IMPROVING ADVERSARIAL ROBUSTNESS VIA SELF-SUPERVISED CONFIDENCE-BASED PERTURBATIONS DENOISING

Yongkang Chen[1], *Ming Zhang*[1,*], *Xiaofeng Zhang*[2], *Wei Kong*[1], *Tong Wang*[1], *Xiaohui Kuang*[1]

[1]NKLSTISS, Institute of Systems Engineering, Academy of Military Sciences, China
[2]Shanghai Jiao Tong University, China

ABSTRACT

Deep neural networks (DNNs) are vulnerable to adversarial examples, and several solutions have been proposed to improve their robustness. Preprocessing methods can mitigate the impact of adversarial perturbations. However, such methods struggle to keep up with continuously evolving attacks. In this work, we propose *self-supervised confidence-based perturbations denoising* (SCPD), which uses a self-supervised adversarial training mechanism to remove adversarial perturbations and restore natural examples. Our method hinges on the premise that the association between natural and adversarial examples can help remove adversarial perturbations, and confidence can guide the generation of well-generalizing adversarial features. Specifically, we first maximize the distortions to the confidence of natural examples to craft adversarial examples, without ground-truth labels. Then, we train a denoising network to learn the association for projecting adversarial examples close to natural ones. Our experiments show that SCPD improves adversarial robustness, surpassing both adversarial training and preprocessing methods, particularly against unseen and adaptive attacks.

***Index Terms*—** Adversarial robustness, self-supervised, denoising, confidence

1. INTRODUCTION

Deep neural networks (DNNs) have achieved significant advancements in numerous domains including image recognition [1], natural language processing [2], and speech recognition [3]. Despite their success, DNNs exhibit a critical vulnerability to adversarial examples [4], where meticulously crafted, virtually imperceptible perturbations to an image can lead the network to produce erroneous predictions. This susceptibility presents substantial risks to a myriad of security-sensitive deep learning applications, underscoring the necessity to devise efficacious strategies to improve the robustness of DNNs against adversarial examples.

Numerous efforts [5, 6, 7] have been undertaken to defend neural networks against adversarial perturbations. Preprocessing defenses, for instance, can improve adversarial robustness and prove effective across a wide spectrum of models and tasks [8]. One such defense involves applying stochastic transformations to inputs before feeding to the model. However, the majority of stochastic transformations are flawed due to the insufficiency of randomness [9]. Other methods, such as JPEG and TVM, which employ conventional image processing [10], have been surpassed in white-box settings [11] and demonstrated limited effectiveness in black-box settings [12]. The primary cause of these failures is the isolated consideration of natural and adversarial examples, neglecting the association between them.

Establishing a association between natural and adversarial examples enables us to exploit the association to restore natural examples, thereby facilitating defense against adversarial examples. Prior methods have investigated the establishment of this association [13, 14, 15], aiming to eliminate adversarial perturbations and restore natural examples. However, these methods grapple with the issue of poor generalization of the crafted adversarial examples, which are either manually crafted or generated from intermediate features. This results in inadequate defense performance against unseen attacks.

In contrast, we propose the crafting of adversarial perturbations in the low-dimensional confidence space, dubbing such perturbations as *confidence-based perturbations* (**CP**). Specifically, we maximize the distortions to the confidence of natural examples without ground-truth labels to craft adversarial examples. Confidence can guide the generation of adversarial perturbations and thus provides valuable information for modeling the well-generalizing features of the adversarial perturbations [16]. These well-generalizing adversarial features can aid in defending against unseen attacks by establishing the association between natural and adversarial examples. To learn this association, we propose *self-supervised confidence-based perturbations denoising* (**SCPD**), a self-supervised adversarial training mechanism to eradicate adversarial perturbations and restore natural examples. Specifically, we first craft adversarial examples using CP. Subsequently, we train a denoising network as the association to project adversarial examples close to natural ones. We summarize our contributions as below.

❶ We propose self-supervised confidence-based perturba-

[*] Corresponding author.

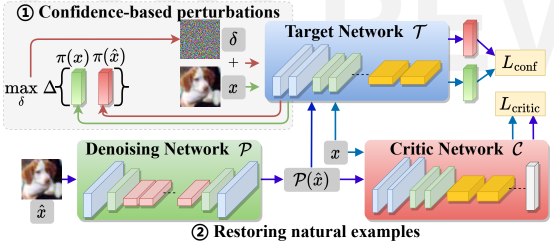

① Confidence-based perturbations
Target Network $\mathcal{T}$
δ
$+$
x
$\pi(x)\ \pi(\hat{x})$
$\max_{\delta} \Delta$
L_{conf}
L_{critic}
Denoising Network $\mathcal{P}$
$\hat{x}$
$\mathcal{P}(\hat{x})$
x
Critic Network $\mathcal{C}$
② Restoring natural examples

Fig. 1: A visual illustration of SCPD. Within self-regulated adversarial training mechanism, we initially craft adversarial examples by distort the confidence of natural examples, utilizing the CP method. Subsequently, we aim to reduce the distance between the output confidences of the natural and adversarial examples, which serves as a guiding principle for the training of the denoising network.

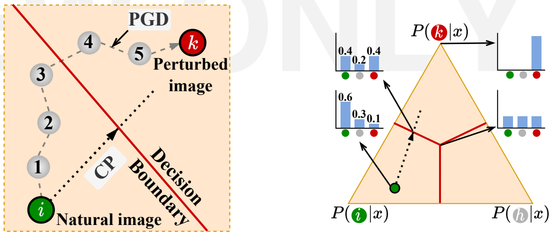
tions denoising (SCPD) to learn the association between natural and adversarial examples and restore natural ones.
❷ We introduce CP to craft adversarial examples in the confidence space. Furthermore, CP can be employed to improve previous methods for adversarial robustness.
❸ Experiments demonstrate that our method improves the robustness against unseen attacks and adaptive attacks when compared to the previous state-of-the-art method.

2. METHODOLOGY

The association between natural and adversarial examples is crucial for improving adversarial robustness. We design the SCPD, which uses self-supervised adversarial training to learn the association for removing adversarial perturbations. As shown in Figure 1, the training procedure can be viewed as optimizing a min-max problem similar to that in standard adversarial training. Given a natural example x, let $\pi(\cdot)$ represent the output confidence from the target model $\mathcal{T}$. We first maximally disrupt $\pi(x)$ to craft an adversarial example $\hat{x}$, where $\hat{x} = x + \delta$ and δ is the adversarial perturbation. Then, we train the denoising network $\mathcal{P}$ by minimizing the difference between $\pi\left(x\right)$ and $\pi\left(\hat{x}\right)$ to project adversarial examples close to natural ones. Note that the parameters of the target model are frozen in the training procedure. Additionally, the denoising network $\mathcal{P}$ and the critic network $\mathcal{C}$ share similarities with the generator and discriminator in a Generative Adversarial Network, respectively. In this paper, the difference is that the denoising network $\mathcal{P}$ is used for denoising rather than reconstruction, and the critic network $\mathcal{C}$ helps improve the texture details of the denoised images.

2.1. Confidence-based perturbations

Given a target network f_θ, its last layer is often a fully connected layer with a size equal to the number of labels, denoted by K, to output predicted confidence that is not normalized. For a given example x and corresponding label y, we have $z = f_\theta(x)$, where z_i denotes logit for class i. The predicted logit can be normalized into predicted confidence through the softmax function. The confidence for class i, denoted by π_i,



(a) (b)

Fig. 2: A visual illustration of our motivation. (a) Comparison between CP and PGD. After 4 iterations, PGD crosses the decision boundary, During this procedure, each instance needs to find the best direction to pass over the boundary again. In contrast, CP maintains a consistent objective once it starts. (b) We use a ternary diagram to represent all possible output confidence of a 3-class classifier. All data points within the region satisfy the condition of $P\left(i|x\right)+P\left(h|x\right)+P\left(k|x\right) = 1$. Different colors are used to represent different classes.

can be computed as follows:

$$\pi_i(x) = P\left(y = i|x\right) = \frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)} \tag{1}$$

It is easy note that Equation (1) obtains a valid confidence distribution, where $\pi_i > 0$ and $\sum_{i=1}^{K}\pi_i = 1$. Usually, the true confidence distribution is encoded using one-hot as a vector p, where $p_i = 1$ if $i = y$ and 0 otherwise. A common attack objective is to maximize the negative cross-entropy loss

$$\mathcal{L}(f_\theta(x), y) = -\sum_{i=1}^{K} p_i \log\pi_i = -z_y + \log(\sum_{i=1}^{K}\exp(z_i)) \tag{2}$$

to update input example, making the predicted confidence for the true class as small as possible. Note that in Equation (2) that ground-truth label is necessary as a anchor. As shown in Figure 2a, we visualize the process of using the PGD method to continuously perturb the natural example of class i and ultimately obtain adversarial example of class k. In each step, the adversary uses sign$(\nabla_x\mathcal{L}(f_\theta(\hat{x}), y))$ to search for the gradient direction that find an instance to violate the decision boundary.

Intuitively, we can expect that the direction of perturbations for each iteration does not depend on ground-truth label but can ultimately cross the decision boundary. We can use confidence to help annotate the anchor of adversarial attack without ground-truth labels. We first define the distance in confidence space to measure the distortions to the confidences of a natural example x as follows:

$$\Delta(\hat{x}, x) = d(\pi(\hat{x}), \pi(x))$$

where $d(\cdot)$ is the distance metric between the confidence of the original and perturbed examples. Note that we use mean square error (MSE) as the distance metric. Although adversarial perturbations may be imperceptible to the human visual system, they have a significant impact in the confidence space.

We propose crafting adversarial examples by solving the maximization problem as follow:

$$\max_{\|\delta\|_\infty \leq \epsilon} \Delta(\hat{x}, x) = d(\pi(\hat{x}), \pi(x))$$

where ϵ denotes the perturbation budget and $\hat{x} = x + \delta$. The distance in confidence space is evidently an effective measure, where the magnitude reflects the degree to which it deviates from the original example in confidence space. Moreover, the maximization process provides an anchor for adversarial attacks. Figure 2b summarizes the key concepts, with data point x for the true label i and k for the confusing class. In an extreme case, if $P(k|x) = 1$, the data point x is positioned in the confusing class. With this scenario in mind, when we maximize $\Delta(\hat{x}, x)$, the data point x will move in the direction furthest away from the starting point, implying the driving force of pushing the data point across the boundary without the need for ground-truth labels. Specifically, the data point x will move towards an automatically selected anchor where $P(k|x) = 1$. Note that once the attack begins, the anchor no longer changes.

2.2. Restoring natural examples

Here, we use a hybrid loss function to train the denoising network $\mathcal{P}$ that restores natural examples from adversarial examples. This loss term comprises confidence loss and critic loss, which we explain below.

Confidence loss. Adversarial examples crafted by the CP method directly influence the output confidence, leading to misleading predictions on the target network. Our method requires the denoising network to learn to minimize the distance between natural and adversarial examples in the confidence space. It should be noted that when using normalization on logits, the average distance between different classes may be on the same scale, leading to poor supervision in practice. Therefore, we use unnormalized confidence that contains rich intra-class and inter-class information, to aid in the restoration of image content. And the confidence loss can be defined as follows:

$$\mathcal{L}_{\text{conf}} = \Delta(\mathcal{T}(\mathcal{P}(\hat{x})), \mathcal{T}(x))$$

where $\mathcal{T}(\cdot)$ denotes the logits from the target network $\mathcal{T}$ and Δ denotes a distance metric which can be mean square error (MSE) or Kullback-Leibler (KL) divergence between the confidence scores of the natural and adversarial example.

Critic loss. Instead of using the standard GAN training objective, we used the RaHinge GAN method [17], which has better convergence performance. The RaHinge GAN loss is adopted as the basic adversarial loss for efficient training, given by:

$$\mathcal{L}_{\text{critic}} = -\log(\sigma(\mathcal{C}(\mathcal{P}(\hat{x})) - \mathcal{C}(x))) \tag{3}$$

Here, σ represents the sigmoid layer. It should be noted that Equation (3) includes both adversarial and natural examples, indicating that the denoising network $\mathcal{P}$ is trained using gradients from both types of examples during training.

Overall training objective. The overall loss objective for denoising network $\mathcal{P}$ is the combination of losses defined on confidence and critic loss:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{conf}} + \beta\mathcal{L}_{\text{critic}}$$

where α and β are positive parameters that trade off each component, with α being much greater than β. The confidence loss focuses on restoring image content and style, while the critic loss restores texture details.

3. EXPERIMENTS

Experimental setup. We assessed our defense using SVHN and CIFAR-10 datasets. Our denoising network employs a DUNET architecture, inspired by previous works [15, 13]. We forgo skip connections in the U-Net to prevent adversarial perturbations. Three target network architectures are used: VGG-19, ResNet-50, and WideResNet. The critic network is based on VGG, with five convolutional blocks and a fully connected layer. Target networks are optimized using SGD [18], with learning rates set to 0.01 (0.1 for WideResNet). Once trained, target model parameters are fixed. The denoising and critic networks use MSE as the distance metric, with $\alpha = 10^3$ and $\beta = 5 \times 10^-3$, and are optimized using Adam [19] with a learning rate of 10^{-3} and delay of 2×10^{-4}.

Defending against unseen types of attacks. First, We evaluate the effectiveness in the most basic and commonly used threat scenario, in which the adversary is unaware of the defense. We compare proposed method with state-of-the-art preprocessing and adversarial training methods. In Figure 3, the visual illustration demonstrates SCPD is effective in removing various adversarial perturbations while preserving texture similar to natural examples. Our method can restore natural examples with high confidence from adversarial examples that was originally misclassified with high confidence. Quantitative analysis in Table 1 indicates that our CP significantly improves upon previous methods, and our SCPD achieves even better robust performance under different attacks. For example, our method reduces the fooling rates of TI_N from 15.50% to 5.79% on the SVHN dataset compared to the previous state-of-the-art. Furthermore, the proposed method achieves the best results under strong AA_N on both the SVHN and CIFAR-10 datasets, which proves the effectiveness of the SCPD.

Defending against strong adaptive attack. We combine BPDA with PGD_N method to bypass preprocessing methods and evaluate our method in this challenging setting. Table 2 reports the robust accuracy against BPDA attack with different iterations on CIFAR-10. These results clearly demonstrates significant robustness gains of our method, with robust accuracy against 50 iterations of BPDA attack improved to 42.57% and 39.71% against 100 iterations of BPDA attack. Compared to previous state-of-the-art method, SCPD generally exhibits the best results, successfully defending against adaptive attacks. Additionally, for CP-HGD and CP-CAFD, the improved versions have better adversarial robustness against adaptive attacks compared to the original designs. We attribute the improvements to perturbation generation strategy instead of hand-crafted or weak ones.

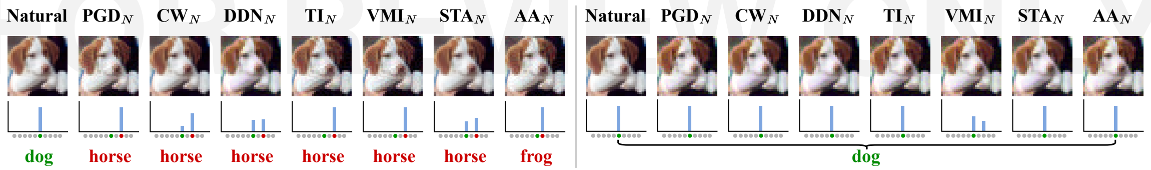


Fig. 3: Visualization of SCPD: One image is chosen, along with the corresponding adversarial and restored examples.

Table 1: Comparison in terms of natural error rate and fooling rate against unseen attacks.

Defense	Natural	Attacks								
		PGD$_N$[20]	PGD$_T$[20]	CW$_N$[21]	DDN$_N$[22]	TI$_N$ [12]	VMI$_N$ [23]	STA$_N$ [24]	STA$_T$ [24]	AA$_N$ [25]
SVHN										
JPEG[26]	**4.50**	95.49	77.87	21.53	9.86	79.56	90.44	36.34	47.02	97.53
TVM[10]	5.10	86.61	64.05	11.34	8.53	77.69	86.69	25.86	35.06	89.64
AT[27]	10.50	14.27	12.40	11.75	11.68	17.15	15.55	13.37	24.57	14.83
TRADES[28]	9.20	11.83	9.96	10.64	10.38	14.12	12.91	**11.95**	23.03	22.56
HGD[13]	6.72	11.07	17.99	16.72	17.14	22.01	17.24	22.70	33.77	19.43
CAFD[15]	7.50	7.22	6.00	10.02	8.47	15.50	11.50	16.44	27.44	6.33
CP-HGD	7.54	6.99	5.74	9.38	8.59	14.15	10.41	14.82	25.95	6.27
CP-CAFD	6.10	6.50	5.99	**7.84**	**7.18**	11.98	8.21	14.69	26.05	5.92
SCPD	6.30	**4.46**	**4.47**	8.80	7.87	**5.79**	**4.06**	15.32	**23.74**	**4.49**
CIFAR-10										
JPEG[26]	14.40	52.11	32.52	19.60	17.68	71.07	75.50	36.90	40.86	49.01
TVM[10]	13.10	87.69	66.03	19.99	15.08	80.30	89.12	44.48	52.34	84.38
AT[27]	21.4	22.56	22.08	21.65	21.63	23.73	23.33	24.82	28.19	22.76
TRADES[28]	19.20	20.41	19.74	19.42	19.31	**21.68**	**21.09**	22.63	26.02	20.41
HGD[13]	14.58	19.23	16.57	15.21	15.09	33.40	25.19	23.51	26.72	18.92
CAFD[15]	11.30	13.40	12.25	11.50	11.52	44.30	30.97	23.58	25.91	13.02
CP-HGD	10.50	16.77	12.03	10.96	10.90	52.39	39.04	22.16	24.28	14.60
CP-CAFD	10.70	13.23	11.50	10.54	10.80	39.91	28.18	22.59	24.20	12.56
SCPD	**10.00**	**11.22**	**10.28**	**9.94**	**10.03**	34.15	23.82	**21.32**	**22.90**	**10.63**

Table 2: Robust accuracy.

Defense	Hand-Crafted	Attacks	Natural	Robust
HGD[13]	✓	BPDA 50 BPDA 100	85.41	19.08 18.54
CAFD[15]	✗	BPDA 50 BPDA 100	89.69	33.22 30.36
CP-HGD	✗	BPDA 50 BPDA 100	89.51	26.50 23.52
CP-CAFD	✗	BPDA 50 BPDA 100	89.30	39.29 36.08
SCPD	✗	BPDA 50 BPDA 100	89.98	42.57 39.71

Table 3: Ablation study.

$\mathcal{L}_{\text{conf}}$	L_{critic}	PGD$_N$	DDN$_N$	TI$_N$	STA$_N$	AA$_N$
SVHN						
✓	✗	5.00	8.23	6.00	16.39	5.64
✗	✓	69.98	17.31	55.83	31.67	77.46
✓	✓	**4.46**	**7.87**	**5.79**	**15.32**	**4.49**
CIFAR-10						
✓	✗	12.02	10.10	35.14	21.64	11.00
✗	✓	90.94	47.04	77.81	66.27	91.71
✓	✓	**11.22**	**10.03**	**34.15**	**21.32**	**10.63**

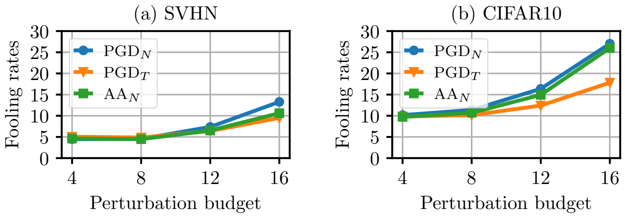


Fig. 4: Fooling rates (percentage) of our method against adversarial examples with different perturbation budget, where lower fooling rates indicate better performance. We evaluate our method against three strong attacks, with the ℓ_∞ norm perturbation budget set within the range of $(4/255, 16/255)$.

Ablation study. We validate the effectiveness of each component in the loss function. Specifically, we train proposed method using $\mathcal{L}_{\text{conf}}$ only, $\mathcal{L}_{\text{critic}}$ only and $\mathcal{L}_{\text{conf}}\&\mathcal{L}_{\text{critic}}$, respectively. The fooling rates with and without $\mathcal{L}_{\text{conf}}$ is reported in Table 3. The results show that when incorporating only the $\mathcal{L}_{\text{critic}}$ loss, SCPD's performance against unseen attacks is significantly impacted. When using only the $\mathcal{L}_{\text{conf}}$ loss, the adversarial robustness of the method under partial attacks slightly drops. This indicates that $\mathcal{L}_{\text{conf}}$ loss primarily contributes to improving the robustness of the method under various attacks and emphasizes the importance of this loss term for removing adversarial perturbations.

Robustness to different perturbation budget. To explore the robustness of the method to different perturbation budget, we set the ℓ_∞ norm perturbation budget within the range of (4/255, 16/255). Figure 4 shows that our method can maintain relatively low fooling rates when the budget is smaller. This indicates that our method is effective in defending against small perturbation attacks of less than 8/255. However, as the perturbation budget increases, the fooling rates of our method gradually rise, particularly when the budget exceeds 12/255, resulting in a significant drop in performance.

4. CONCLUSION

This work focuses on designing a preprocessing method for adversarial robustness against different types of unseen attacks and adaptive attacks. Based on the fact that the association between natural and adversarial examples can help remove adversarial perturbations and confidence can guide the generation of well-generalizing adversarial features. we propose a *self-supervised confidence-based perturbations denoising* method to remove adversarial perturbations and restore natural examples. Specifically, we first maximize the distortions of confidence of natural examples to craft adversarial examples and this process is independent of the ground-truth labels. Then, under self-supervised adversarial training mechanism, a denoising network learns to project adversarial examples close to natural ones. To guide the learning of the denoising network, we propose a loss function that minimizing the logits distance between the perturbed examples and natural examples. Experimental results demonstrate that our proposed method presents superior effectiveness against unseen threat models, such as different ℓ_p norms or perturbation budget, especially for adaptive attacks.

5. REFERENCES

[1] Xuehui Wang, Kai Zhao, Ruixin Zhang, Shouhong Ding, Yan Wang, and Wei Shen, "Contrastmask: Contrastive learning to segment every thing," in *Proc. of CVPR*, 2022.

[2] Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan, "Planning with large language models for code generation," in *Proc. of ICLR*, 2023.

[3] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Unsupervised speech recognition," in *Proc. of NeurIPS*, 2021.

[4] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *Proc. of ICLR*, 2014.

[5] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu, "Understanding robust overfitting of adversarial training and beyond," in *Proc. of ICML*, 2022.

[6] Gaurav Kumar Nayak, Ruchit Rawal, and Anirban Chakraborty, "De-crop: Data-efficient certified robustness for pretrained classifiers," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[7] Florian Tramèr, "Detecting adversarial examples is (nearly) as hard as classifying them," in *Proc. of ICML*, 2022.

[8] Changhao Shi, Chester Holtz, and Gal Mishne, "Online adversarial purification based on self-supervised learning," in *Proc. of ICLR*, 2021.

[9] Yue Gao, Ilia Shumailov, Kassem Fawaz, and Nicolas Papernot, "On the limitations of stochastic preprocessing defenses," in *Proc. of NeurIPS*, 2022.

[10] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten, "Countering adversarial images using input transformations," in *Proc. of ICLR*, 2018.

[11] Anish Athalye, Nicholas Carlini, and David A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. of ICML*, 2018.

[12] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. of CVPR*, 2019.

[13] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. of CVPR*, 2018.

[14] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli, "A self-supervised approach for adversarial robustness," in *Proc. of CVPR*, 2020.

[15] Dawei Zhou, Nannan Wang, Chunlei Peng, Xinbo Gao, Xiaoyu Wang, Jun Yu, and Tongliang Liu, "Removing adversarial noise in class activation feature space," in *Proc. of ICCV*, 2021.

[16] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry, "Adversarial examples are not bugs, they are features," in *Proc. of NeurIPS*, 2019.

[17] Alexia Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *Proc. of ICLR*, 2019.

[18] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. of ICML*, 2013.

[19] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. of ICLR*, 2018.

[21] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *Proc. of SP*, 2017.

[22] 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 *Proc. of CVPR*, 2019.

[23] Xiaosen Wang and Kun He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proc. of CVPR*, 2021.

[24] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song, "Spatially transformed adversarial examples," in *Proc. of ICLR*, 2018.

[25] Francesco Croce and Matthias Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. of ICML*, 2020.

[26] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *ArXiv preprint*, 2017.

[27] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of ICLR*, 2015.

[28] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. of ICML*, 2019.