv} also generalizes to new attack algorithms. As discussed in Section 5.1, we only include PGD adversarial examples in our training set for C_{adv}. However, on adversarial examples generated by the AutoAttack library, the classification accuracy of C_{adv} still holds up. In particular, the accuracy is $> 95\%$ across all the individual test sets created. These results suggest two important findings that validate our results in Theorem 1 — independent of **(a)** the model to be attacked; and **(b)** the algorithm for generating the optimal adversarial perturbation, the optimal adversarial images for a given ℓ_p region follow similar distributions. We present the full results in Appendix H.2.

Categorizing common corruptions. CIFAR-10-C is a benchmark consisting of 19 different types of common corruptions [Hendrycks and Dietterich, 2019]. For each image in the original CIFAR-10 test set, CIFAR-10-C includes im-

Table 1: Generalization results when C_{adv} is trained on different Noise, Blur, Weather and Digital corruptions (Severity=5). Test is performed on Speckle Noise + Gaussian Blur + Spatter + Saturate.

Trained On	Accuracy
Impulse + Defocus Blur + Snow + Brightness	70.4%
+ Gaussian + Glass Blur + Fog + Contrast	80.1%
+ Shot + Motion Blur + Frost + Elastic Trans	85.6%
+ Zoom Blur + JPEG Compression + Pixelate	93.5%
+ Speckle + Gaussian Blur + Spatter + Saturate	99.8%

ages with different corruptions. To train the corruption classifier, we split CIFAR-10-C, so that each corruption type has 9K training samples, and 1K for testing. For corruptions of the highest severity, we observe that our corruption classifier achieves greater than 99% test accuracy on the test split. Details about the architecture are deferred to Appendix D. This demonstrates that our perturbation classifier is applicable to both ℓ_p adversarial perturbations and semantic common corruptions. We discuss detailed results of corruption classification at various severity levels in Appendix H.3.

Generalization to unseen corruptions. We further evaluate the generalization of the perturbation classifier to unseen corruption types. Specifically, different from the above setting of classifying corruption types, now our classifier categorizes all corruption types into 4 categories — noise, blur, digital, and weather (as defined in the CIFAR-10-C benchmark). We evaluate the model performance on 4 held-out corruption types, 1 for each category, and select these corruption types following the model validation setting in Hendrycks and Dietterich [2019]. From the remaining 15 corruption types, we vary the number of corruptions included for training, and present the results in Table 1. We observe that even if we do not train the perturbation classifier on the same corruption types for testing, the classifier still obtains a high generalization accuracy ($> 90\%$). These results demonstrate that perturbation classification is effective even for unseen perturbations.

6.2 ROBUSTNESS TO ℓ_p ATTACKS

Baselines. We compare PROTECTOR with the state-of-art defenses against the union of $\ell_1, \ell_2, \ell_\infty$ adversaries. For Tramèr and Boneh [2019], we compare two variants of adversarial training: (1) the **MAX** approach, where for each image, among different perturbation types, the adversarial sample that leads to the maximum increase of the model loss is augmented into the training set; (2) the **AVG** approach, where adversarial examples for all perturbation types are included for training. We also compare with **MSD** [Maini et al., 2020], which modifies the standard PGD attack to incorporate the union of multiple perturbation types within the steepest decent. In addition, we evaluate $\mathbf{M}_{\ell_1}, \mathbf{M}_{\ell_2}, \mathbf{M}_{\ell_\infty}$ models trained with $\ell_1, \ell_2, \ell_\infty$ perturbations separately, as described in Appendix D.

Attack evaluation. We evaluate against the strongest attacks in the adversarial examples literature, and with adaptive attacks specifically designed for PROTECTOR (Section 5.4). We perform standard PGD attacks along with attacks from the AutoAttack library [Croce and Hein, 2020], which achieves the state-of-art adversarial error rates against multiple recently published models. The radius of the $\{\ell_1, \ell_2, \ell_\infty\}$ perturbation regions is $\{10, 2, 0.3\}$ for the MNIST dataset and $\{10, 0.5, 0.03\}$ for the CIFAR-10 dataset. We present the full details of attack algorithms in Appendix F.

Following prior work, we evaluate models on adversarial examples generated from the first 1000 images of the test set for MNIST and CIFAR-10. Our main evaluation metric is the accuracy on *all attacks* – a given input is a failure case if any of the attack algorithm in our suite successfully fools the model.

Results. In Table 2, we summarize the worst-case performance against all attacks of a given perturbation type for MNIST and CIFAR-10 datasets. In particular, "Ours" denotes the robustness of PROTECTOR against the adaptive attacks described in Section 5.4, and "Ours*" denotes the robustness of PROTECTOR against standard attacks based on Equation 3. The adaptive strategy effectively reduces the overall accuracy of PROTECTOR by $2 - 5\%$, showing that incorporating the gradient and prediction information of all second-level predictors results in a stronger attack.

PROTECTOR outperforms all baselines by 6.4% on MNIST, and 10% on CIFAR-10 in terms of the *all attacks* metric, even when evaluated against a strong adaptive adversary. Compared to the previous state-of-art defense against multiple perturbation types (MSD), the accuracy gain on ℓ_∞ attacks is especially notable, i.e., around 15%. In particular, if we compare the performance on each individual attack algorithm, as shown in Appendix J.1 and J.2 for MNIST and CIFAR-10 respectively, the average accuracy gain is $\sim 15\%$ for both datasets. These results demonstrate that PROTECTOR considerably mitigates the trade-off in the accuracy for individual attacks. Further, PROTECTOR retains a 7% higher CIFAR-10 accuracy on *clean images*, as opposed to past defenses that sacrifice benign accuracy for robustness to multiple perturbation types.

6.3 ROBUSTNESS TO NON-ℓ_p ATTACKS

We demonstrate how PROTECTOR can be extended to perturbation types beyond those restricted to ℓ_p types. Laidlaw et al. [2021] evaluate the robustness of various adversarial defenses against attacks $\mathcal{A} \in \mathcal{S} = \{\ell_2, \ell_\infty, \text{StAdv}, \text{ReColor}\}$ on CIFAR-10. We directly compare PROTECTOR with the pre-trained models for each

Table 2: Worst-case accuracies against different ℓ_p attacks: (a) MNIST; (b) CIFAR-10. *Ours* represents PROTECTOR against the adaptive attack strategy (Eq 4), and *Ours** is the standard setting.

MNIST	M_{ℓ_∞}	M_{ℓ_2}	M_{ℓ_1}	MAX	AVG	MSD	Ours	Ours*
Clean accuracy	99.2%	98.7%	98.8%	98.6%	99.1%	98.3%	98.9%	98.9%
ℓ_∞ attacks ($\epsilon = 0.3$)	90.2%	2.6%	0.0%	39.0%	57.8%	63.5%	78.1%	79.0%
ℓ_2 attacks ($\epsilon = 2.0$)	9.5%	72.3%	47.8%	58.5%	58.6%	65.7%	66.6%	72.3%
ℓ_1 attacks ($\epsilon = 10$)	18.8%	70.6%	77.5%	41.8%	46.1%	64.3%	68.1%	72.5%
All attacks	7.3%	2.6%	0.0%	29.1%	37.1%	57.2%	**63.6%**	**67.2%**

(a)

CIFAR-10	M_{ℓ_∞}	M_{ℓ_2}	M_{ℓ_1}	MAX	AVG	MSD	Ours	Ours*
Clean accuracy	89.5%	93.9%	89.0%	81.0%	84.6%	81.7%	89.0%	89.0%
ℓ_∞ attacks ($\epsilon = 0.03$)	59.3%	34.8%	35.0%	34.9%	39.7%	43.7%	56.1%	58.4%
ℓ_2 attacks ($\epsilon = 0.5$)	64.6%	77.2%	71.5%	61.8%	65.5%	64.5%	69.3%	69.4%
ℓ_1 attacks ($\epsilon = 10$)	27.6%	45.3%	60.9%	43.7%	60.0%	56.1%	57.9%	59.5%
All attacks	27.6%	32.9%	35.0%	31.5%	39.3%	43.5%	**53.5%**	**54.9%**

(b)

Table 3: Worst-case accuracies against ℓ_∞ ($\epsilon = 0.003$), ℓ_2 ($\epsilon = 0.5$), spatial and recolor attacks. *Ours* represents PROTECTOR against the adaptive attack strategy (Eq 4), and *Ours** is the standard setting. PAT [Laidlaw et al., 2021] is trained using perceptual adversarial training.

CIFAR-10	M_{ℓ_∞}	M_{ℓ_2}	M_{StAdv}	M_{ReColor}	MAX	AVG	PAT	Ours	Ours*
Clean acc.	89.5%	93.9%	86.2%	93.4%	84.0%	86.8%	71.6%	89.5%	89.5%
ℓ_∞ attacks	59.3%	34.8%	0.1%	8.5%	25.8%	42.1%	29.8%	58.2%	59.1%
ℓ_2 attacks	64.6%	77.2%	10.0%	34.8%	44.2%	64.8%	54.1%	57.0%	57.2%
StAdv	5.7%	0.2%	68.9%	0.0%	46.2%	27.8%	58.4%	50.4%	55.7%
ReColor	85.5%	84.0%	52.1%	86.8%	77.4%	80.5%	70.9%	85.2%	85.3%
All attacks	5.4%	0.2%	0.1%	0.0%	24.0%	21.5%	27.8%	**40.9%**	**41.9%**

individual defense provided in their work. This includes their defense based on perceptual adversarial training (**PAT**) and the **MAX**, **AVG** models, along with perturbation-specific robust models $M_{\mathcal{A}}$. Specifically, as discussed in Section 5.1, we train a perturbation classifier that classifies adversarial examples as belonging to one of the two classes: $\{\{\ell_\infty, \ell_2, \text{ReColor}\}, \text{StAdv}\}$. We use two individual robust predictors: $\{M_{\ell_\infty}, M_{\text{StAdv}}\}$. The choice is once again made based on the robust accuracy of M_{ℓ_∞} models against $\{\ell_\infty, \ell_2, \text{ReColor}\}$ attacks as also presented in Table 3. This ability to combine attacks also represents positively on the scalability of PROTECTOR. PROTECTOR improves by 13.1% against the union of all attacks. Importantly, PROTECTOR preserves a high accuracy against benign samples, whereas PAT classifies only 71.6% of unperturbed samples correctly, which makes it difficult to adopt it in real-world settings.

7 CONCLUSION

In this work, we introduce the problem of categorizing perturbation types. We theoretically demonstrate that adversarial inputs of different attack types are separable, and empirically validate our claims on different ℓ_p and non-ℓ_p attacks. In addition to categorizing them with high accuracy, the perturbation categorizer also generalizes to *unseen* corruptions of the same category.

PROTECTOR performs perturbation type categorization to achieve robustness against the union of multiple perturbation types. We theoretically examine the existence of a natural tension for any adversary trying to fool our model—between fooling the attack classifier and the specialized robust predictors. Our empirical results on MNIST and CIFAR-10 datasets complement our theoretical analysis, showing that PROTECTOR outperforms existing defenses against multiple ℓ_p and non-ℓ_p attacks by over 5%, while showing gains of over $\sim 15\%$ on average and clean accuracy metrics.

Our work serves as a stepping stone towards the goal of universal adversarial robustness, by dissecting multiple adversarial objectives into individually solvable pieces and combining them via PROTECTOR. In its present form, PROTECTOR requires the knowledge of each individual attack type that we want to be robust against—to train the perturbation classifier. This limitation opens up various avenues for future work, including the new problem of perturbation categorization by defining sub-classes of adversarial attack types, and training generative models to synthesize diverse perturbations.

Acknowledgements

This material is in part based upon work supported by the National Science Foundation under Grant No. TWC-1409915, Berkeley DeepDrive, and DARPA D3M under Grant No. FA8750-17-2-0091. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Xinyun Chen is supported by the Facebook Fellowship.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and D. A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Sye_OgHFwH`.

J. Buckman, Aurko Roy, Colin Raffel, and Ian J. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11190–11201, 2019.

Alessandro Cennamo, Ido Freeman, and Anton Kummert. A statistical defense approach for detecting adversarial examples. In *Proceedings of the 2020 International Conference on Pattern Recognition and Intelligent Systems*, pages 1–7, 2020.

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

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. *arXiv preprint arXiv:1909.03418*, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Advances in Neural Information Processing Systems*, pages 1635–1646, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017. URL `https://arxiv.org/abs/1607.02533`.

Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.

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

Pratyush Maini, Eric Wong, and J. Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, 2020.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming, 2020.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. *arXiv preprint arXiv:2002.11569*, 2020.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. In *International Conference on Learning Representations*, 2018.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. *Advances in Neural Information Processing Systems*, 31:8312–8323, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pages 5866–5876, 2019.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Haotao Wang, Tianlong Chen, Shupeng Gui, Ting-Kuei Hu, Ji Liu, and Zhangyang Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. In *NeurIPS*, 2020.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=MIDckA56aD`.

Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected Sinkhorn iterations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6808–6817. PMLR, 09–15 Jun 2019.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3905–3911, 2018a.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018b. URL `https://openreview.net/forum?id=HyydRMZC-`.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, pages 13276–13286, 2019a.

Xuwang Yin, Soheil Kolouri, and Gustavo K Rohde. Adversarial example detection and classification with asymmetrical adversarial training. *arXiv preprint arXiv:1905.11475*, 2019b.

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*, pages 7472–7482, 2019.