ADD-DEFENSE: TOWARDS DEFENDING WIDESPREAD ADVERSARIAL EXAMPLES VIA PERTURBATION-INVARIANT REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

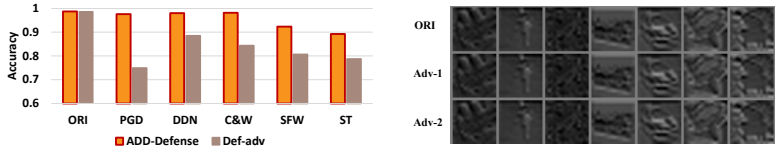
Due to vulnerability of machine learning algorithms under adversarial examples, it is challenging to defend against them. Recently, various defenses have been proposed to mitigate negative effects of adversarial examples generated from known attacks. However, these methods have obvious limitations against unknown attacks. Cognitive science investigates that the brain can recognize the same person with any expression by extracting invariant information on the face. Similarly, different adversarial examples share the invariant information retained from original examples. Motivated by this observation, we propose a defense framework ADD-Defense, which extracts the invariant information called *perturbation-invariant representation* (PIR) to defend against widespread adversarial examples. Specifically, realized by adversarial training with additional ability to utilize perturbation-specific information, the PIR is invariant to known attacks and has no perturbation-specific information. Facing the imbalance between widespread unknown attacks and limited known attacks, the PIR is expected to generalize well on unknown attacks via being matched to a Gaussian prior distribution. In this way, the PIR is invariant to both known and unknown attacks. Once the PIR is learned, we can generate an example without malicious perturbations as the output. We evaluate our ADD-Defense using various pixel-constrained and spatially-constrained attacks, especially BPDA and AutoAttack. The empirical results illustrate that our ADD-Defense is robust to widespread adversarial examples.

1 INTRODUCTION

Machine learning algorithms have made outstanding achievements in many fields, such as computer vision (He et al., 2016), natural language processing (Sutskever et al., 2014) and speech recognition (Hinton et al., 2012). However, many algorithms are fragile to the adversarial examples - inputs generated by adding imperceptible but malicious perturbation on original examples (Goodfellow et al., 2014). Many pixel-constrained and spatially-constrained perturbations have been proposed, such as PGD (Madry et al., 2017), DDN (Rony et al., 2019), ST (Xiao et al., 2018), BPDA (Athalye et al., 2018) and AutoAttack (Croce & Hein, 2020), which makes the machine learning algorithms output misleading predictions.

Various defenses have been proposed to protect machine learning algorithms (Goodfellow et al., 2014; Tramèr et al., 2017; Wu et al., 2019), which can mitigate the interference of adversarial examples generated from known attacks. However, in the real world, attackers can use multiple attacks and even create unknown attack mechanisms to defeat machine learning algorithms. Generally, there are more unknown attacks than known attacks. Note that in this paper, we call the attacks used to train defenses as known attacks, and other attacks as unknown attacks. Many defenses trained on adversarial examples generated from known attacks may have limitations in reaching stable performances when facing unknown attacks (Papernot et al., 2016; Tramèr et al., 2017). As shown in Fig 1(a), the accuracy of Def-adv fluctuates significantly on unknown untargeted attacks. Thus, how to effectively defend against multiple and even unknown attacks deserves to investigate deeply.

Cognitive science gives us some inspiration to solve this problem. Behavioural brain researches (Mishkin & Ungerleider, 1982; Kanwisher et al., 1997) show that the brain can recognize the same



(a) (b)

Figure 1: Performances of adversarial training (Def-adv) and our ADD-Defense on MNIST (a) and the illustration of the invariant information (b). The defenses are trained with adversarial examples generated from untargeted FGSM attack. "Adv" denotes adversarial examples.

person even if the person shows different or even unseen expressions, because the brain can extract invariant information on the face. Similarly, different adversarial examples share the invariant information retained from original examples, such as the semantic features shown in Fig 1(b). Motivated by the research, we propose a defense framework ADD-Defense, which extracts the shared invariant information called perturbation-invariant representation (PIR) to defend against widespread adversarial examples. Once such a representation is learned, given an adversarial example generated from an unknown attack, our defense can remove its unique perturbation and extract the invariant information, and further generate an examples without malicious perturbations from the representation. As shown in Fig 1(a), our ADD-Defense shows a more stable performance on widespread adversarial examples.

In order to defend the widespread adversarial examples, our PIR is supposed to be an unified representation and have no perturbation-specific information generated from both known attacks and unknown attacks. A unified representation can be obtained simply by minimizing the distance between different representations, but this mechanism cannot remove the perturbation-specific information. To solve this problem, we introduce a perturbation discriminator to disentangle and remove the underlying perturbation-specific information in the representation by adversarial training. In addition, the imbalance between widespread unknown attacks and limited known attacks has a negative impact on removing perturbation-specific information generated from unknown attacks. We reduce this impact by matching the learned PIR to a Gaussian prior distribution, so that our defense is expected to generalize well on unknown attacks. Moreover, we introduce a class classifier to ensure that the PIR retains the correct class-specific information. The classifier simply takes the PIR as the input and predicts the class label. Empirical results on pixel-constrained attacks (Fig 3) and spatially-constrained attacks (Fig 4), especially BPDA (Athalye et al., 2018) and AutoAttack (Croce & Hein, 2020) (Fig 5) illustrate that our ADD-Defense has robust performance to widespread adversarial examples.

This paper makes the following three contributions:

- We propose a defense framework ADD-Defense, which learns a *perturbation-invariant representation* (PIR) for defending against widespread adversarial examples based on limited known attacks.
- The PIR retains the invariant information and has no perturbation-specific information generated from *known and unknown* attacks, which is realized by a perturbation discriminator and a Gaussian prior.
- Our defense is consistently effective on defending malicious pixel-constrained and spatially-constrained perturbations. More importantly, it also has a great performance on advanced BPDA and benchmark AutoAttack.

The rest of the paper is organized as follows. We present the related defense strategies in Section 2 and introduce our defense in section 3. Then, empirical results based on various attacks and extensibility evaluations are presented in Section 4.

2 Related work

In this section, we mainly introduce related defense strategies and discuss their advantages and limitations. In addition, some attacks are stated in Appendix A.

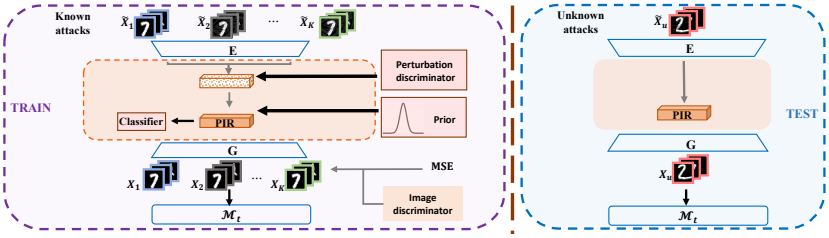


Figure 2: An overview of ADD-Defense. We remove perturbation-specific inforamtion generated from known attacks by a perturbation discriminator. Next, we make the representation generalize for unknown attacks by matching it with a Gaussian prior. In this way, we can obtain a perturbation-invariant representation (PIR). A classifier is exploited to preserve class-specific information. Input an adversarial example $\tilde{x}_u$ generated from an unknown attack, we can extract the PIR and utilize it to generate an example which have a correct prediction for the target model $\mathcal{M}_t$.

Augmenting training data: Adversarial training is a widely used strategy for defending malicious perturbation by augmenting the training data with adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014; Madry et al., 2017; Wu et al., 2019; Xie et al., 2020). This strategy often improves the robustness of the target model when the attacker chooses the same attack as that used to generate adversarial examples in the augmented dataset. However, the strategy usually does not perform as well when the attacker utilize other malicious perturbations and it tends to cause *gradient mask* (Papernot et al., 2016; Tramèr et al., 2017), which would make the robustness more inclined to white-box attacks than black-box attacks.

Modifying the target model: Defensive distillation (Papernot & McDaniel, 2017) uses a variant of distillation (Hinton et al., 2015) to train the target model in two rounds. It learns a smoother and reduce the gradient amplitude around the input point, which makes it difficult for attackers to generate adversarial examples. However, research shows that it fails to fully resist black-box attacks transferred from other networks. Randomized smoothing (Cohen et al., 2019) strategy use randomized smoothing to turn a classifier that classifies well under Gaussian noise into a new classifier which is robust to perturbations under the L^2. But it exhibits limited effectiveness against some spatially-constrained attacks (Wu et al., 2019), such as RP2 attack (Eykholt et al., 2018).

Adding additional mechanism: Adding an additional defense mechanism to the input of the target model is another strategy. A detection mechanism Lid (Ma et al., 2018) can detect adversarial examples, but it cannot remove malicious perturbations. Utilizing a generative network to learn a mapping strictly from adversarial examples to original examples such as APE-GAN (Shen et al., 2017) also can defend the target model effectively, but it has no mechanism to guarantee a stable performance against multiple or unknown attacks. Our defense can achieve this guarantee by learning a perturbation-invariant representation.

3 Our Approach

In this paper, the fundamental purpose of defense is to remove widespread malicious perturbations, the critical problem of which is learning a perturbation-invariant representation (PIR). Once such a representation is learned, our ADD-Defense can remove the malicious perturbation generated from both known and unknown attacks, and generate examples which have no malicious perturbations and can be classified correctly by the target machine learning algorithm.

Notation: Suppose there are K known attack for a target model $\mathcal{M}_t$, we denote by $\tilde{X}_k = [\tilde{x}_{k_1}, \ldots, \tilde{x}_{k_N}]^\top$ the adversarial examples generated from attack $k \in \{1, \ldots, K\}$ and by $X_k = [x_{k_1}, \ldots, x_{k_N}]^\top$ the restored examples without malicious perturbations, where k_N is the number of adversarial examples generated from the attack k. We call the normal examples from benchmark datasets as original examples. The original examples are denoted by $\tilde{X}_0 = [\tilde{x}_{0_1}, \ldots, \tilde{x}_{0_N}]^\top$ and their corresponding class labels are denoted by $Y = [y_1, \ldots, y_N]^\top$, where $y_i \in \mathbb{R}^{m \times 1}$ is an one-hot vector and m is the number of classes of original examples.

3.1 PERTURBATION-INVARIANT REPRESENTATION

The PIR is expected to have following properties. One is that the representation should be invariant to known attacks, that is, the representation should be unified and have no perturbation-specific information generated from known attacks. Another one is that the invariant representation should be generalized to unknown attacks. It is noteworthy that if the representations among known attacks are still different or have much perturbation-specific information, the generalization of the defense would have poor performance against adversarial examples generated from unknown attacks. The last one is that the representation should acquire class-specific information, which would be useful to have a correct prediction for an input with any malicious perturbation.

To learn a unified representation across known attacks, one can simply utilize a encoder to extract representations in latent space from inputs and apply a measure to align them among different attacks. We have encoder E take the examples $\tilde{X}$ as inputs and derive their representations Z, which then are aligned via the Maximum Mean Discrepancy (MMD) measure. The distance between multiple representations can be computed in reproducing kernel Hibert space $\mathcal{H}$ by a extended multi-domain MMD measure deduced in Appendix B. The objective function of distance between representations is formulated as:

$$\mathcal{L}_{mmd} = \frac{1}{K^2} \sum_{k,l=0}^{K-1} \mathrm{MMD}\left(Z_k, Z_l\right), \tag{1}$$

where $\mathrm{MMD}\left(Z_k, Z_l\right) = \|\mu_{P_k} - \mu_{P_l}\|_{\mathcal{H}}$ is the distance between two representation, P_i is the distribution of representation Z_i and μ_P is a mean map operation to map representations to the $\mathcal{H}$.

However, the aforementioned technical is not guaranteed to remove the perturbation-specific information from the unified representation, since the MMD measure aligns the representations simply without considering the perturbation-specific information. To address the problem, we extend above mechanism to remove the perturbation-specific information from the the representation. This is achieved by exploiting perturbation discrimination in the latent space to compose a adversarial network with the encoder E. More precisely, as shown in Fig 2, the introduced perturbation discriminator D_P only takes the representation as input and produces the perturbation label prediction. The D_P is trained with perturbation-specific label Y_k^p. As the adversary of D_P, the encoder E aims to confuse D_P from correctly predicting the perturbation label and make the prediction of D_P tend to perturbation-confused label Y_ζ^p, so that the perturbation-specific information in the unified representation can be removed and the representation is invariant for perturbations generated by different known attacks. The objective function of removing perturbation-specific information is defined as:

$$\mathcal{L}_p = \sum_{k=1}^{K} \left[Y_k^p \cdot \log \mathrm{soft}\left(D_P\left(E\left(\tilde{X}_k\right)\right)\right) - Y_\zeta^p \cdot \log \mathrm{soft}\left(D_P\left(E\left(\tilde{X}_k\right)\right)\right)\right], \tag{2}$$

where perturbation-specific label is implemented by an one-hot perturbation-specific vector $Y_k^p = [\xi_1, \ldots, \xi_K]^\top$, in which ξ_i is set to $\xi_i = 1$ if $i = k$, otherwise it is set to $\xi_i = 0$. The perturbation-confused label is implemented by $Y_\zeta^p = [\xi_1, \ldots, \xi_K]^\top$, in which ξ_i is set to $\xi_i = \frac{1}{K}$.

Our PIR is also suppose to be invariant to adversarial examples generated from unknown attacks. One problem we consider is the imbalance between adversarial examples generated from limited known attacks and those generated from widespread unknown attacks, which leads a risk that the above invariant representation may be overfitted to known attacks and thus the representation generalize poorly for unknown attacks. Inspired by the method of domain generalization (Li et al., 2018), we introduce a prior distribution to regularize the representation distribution P with the Jensen-Shannon Divergence (JSD). In the latent space, the adversarial examples are actually mapped to Gaussian distributions with different mean and variance, and representations are drawn from these distributions, we thus adopt a Gaussian distribution $\mathcal{N}(0, I)$ according to previous work (Larsen et al., 2016; Kingma & Welling, 2013) as the prior distribution. The objective function of generalization is defined as:

$$\mathcal{L}_g = JSD\left(P_1, \cdots, P_K\right) = \frac{1}{K} \sum_{k=1}^{K} KL\left(P_k \| \bar{P}\right), \tag{3}$$

where P_i is the distribution of input examples $\tilde{X}_i$. The JSD measure is the average of KL-divergences of each distribution from the average distribution $\bar{P}$, here, we set $\bar{P} = \mathcal{N}$. In addition, in order to make PIR preserve more class-specific information from input examples, a class classifier C in latent space is introduced to identify the class of the input example from the invariant representation. The objective function of classifier C is written as:

$$\mathcal{L}_c = -\sum_{k=1}^{K} Y \cdot \log \text{soft}\left(C\left(E\left(\tilde{X}_k\right)\right)\right), \tag{4}$$

where Y is the true class labels of the input examples. Our defense can learn the PIR by jointly optimization the three components, and the optimization problem for learning a PIR is expressed as:

$$\min_{E} \max_{D_P} \mathcal{L}_{mmd} + \lambda_1 \mathcal{L}_g + \lambda_2 \mathcal{L}_p + \lambda_3 \mathcal{L}_c. \tag{5}$$

3.2 Examples without malicious perturbations

Once the PIR is learned, we further utilize a generator G in our defense to preserve the recovery ability of our PIR and generate restored examples which have no malicious perturbation. Here, we take the original examples $\tilde{X}_0$ as the target examples to supervise the generator G with Mean Square Error (MSE) loss: $\mathcal{L}_{mse} = \sum_{k=1}^{K} \left\| X_k - \tilde{X}_0 \right\|_2^2$, where X_k denotes the restored examples generated from the PIR. Moreover, as noted in Zhao et al. (2017), the generator based on MSE measure tends to generate blurry examples, which would lead to misleading classification. To overcome the above limitation, we additionally introduce an image discriminator D_I in our defense to form a adversarial network with generator G. The discriminator D_I can improve the images quality of the restored examples X_k. We define the objective function for adversarial learning between image discriminator and generator G as:

$$\mathcal{L}_{adv} = \mathbb{E}_{\tilde{x}_0 \sim P_{\text{data}}(\tilde{x}_0)} \left[\log D_I\left(\tilde{x}_0\right)\right] - \mathbb{E}_{\tilde{x}_k \sim P_{\text{data}}(\tilde{x}_k)} \left[\log D_I\left(G\left(E\left(x_k\right)\right)\right)\right], \tag{6}$$

where $\tilde{x}_0 \in \cup_{k=1}^{K} \tilde{X}_0$ denotes the original example and $x_k \in \cup_{k=1}^{K} X_k$ denotes the restored example. The optimization problem for generating restored examples is expressed as:

$$\min_{G\ E} \max_{D_I} \mathcal{L}_{adv} + \theta \mathcal{L}_{mse}. \tag{7}$$

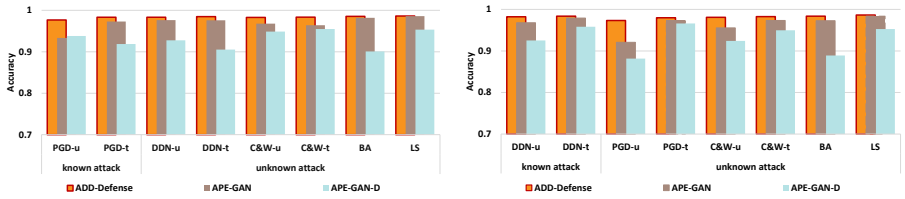
For the reconstruction of original examples, since the invariant information extracted from adversarial examples by our defense is retained from original examples, the PIR is expected to be as effective for original examples as adversarial examples. Moreover, the original examples can also be utilized to train our defense and the PIR extract the invariant information shared by the original examples and the adversarial examples.

4 Experiment

Experiments on malicious pixel-constrained and spatially-constrained perturbations are implemented in Section 4.1, 4.2. The evaluations against two advanced attacks: BPDA and benchmark AutoAttack are presented in Section 4.3. In addition, some additional evaluations have been made in Section 4.4 to indicate that our defense has great extensibility. Details of the adversarial examples, target models and the architecture of our defense are given in the Appendix C.

4.1 Defense against pixel-constrained perturbation

Pixel-constrained perturbations are generated by manipulating the pixel values directly on the whole example by leveraging the L^p distance for penalizing perturbations. The effectiveness of our defense against pixel-constrained perturbations is evaluated on four datasets: MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011). We select two attacks as known attacks to generate adversarial examples for training our defense together with the original examples. The other attacks are regarded as unknown attacks to generated adversarial examples for testing defenses.



(a) (b)

Figure 3: Performances of defenses on MNIST. The known attacks are the PGD attacks with $\epsilon = 0.15$ through 40 iterations (a) and DDN attacks (b).

Table 1: Performances on Fashion-MNIST (top), CIFAR-10 (middle) and SVHN (bottom).

Model	PGD_u	PGD_t	$\text{C\&W}_u$	$\text{C\&W}_t$	BA_u	LS_u	DDN_u	DDN_t
No defense	0	0.0648	0.0157	0.0567	0	0.058	0	0
APE-GAN	0.4790	0.7730	0.8763	0.8798	0.8507	0.8647	0.8757	0.8724
ADD-Defense	**0.8190**	**0.8652**	**0.8767**	**0.8808**	**0.8823**	**0.8843**	**0.8767**	**0.8804**
No attack: 0.9130 Rec-APE: 0.8730 Rec-ADD:0.8867								

Model	PGD_u	PGD_t	$\text{C\&W}_u$	$\text{C\&W}_t$	BA_u	LS_u	DDN_u	DDN_t
No defense	0	0	0	0.0374	0	0.12	0	0
APE-GAN	0.5217	0.6576	0.7247	0.7159	0.7267	0.6967	0.7313	0.7148
ADD-Defense	**0.6027**	**0.6623**	**0.7333**	**0.7210**	**0.7302**	**0.7260**	**0.7373**	**0.7253**
No attack: 0.9230 Rec-APE: 0.7527 Rec-ADD:0.7406								

Model	PGD_u	PGD_t	$\text{C\&W}_u$	$\text{C\&W}_t$	BA_u	LS_u	DDN_u	DDN_t
No defense	0.0187	0.0497	0	0	0	0.0087	0	0
APE-GAN	0.6582	0.8908	0.9186	0.9233	0.9212	0.8750	0.9198	0.9271
ADD-Defense	**0.8331**	**0.9186**	**0.9294**	**0.926**	**0.9308**	**0.8940**	**0.9308**	**0.9284**
No attack: 0.9600 Rec-APE: 0.9372 Rec-ADD:0.9374								

We present the distribution of adversarial examples in Fig 6 in Appendix D, our ADD-Defense has rectified the modification affected by malicious perturbations. We find that the advanced APE-GAN which also utilizes the generative model can effectively defend against unknown attacks as shown in Fig 3(a), compared with it, our defense has more stable accuracy. In order to explain that the improvement of the defense effect is not due to the deepening of network, we construct a deeper network APE-GAN-D based on APE-GAN. The results of APE-GAN-D illustrate that simply deepening the network my does not improve the stability of defense. The Fig 3(b) in Appendix C.1 presents the accuracy trained on DDN attacks. Fig 7(a) shows a comparison of the restored examples against the untargeted L^2-C&W attack.

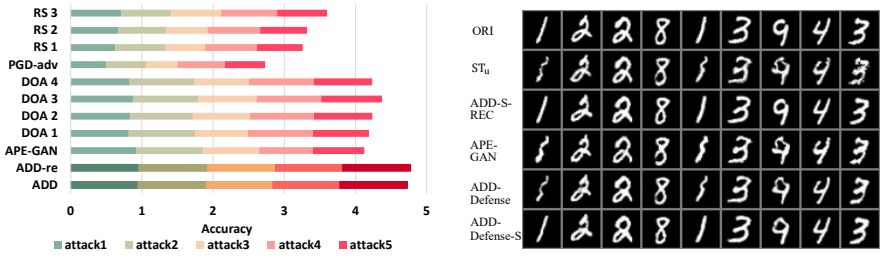
We also evaluate the accuracy on Fashion-MNIST (Table 1 (top)), CIFAR-10 (Table 1 (middle)) and SVHN (Table 1 (bottom)) datasets. The original examples and adversarial examples generated from L^∞-PGD attack are regarded as training examples. The Fig 7(b) shows the restored examples against the untargeted DDN attack. It can be seen that our defense have an effective and relatively stable performance. We further conduct experiments to understand the impact of the different components as shown in Fig 12 in Appendix D. We can find that using perturbation discriminator is helpful to remove malicious perturbation generated from widespread attacks, and using a prior distribution is helpful to learn a invariant representation from adversarial examples generated from unknown attacks. The details of experimental results in this section are shown in the Appendix D.

4.2 Defense against spatially-constrained perturbations

In addition to pixel-constrained perturbations, some attacks focuse on spatially-constrained perturbations, which mimic non-suspicious vandalism or art. The RP2 (Eykholt et al., 2018) attack generates malicious perturbations by greatly modifying pixel values in a limited space. We evaluate our defense compared with APE-GAN, randomized smoothing (RS) (Cohen et al., 2019), L^∞-PGD adversarial training and DOA adversarial training (Wu et al., 2019) on LISA dataset (Jensen et al., 2016) as shown in Fig 4(a). Our defense has effective and stable performance when facing the unknown attacks. We further retrain the target model based on the reconstructed examples generated from original examples and obtain a improved accuracy.

Table 2: Performances of defenses against spatially-constrained attacks. We retrain our defense with the adversarial examples generated from untargeted ST attack and targeted PGD attack, and obtain the ADD-Defense-S.

Model	No attack	ST_u	ST_u	SFW_u
Ori	0.9889	0.0025	0	0.3345
APE-GAN	0.9875	0.7954	0.7164	0.8934
ADD-Defense	**0.9867**	0.8930	0.8736	0.9274
ADD-Defense-S	0.9798	**0.9348**	**0.9206**	**0.9368**



(a) (b)

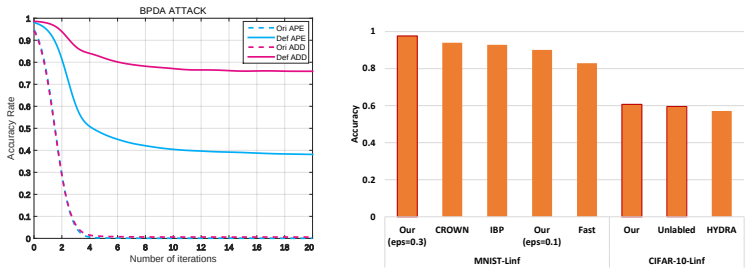
Figure 4: Performances of defenses against RP2 attack (a) and the restored examples against ST_u attack (b). The known attacks are attack 1 and attack2, which generate different adversarial patches. We also retrain the target model based on the reconstructed examples generated from our defense and get further improved results as shown in ADD-re. The "ADD-S-rec" denotes the reconstructed examples by the ADD-Defense-S.

In addition, we evaluate our defense on spatial-transformation attack ST (Xiao et al., 2018) and geometrical-transformation attack SFW (Wu et al., 2020) on MNIST dataset. As shown in Table 2, our defense trained on L^∞-PGD attack has a more effective and robust performance, but it dose not remove the spatially-constrained perturbations thoroughly. This may be due to the lack of adversarial examples generated from spatially-constrained attacks in our training process. Thus, we train our defense again based on adversarial examples generated from untargeted ST attack and targeted PGD attack, we name the retrained framework as ADD-Defense-S. It can be seen from Fig 4(b) and Table 2 that the shape of the digit is corrected on the sixth row, and the accuracy of the target model against spatially-constrained attacks is improved while our defense remains high accuracy 0.9634 and 0.9749 against untargeted and targeted PGD attack respectively. Moreover, we evaluate the influence of maximum perturbation value. Overall, within a appropriate range, such as when the maximum perturbation value is in the range of 0 to 0.2 on MNIST dataset, the defense trained on adversarial examples generated from the PGD attack with $\epsilon = 0.15$ can have a relatively stable performance. The detailed results in this section are shown in Appendix E.

4.3 Defense against BPDA and AutoAttack

The Backward Pass Differentiable Approximation (BPDA) attack is proposed to effectively attack the defenses which utilize a obfuscated gradient. The obfuscated gradient is a phenomenon exhibited by many defenses that makes standard gradient-based attacks fail to generate adversarial examples. The BPDA can also evaluate the dependence of defenses on the obfuscated gradient. The results as shown in Fig 5(a) indicate that our defense can resist BPDA effectively, which means that our defense has less dependence on the obfuscated gradient and is more secure. An advanced attack AutoAttack proposes two extensions of the PGD attack and combine them with two complementary existing attacks to form a parameter-free, computationally affordable and user-independent ensemble of attacks. The results of defenses[1] against L^∞-AutoAttack are shown in Fig 5(a). On MNIST dataset, our defense trained on original examples and adversarial examples generated from L^∞-PGD with $\epsilon = 0.15$ achieves great results. Considering that the adversarial examples based on $\epsilon = 0.3$ has far greater interference in visual perception than the adversarial examples based on $\epsilon = 0.15$, we train our defense again based on L^∞-PGD with $\epsilon = 0.3$ and it achieves a better performance.

[1] https://github.com/fra31/auto-attack



(a) (b)

Figure 5: Performances of defenses against BPDA on MNIST (a) and AutoAttack (b). For BPDA attack, the defenses are trained based on L^∞-PGD with $\epsilon = 0.15$. The dotted line represents the accuracy of the attacked target model and the solid line represents the accuracy of the target model using defenses. For AutoAttack, our defense is trained based on L^∞-PGD with $\epsilon = 0.15$ and 0.3 on MNIST and L^∞-PGD with $\epsilon = 0.03$ on CIFAR-10.

On CIFAR-10 dataset, the malicious perturbations are generated based on $\epsilon = 8/255$. Our defense trained on original examples and adversarial examples generated from L^∞-PGD with $\epsilon = 0.03$ also has great performance. The detailed results in this section are shown in Appendix F.

4.4 Extensibility evaluations

Besides the accuracy directly reflects the effectiveness in removing malicious perturbations, we also demonstrate that our ADD-Defense has good extensibility on the following evaluations:

Hardness inversion: Hardness inversion is a negative phenomenon that the robustness of a defense is higher for a strictly more powerful attack (Gilmer et al., 2018). Hardness inversion is initially used to evaluate the performance of a attack, it can also be taken to evaluate the robustness of a defense. We implement this evaluation on MNIST dataset as shown in Appendix G.1, and the results show that hardness inversion does not occur.

Local intrinsic dimensionality (Lid): Lid can be utilized to train a binary classifier for distinguishing the adversarial examples from original examples. In this way, we can evaluate our defense on hidden features rather than the accuracy. When we use it to distinguish the restored examples from original examples, the binary classifier get a low recall rate, which reflects that our defense can effectively eliminate the malicious perturbations. The detailed process are presented in Appendix G.2

Examples with non-malicious perturbation: Our defense focuses on removing malicious perturbations and get a perturbation-invariant representation in latent space rather than just learning a mapping from adversarial examples to original examples. In addition, an example with perturbation does not mean that it can defeat the target model. We call the perturbation that cannot attack the target model as the non-malicious perturbation. We design an exploratory experiment in Appendix G.3 and demonstrate that the malicious perturbations have been eliminated in our PIR, and examples restored from the representation can be classified correctly by the target model even if the restored examples have some perturbations.

5 Conclusion

In this paper, we propose a defense framework ADD-Defense for defending machine learning algorithms against widespread attacks, the crucial problem of which is learning a perturbation-invariant representation by jointly optimizing a encoder, a perturbation discriminator, generalization function and a class classifier. Once the PIR is learned, we can generate an example without malicious perturbations as the output. Experimental results demonstrate that our defense can effectively remove malicious perturbation generated from both known attacks and unknown attacks. One limitation of our defense is that even if the reconstructed examples maintain a good accuracy, but they have more misleading classification than the original examples. This limitation is elaborated in Appendix H and we will study how to reduce this negative impact in the future work.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420, 2018.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pp. 39–57. IEEE, 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, pp. 11192–11203, 2019.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. arXiv preprint arXiv:1902.02918, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. arXiv preprint arXiv:2003.01690, 2020.

Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623, 2019.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1625–1634, 2018.

Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. arXiv preprint arXiv:1807.06732, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. arXiv preprint arXiv:1810.12715, 2018.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In Advances in neural information processing systems, pp. 513–520, 2007.

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.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6):82–97, 2012.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

Morten Bornø Jensen, Mark Philip Philipsen, Andreas Møgelmose, Thomas Baltzer Moeslund, and Mohan Manubhai Trivedi. Vision for looking at traffic lights: Issues, survey, and perspectives. IEEE Transactions on Intelligent Transportation Systems, 17(7):1800–1815, 2016.

Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. Journal of neuroscience, 17(11):4302–4311, 1997.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

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

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In International conference on machine learning, pp. 1558–1566. PMLR, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409, 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. arXiv preprint arXiv:1801.02613, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Mortimer Mishkin and Leslie G Ungerleider. Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. Behavioural brain research, 6(1):57–77, 1982.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In International Conference on Machine Learning, pp. 10–18, 2013.

Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. arXiv preprint arXiv:1612.06299, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Nicolas Papernot and Patrick McDaniel. Extending defensive distillation. arXiv preprint arXiv:1705.05264, 2017.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. arXiv preprint arXiv:1611.03814, 2016.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. arXiv preprint arXiv:1707.04131, 2017.

Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4322–4330, 2019.

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. On pruning adversarially robust neural networks. arXiv preprint arXiv:2002.10509, 2020.

Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. arXiv preprint arXiv:1707.05474, 2017.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In International Conference on Algorithmic Learning Theory, pp. 13–31. Springer, 2007.

Chang Song, Hsin-Pai Cheng, Huanrui Yang, Sicheng Li, Chunpeng Wu, Qing Wu, Yiran Chen, and Hai Li. Mat: A multi-strength adversarial training method to mitigate adversarial attacks. In 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 476–481. IEEE, 2018.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pp. 3104–3112, 2014.

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, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994, 2020.

Kaiwen Wu, Allen Houze Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. arXiv preprint arXiv:2008.02883, 2020.

Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. arXiv preprint arXiv:1909.09552, 2019.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 819–828, 2020.

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. arXiv preprint arXiv:1906.06316, 2019.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. 2017.

Appendices

A RELATED WORK

Table 3: The summary of attacks: PGD (Madry et al., 2017), C&W (Carlini & Wagner, 2017), BA (Brendel et al., 2017), LS (Narodytska & Kasiviswanathan, 2016), DDN (Rony et al., 2019), RP2 (Eykholt et al., 2018), ST (Xiao et al., 2018) and SFW (Wu et al., 2020). The subscPIRt "*u*" indicates untargeted attack and the subscPIRt "*t*" indicates targeted attack.

Attack	Knowledge		Goal	Perturbation
PGD_u	white box	gradient-based	untargeted	pixel-constrained
PGD_t	white box	gradient-based	targeted	pixel-constrained
$C\&W_u$	white box	gradient-based	untargeted	pixel-constrained
$C\&W_t$	white box	gradient-based	targeted	pixel-constrained
BA_u	black-box	decision-based	untargeted	pixel-constrained
LS_u	black-box	score-based	untargeted	pixel-constrained
DDN_u	white box	gradient-based	untargeted	pixel-constrained
DDN_t	white box	gradient-based	targeted	pixel-constrained
$RP2_u$	white box	gradient-based	untargeted	spatially-constrained
ST_u	white box	score-based	untargeted	spatially-constrained
ST_t	white box	score-based	targeted	spatially-constrained
SFW_u	white box	gradient-based	untargeted	spatially-constrained

The adversarial examples as shown in Table 3 can be described in three ways: 1) *white-box and black-box attacks*: Considering the knowledge of the target model parameters, the white-box attacks can utilize the all knowledge when the black-box attacks can only query the results of the target models. 2) *targeted and untargeted attacks*: There is a significant distinction between situations where the objective is to induce the target model to produce a specific error versus those where any error suffices (Gilmer et al., 2018). Following Papernot et al. (2016), the former is referred as targeted attack and the latter is referred as untargeted attack. 3) *pixel-constrained and spatially-constrained perturbation*: many advanced attack models manipulate the pixel values directly on the whole example by leveraging the L^p distance for penalizing perturbations to generate adversarial examples, when some attackers focus on non-suspicious perturbation that mimic vandalism or art to reduce the likelihood of detection by a casual observer, such as spatially-constrained perturbations, which break the limitation of small L^p distance measures.

attacks usually aim to find a indistinguishable perturbation δ to generate a adversarial example $x_{adv} = x + \delta$ and the δ is expected to be small enough that the x_{adv} can remain undetectable. In addition, some attacks generate non-suspicious adversarial examples, which have no strict constraints on the perturbation value as long as it would appear to a human to be a real input (Gilmer et al., 2018). Here, we provide a brief description of six utilized attacks in experiments:

A.1 PROJECT GRADIENT DESCENT ATTACK (PGD)

Given an input x and its corresponding true label y, the perturbation δ is set to:

$$\delta = \varepsilon * \operatorname{sign}\left(\nabla_x J(x, y)\right). \tag{8}$$

The PGD attack (Madry et al., 2017) computes the perturbation iteratively and obtain the adversarial example:

$$\tilde{x}^{t+1} = \prod_{x+s} \left(\tilde{x}^t + \alpha \cdot \operatorname{sign}\left(\nabla_x J\left(\tilde{x}^t, y\right)\right)\right). \tag{9}$$

It determines the direction with the sign of the gradient to change the corresponding pixel value. In addition, the perturbation δ can be applied to targeted attack by modifying the cost function J and replacing label y with target label.

A.2 THE CARLINI-WAGNER ATTACK (C&W)

Carlini & Wagner (2017) propose three attacks for the L^0, L^2, L^∞ distance metric.In this paper, we just consider the L^2 attack and the objective is set to :

$$\begin{gathered}\min_\delta \left[\|\tilde{x} - x\|_2^2 + c \cdot f(\tilde{x})\right], \\ \text{where } f(\tilde{x}) = \max\left(\max_{i \neq t}\{Z(\tilde{x})_i\} - Z(\tilde{x})_t, -\kappa\right), \\ \text{and } \tilde{x} = \tfrac{1}{2}(\tanh(\operatorname{arctanh}(x) + \delta) + 1),\end{gathered} \tag{10}$$

where $Z(\tilde{x})_i$ is the logits corresponding to the $i-th$ class, κ is used to control the confidence of adversarial examples and c is a constant. The model is an effective optimization-based attack and it shows less perturbations than PGD.

A.3 DECOUPLING DIRECTION AND NORM ATTACK (DDN)

The DDN attack (Rony et al., 2019) is proposed to solve the difficulty of finding the appropriate constant c for the optimization in C&W attack. The norm is constrained by projecting the malicious perturbation δ on an $\epsilon - sphere$ around the original example x rather than impose a penalty on the L^2 during the optimization. Then, the L^2 is modified through a binary decision, and in the iterative process of training, if the example x_k is not adversarial at step k, the norm is increased for step $k+1$, otherwise decreased. The DDN attack has similar attack effect with C&W attack, but the former is faster than the latter.

A.4 BOUNDARY ATTACK (BA)

The BA model (Brendel et al., 2017) is a powerful decision-based attack that solely relies on the final decision of the model. The attack is initialized from a point that is already adversarial and then performs a random walk along the boundary between the adversarial and the the non-adversarial region such that it stays in the adversarial region and the distance towards the target example is reduced. The perturbations δ^k has a relation with the distance between the perturbed example towards the original input and needs to reduce the distance:

$$\begin{gathered}\left\|\delta^k\right\|_2 = \eta \cdot d\left(x, \tilde{x}^{k-1}\right), \\ \text{and } d\left(x, \tilde{x}^{k-1}\right) - d\left(x, \tilde{x}^{k-1} + \delta^k\right) = \epsilon \cdot d\left(x, \tilde{x}^{k-1}\right),\end{gathered} \tag{11}$$

where η and ϵ are related hyper parameters. The attack is simple and requires neither gradients nor probabilities, but it spends more time cost.

A.5 LOCAL SEARCH ATTACK (LS)

The LS attack (Narodytska & Kasiviswanathan, 2016) craft adversarial examples by carefully constructing a small set of pixels to perturb by using the idea of greedy local-search. The objective function which equals the probability assigned by the target model that the input example x belongs to class $c(x)$, is set to :

$$f_{c(x)}(x) = o_{c(x)}, \tag{12}$$

where o_j denotes the probability as determined by the target model that example x belongs to class j. In each round of iterations, it finds some pixel locations to perturb using the $f_{c(x)}(x)$ and then applies a transformation function to these selected pixels to construct a perturbed example. It terminates if it succeeds to push the true label below the $k-th$ place in the confidence score vector at any round. Otherwise, it proceeds to the next round.

A.6 ROBUST PHYSICAL PERTURBATIONS (RP2)

The RP2 attack (Eykholt et al., 2018) can attack the target physically by synthesizing non-suspicious adversarial patches, which belongs to spatially constrained perturbations. The objective function is set to:

$$\min_\delta \lambda\|\delta\|_p - J(f(x+\delta), y), \tag{13}$$

where λ is a hyperparameter that controls the regularzation of the distortion. In this paper, the utilized RP2 attack restricts the terms of the objective function to operate exclusively on masked pixels and the function is modified to:

$$\min_{\delta} \lambda \left\| M_x \cdot \delta \right\|_p + \sum_{i=1}^{k} J\left(f\left(x_i + M_x \cdot \delta \right), y^* \right), \tag{14}$$

where M_x is a perturbation mask matrix.

B DERIVATION OF MMD FOR MULTI-ATTACK

Theorem 1. (Muandet et al., 2013): $\bar{P}$ and P_k denote the probability across the K attacks domain and the probability of attack domain $k \in \{0, 1, \cdots K-1\}$ respectively, $\mu_{\bar{P}}$ and μ_{P_k} denote the mean embedding element across all attack domains and that for attack domain k. The distribution variance $\frac{1}{k}\sum_{k=0}^{K-1} \left\| \mu_{P_k} - \mu_{\bar{P}} \right\| = 0$ if and only if $P_0 = P_1 = \cdots = P_{K-1}$.

The MMD loss in RKHS is formulated as (Gretton et al., 2007) $\mathrm{MMD}\left(Z_k, Z_l\right) = \left\| \mu_{P_k} - \mu_{P_l} \right\|_{\mathcal{H}}$, where $\mu_P = \mathbb{E}_{Z\sim P}[\phi(Z)] = \mathbb{E}_{Z\sim P}[k(Z,\cdot)]$ is the mean embedding element of distribution P with a feature mapping $\phi(\cdot) : \mathbb{R}^d \to \mathcal{H}$ and a kernel function $k\left(\cdot,\cdot\right)$ (Smola et al., 2007). We derive the MMD for representation distribution from various attacks according to Theorem 1. The $\mu_{\bar{P}}$ can be equal to $\frac{1}{K}\sum_{l=0}^{K-1} \mu_{P_l}$ and the variance for multiple distributions is denoted as:

$$\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1} \left\| \mu_{P_k} - \mu_{\bar{P}} \right\|_{\mathcal{H}} &= \frac{1}{K}\sum_{k=0}^{K-1} \left\| \frac{1}{K}\sum_{l=0}^{K-1} \mu_{P_k} - \frac{1}{K}\sum_{l=0}^{K-1} \mu_{P_l} \right\|_{\mathcal{H}} \\
&= \frac{1}{K}\sum_{k=0}^{K-1} \left\| \frac{1}{K}\sum_{l=0}^{K-1} \left(\mu_{P_k} - \mu_{P_l} \right) \right\|_{\mathcal{H}} \leq \frac{1}{K^2}\sum_{k,l=0}^{K-1} \mathrm{MMD}\left(Z_k, Z_l\right).
\end{aligned}$$

We take the upper bound of the variance as the loss $\mathcal{L}_m md$ and Use an empirical estimation as a substitute for the mean embedding element: $\mu_{P_k} := \frac{1}{N}\sum_{n=1}^{N} \phi\left(z_k^n\right)$ (Gretton et al., 2007). The MMD measure is written as follows:

$$\begin{aligned}
\mathrm{MMD}\left(Z_k, Z_l\right) &= \left\| \frac{1}{N}\sum_{n=1}^{N} \phi\left(z_k^n\right) - \frac{1}{N}\sum_{m=1}^{M} \phi\left(z_l^m\right) \right\|_{\mathcal{H}} \\
&= \left[\frac{1}{N^2}\sum_{n,n'=1}^{N} k\left(z_l^n, z_l^{n'}\right) - \frac{2}{MN}\sum_{n,m=1}^{N,M} k\left(z_k^n, z_l^m\right) + \frac{1}{M^2}\sum_{m,m'=1}^{N} k\left(z_l^m, z_l^{m'}\right) \right]^{\frac{1}{2}},
\end{aligned}$$

where M and N denote the batch size of training data. The $k\left(\cdot,\cdot\right)$ is the radial basis function (RBF): $k\left(x, x'\right) = \exp\left(-\frac{1}{2\sigma^2}\left\| x - x' \right\|^2\right)$, where σ is the bandwidth parameter.

C EXPERIMENT PREPARATION

C.1 DATASETS AND ADVERSARIAL EXAMPLES

We use five image datasets in this experiment, we select 60000 the training images as a training set and 10000 the testing images as testing set in MNIST, Fashion-MNIST, CIFAR-10 and SVHN dataset. We select 3508 the training images as a training set and 1164 testing images as testing set in LISA. The untargeted and targeted L^∞-PGD, L^2-C&W, L^∞-DDN, ST and untargeted LS attacks are implemented by utilizing Advertorch Toolbox (Ding et al., 2019). The untargeted BA attack is implemented by Foolbox Library (Rauber et al., 2017). The specific parameters of each attack can be found in (Ding et al., 2019; Rauber et al., 2017). The untargeted RP2 and SFW can be implemented from the open source code [2] [3]. Here, we present the main parameters for each attack:

[2] https://github.com/tongwu2020/phattacks/tree/master/sign/experiment

[3] https://github.com/watml/fast-wasserstein-adversarial

C.1.1 PGD ATTACK

MNIST: eps (maximum distortion): 0.15, nb_iter (number of iterations): 40;

Fashion-MNIST: eps : 0.15, nb_iter : 40;

CIFAR-10: eps : 0.03, nb_iter : 40;

SVHN: eps : 0.025, nb_iter : 40,

C.1.2 C&W ATTACK

MNIST: $binary\_search\_steps$ (number of binary search times to find the optimum): 9, $max\_iterations$ (the maximum number of iterations): 300;

Fashion-MNIST: $binary\_search\_steps$: 9, $max\_iterations$: 200;

CIFAR-10: $binary\_search\_steps$: 9, $max\_iterations$: 50;

SVHN: $binary\_search\_steps$: 9, $max\_iterations$: 200,

C.1.3 DDN ATTACK

MNIST: nb_iter (number of iterations): 100, $gamma$ (factor to modify the norm at each iteration): 0.05;

Fashion-MNIST: nb_iter: 100, $gamma$: 0.05;

CIFAR-10: nb_iter: 100, $gamma$: 0.05;

SVHN: nb_iter: 200, $gamma$: 0.05,

C.1.4 LS ATTACK

MNIST: p (parameter controls pixel complexity): 10, r (perturbation value): 1.5, t (the number of pixels perturbed at each round): 300;

Fashion-MNIST: p: 10, r: 1.5, t: 300;

CIFAR-10: p: 10, r: 1.5, t: 15;

SVHN: p: 10, r: 1.5, t: 300,

C.1.5 BA ATTACK

MNIST: $steps$ (Maximum number of steps to run): 5000;

Fashion-MNIST: $steps$: 5000;

CIFAR-10: $steps$: 2000;

SVHN: $steps$: 5000,

C.1.6 ST ATTACK

$max_iterations$: (Maximum number of iterations): 5000, $search_steps$: (number of search times to find the optimum): 20,

C.1.7 SFW ATTACK

eps (maximum distortion): 0.15, nb_iter (number of iterations): 300.

C.2 TARGET MODELS

The target model on MNIST dataset is the LeNet network (LeCun et al., 1998) embedded in the Advertorch Toolbox, the target model on Fashion-MNIST dataset is composed of 3 convolutional

networks with maxpool layer and a fully-connected layer. [4]. A ResNet-110 network[5] (He et al., 2016) is utilized to classify on CIFAR-10 dataset. On SVHN dataset, a network composed of 4 convolutional layers with maxpool layer and dropout (0.3) is trained [6] to classify the real-world digits. The target model on LISA is composed of 3 convolutional layers and one fully-connected layer. [7].

The accuracy Acc_u against untargeted attack and the accuracy Acc_t against targeted attack are formulated as:

$$Acc_u = num_u / NUM,$$

$$Acc_t = (num_t - num_r) \,/\, (NUM - num_r)\,,$$

where num_u, num_t demote the the number of examples correctly classified, and the num_r represents the number of target items, for example, the goal of the targeted attack is to have all examples classified into class 3, and num_r is the number of examples which belong to class 3 in all correctly classified examples.

C.3 Architecture of our defense

The architecture of our defense is given in Table C.3. In what follows:

- Conv(m, k, s, p) refers to a convolutional layer with m feature maps, filter size $k \times k$, stride s and padding p,
- Deconv(m, k, s, p) refers to a convolutional layer with m feature maps, filter size $k \times k$, stride s and padding p,
- FC(m) refers to a fully-connected layer with m outputs,
- LeakyReLU refers to the leaky version of the Rectified Linear Unit.

Table 4: The architecture of our defense

E	G	D_P	D_I
Conv(128, 4, 2, 1)	Deconv(1024, 4, 2, 1)	FC(1024)	Conv(32, 4, 2, 1)
LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Conv(256, 4, 2, 1)	Deconv(512, 4, 2, 1)	FC(256)	Conv(64, 4, 2, 1)
LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Conv(512, 4, 2, 1)	Deconv(256, 4, 2, 1)	FC(64)	Conv(128, 4, 2, 1)
LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Conv(1024, 4, 2, 1)	Deconv(128, 4, 2, 1)	FC(3)	Conv(256, 4, 2, 1)
LeakyReLU	LeakyReLU		LeakyReLU
Conv(2048, 4, 2, 1)	Deconv(3, 4, 2, 1)		FC(512)
			LeakyReLU
			FC(1)

D Defense against pixel-constrained perturbation

Fig 6 shows the change of distributions on MNIST dataset by using t-distributed stochastic neighbor embedding (t-SNE) (Maaten & Hinton, 2008). The malicious perturbations generated from PGD attack modify the distribution of original examples, and our defense has largely rectified the modification. The positive parameters are set as $\lambda_1 = e^{-4}, \lambda_2 = e^{0}, \lambda_3 = e^{-3}, \theta = e^{3}$ on MNIST, Fashion-MNIST and SVHN, and $\lambda_1 = e^{-6}, \lambda_2 = e^{-1}, \lambda_3 = e^{-3}, \theta = e^{1}$ on CIFAR-10.

Fig 8(a) shows adversarial examples generated by various attacks. The results of different defense are presented in Fig 8(b), 8(c). Fig 9(a), Fig 10(a) and 11(a) shows adversarial examples generated by various attacks on Fashion-MNIST, CIFAR-10 and SVHN. The results of our ADD-Defense and APE-GAN are presented in Fig 9(b), 9(c), 10(b), 10(c), 11(b) and 11(c). Fig 7(c) shows the

[4] https://github.com/GunjanChhablani/CNN-with-FashionMNIST

[5] https://github.com/tongwu2020/phattacks/tree/master/cifar/ori200

[6] https://github.com/aaron-xichen/pytorch-playground

[7] https://github.com/tongwu2020/phattacks/tree/master/sign/experiment

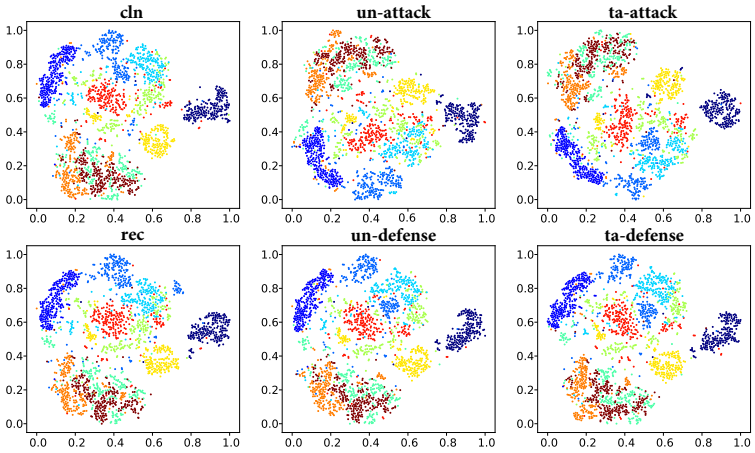
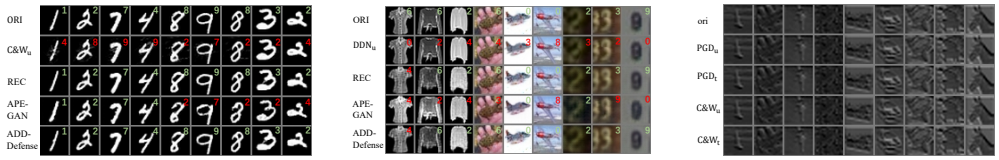


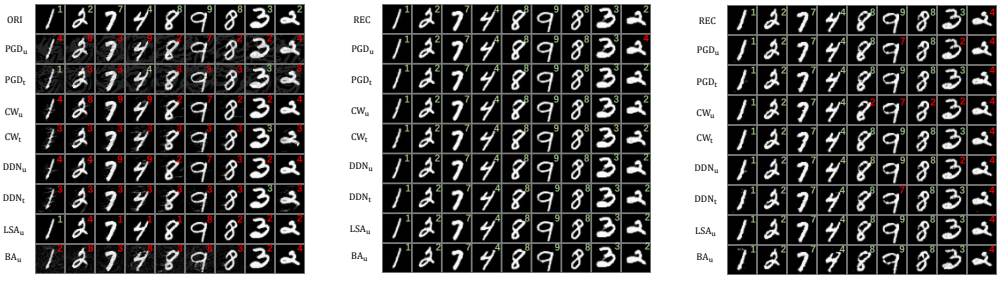
Figure 6: Illustration of examples distribution. “un” indicates untargeted attack and ”ta” indicates targeted attack. The top images are the distributions of original examples and adversarial examples. The bottom images are the reconstructed and restored examples by our defense. Different colors represent the different classes. It can be seen that our defense corrects the change of the distribution of adversarial examples.



(a) (b) (c)

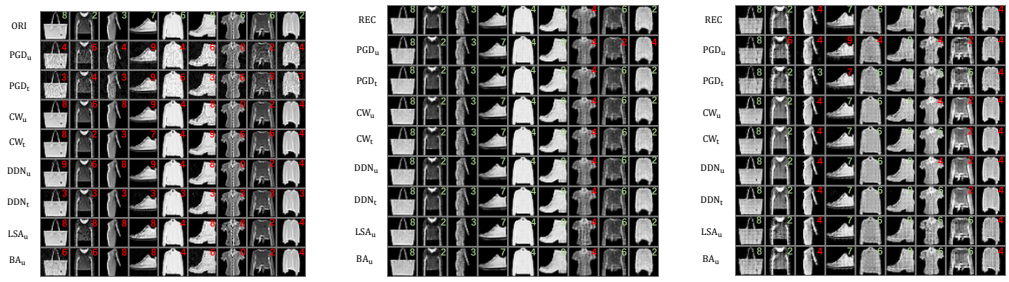
Figure 7: The illustration of restored examples on MNIST (a) and CIFAR-10 (b). The red superscript represents the wrong classification and the green superscript represents the correct classification. The representations with the size of 16 * 16 are taken from the 10-th channel of first activation layer of the encoder E (c).

output representation of the first activation layer of the encoder E on CIFAR-10 dataset. The input images are the original examples and adversarial example generated by untargeted and targeted PGD and C&W attack. It is difficult to distinguish the unique perturbation of each attack, and the representation of each attack are very similar, which means the specific perturbations of each attacks are eliminated . The Fig 12 shows the impact of different components on performance.



(a) (b) (c)

Figure 8: Adversarial examples generated by various attacks on MNIST dataset (a). The restored images of our defense (b) and APE-GAN (c) on MNIST dataset.



(a) (b) (c)

Figure 9: Adversarial examples generated by various attacks on Fashion-MNIST dataset (a). The restored images of our defense (b) and APE-GAN (c) on Fashion-MNIST dataset.

(a) (b) (c)

Figure 10: Adversarial examples generated by various attacks on CIFAR-10 dataset (a). The restored images of our defense (b) and APE-GAN (c) on CIFAR-10 dataset.

(a) (b) (c)

Figure 11: Adversarial examples generated by various attacks on SVHN dataset (a). The restored images of our defense (b) and APE-GAN(c) on SVHN dataset.

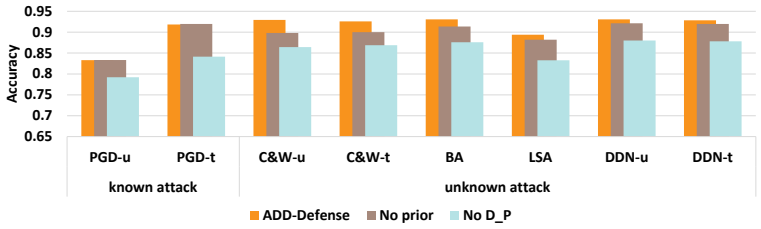
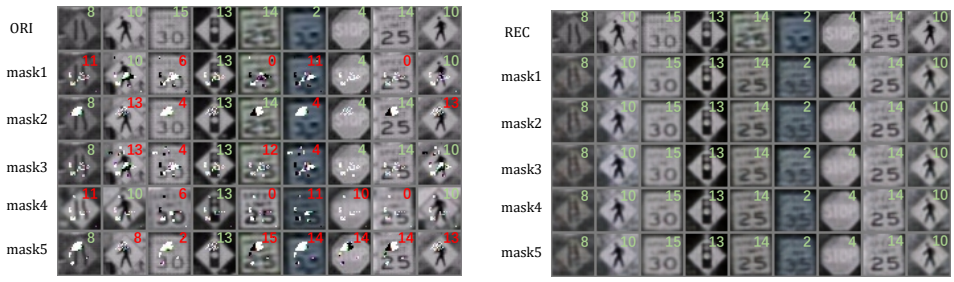


Figure 12: Performance against attacks on SVHN. The defense is removed some different components, such as the perturbation discriminator and the prior distribution.



(a) (b)

Figure 13: Adversarial examples generated with different adversarial patches (a) and the restored images of our defense (b) on LISA dataset.

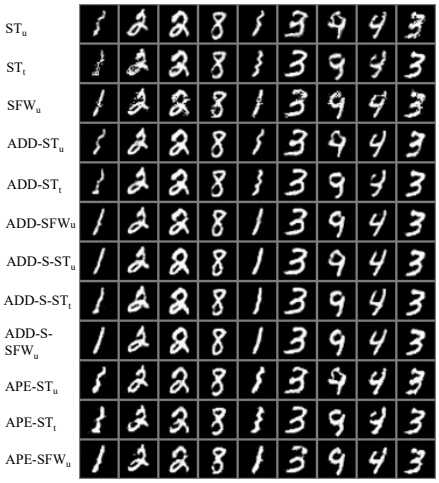


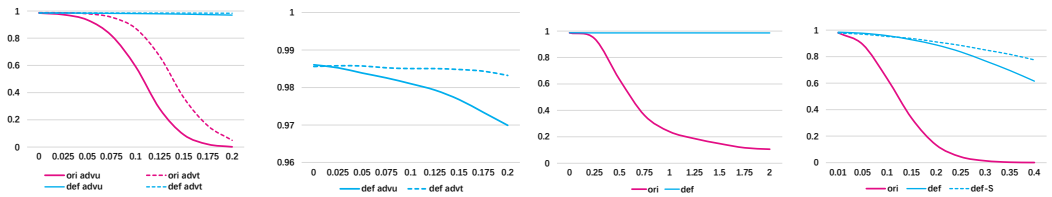
Figure 14: Adversarial examples generated from ST and FSW attack on MNIST on the top three rows. The restored examples of our defense ADD-Defense, improved framework ADD-Defense-S and APE-GAN are shown on the next rows.

E DEFENSE AGAINST SPATIALLY-CONSTRAINED PERTURBATIONS

Fig 13(a) shows adversarial examples generated by RP2 on LISA dataset and the results of our ADD-Defense are presented in Fig 13(b). The positive parameters are set as $\lambda_1 = e^{-5}, \lambda_2 = e^0, \lambda_3 = e^{-3}, \theta = e^3$. Five different masks are utilized to cause five different RP2 attacks. The three different sigma values in RS (Section 4.2) are 0.25, 0.5 and 1 respectively, the four strategies in DOA are (sticker size with 10 * 5, exhaustive search), (sticker size with $10 * 5$, gradient Based search), (sticker size with $7*7$, exhaustive search) and (sticker size with $7*7$, gradient Based search).

Fig 14 shows adversarial examples generated by ST attack and FSW attack on MNIST dataset, whose perturbation produces deformation. The positive parameters are set as $\lambda_1 = e^{-4}, \lambda_2 = e^0, \lambda_3 = e^{-3}, \theta = e^3$. The number of search times of ST attack is set to 20 and the perturbation of SFW$_u$ is set to 0.15. The results of our ADD-Defense, ADD-Defense-S and APE-GAN are also presented in Fig 14.

We evaluate the influence of maximum perturbation values on accuracy. Fig 15 shows that the increase of perturbation generated by white-box attack PGD and black-box attack LS$_u$ has little effect on accuracy, and the increase of perturbation generated by geometric attack SFW$_u$ has obvious negative effect on accuracy. However, with the increase of the perturbation value, adversarial examples become more unnatural and perceptible, and they even cause obvious classification errors in human perception, as shown in Fig 16.



(a) PGD (b) PGD (c) LS_u (d) SFW_u

Figure 15: Performance against attacks with various maximum perturbation values. The rose line represents the accuracy of the target model to the adversarial examples, and the cerulean line represents the accuracy after defense. Fig (b) is an enlarged view of the cerulean part in Fig (a)

Figure 16: Misclassified images with deception. The perturbation value of SFW_u is 0.40. The images in first, sixth, eighth and ninth columns are easily mistaken for the number "8, 9, 3, 9" in visual perception.

F Defense against BPDA and AutoAttack

The Backward Pass Differentiable Approximation (BPDA) attack (Athalye et al., 2018) is proposed to attack the defenses effectively which utilize obfuscated gradients. The adversarial examples and restored examples against BPDA attack with different maximum number of iterations are shown in Fig 17. The accuracy of defenses against AutoAttack is shown in Table 5.

Table 5: Performance of defenses against AutoAttack, including CROWN (Zhang et al., 2019), IBP (Gowal et al., 2018), Fast (Wong et al., 2020), Unlabled (Carmon et al., 2019) and HYDRA (Sehwag et al., 2020).

MNIST-Linf					CIFAR-10-Linf		
Our (eps=0.3)	CROWN	IBP	Our (eps=0.1)	Fast	Our	Unlabled	HYDRA
0.9765	0.9396	0.9283	0.9011	0.8293	0.6071	0.5953	0.5714

G Extensibility evaluations

G.1 Hardness inversion

Hardness inversion occurs when the reported robustness is higher for a strictly more powerful attacker (Gilmer et al., 2018). In general, we would expect defense methods to become less effective as the adversary has fewer limitations, since a more powerful adversary can always mimic a weaker one. One example of hardness inversion is when a paper reports higher robustness to white-box attacks than black-box attacks. Another example would be reporting higher accuracy against an untargeted attacker than a targeted attacker (Song et al., 2018). Although hardness inversion is initially used to evaluate attack performance, it can also be taken to evaluate the robustness of defense: the defense appears should be more robust to a more rigorous attacker (Gilmer et al., 2018). Fig 18(a) shows the radar map of the performance of our defense against targeted and untargeted attack on the MNIST data set, and Fig 18(b) shows the that against black-box and white-box attack. Slightly different from the Acc_t in Appendix C.2, in Fig 18(a), in order to compare the effect fairly, we remove all examples which belong to the target class for attackers, and the accuracy Acc_u and Acc_t

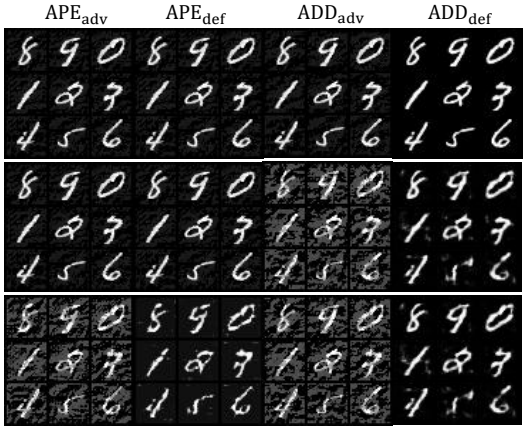


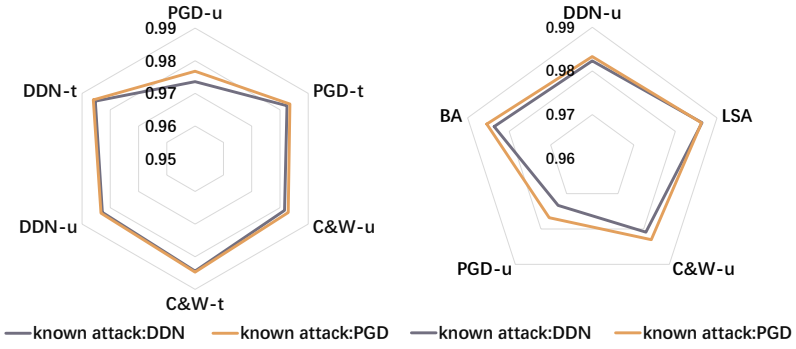
Figure 17: The images in BPDA experiment. The subscPIRt "adv" denotes the adversarial examples of BPDA for different defenses. The subscPIRt "def" denotes the restored images of defense. From the first row to the third row, the maximum number of iterations is 1,10 and 20 respectively.

are modified as:

$$Acc_u = (num_u - num_c) \,/\, (NUM - num_c)$$

$$Acc_t = (num_t - mum_c) \,/\, (NUM - num_c)$$

where num_c denotes the number of examples belong to target class for attackers. The results showed that the hardness inversion does not occur in the two evaluations



(a) target&untarget (b) white-box&black-box

Figure 18: Illustration of hardness inversion on MNIST dataset. "-u" indicates untargeted attack and "-t" indicates targeted attack. The two colored lines represent the results based on different known attacks respectively. Fig(a) shows the results against targeted and untargeted attacks and Fig(b) shows the results against white-box and black-box attacks. The accuracy against targeted attacks and black-box attacks is higher than that against untargeted attacks and white-box attacks.

G.2 Local intrinsic dimensionality (Lid)

Lid (Ma et al., 2018) can reveal the essential difference between normal examples and adversarial examples, and can be used to distinguish them. The negative examples is composed of the original examples and the noise examples, and the positive examples is composed of the adversarial examples. Both positive and negative examples account for half of the testing data. Recall rate can reflect the probability that the adversasrial examples are detected. We use Lid to evaluate the difference between the original examples and the restored examples by our defense. The binary classifier based on Lid can not effectively distinguish the restored examples from the original examples in view of the result that the recall rate of Lid for restored examples is close to 0, which reflects that our defense

can effectively eliminate the malicious perturbations. The Table 6 shows the ROC score, precision and recall of binary classification based on Lid.

Table 6: Results of binary classification based on Lid. The attack for training is L^2 − C&Wattack, the "adv" indicates the identification results of the binary classifier based on Lid for the adversarial examples. The "def" indicates the identification results of the binary classifier based on Lid for the restored examples

Dataset	Data	ROC-AUC	Precision	Recall
MNIST	adv	0.9901	0.9787	0.8378
	def	0.5775	0.4098	0.0126
CIFAR	adv	0.9859	0.9268	0.9398
	def	0.5874	0.5331	0.0848

G.3 EXAMPLES WITH NON-MALICIOUS PERTURBATION

The previous experiments are to eliminate perturbation and generate original example similar to original examples. Our defense focuses on removing perturbations in latent space rather than just generating original examples. Moreover, each adversarial example is calculated by the attacker on each single example, and an example with perturbation does not mean that it can interfere the target model. We call the perturbation that cannot attack the target model as non-malicious perturbation. We design an exploratory experiment on MNIST dataset: As an extreme case, we replace original examples with adversarial examples generated by untargeted C&W attack as target examples, and use the original examples and adversarial examples generated by targeted and untargeted C&W attacks as the training data. We then repeat this experiment using DDN attack. The Table 7 indicate that most of the generated examples have no malicious perturbation. This also illustrates that learning a mapping to the data distribution composed of adversarial examples which is generated for each specific original example separately, is not necessarily beneficial for attackers to generate effective malicious perturbation.

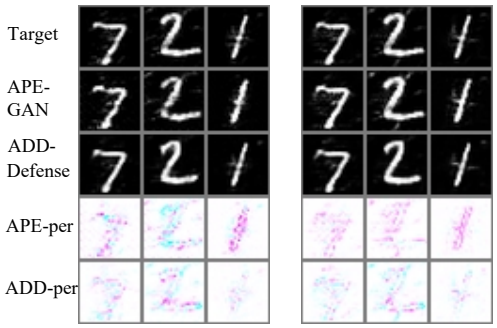


Figure 19: The examples with non-malicious perturbation. The target images in figure (left) are the adversarial examples generated by untargeted C&W attack. The target images in figure (right) are the adversarial examples generated by untargeted DDN attack. The images on fourth and fifth rows reflect the difference in pixels between the generated images and the target images. The blue pixels indicate that the pixel value in generated images is lower than that in the target images, and red pixels indicates that pixel value in generated images is higher than that in the target images.

Compared with APE-GAN, our results have higher accuracy. This may be due to the positive role of PIR in removing perturbation and extracting invariant information, or it may be that our defense is not as good as APE-GAN in learning the distribution of target examples with malicious perturbations, resulting in a bigger gap between the generated examples and the target examples. We notice that although the example generated by APE-GAN has more obvious perturbations in visual perception, they are not completely consistent with the perturbations in the target examples. Thus, we calculate the gap for each pixel in all examples. As shown in Fig 19, the gap between the examples generated by APE-GAN and the target examples is greater than our defense, which indicates that our

Table 7: Accuracy of the target model to the examples with non-malicious perturbations. The target images on first row is the adversarial examples generated by untargeted C&W attack. The accuracy of reconstructed images (”-rec”) and restored images (”-def”) against untargeted C&W attack is presented. Similarly, the accuracy against DDN attack is presented on the second row.

Model	APE-rec	APE-def	ADD-rec	ADD-def
C&W	0.9647	0.6808	**0.9699**	**0.9544**
DDN	0.9323	0.7425	**0.9739**	**0.9608**

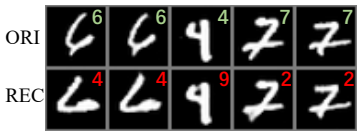


Figure 20: The misclassified reconstructed images on MNIST dataset. The images reconstructed by our defense are shown on the second row. In visual perception, the number reflected in the image is similar to the error class.

defense learns better mapping than APE-GAN and the design of eliminating malicious perturbation in potential space is effective and robust.

H Limitation and Future work

We notice that our defense has a slight negative effect on the target model for the original examples. At the same time, as shown in Fig 20, we find that many of the misclassified examples are visually deceptive and the slight deformation and blurring of the reconstructed images mislead the target model. We also find that some of the original examples which is misclassified by the target model are classified correctly after defense, so, the sensitivity of the target model to deformation and blurring also has a impact on accuracy. For future work, we will further study the effect of retraining the target model based on the reconstructed image, as tested in section 4.2, and further reduce the deformation and blur of the image generated by our defense.