API-Net: Robust Generative Classifier via a Single Discriminator

Xinshuai Dong[1], Hong Liu[1], Rongrong Ji[1] *, Liujuan Cao[1], Qixiang Ye[2], Jianzhuang Liu[3], and Qi Tian[4]

[1] Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics, Xiamen University
[2] University of Chinese Academy of Sciences
[3] Noah's Ark Lab, Huawei Technologies
[4] Huawei Cloud BU

Abstract. Robustness of deep neural network classifiers has been attracting increased attention. As for the robust classification problem, a generative classifier typically models the distribution of inputs and labels, and thus can better handle off-manifold examples at the cost of a concise structure. On the contrary, a discriminative classifier only models the conditional distribution of labels given inputs, but benefits from effective optimization owing to its succinct structure. This work aims for a solution of generative classifiers that can profit from the merits of both. To this end, we propose an *Anti-Perturbation Inference* (API) method, which searches for anti-perturbations to maximize the lower bound of the joint log-likelihood of inputs and classes. By leveraging the lower bound to approximate Bayes' rule, we construct a generative classifier *Anti-Perturbation Inference Net* (API-Net) upon a single discriminator. It takes advantage of the generative properties to tackle off-manifold examples while maintaining a succinct structure for effective optimization. Experiments show that API successfully neutralizes adversarial perturbations, and API-Net consistently outperforms state-of-the-art defenses on prevailing benchmarks, including CIFAR-10, MNIST, and SVHN. [1]

Keywords: Deep Learning · Neural Networks · Adversarial Defense · Adversarial Training · Generative Classifier

1 Introduction

Deep neural networks (DNNs) have achieved unprecedented success in a wide range of applications [11], [18], [34], [48], [40], [14]. However, they are strikingly susceptible to adversarial examples [41]. The latest attack techniques can generate adversarial perturbations that are seemingly innocuous to humans but easily fool DNNs [12], [33], [4], raising grand challenges to advanced machine learning systems where DNNs are widely deployed [19], [27], [5], [1].

* Corresponding author rrji@xmu.edu.cn.
[1] Our code is available at github.com/dongxinshuai/API-Net.

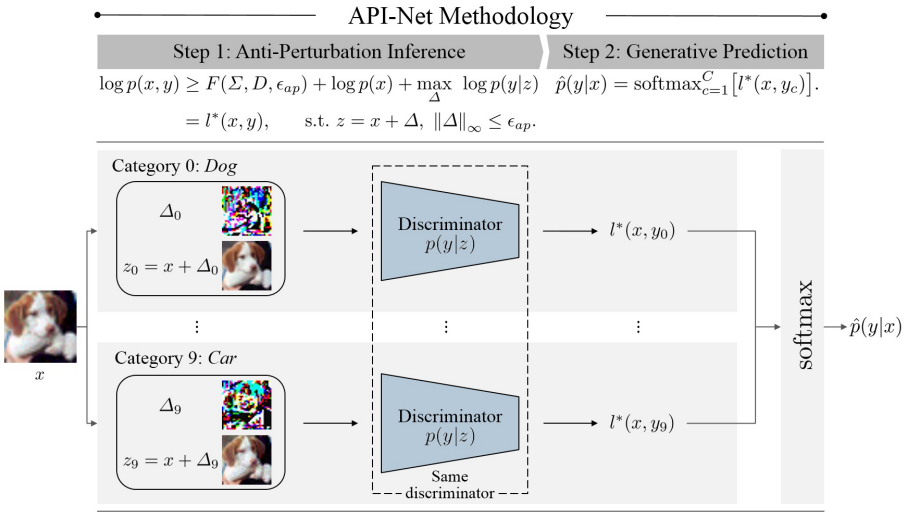


Fig. 1: Overview of our inference procedure. 1) For each sample, search for the anti-perturbation Δ to maximize the lower bound of the joint log-likelihood. 2) Leverage the lower bound $l^*(x, y)$ as an approximation of the log-likelihood for generative prediction using Bayes' rule (best view in color with zooming in)

This phenomenon has attracted increased attention, with focuses on both attack methods and defenses. Attack methods intend to cause the failure of DNNs in their task, by maliciously modifying the input data. Such methods include Fast Gradient Sign Method (FGSM) [12], DeepFool [30], Projected Gradient Descent (PGD) attack [27], C&W attack [4], Universal Perturbations [29], [24], and Wasserstein distance-based attack [43]. The prevailing "off-manifold" conjecture deems adversarial examples as outliers near but away from a class-related manifold [41], [12], [10], [39], though challenged by [10].

To defend against adversarial attacks, several methods have been proposed. Some focus on the detection [28], [21], [26], [23], [25]. Another prominent category aims to enhance the accuracy of DNNs under attacks, which is the focus of this study. Among the work concerning the robust accuracy of classifiers, adversarial training is currently one of the most reliable [2]. Instead of minimizing the loss evaluated at vanilla inputs, adversarial training augments the training data with adversarially perturbed inputs, and builds defense that is shown to be the few resistant to the newest attacks [41], [12], [27], [2].

While the robustness of discriminative classifiers is extensively investigated, few works involve the robustness of generative classifiers [32]. Generative classifiers are intuitively more robust to adversarial examples in that they learn the distribution of inputs and classes, and thus can make meaningful predictions by checking whether the class-specific features are present in the inputs. By comparing the joint log-likelihood of a given input and each class, which relates to the "distance" of the input to a class-specific data manifold, a generative

classifier can estimate the prediction probabilities well even when its inputs are off-manifold examples. However, the classical generative classifiers, *e.g.*, naive Bayes, and linear discriminant analysis [8], perform poorly even on vanilla image classifications, failing to illustrate the robustness of generative classifiers.

Recent advances have been filling up this vacancy. For example, leveraging deep Latent Variable Model (LVM) to construct generative adversarial defense and detection [22]. However, the inference model of the deep LVM is itself a neural network and can be vulnerable. To tackle this problem, [37] proposed optimization-based inference, which substitutes the expectation under the inference model with a maximum likelihood sample to avoid stochastic sampling and to bypass the vulnerabilities. Though effective, current generative solutions are still perplexed by two problems: 1) To learn the joint distribution of samples and classes, a generative classifier often contains multiple components, which brings difficulties to optimization. 2) Adversarial training, which is very useful to build robustness for discriminative classifiers, is hard to apply to current generative models directly. On the contrary, a discriminative classifier only models the distribution of classes conditioned on inputs, and thus takes advantage of its succinctness for effective optimization. Hence, the main question this work aims to address is: is there a solution that can benefit from the merits of both types?

Inspired by [36], which illustrates that a single robust discriminator can be a powerful tool to perform low-level image synthesis tasks, such as inpainting and denoising, we propose to construct a structurally concise generative classifier based on such generative capabilities of a robust discriminator. Specifically, we propose an *Anti-Perturbation Inference* (API) approach and derive a tractable lower bound of the joint log-likelihood of inputs and classes. We use API to search for the anti-perturbation that neutralizes the potential perturbation to maximize the lower bound, and then leverage the lower bound to approximate Bayes' rule for generative predictions. Hence we build a generative classifier, API-Net, upon a single discriminator. API-Net benefits from the merits of both discriminative models and generative models: 1) Its generative properties facilitate modeling class-related manifolds to handle off-manifold examples. 2) Its concise structure ensures effective optimization and thus helps it make full use of adversarial training to gain robustness. We show the inference framework of API-Net in Fig. 1 and summarize the major contributions of this paper as follows:

• We propose a novel anti-perturbation inference approach and derive a lower bound of the joint log-likelihood of inputs and classes. By maximizing the lower bound, we obtain the anti-perturbation that can neutralize adversarial noise.

• Based on the proposed API method, we leverage the lower bound to approximate Bayes' rule and hence build API-Net, a novel generative classifier upon a single discriminator. API-Net takes advantage of both generative and discriminative classifiers to achieve robustness.

• Experiments on multiple prevailing benchmarks show that our approach consistently outperforms state-of-the-art methods with significant margins (*e.g.*, we achieve 63.13% accuracy under PGD-40 attacks on CIFAR-10, while the state-of-the-art is 55.40%).

2 Related Work

For a large amount of work, we focus on the most related ones, which can be classified into three categories, adversarial training, preprocessing-based defenses, and robust generative classifiers.

Adversarial training. Adversarial training can be regarded as a special kind of data augmentation by generating and leveraging adversarial examples during training [12], [19]. For each mini-batch of samples, adversarial images are generated, and further utilized to update the neural networks' parameters [31], [12]. [27] suggests using Projected Gradient Descent (PGD) for adversary generation and currently it is one of the most effective ways to defend against adversarial attacks.

Preprocessing-based defenses. This line of methods aims to destroy the structure of adversarial noise or project the adversarial examples into a learned manifold. Typical methods include image discretization [7], re-scaling [44], feature squeezing [45], thermometer encoding [3], neural-based transformations [38], [35], and matrix estimation [46]. However, most of these defenses rely on obfuscated gradients which can be circumvented by applying the Backward Pass Differentiable Approximation (BPDA) based attacks [2]. Our approach can also be deemed as having non-differentiable preprocessing and should be tested under BPDA-based attacks for rigorous evaluations.

Robust generative classifiers. Generative classifiers are considered more robust if the "off-manifold" conjecture on adversarial examples holds. Following this line, there is a trend of study on the robustness of generative classifiers. Deep Bayes examines the robustness of different factorization structures of deep LVM [22] and builds generative adversarial defense and detection. [37] leverages Variational Auto-Encoder (VAE) [16] to approximate the joint log-likelihood and proposes an optimization-based inference method to circumvent the vulnerable inference model. Despite the effectiveness, existing generative classifiers are puzzled by their complicated structures, which impedes not only effective optimization but also obtaining further robustness through adversarial training.

3 The Proposed Method

In this section, we first introduce the basic task setting and the underlying motivation of our method. We then propose the API approach, which leads to our generative classifier, API-Net. Finally, we present the objective function and specify the optimization procedure for API-Net.

3.1 Motivation

Denote adversarial or vanilla examples as $x \in \mathbb{R}^D$ and class labels as $y \in \{y_c | c = 1, ..., C\}$, where y_c is the one-hot encoding vector for class c. The focus of this work is to build a robust classifier that can maintain high accuracy under adversarial attacks. The attacks aim at generating a perturbation δ given x such that

$x + \delta$ can fool a classifier while the perturbation keeps quasi-imperceptible to humans. To guarantee the perturbation to be quasi-imperceptible, δ is usually bounded by $\|\delta\|_p \leq \epsilon$, where ϵ is a small constant and $\|\cdot\|_p$ is the l_p norm. In this paper, we mainly focus on defense against l_∞-bounded attacks, though our method can also be extended to other l_p-bounded scenario.

To solve the classification problem under attacks, recent research has explored the potential of generative classifiers [22], [37]. Instead of building $p(y|x)$ directly, a generative classifier typically predicts labels using Bayes' rule:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \text{softmax}_{c=1}^{C}\big[\log p(x, y_c)\big], \tag{1}$$

where $\text{softmax}_{c=1}^{C}$ denotes the softmax operation over the C axes. Generative classifiers can better estimate the prediction probabilities to handle off-manifold examples, in that they model the joint distribution and then explicitly consider the distance between the sample and each class-related manifold.

However, though considered to be robust owing to such merit, generative classifiers are often perplexed by their complicated structure. To model the joint log-likelihood for a generative classifier, it often necessitates introducing a latent variable z; the resulting probabilistic graphical model contains multiple components, which not only complicates the implementation but also hinders effective optimization [42]. On the contrary, a discriminative classifier directly models the conditional distribution $p(y|x)$, and thus takes advantage of keeping a concise structure as well as optimizing the quantity of direct interest. Hence, for a better solution of robust generative classifiers, this work makes an attempt in providing a design that can merit from both types.

Inspired by [36] which shows the capabilities of a single robust discriminator to perform image synthesis, we propose to leverage such generative capabilities to build API-Net, a structurally succint generative classifier. To be concrete, a robust conditional distribution of classes given inputs can be leveraged to generate gradients in the input space, and to direct a searching procedure to approach a class-related data manifold. Based on such properties, we derive a tractable lower bound of the joint log-likelihood, which can be further used by API-Net to approximate Bayes' rule for generative predictions. Different from [37], a generative classifier that customizes a VAE for each class, the structure of API-Net entails parameterizing only a single discriminator. The overall process, as shown in Fig. 1, is detailed in what follows.

3.2 Anti-Perturbation Inference Net

To learn the joint distribution $p(x, y)$, we leverage variational inference [16] to introduce a latent variable z and a inference model $q(z|x, y)$. Therefore, a lower bound of the joint log-likelihood can be formulated as (please see Appendix A.1 for the full derivation):

$$\log p(x, y) \geq \mathop{\mathbb{E}}_{q(z|x,y)}[\log p(x, y, z)]. \tag{2}$$

Anti-Perturbation Inference. We then introduce anti-perturbation inference, whose foundation falls in the definition of the latent variable z. We define z as the vanilla sample without any noise, which is in contrast to x that might contain adversarial perturbation (we never know in advance). We aim at an inference procedure: it generates Δ that can nullify the potential adversarial noise, and $z = x + \Delta$ can approach the unpolluted input. Therefore, we term this method *anti-perturbation inference*.

According to the meaning of each variable, we define a directed-graph model with structure:

$$p(x, y, z) = p(y|z)p(z|x)p(x), \tag{3}$$

which suggests that the unpolluted sample z depends on input x, and the class y depends on the unpolluted sample z. Similar to manifold projection defenses [38], [35], we can parameterize the inference model $q(z|x, y)$ by a neural network to calculate the expectation in Eq. 2. Nonetheless, such an inference model is itself a neural network and thus vulnerable [2].

To bypass the vulnerabilities, we follow [37] to leverage optimization-based inference to substitute the expectation under the inference model $q(z|x, y)$:

$$\log p(x, y) \geq \max_{\Delta} \ \log p(y|z)p(z|x)p(x), \tag{4}$$

$$\text{s.t. } z = x + \Delta, \ \|\Delta\|_\infty \leq \epsilon_{ap}, \tag{5}$$

where the small constant ϵ_{ap} bounds the anti-perturbation Δ. This is because we have the prior that the anti-perturbation does not need to be very large to counter the potential adversarial perturbation which is bounded by l_∞ with a small constant ϵ (Section 4.2 shows our defense does not over-fit ϵ).

Besides, owing to the restricted anti-perturbation, we can further simplify the lower bound in Eq. 4 by leveraging the following Lemma (the proof of which and the specific definition of F can be found in Appendix A.2.):

Lemma 1. *Let $p(z|x)$ be a Gaussian, $\mathcal{N}(z|x, \Sigma)$. If $\|z - x\|_\infty \leq \epsilon_{ap}$, then we have $\log p(z|x) \geq F(\Sigma, D, \epsilon_{ap})$, where D denotes the dimension of x and z, and F is a function irrelevant to y.*

According to Lemma 1, a new lower bound $l^*(x, y)$ of the joint log-likelihood can be obtained as follows:

$$\log p(x, y) \geq F(\Sigma, D, \epsilon_{ap}) + \log p(x) + \max_{\Delta} \ \log p(y|z) \tag{6}$$

$$= l^*(x, y), \tag{7}$$

$$\text{s.t. } z = x + \Delta, \ \|\Delta\|_\infty \leq \epsilon_{ap}. \tag{8}$$

Generative Prediction. To make generative predictions, we take $l^*(x, y)$ into Eq. 1 to approximate Bayes' rule. We can rule out the label-irrelevant terms and

formulate the generative prediction as:

$$p(y|x) \approx \text{softmax}_{c=1}^{C}\big[l^*(x, y_c)\big] \tag{9}$$

$$= \text{softmax}_{c=1}^{C}\big[\max_{\Delta} \ \log p(y_c|z)\big], \tag{10}$$

$$\text{s.t. } z = x + \Delta, \ \|\Delta\|_\infty \leq \epsilon_{ap}. \tag{11}$$

Based on Eqs. 10 and 11, we here construct the generative classifier API-Net. We parameterize $p(y|z)$ with an adversarially robust neural network with parameter θ as $p_\theta(y|z)$, and the resulting generative classifier $\hat{p}(y|x;\theta)$ based on $p_\theta(y|z)$ is formulated as:

$$\hat{p}(y|x;\theta) = \text{softmax}_{c=1}^{C}\big[\max_{\Delta} \ \log p_\theta(y_c|z)\big], \tag{12}$$

$$\text{s.t. } z = x + \Delta, \ \|\Delta\|_\infty \leq \epsilon_{ap}, \tag{13}$$

where $\hat{p}(y|x;\theta)$ depends on the underlying $p_\theta(y|z)$ thus conditioned on θ.

Eqs. 12 and 13 define the proposed API-Net. As a generative classifier, API-Net makes generative predictions by comparing between log-likelihood of classes, which facilitates tackling off-manifold examples. In contrast to previous solutions of generative classifier [22], [37], API-Net can be implemented with rather minimal effort and can take advantage of effective optimization, since it is built upon only a single conditional distribution $p_\theta(y|z)$. Besides, by maximizing the lower bound of the joint log-likelihood, Δ strives to neutralize the adversarial noise and $z = x + \Delta$ seeks to approach the vanilla sample to defend against attacks (the effectiveness of which is shown in Section 4.3).

3.3 Optimization

A key ingredient of API-Net is the image synthesis ability, which is achieved by making the underlying $p_\theta(y|z)$ robust [36]. By building API-Net upon an off-the-shelf robust discriminator, we can achieve additional robustness without training (validated in Section 4.3). Next, we introduce the training objective function of API-Net towards further robustness.

Objective Function. A typical objective function for generative models is to maximize the joint log-likelihood. However, the essential performance we consider in this work is the classification accuracy, which can often be enhanced by training models discriminatively to gain more powerful discrimination [15], [20]. We thereby treat API-Net as a whole and minimize the expectation of cross-entropy loss under the data distribution $\mathcal{D}$ to optimize θ:

$$\min_{\theta}\big[\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[-\log \hat{p}(y|x;\theta)]\big]. \tag{14}$$

This objective function is also beneficial for API-Net to incorporate adversarial training, which is initially designed for a discriminative loss, to gain further

Algorithm 1 API-Net Training

Input: dataset $\mathcal{D}$, number of categories C, ϵ_{ap} for anti-perturbation, ϵ_{train} for adversarial training, parameters of PGD for anti-perturbation and for adversarial training.
Output: Parameters θ.

repeat
  for random mini-batch $\{x_i, y_i\}_{i=1}^{n} \sim \mathcal{D}$ **do**
    for every x_i, y_i in the mini-batch (in parallel) **do**
      Solve δ in Eqs. 15 and 16 by PGD using gradient approximation;
      for $c = 1$ to C **do**
        Solve Δ_c in Eqs. 12 and 13 by PGD for anti-perturbation inference;
      end for
    end for
    Compute the loss defined in Eqs. 15 and 16 and then update θ;
  end for
until the training converges.

robustness. We absorb adversarial training as a data augmentation technique to formulate the final objective function of API-Net:

$$\min_{\theta} \Big[\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\max_{\delta} -\log \hat{p}(y|x+\delta;\theta)] \Big], \tag{15}$$

$$\text{s.t. } \|\delta\|_{\infty} \leq \epsilon_{train}, \tag{16}$$

where ϵ_{train} sets the allowed perturbation budget for adversarial training (different from ϵ_{ap} which bounds the anti-perturbation).

Optimization Procedure. We show the overall training process of API-Net in Algorithm 1. Projected gradient descent (PGD) [6], [9], [27] is employed for the optimization of Δ and δ. For the adversarial training defined in Eqs. 15 and 16 and the evaluation of our method under gradient-based attacks, as $\hat{p}(y|x;\theta)$ is non-differentiable with respect to x, we leverage the following two strategies to approximate the gradients:

(1) Backward Pass Differentiable Approximation (BPDA) [2]. Since ϵ_{ap} is a small constant, we approximate the derivative of z with respect to x as the derivative of the identity function: $\bigtriangledown_x z \approx \bigtriangledown_x x = 1$ for backward passes.

(2) Forward and Backward Differentiable Approximation. As ϵ_{ap} is small, we simply set $z = x$ to calculate the gradient for both forward and backward passes.

During training, the second strategy is used considering the computational efficiency. For evaluation, we conduct attacks based on both strategies for a rigorous examination of the proposed method.

4 Experiments

In this section, we first present the experimental settings. We then evaluate the overall robustness of the proposed API-Net and compare it with state-of-the-arts in Section 4.2. We finally conduct ablation studies in Section 4.3.

Table 1: Accuracy (%) under white-box attacks on CIFAR-10 with $\epsilon = 8/255$

Method	Architecture	Clean	FGSM	PGD-40	PGD-100	C&W-40	C&W-100
Standard	ResNet18	**94.46**	24.24	0.00	0.00	0.00	0.00
Madry	ResNet18	82.15	61.83	47.52	47.29	46.78	46.66
ME-Net	ResNet18	84.00	-	55.40	53.50	-	-
Trades	ResNet18	82.83	64.14	52.08	51.97	49.05	48.94
API-Net	ResNet18	81.25	**65.71**	**63.13**	**62.87**	**55.73**	**54.57**

Table 2: Accuracy (%) under white-box attacks on SVHN with $\epsilon = 8/255$

Method	Architecture	Clean	FGSM	PGD-40	PGD-100	C&W-40	C&W-100
Standard	ResNet18	**96.62**	45.23	0.84	0.52	0.90	0.62
Madry	ResNet18	94.30	74.55	53.37	52.91	51.95	51.83
ME-Net	ResNet18	87.60	-	71.90	69.80	-	-
Trades	ResNet18	91.06	72.83	58.21	57.83	54.71	54.65
API-Net	ResNet18	87.72	**80.34**	**74.36**	**73.68**	**62.51**	**60.25**

4.1 Experimental Settings

Datasets. The experiments are performed on CIFAR-10, SVHN, and MNIST.

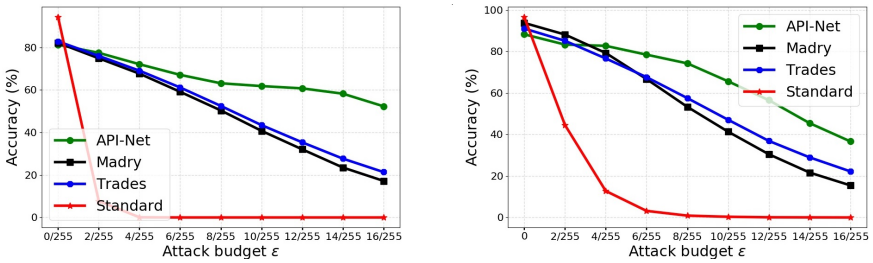
Compared Methods. *Standard:* standard training approach using clean images [17]. *Madry:* adversarial training based defense using PGD [27]. *ME-Net:* preprocessing-based defense with Matrix-Estimation [46]. We plot its accuracy under BPDA-based attacks. *Trades:* adversarial training based approach with KL-divergence-based adversarial examples generation and regularization [47]. We plot the performance of Trades under its best setting where $\frac{1}{\lambda} = 6$.

Implementation Details. We implement the Standard, Madry, and Trades methods, and report the robust accuracy of ME-Net according to [46]. We set the pixel values in $[0, 1]$, and use PGD [27] of 7 iterations with $\epsilon_{train} = 8/255$ and step-size 0.007 on CIFAR-10 and SVHN, and PGD of 40 iterations with $\epsilon_{train} = 76.5/255$ and step-size 0.01 on MNIST for adversarial training. We first leverage Madry's method to train the underlying $p_\theta(y|z)$ for a guarantee of generative capabilities and then train API-Net following Eqs. 15 and 16. To align with past work, we apply data augmentation on CIFAR-10 and SVHN datasets following [13] and do not apply any data augmentation on MNIST.

Parameters of API. We set $\epsilon_{ap} = 14/255$ for CIFAR-10, and $\epsilon_{ap} = 12/255$ for SVHN and MNIST. We use PGD of 8 iterations with step-size 0.007 for CIFAR-10 and SVHN, and PGD of 8 iterations with step-size 0.01 for MNIST.

Table 3: Accuracy (%) under white-box attacks on MNIST with $\epsilon = 76.5/255$

Method	Architecture	Clean	FGSM	PGD-40	PGD-100	C&W-40	C&W-100
Standard	LeNet	**99.16**	-	0.15	0.05	-	-
ME-Net	LeNet	97.40	-	94.00	91.80	-	-
API-Net	LeNet	98.30	-	**94.22**	**92.09**	-	-
Standard	SmallCNN	**99.41**	50.72	1.50	0.00	0.10	0.00
Madry	SmallCNN	99.31	97.86	96.64	95.68	96.77	95.62
Trades	SmallCNN	99.15	97.95	96.81	96.02	96.91	95.98
API-Net	SmallCNN	99.21	**98.39**	**97.10**	**96.35**	**97.17**	**96.34**



(a) CIFAR-10 (b) SVHN

Fig. 2: Accuracy under PGD-40 attacks with ϵ varying from 0/255 to 16/255

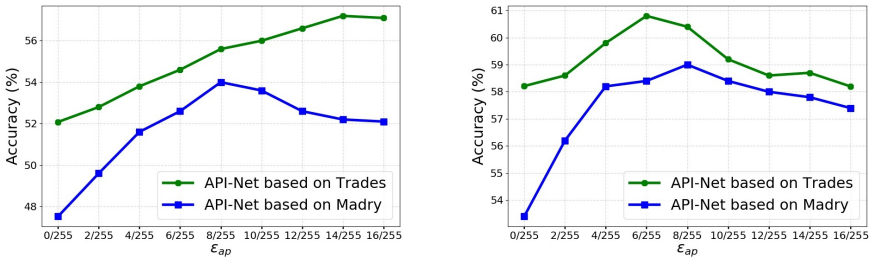
Attack Details. We mainly focus on l_∞-bounded white-box attacks. The white-box attacks are deemed as the most powerful attacks since the attacker has full information about the defense model under this setting. We leverage FGSM [12] and two currently strongest gradient-based attacks: PGD and C&W (l_∞-bounded, k=50) with T iterations (PGD-T and C&W-T) and random restart [27], [4]. Aligned with past work, we mainly focus on the performance under attacks with $\epsilon = 8/255$ on CIFAR-10 and SVHN, and $\epsilon = 76.5/255$ on MNIST. As defined in Section 3.3, we use two strategies to approximate the gradient for the attacks and report the worst accuracy for strict evaluation.

4.2 Robustness

Accuracy under Attacks across Datasets. We compare the robust accuracy of API-Net with those of state-of-the-art defense methods. The results on CIFAR-10, SVHN and MNIST are respectively shown in Tab. 1, Tab. 2, and Tab. 3, which clearly show that API-Net surpasses the state-of-the-art methods against multiple white-box attacks of different iterations. In particular, we surpass the runner-up method by 8% under the most prevailing PGD attack on CIFAR-10 and by 4% on SVHN. These quantitative results demonstrate the outstanding robust performance of API-Net.

Table 4: Ablation study on training and initialization. The accuracy (%) is reported under the PGD-40 attack with $\epsilon = 8/255$

Initialization	Dataset	Learning rate	Trades	ME-Net
Random	CIFAR-10	0.1	**52.08**	**55.40**
From Madry	CIFAR-10	0.1	50.08	52.92
From Madry	CIFAR-10	0.01	47.24	48.55
From Madry	CIFAR-10	0.001	50.52	50.03
Method	Dataset	Initialization	w.o. train	with train
API-Net	CIFAR-10	Madry	52.08	**63.13**
API-Net	SVHN	Madry	58.58	**74.36**



(a) CIFAR-10 (b) SVHN

Fig. 3: Robustness of API-Net based on off-the-shelf robust discriminators without further training under the PGD-40 attack with $\epsilon = 8/255$

Accuracy under Attacks with Different ϵ. In this section, we evaluate the accuracy under the PGD-40 attack with ϵ varying from 0 to 16/255 with interval 2. As shown in Fig. 2, the proposed API-Net achieves leading robustness. It verifies that the robustness of API-Net is not based on over-fitting a specific attack ϵ. Rather, when ϵ increases, the accuracy of our method declines at a slower rate compared to the state-of-the-arts, though all the methods are trained with the same $\epsilon_{train} = 8/255$. This experiment demonstrates the potential of our method to be more applicable to real-life machine learning systems where the bound on perturbations cannot be known in advance.

4.3 Ablation Study

API-Net Based on off-the-Shelf Robust Discriminator. In this section, we investigate the robustness gain merely owing to the design of API-Net. To this end, we initialize the underlying $p_\theta(y|z)$ of API-Net with off-the-shelf robust models, Madry, and Trades, and then test the accuracy under attacks without any training. We plot the accuracy under PGD-40 attacks with ϵ_{ap} varying from

(a) Classes 0 to 4 (b) Classes 5 to 9

Fig. 4: Qualitative results to show the effectiveness of the proposed API. **Column 1:** vanilla images. **Column 2:** adversarially perturbed images as input x. **Column 3:** z generated by API conditioned on $y_{(t+1) \bmod C}$. **Column 4:** z generated conditioned on true label y_t (best view in color with zooming in)

0 to 16/255. We note that when $\epsilon_{ap} = 0$, API-Net degenerates to the original discriminative classifier and its accuracy corresponds to the off-the-shelf model.

As shown in Fig. 3, our proposed API-Net can be directly deployed to the off-the-shelf robust discriminators to obtain additional robustness. Specifically, API-Net provides additional gains of approximate 5% robust accuracy on CIFAR-10 and 4% on SVHN. It demonstrates that the proposed API-Net makes better use of the underlying discriminative distribution to build robustness.

Training and Initialization. We here investigate the effectiveness of the proposed API-Net training schedule. We compare the robust accuracy between API-Net initialized with Madry without training and API-Net initialized with Madry plus training. As shown in Tab. 4, the training schedule contributes to about 11% gain in robust accuracy on CIFAR-10 and 15% on SVHN. We also examine the effect of initialization. We train Trades and ME-Net on CIFAR-10 with initialization from Madry's model and try different learning rates to ensure a good convergence. As shown in Tab. 4, the initializations do not advance the robustness neither for Trades nor ME-Net.

Visualization of API. To emphasize the consistency, we use ten images chosen from the first one of each class in the test set of SVHN. We aim to qualitatively demonstrate how anti-perturbations work. To this end, we apply PGD attacks to generate adversarial examples as inputs and conduct API with $\epsilon_{ap} = 12/255$ to

Table 5: Ablation study on the optimal ϵ_{ap}. The accuracy (%) is reported under the PGD-40 attack

ϵ_{ap}	6/255	8/255	10/255	12/255	14/255	16/255
CIFAR-10	55.09	58.68	60.90	62.32	**63.13**	62.83
SVHN	61.14	69.20	72.45	**74.40**	73.98	73.39
MNIST	96.23	96.49	96.72	**97.12**	96.32	96.69

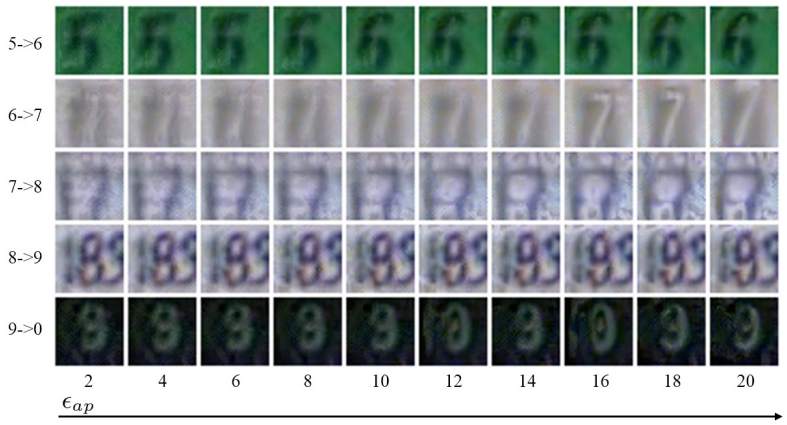


Fig. 5: Visualization of how z changes with ϵ_{ap} conditioned on $y_{(t+1) \bmod C}$. From left to right, ϵ_{ap} varies from 2 to 20 (best view in color with zooming in)

obtain z. As shown in Fig. 4, when conditioned on the true label y_t, $t \in [C]$, the anti-perturbation effectively counters the adversarial noise, leading to z (Fig. 4 column 4) that is very similar to the vanilla image (Fig. 4, column 1).

On the contrary, when conditioned on a wrong label, *e.g.*, $y_{(t+1) \bmod C}$, the resulting z would be dubious (Fig. 4 column 3). This is beneficial since it would lead to a low $p_\theta(y|z)$ and thus a low $l^*(x, y)$, which results in a low prediction probability for this wrong class. We also notice that some images in the column 3 of Fig. 4 start to generate features of the wrong class $y_{(t+1) \bmod C}$. This reveals the importance of an appropriate value of ϵ_{ap}, which should be dataset-related, to preserve the original global structure of each image.

Optimal Searching Scope of Anti-Perturbation. Intuitively, we consider two points concerning the optimal value of ϵ_{ap}: 1) it should be large enough to ensure a powerful Δ to counter potential perturbations. 2) it should be limited to prevent z from being a plausible image of a wrong class. To qualitatively analyze, we change ϵ_{ap} from 2 to 20 and visualize z conditioned on $y_{(t+1) \bmod C}$. As shown in Fig. 5, when ϵ_{ap} increases to 12/255, z begins to contain plausible features of class $y_{(t+1) \bmod C}$, which indicates ϵ_{ap} should be no more than 12/255. We then

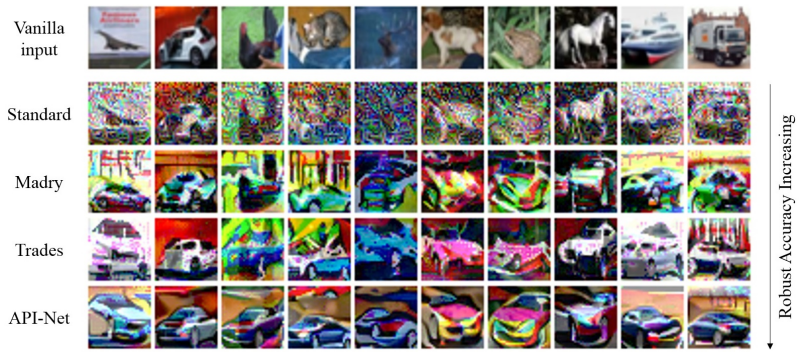


Fig. 6: Visualization of large-ϵ adversarial examples on CIFAR-10. Images are manipulated into being classified as a car (best view in color with zooming in)

quantitatively analyze. As shown in Tab. 5, on SVHN, API-Net performs the best when $\epsilon_{ap} = 12/255$, which is consistent with the qualitative results.

Hidden Representation. We here explore the learned hidden representation of API-Net by leveraging the gradients and see what convinces API-Net most. We employ PGD to manipulate images from CIFAR-10 into being classified as a car from each model's perspective. We set a large $\epsilon = 80/255$ to alter the global structure and generate salient patterns, and run 1000 iterations to ensure a good convergence. As shown in Fig. 6, based on the gradients provided by API-Net, highly plausible patterns are generated, both in terms of structure and texture. These suggest that API-Net does not rely on obfuscated gradients. Rather, API-Net has learned representations consistent best with human perception.

5 Discussion and Conclusion

Despite the success in numerous applications, DNNs' performance is far from robust compared to that of a human. In this work, we made an attempt in providing a solution, API-Net, that can profit from the merits of both discriminative and generative classifiers to improve the robustness. The experiments showed that API-Net outperforms state-of-the-art defenses and generates gradients that result in perceptually meaningful representations. We hope that this work can be a stepping stone towards reliable DNNs for real-life machine learning applications.

Acknowledgements

This work is supported by the Nature Science Foundation of China (No.U1705262, No.6177244, No.61572410, No.61802324 and No.61702136), National Key R&D Program (No.2017YFC0113000, and No.2016YFB1001503), Key R&D Program of Jiangxi Province (No. 20171ACH80022) and Natural Science Foundation of Guangdong Provice in China (No.2019B1515120049).

References

1. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.J., Srivastava, M., Chang, K.W.: Generating natural language adversarial examples. In: EMNLP (2018)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML (2018)
3. Buckman, J., Roy, A., Raffel, C., Goodfellow, I.: Thermometer encoding: One hot way to resist adversarial examples. In: ICLR (2018)
4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: SP. IEEE (2017)
5. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: SPW. IEEE (2018)
6. Cauchy, A.: Méthode générale pour la résolution des systemes d'équations simultanées. Comp. Rend. Sci. Paris **25**(1847), 536–538 (1847)
7. Chen, J., Wu, X., Liang, Y., Jha, S.: Improving adversarial robustness by data-specific discretization. CoRR, abs/1805.07816 (2018)
8. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics **7**(2), 179–188 (1936)
9. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval research logistics quarterly **3**(1-2), 95–110 (1956)
10. Gilmer, J., Metz, L., Faghri, F., Schoenholz, S.S., Raghu, M., Wattenberg, M., Goodfellow, I.: Adversarial spheres. arXiv preprint arXiv:1801.02774 (2018)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine (2012)
15. Holub, A., Perona, P.: A discriminative framework for modelling object classes. In: CVPR (2005)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)
19. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
20. Lasserre, J.A., Bishop, C.M., Minka, T.P.: Principled hybrids of generative and discriminative models. In: CVPR (2006)
21. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. In: ICCV (2017)
22. Li, Y., Bradshaw, J., Sharma, Y.: Are generative classifiers more robust to adversarial attacks? In: ICML (2019)
23. Li, Y., Gal, Y.: Dropout inference in bayesian neural networks with alpha-divergences. In: ICML (2017)

24. Liu, H., Ji, R., Li, J., Zhang, B., Gao, Y., Wu, Y., Huang, F.: Universal adversarial perturbation via prior driven uncertainty approximation. In: ICCV (2019)
25. Louizos, C., Welling, M.: Multiplicative normalizing flows for variational bayesian neural networks. In: ICML (2017)
26. Lu, J., Issaranon, T., Forsyth, D.: Safetynet: Detecting and rejecting adversarial examples robustly. In: ICCV (2017)
27. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
28. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. In: ICLR (2017)
29. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR (2017)
30. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR (2016)
31. Na, T., Ko, J.H., Mukhopadhyay, S.: Cascade adversarial machine learning regularized with a unified embedding. arXiv preprint arXiv:1708.02582 (2017)
32. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: NeurIPS (2002)
33. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: EuroS&P. IEEE (2016)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
35. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 (2018)
36. Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Image synthesis with a single (robust) classifier. In: NeurIPS (2019)
37. Schott, L., Rauber, J., Bethge, M., Brendel, W.: Towards the first adversarially robust neural network model on mnist. In: ICLR (2019)
38. Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N.: Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In: ICLR (2018)
39. Stutz, D., Hein, M., Schiele, B.: Disentangling adversarial robustness and generalization. In: CVPR (2019)
40. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NeurIPS (2014)
41. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2013)
42. Tramer, F., Carlini, N., Brendel, W., Madry, A.: On adaptive attacks to adversarial example defenses. arXiv preprint arXiv:2002.08347 (2020)
43. Wong, E., Schmidt, F.R., Kolter, J.Z.: Wasserstein adversarial examples via projected sinkhorn iterations. In: ICML (2019)
44. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. In: ICLR (2018)
45. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
46. Yang, Y., Zhang, G., Katabi, D., Xu, Z.: Me-net: Towards effective adversarial robustness with matrix estimation. In: ICML (2019)
47. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML (2019)

48. Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: Freeanchor: Learning to match anchors for visual object detection. In: NeurIPS (2019)