TOWARDS COUNTERACTING ADVERSARIAL PERTURBATIONS TO IMPROVE ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Studies show that neural networks are susceptible to adversarial attacks. This exposes a potential threat to neural network-based artificial intelligence systems. We observe that the probability of the correct result outputted by the network increases by applying small perturbations generated regarding labels other than the original predicted one to adversarial examples. Based on this observation, we propose a method of counteracting adversarial perturbations to improve adversarial robustness. In our method, we randomly select a number of labels and generate small perturbations regarding these labels. The generated perturbations are added together and then clamped to a specified space. The obtained perturbation is finally added to the adversarial example counteract the adversarial perturbation contained in the example. The proposed method is applied at inference time and does not require retraining the model. We validate the proposed method on CIFAR-10 and CIFAR-100. The experimental results demonstrate that our method effectively improves the defense performance of the baseline methods, especially against attacks that use more iterations.

1 INTRODUCTION

Deep neural networks (DNNs) have become the dominant approach for various tasks including image understanding, natural language processing and speech recognition. However, recent studies demonstrate that neural networks are vulnerable to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). That is, these network models make an incorrect prediction with high confidence for inputs that are only slightly different from correctly predicted examples. This reveals a potential threat to neural network-based artificial intelligence systems, many of which have been widely deployed in real-world applications.

The adversarial vulnerability of neural networks reveals fundamental blind spots in the learning algorithm. Even with advanced learning and regularization techniques, neural networks are not learning the true underlying distribution of the training data, although they can obtain extraordinary performance on the test set. This phenomenon is now attracting much research attention. There have been increasing studies attempting to explain networks' adversarial vulnerability and develop methods to resist against adversarial examples (Madry et al., 2018; Zhang et al., 2020; Pang et al., 2020). While much progress has been made, most existing studies still remain preliminary. Because it is difficult to construct a theoretical model to explain the adversarial perturbation generating process, defending against adversarial attacks is still a challenging task.

Existing methods of resisting adversarial perturbations perform defense in the training procedure or at inference time. Training stage defense methods attempt to increase model capacity to improve adversarial robustness. One of the commonly used methods is adversarial training, in which a mixture of adversarial and clean examples are used to train the neural network. The adversarial training method can be seen as minimizing the worst case loss when the training example is perturbed by an adversary (Goodfellow et al., 2015). Adversarial training requires an adversary to produce adversarial examples in the training procedure. This can significantly increase the training time. Adversarial training can also result in reduced performance on clean examples. Lamb et al. (2019) recently introduced interpolated adversarial training (IAT) that incorporates interpolation-based training into the adversarial training framework. The IAT method helps to improve performance on clean examples while preserving adversarial robustness.

As to inference time defense methods, the main idea is to transfer adversarial perturbations such that the obtained inputs are no longer adversarial. Tabacof & Valle (2016) studied the use of random noise like Gaussian noise and heavy-tail noise to resist adversarial perturbations. Xie et al. (2018) introduced to apply two randomization operations, *i.e.,* random resizing and random zero padding, to inputs to improve adversarial robustness. Guo et al. (2018) investigated the use of randomly cropping and rescaling to transfer adversarial perturbations. More recently, Pang et al. (2020) proposed the mixup inference method that uses the interpolation between the input and a randomly selected clean image for inference. This method can shrink adversarial perturbations somewhat by applying interpolation. Inference time defense methods can be directly applied to off-the-shelf network models without retraining or finetuning. This can be much efficient as compared to training stage defense methods.

Though adversarial perturbations are not readily perceivable by a human observer, it is suggested that adversarial examples are outside the natural image manifold (Hu et al., 2019). Previous studies have suggested that adversarial vulnerability are caused by the locally unstable behavior of classifiers on data manifolds (Fawzi et al., 2016; Pang et al., 2018). Pang et al. (2020) also suggested that adversarial perturbations have the locality property and could be resisted by breaking the locality. Existing inference time defense approach mainly use simple stochastic transformations like mixup and random cropping and rescaling to break the locality. In this research, we observe that applying small perturbations that are generated regarding categories other than the predicted one to the adversarial example helps to counteract the adversarial effect. Motivated by this observation, we proposed a method that uses small perturbations to resist adversarial perturbations. In the proposed method, we first generate small perturbation regarding a number of randomly selected categories using first-order gradient. These small perturbations are added together and projected onto a specified space before applied to the adversarial example. Our method can be used as a preliminary step before applying existing inference stage defense methods.

As far as we know, this is the first research on employing perturbations that generated using local gradient to resist adversarial attacks. We evaluate the proposed method on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) against both targeted and untargeted attacks. The experimental results show that our method is effective in improving performance of several methods that were recently developed for defense at inference time.

The contributions of this paper can be summarized as follows:

- We propose a method that uses small perturbations generated by employing local gradient to resist adversarial perturbations. We show that the proposed method is effective in counteracting adversarial perturbations to improve neural networks' robustness.
- Extensive experiments are conducted on CIFAR-10 and CIFAR-100 to validate the proposed method. The experimental results show that the proposed method is complementary to the baseline methods to improve defense performance for both untargeted and targeted attacks and that the propsoed method performs well in resisting against attacks that use more interations.

2 Preliminary

2.1 Adversarial Examples

We consider a neural network $f(\cdot)$ with parameters $\boldsymbol{\theta}$ that outputs a vector of probabilities of $L = \{1, 2, ..., l\}$ categories. In supervised learning, empirical risk minimization (ERM) (Vapnik, 1998) has been commonly used as the principle to optimize the parameters by using a training set. Given an input $\boldsymbol{x}$, the network the network makes a prediction $c(\boldsymbol{x}) = \arg\max_{j \in L} f_j(\boldsymbol{x})$. The prediction is correct if $c(\boldsymbol{x})$ is the same as the actual target $c^*(\boldsymbol{x})$.

Unfortunately, ERM trained neural networks are vulnerable to adversarial examples, inputs formed by applying small but intentionally crafted perturbations (Szegedy et al., 2014; Madry et al., 2018). That is, an adversarial example x' is close to a clean example x under a distance metric, *e.g.,* ℓ_∞ distance, but the neural network outputs an incorrect result for the adversarial example x' with high confidence. In most cases, the difference between the adversarial example and clean example is not readily recognizable to humans.

2.2 ATTACK METHODS

Existing attack methods can be categorized into white-box attacks and black-box attacks. We focus on defending against white-box attacks, wherein the adversary has full access to the network model including the architecture and weights. We consider two attack methods: the fast gradient sign (FGSM) method and the projected gradient descent (PGD) method.

The FGSM method is a one-step attack method. It produces adversarial perturbations that yield the highest loss increase in the gradient sign direction. Let $\boldsymbol{x}$ be the input to a network model, y the label associate with $\boldsymbol{x}$ and $L(\boldsymbol{\theta}, \boldsymbol{x}, y)$ be the loss function for training the neural network. The FGSM method a produces max-norm constrained perturbation as follows:

$$\eta = \epsilon \text{sign}(\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)), \tag{1}$$

where ϵ denotes the max-norm constraint. This method was developed based on the view that the primary cause of neural networks' adversarial vulnerability is their linear nature. The required gradient can be computed efficiently using backpropagation.

The PGD method is a multistep attack method that iteratively applies projected gradient descent on the negative loss function (Kurakin et al., 2016) to the input $\boldsymbol{x}$ as follows:

$$\boldsymbol{x}^{t+1} = \Pi_{\boldsymbol{x}+\mathcal{S}}(\boldsymbol{x}^t + \alpha \text{sign}(\nabla_{\boldsymbol{x}^t} L(\boldsymbol{\theta}, \boldsymbol{x}^t, y))), \tag{2}$$

where α denotes the step size and Π denotes the projection operator that projects the perturbed input onto $\boldsymbol{x} + \mathcal{S}$. We consider projecting the perturbed input onto a predefined ℓ_∞ ball from the original input. The PGD attack method can be seen as a multistep FGSM method. It is a much strong adversary that reliably causes a variety of neural networks to misclassify their input.

3 METHODOLOGY

While many studies have been conducted on defense against adversarial attacks at inference time, these studies have not considered using gradient information to resist adversarial perturbations. Previous work has suggested that the cause of neural networks' adversarial vulnerability is their linear nature (Goodfellow et al., 2015). It would be more effective to use first-order gradient information to counteract the adversarial perturbations such that resulted perturbations no longer leads the model to make an incorrect prediction.

Adversarial perturbations are small crafted perturbations that minimally affect the visual quality of inputs but cause the neural network to misclassify the inputs in favor of an incorrect answer with high probability. We show that this effect can be counteracted by applying perturbations that are generated regarding labels other than the predicted one by using first-order gradient. An illustration of this phenomenon is shown in Figure **??**. We see that by adding perturbations that are generated regarding non-predicted labels to the input, the prediction probability for the correct category increases and that for the incorrect label is suppressed.

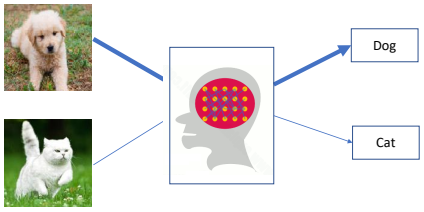

Figure 1: An illustration

Based on this phenomenon, we propose a method of counteracting adversarial perturbations to improve adversarial robustness. In the proposed method, we generate small perturbations regarding a number of randomly selected labels and apply these perturbations to the input before feeding it the neural network for prediction. Let $\boldsymbol{x}$ be the input to a model, which can be an adversarial example or clean example. We randomly select N class labels and generate small perturbations regarding the N selected labels using local first-order gradient information. These N small perturbations are added together and then projected onto a ℓ_∞-bounded space before applying to the input. This process can be formulated as follows:

Algorithm 1 Framework of ensemble learning for our system.

Input:
The set of positive samples for current batch, P_n;
Output:
1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n;
2: $E_n = E_{n-1} \cup E$;
3: Classifying samples in $U_n - T_n$ by E_n;
4: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;
5: **return** E_n;

$$\tilde{x} = \boldsymbol{x} + \Pi_{\boldsymbol{x}+\mathcal{C}}(\beta \sum_{i=1}^{N} \eta_i(\boldsymbol{x})), \tag{3}$$

where $\eta_i(\boldsymbol{x})$ denotes the perturbation generated regarding the i-th selected class label, $\mathcal{C} = \{\boldsymbol{t} | \, \|\boldsymbol{t} - \boldsymbol{x}\|_\infty \leq \mu\}$ is a μ bounded ℓ_∞ space and β is a hyperparameter. The one-step PGD method is used to generate the small perturbation. This is the same as using the FGSM method and empirically achieves better performance than using multiple steps. The perturbations can be generated in an untargeted or targeted manner. The combined perturbation is constrained to the space $\mathcal{C}$. This can ensure that the obtained input is visually similar to the original image. We detail the procedure for counteracting adversarial perturbations in Algorithm **??**.

Discussion and Analysis Adversarial examples exposes underlying flaws in the training algorithms. While much progress has been made in resisting against adversarial attacks, it is still difficult to theoretically understand neural networks' vulnerability to adversarial examples. Previous work (Athalye et al., 2018) has suggested that the adversarial perturbation δ can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min \|\delta\|_p, \\ &s.t.\ c(\boldsymbol{x} + \delta) \neq c^*(\boldsymbol{x}), \|\delta\|_p \leq \mu, \end{aligned} \tag{4}$$

where μ is a hyperparameter constraining the size of the perturbation. This problem can be effectively solved by gradient descent-based attack methods like PGD and FGSM that reliably cause neural networks to output an incorrect result. These attack methods typically use local first-order gradient information to find the optimal solution. Because state-of-the-art neural networks usually have many parameters, perturbations obtained with these attack methods may overfit to inputs. Therefore, perturbing and transferring these adversarial perturbation could be an effective way to suppress the adversarial effect. Unlike previous random transformation-based methods, we employ the use of local first-order gradient information to counteract the adversarial effect. We show that the proposed method is effective in improving defense performance, especially against strong iterative optimization-based attack methods.

Let $\boldsymbol{x}_0$ be an clean example and δ be the adversarial perturbation. In our method, the following input is taken as input to the neural network:

$$\boldsymbol{x}_0 + \delta \cdot \mathbf{1}_{z(\boldsymbol{x}_0)} + \Pi_{\mathcal{C}}(\beta \sum_{i=1}^{N} \eta_i), \text{ where } \mathbf{1}_{z(\boldsymbol{x}_0)} = \begin{cases} 0, & \boldsymbol{x}_0 \text{ is subject to adversarial attack,} \\ 1, & \boldsymbol{x}_0 \text{ is not subject to adversarial attack.} \end{cases} \tag{5}$$

The perturbation η_i generated to counteract the adversarial perturbation should be small, otherwise it would be a new adversarial perturbation. This would essentially have no effect in counteracting the adversarial perturbation. Adversarial training that has shown to be effective to improve adversarial robustness usually employ a first-order adversarial like PGD to provide adversarial examples. Adversarial training usually use an adversary that uses local gradient information to provide adversarial examples for training. These adversarial examples help to regularize the model to be resistant to adversarial examples. We show through experiments that our method is complementary to adversarial training to improve overall defense performance against both untargeted and targeted attacks.

Table 1: Classification accuray (%) on adversarial exmples crafted on the test set of CIFAR-10. We report performance under PGD attacks with 10, 50 and 200 iterations. The numbers in parentheses are performance improvements achieved by applying our method.

Method	Untargeted attacks			Targeted attacks		
	PGD_{10}	PGD_{50}	PGD_{200}	PGD_{10}	PGD_{50}	PGD_{200}
Mixup	3.6	3.2	3.1	≤ 1	≤ 1	≤ 1
Ours + Mixup	26.5	27.2	31.7	55.3	59.7	63.1
Xie et al. (2018) + Mixup	23.0	19.6	19.1	38.4	31.1	25.2
Ours + Xie et al.'s (2018) + Mixup	$35.5_{(+12.5)}$	$36.4_{(+16.8)}$	$41.3_{(+22.2)}$	$60.3_{(+21.9)}$	$63.7_{(+32.6)}$	$66.1_{(+40.9)}$
Guo's	31.2	28.8	28.3	57.8	49.1	48.9
Ours + Guo's	$50.5_{(+19.3)}$	$51.8_{(+23.0)}$	$56.2_{(+27.9)}$	$74.1_{(+16.3)}$	$78.0_{(+28.9)}$	$79.8_{(+30.9)}$
MI-OL Pang et al. (2020)	26.1	18.8	18.3	55.6	51.2	50.8
Ours + MI-OL Pang et al. (2020)	$38.8_{(+12.7)}$	$38.6_{(+19.8)}$	$41.5_{(+23.2)}$	$55.3_{(-0.3)}$	$60.0_{(+8.8)}$	$63.0_{(+12.2)}$
IAT	46.7	43.5	42.5	65.6	62.5	61.9
Ours + IAT	60.1	60.8	60.9	68.0	69.1	68.9
Xie et al. (2018) + Mixup	59.7	58.4	57.9	71.1	69.7	69.3
Ours + Xie's + Mixup	$68.7_{(+9.4)}$	$69.2_{(+10.8)}$	$69.3_{(+11.4)}$	$76.7_{(+5.6)}$	$76.6_{(+6.9)}$	$76.6_{(+7.3)}$
Guo	60.9	60.7	60.3	73.2	72.1	71.6
Ours+Guo	$67.2_{(+6.3)}$	$68.1_{(+7.4)}$	$68.2_{(+7.9)}$	$72.6_{(-0.6)}$	$72.4_{(+0.3)}$	$72.8_{(+1.2)}$
MI-OL (Pang et al., 2020)	64.5	63.8	63.3	75.3	74.7	74.5
Ours + MI-OL (Pang et al., 2020)	$68.6_{(+4.1)}$	$68.9_{(+5.1)}$	$68.8_{(+5.5)}$	$72.4_{(-2.9)}$	$72.8_{(-2.9)}$	$72.9_{(-1.6)}$

The proposed method is applied at inference time. It can be applied to off-the-shelf models without retraining or finetuning them. The required gradients for generating defense perturbations can be computed efficiently in parallel using backpropagation. This would not increase too much time for inference.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We conduct experiments on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). ResNet-50 (He et al., 2016) is used as the network model. We validate the proposed method for models trained with two methods: Mixup (Zhang et al., 2018) and interpolated adversarial training (IAT) (Lamb et al., 2019). For fair performance comparison, we follow the same experimental setup as Pang et al. (2020) to train the models. The training procedure is performed for 200 epochs with a step size of 64. The learning rate is initialized to 0.1 and divided by a factor of 10 at epoch 100 and 150. The values used for interpolation are sampled from $Beta(1, 1)$ for both Mixup and IAT. The ratio between clean examples and adversarial examples used in IAT is set to 1:1. The untargeted PGD_{10} method is used to generate adversarial examples in IAT. The values of ϵ and step size are set to $8/255$ and $2/255$, respectively.

We experiment against both untargeted and targeted PGD attacks with different settings. For defense purposes, the one-step PDG method is used to generate perturbations to resist adversarial perturbations. Unless stated otherwise, defense perturbations are generated in a targeted fashion and the size of defense perturbations and number of randomly selected labels are set to $4/255$ under ℓ_∞ distance and 9, respectively. The value of β is set to 1. For each experiment, we run our model for three times and report the mean accuracy as the performance. The proposed method is implemented in Pytorch and all experiments are conducted on one GPU.

Baselines Three methods that were recently developed for inference time defense are used as baselines. These three methods are Xie et al.'s (2018), Guo et al.'s (2018) and MI-OL (mixup inference with nonpredicted labels) (Pang et al., 2020). We compare the performance our method and the baselines and show results of the joint use of our method and the baselines to resist adversarial examples.

Table 2: Classification accuray (%) on adversarial exmples crafted CIFAR-10. We report performance under PGD attacks with 10, 50 and 200 iterations. For IAT models with targeted attacks, defense perturbations are generated in an untargeted manner. The numbres in parentheses are performance improvements achieved by applying our method

Methods	Untargeted attacks			Targeted attcks		
	PGD_{10}	PGD_{50}	PGD_{200}	PGD_{10}	PGD_{50}	PGD_{200}
Mixup	5.5	5.3	5.2	≤ 1	≤ 1	≤ 1
Ours + Mixup	9.2	12.3	14.3	31.8	32.4	33.8
Xie's (2018) + Mixup	9.6	7.6	7.4	30.2	22.5	22.3
Ours + Xie's + Mixup	$19.3_{(+9.7)}$	$21.1_{(+13.5)}$	$23.4_{(+16.0)}$	$40.9_{(+10.7)}$	$40.7_{(+18.2)}$	$41.2_{(+18.9)}$
Guo's	13.1	10.8	10.5	33.3	26.3	26.1
Ours + Guo's	$28.3_{(+14.8)}$	$30.9_{(+20.1)}$	$32.9_{(+22.4)}$	$50.8_{(+17.5)}$	$50.9_{(+24.6)}$	$51.7_{(+25.6)}$
MI-OL Pang et al. (2020)	12.6	9.4	9.1	37.0	29.0	28.7
Ours + MI-OL Pang et al. (2020)	$19.0_{(+6.4)}$	$20.1_{(+10.6)}$	$21.9_{(+12.8)}$	$36.5_{(-0.5)}$	$35.9_{(+6.9)}$	$36.9_{(+8.2)}$
IAT	26.6	24.1	24.0	52.0	50.1	49.8
Ours + IAT	32.6	33.9	34.6	54.3	53.2	53.2
Xie et al. (2018) + Mixup	42.2	41.5	41.3	57.1	56.3	55.8
Ours + Xie's + Mixup	$46.8_{(+4.6)}$	$48.1_{(+6.6)}$	$48.4_{(+7.1)}$	$58.7_{(+1.6)}$	$57.5_{(+1.2)}$	$57.4_{(+1.6)}$
Guo's	36.2	33.7	33.3	53.8	52.4	52.2
Ours + Guo's	$40.9_{(+4.7)}$	$42.1_{(+8.4)}$	$42.2_{(+8.9)}$	$55.0_{(+1.2)}$	$53.6_{(+1.2)}$	$53.7_{(+1.5)}$
MI-OL Pang et al. (2020)	43.8	42.8	42.5	58.1	56.7	56.5
Ours + MI-OL Pang et al. (2020)	$45.4_{(+1.6)}$	$46.3_{(+3.5)}$	$46.8_{(+4.3)}$	$57.5_{(-0.6)}$	$56.7_{(+0.0)}$	$57.4_{(+0.9)}$

4.2 Experimental Results

We validate the proposed method against oblivious-box attacks (Carlini & Wagner, 2017). That is the adversary does not know about the existence of the defense mechanism, and adversarial example are generated only based on targeted network models. We evaluate the performance of defenses on the entire test set. Table 1 and Table 2 report the quantitative results on CIFAR-10 and CIFAR-100, respectively, demonstrating the effectiveness of the proposed method in improving defense performance. We see from Table 1 that the proposed method significantly helps to improve defense performance of the baseline methods against untageted attacks, achieving at least 12.5% and 4.1% performance gains for mixup and IAT trained models, respectively. For defending against targeted attacks, the proposed method performs well in combination with Xie et al.'s and Guo et al.'s for mixup trained models, and it performs well together with Xie et al.'s compared to with Guo et al.'s and MI-OL. It can be seen from Table 2 that, as with on CIFAR-10, the proposed method also helps improve defense performance against untargeted attacks on CIFAR-100, achieving at least 6.4% and 1.6% performance improvements for mixup and IAT trained models, respectively. For defending against targeted attacks, the use of our method is effective to improve defense performance when combined with Xie et al.'s and Guo et al.'s methods. We can also make the following three observations from the quantitative results.

1. In most cases, the proposed method improves defense performance of the baseline methods. Especially for untargeted attacks in different settings, our method helps to improve defense performance of the baselines. This shows that our method is complementary to the baselines to resist adversarial perturbations. Among the three baseline methods, the joint use of our method with Xie et al.'s and Guo et al.'s methods performs well compared to with the MI-OL method. This could be because the perturbation used to counteract adversarial perturbations is reduced due to the interpolation operation in MI-OL.

2. The proposed method performs well against strong PGD attacks that use more iterations. Previous studies show that adversarial perturbations that are generated using more iterations are difficult to resist. The results of the baselines also show that more iteration PGD attacks result in reduced performance. It is worth noting that the proposed method achieves improved performance for defending against most strong PDG attacks. And for the remaining attacks, the use of more iterations results in comparable performance as the use of less iterations. The results suggest that adversarial perturbations generated using more iterations can be easily counteracted by employing the use of local first-order gradient information.

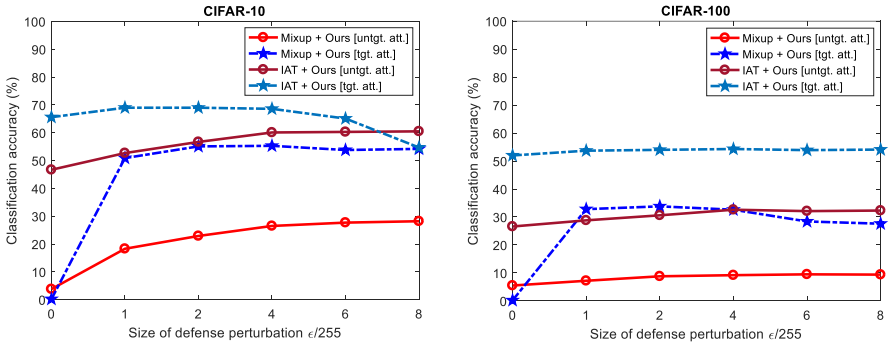

Figure 2: Impact of the size of defense perturbation on classification accuracy (%). We report performance of resisting against both untargeted and targeted PGD_{10} attacks.

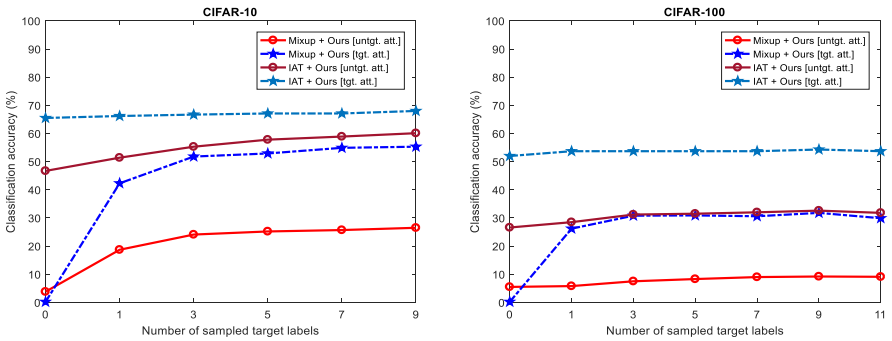

Figure 3: Impact of the number of randomly selected target labels used in our method on classification accuracy (%). We report performance of resisting both untargeted and targeted PGD_{10} attacks.

3. For defending against targeted PGD_{50} and PGD_{200} attacks on CIFAR-10, our method together with Guo et al.'s based on mixup trained models achieve higher performance that those obtained based on IAT trained models, improving the classification accuracy 1.4% and 3.2%, respectively. Overall, our method together with Guo et al.'s achieve better or comparable performance than pure interpolated adversarial training. As far as we know, we are the first to outperform adversarial training-obtained models using inference time methods. This shows that it is promising that adversarial training could be unnecessary if proper perturbations are applied to adversarial examples.

Next, we analyse the impact of the step size used in the one-step PGD method to generate defense perturbations on defense performance. We experiment on CIFAR-10 and CIFAR-100 resisting both untargeted and targeted PGD_{10} attacks. The experimental results are shown in Figure 1. We see that the step size affects differently for untargeted and targeted attacks. The performance improves as the step size increases from 1 to 8 for untargeted tasks on the two datasets. For targeted attacks, the performance improves as the step size increases from 1 to 4 but starts to reduce or maintain similar performance as the step size further increases.

We also analyse the impact of the number of selected class labels used in our method on defense performance. Figure 2 demonstrates the results of resisting untargetd and targeted PGD_{10} attacks on CIFAR-10 and CIFAR-100. We see that the performance improves for both untargeted and targeted attacks as the number increases from 1 to 9 on CIFAR-10. On CIFAR-100, the performance also improves as the number increases from 1 to 9 but begins to drop or remain similar as the number further increases.

Discussion on type of defense perturbations In our experiments, perturbations that we use to counteract adversarial perturbations are generated in a targeted manner other than for targeted attacks on IAT trained models on CIFAR-100, defense perturbations are generated in an untargeted manner. Overall, untargeted adversarial perturbations can be effectively counteracted using perturbations

Table 3: Classification accuracy (%). We report performance regarding targeted attacks and mixup trained models.

Methods	CIFAR-10			CIFAR-100		
	PGD_{10}	PGD_{50}	PGD_{200}	PGD_{10}	PGD_{50}	PGD_{200}
Guo et al.'s (2018)	57.8	49.1	48.9	33.3	26.3	26.1
Gaussian noise + Guo et al.'s	63.9	63.8	63.3	34.0	30.3	29.0
Random rotation + Guo et al.'s	63.8	63.6	63.2	40.8	36.4	35.1
Xie et al.'s + Guo et al.'s	59.1	58.3	58.2	41.1	35.8	33.7
MI-OL (Pang et al., 2020) + Guo et al.'s	62.9	62.6	65.1	37.6	34.0	35.7
Ours + Guo et al.'s	74.1	78.0	79.8	50.8	50.9	51.7

generated in a targeted manner by our method. The results also suggest that adversarial training has an unstable behavior for different data distributions.

Discussion on number of iterations defense perturbations We use the FGSM method to generate defense perturbations. This is the same as the use of one step PGD method. We experimented using PGD with multiple steps to generate perturbations. However, this lead to significantly reduced performance.

Table 4: Classification accuray (%) on clean examples.

Method	CIFAR-10		CIFAR-100	
	Mixup	IAT	Mixup	IAT
ResNet-50 (w/o defense)	3.6	3.2	≤ 1	≤ 1
Xie's				
Guo's				
MI-OL				
Ours (targted perturbation)	61.2	75.3	8.3	47.4
Ours (untargted perturbation)	87.1	88.3	66.0	65.1

We compare the performance of different method in combination with. We experiment using Guo et al.'s method as the base method and report perfomance regarding targeted PGD_{10} attacks. The results are shown in Table 3. We see that our method significantly improves the performance compared to Xie et al.'s, MI-OL, as well as random rotation and Gaussian noise.

We further evaluate our method on clean examples. Table 4 compares the performance of our method and the baselines. We see that the use of untargeted perturbation yields improved performance compared to the baselines.

5 Related Work

There have been a number of work on defending against adversarial perturbations at the inference time. In many transformation-based methods, the transformation is stochastic (Prakash et al., 2018; Zhang et al., 2018). Existing transformation defenses benefit from improved performance by taking the majority vote across samples of random transformations. Mixup inference is similar to our work. This method combines a test image with a number of clean training images to shrink the adversarial perturbation. In contrast to their work, we use designed perturbation .

Stochastic transformation-based defenses have shown considerable success in recovering from adversarial attacks. such that the transformed adversarial image would no longer be adversarial (Kou et al., 2019).

Xie et al. (2018) proposed a defense method that performs mitigate adversarial effects by adding a random resizing layer and a random padding layer. Pixel deflection (Prakash et al., 2018) corrupts an image by locally redistributing pixels. (Kou et al., 2019) train a separate compact distribution classifier to recognize patterns in the distributions of softmax probabilities of clean images. Transformation-based methods also lead to reduced performance on clean inputs.

though no model has yet succesfully done so while maintaining state of the art accuracy on clean inputs.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International Conference on Machine Learning*, 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*, pp. 3–14, 2017.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pp. 1632–1640, 2016.

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

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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*, pp. 1635–1646, 2019.

Connie Kou, Hwee Kuan Lee, Ee-Chien Chang, and Teck Khim Ng. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *International Conference on Learning Representations*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 95–103, 2019.

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

Tianyu Pang, Chao Du, and Jun Zhu. Max-mahalanobis linear discriminant analysis networks. *International Conference on Machine Learning*, 2018.

Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. *International Conference on Learning Representations*, 2020.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8571–8580, 2018.

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

Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 426–433. IEEE, 2016.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *International Conference on Learning Representations*, 2018.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *International Conference on Learning Representations*, 2020.