Detecting Adversarial Data Using Perturbation Forgery

Supplementary Material

1. Assumption and Proofs

1.1. Assumption of Adv-noise Distribution

A common assumption for instance feature distribution is to model it as a Gaussian distribution [5]. Obtained by subtracting from clean data and adversarial data, adversarial noises naturally form a Gaussian distribution. Specifically, assuming the clean data follows a Gaussian distribution, the generation of adversarial samples can be represented as: $\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}, \mathbf{x} \sim \mathcal{N}$. Since adversarial data generated by an untargeted attack shares the same classes as the clean data and has a consistency with clean data on vision, it also follows a Gaussian distribution $\hat{\mathbf{x}} \sim \mathcal{N}$. Consequently, the adversarial noise forms a Gaussian distribution:$\boldsymbol{\eta} = \hat{\mathbf{x}} - \mathbf{x}, \boldsymbol{\eta} \sim \mathcal{N}$. For convenience of expression, we assume that noise r.v $\boldsymbol{\eta}$ satisfied truncated Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}_d)$ with it's truncated interval is $[-\epsilon, \epsilon]^d$.

1.2. Theorem Proofs

Theorem 1. *Let $\mathcal{P}_a$ be the distribution set composed of all the adv-noise distributions. Given independent noise distributions $Q_i, i \in N^+$. For $\forall\ i \neq j$, Q_i and Q_j are proximal noise distributions if the following conditions are met.*

1) if $Q_i, Q_j \in \mathcal{P}_a$.

2) if $\exists\ \epsilon_{\boldsymbol{\mu}}, \epsilon_\sigma > 0$ s.t. $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| < \epsilon_{\boldsymbol{\mu}}$ and $|\sigma_i - \sigma_j| < \epsilon_\sigma$ where the $\|\cdot\|$ is a Euclidean norm on $\mathbb{R}^n$.

Proof. Denoting the distribution of natural sample $\mathbf{x}$ as P, we use the 1-Wasserstein distance as the metric $d(\cdot,\cdot)$, to prove the theorem.

1) We denote the independent adv-noise distribution of η_i as $Q_i \in \mathcal{P}_a$ where $\eta_i = \hat{\mathbf{x}}_i - \mathbf{x}$, and a zero-distribution as P where its mean and variance are zero vectors. Then the 1-Wasserstein distance between η_i and η_j can be bound by its metric property and dual form:

$$\begin{aligned} W_1(Q_i, Q_j) \leq & W_1(Q_i, P) + W_1(Q_j, P) \\ \leq & \sup_{\|g\|_L \leq 1} \mathbb{E}_{\boldsymbol{\eta}_i \sim Q_i}[g(\boldsymbol{\eta}_i)] - \mathbb{E}_{\mathbf{x}\sim P}[g(\mathbf{x})] \\ & + \sup_{\|g\|_L \leq 1} \mathbb{E}_{\boldsymbol{\eta}_j \sim Q_j}[g(\boldsymbol{\eta}_j)] - \mathbb{E}_{\mathbf{x}\sim P}[g(\mathbf{x})] \\ \leq & 2\epsilon, \end{aligned}$$

where $\|g\|_L$ is the Lipschitz constant of function g w.r.t. the norm induced by the metric here.

2) Consider two random noise $\boldsymbol{\eta}_i \sim \mathcal{N}_i(\boldsymbol{\mu}_i, \sigma_i^2\mathbf{I}_d)$ and $\boldsymbol{\eta}_j \sim \mathcal{N}_j(\boldsymbol{\mu}_j, \sigma_j^2\mathbf{I}_d)$. While it's mean and variance satisfied that $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \leq \epsilon$ and $|\sigma_i - \sigma_j| \leq \epsilon$. then the 2-Wasserstein distance between η_1 and η_2 is

$$\begin{aligned} W_2\left(\boldsymbol{\eta}_i, \boldsymbol{\eta}_j\right)^2 = & \left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2 + \operatorname{trace}(\sigma_i^2\mathbf{I}_d + \sigma_j^2\mathbf{I}_d \\ & - 2\left((\sigma_j^2\mathbf{I}_d)^{1/2}\sigma_i^2\mathbf{I}_d(\sigma_j^2\mathbf{I}_d)^{1/2}\right)^{1/2}) \\ \leq & \epsilon_{\boldsymbol{\mu}}^2 + d \cdot \epsilon_\sigma^2, \end{aligned}$$

where $(\sigma_i^2\mathbf{I}_d)^{1/2}$ denotes the principal square root of $\sigma_i^2\mathbf{I}_d$. □

Corollary 1. *1) All noise distributions proximal to the known adversarial distribution $p(\boldsymbol{\eta})$ form an open ε-ball centered on $p(\boldsymbol{\eta})$, denoting as $\mathcal{P}_\varepsilon = \{p(\boldsymbol{\eta}_i) | d(p(\boldsymbol{\eta}_i), p(\boldsymbol{\eta})) < \varepsilon)\}$, and also form a metric space $(\mathcal{P}_\varepsilon, d)$.*

2) The adv-noise distribution set is also located in this ball: $\mathcal{P}_a \subset \mathcal{P}_\varepsilon$.

3) Based on the properties of the complete metric space $(\mathcal{P}_\varepsilon, d)$, for all subset $\mathcal{P}_i \subset \mathcal{P}_\varepsilon$, there exists a finite open covering of the subset: $\mathcal{P}_i \subset \{G_c\}_{c\in\mathbb{I}}$.

Proof. 1) Consider 1-Wasserstein distance as metric d, for $\forall Q_i, Q_j \in \mathcal{P}_\varepsilon$, we have

$$W_1(Q_i) \geq 0. \quad W_1(Q_i, Q_j) = 0\ if\ and\ only\ if\ Q_i = Q_j. \tag{1}$$

$$W_1(Q_i, Q_j) = W_1(Q_j, Q_i). \tag{2}$$

$$W_1(Q_i, Q_j) \leq W_1(Q_i, P) + W_1(Q_j, P). \tag{3}$$

Therefore, $(\mathcal{P}_\varepsilon, d)$ is a metric space.

2) It is obvious according to the definition of $\mathcal{P}_\varepsilon$.

3) For any subset $\mathcal{P}_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\} \subset \mathcal{P}_\varepsilon$, we can construct several spherical subsets $\{\mathcal{P}_{ij}\}$ with p_{i1} as the center and $|\varepsilon - d(p_{i1}, p(\boldsymbol{\eta}))|$ as the radius. $\{\mathcal{P}_{ij}\}$ is a open covering of $\mathcal{P}_i$.

□

2. Experimental Details

2.1. Dataset

CIFAR10 CIFAR10 consists of a training set of 50,000 images and a test set of 10,000 images, each with a resolution of 32x32 pixels. These images are divided into 10 different classes. The classes represent everyday objects such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

Table 1. Detection AUROC against various gradient-based attacks on CIFAR-10 ($\epsilon = 4/255$).

Detector	BIM	PGD	RFGSM	DIM	MIM	NIM	VNIM	SNIM	AA
SPAD [20]	0.9951	0.9938	0.9935	0.9952	0.9966	0.9985	0.9947	0.9963	0.9974
EPSAD [26]	0.9999	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
ours	0.9965	0.9965	0.9968	0.9947	0.9980	0.9980	0.9976	0.9964	0.9988

Table 2. Comparison of AUROC scores of detecting gradient-based attacks on ImageNet100 ($\epsilon = 4/255$).

Detector	BIM	PGD	RFGSM	DIM	MIM	NIM	VNIM	SNIM	AA
LID [14]	0.9782	0.9750	0.9637	0.8942	0.9146	0.8977	0.8865	0.8752	0.9124
LiBRe [3]	0.9259	0.9548	0.9769	0.9243	0.8725	0.9013	0.8862	0.8694	0.8653
SPAD [20]	0.9846	0.9851	0.9833	0.9815	0.9820	0.9823	0.9891	0.9877	0.9890
EPSAD [26]	**0.9998**	**0.9989**	**0.9992**	**0.9923**	0.9918	**0.9972**	**0.9946**	0.9908	**0.9998**
ours	0.9911	0.9912	0.9911	0.9863	**0.9931**	0.9934	0.9878	**0.9914**	0.9941

ImageNet100 ImageNet-100 is a subset of ImageNet [1], comprising 100 classes selected from the original ImageNet dataset, each containing a substantial number of high-resolution images, with 1,300 images per class.

Makeup Makeup comprises images of faces with and without makeup. These images are collected from various sources to ensure a diverse representation of facial features, makeup styles, and skin tones. There are 333 before-makeup images and 302 after-makeup images.

CelebA-HQ CelebA-HQ dataset is a high-quality version of the original CelebA dataset.CelebA-HQ consists of 30,000 high-resolution images of celebrity faces, derived from the CelebA dataset through a progressive GAN-based upsampling and quality enhancement process. Each image in CelebA-HQ is 1024x1024 pixels.

LFW LFW contains 13,233 labeled images of 5,749 different individuals collected from the internet, each image is labeled with the name of the person pictured. LFW images vary widely in terms of lighting, facial expression, pose, and background, closely reflecting real-world conditions. The images are provided in a resolution of 250x250 pixels, and the faces are roughly aligned based on the eye coordinates.

2.2. Adversarial Attacks

Gradient-based attacks We use Torchattacks [9] to implement adversarial attacks. The victim models used for adversarial data generation are ResNet-50 [6] for ImageNet, WideReseNet-28 for CIFAR10 [10], and ArcFace [2] for face attacks.

Generative-model-based attacks For CDA, we use the pre-trained generator which has been trained on ImageNet with a relativistic loss against ResNet152. For TTP, we use the pre-trained-generators trained against ResNet50 for 8 target labels, they are: Grey Owl(24), Goose(99), French Bulldog(245), Hippopotamus(344), Cannon(471), Fire Engine(555), Parachute(701), Snowmobile(802). We generate 1000 adversarial examples for each label and randomly select 125 adversarial examples for each label. For M3D, its settings are consistent with TTP. For Diff-PGD, we use the global attack to craft adversarial examples, target classifier is ResNet50, the diffusion model accelerator is ddim50, the reverse step in SDEdit is 3, and the iteration number of PGD is 10, the step size is 2. For Diff-attack, the target classifier is ResNet50, DDIM sample steps are set as 20, and iterations to optimize the adversarial image are set as 30. For the methods mentioned above, we all use the authors' implementation to craft adversarial examples.

Face attacks For face attacks, 1 gradient-based attack TIPIM [24], two GAN-based attacks including Adv-Makeup [25] and AMT-GAN [8], and 3 physical attacks including Adv-Sticker [18], Adv-Glasses [16], and Adv-Mask [18] produce face adversarial samples by attacking the victim face model ArcFace [2].

2.3. Baselines

Adversarial detection For LiBRe[1], we trained a ResNet-50 on ImageNet100 and finetuned it using the authors' implementation, the parameters were kept consistent with theirs. For EPSAD[2], we use the pre-trained models provided by the authors and follow the instructions by the authors to get the detection results. Due to the time-consuming nature of computing EPS, we evaluated 1,000 images for each attack method on ImageNet-100.

[1]https://github.com/thudzj/ScalableBDL/tree/efficient/exps
[2]https://github.com/ZSHsh98/EPS-AD

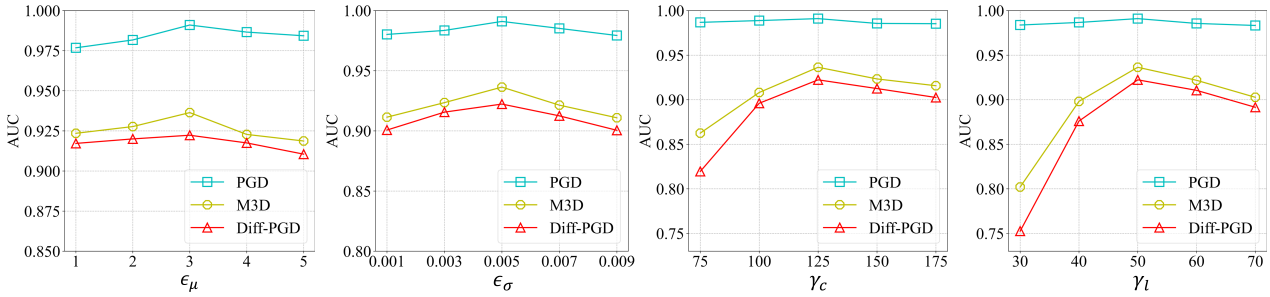
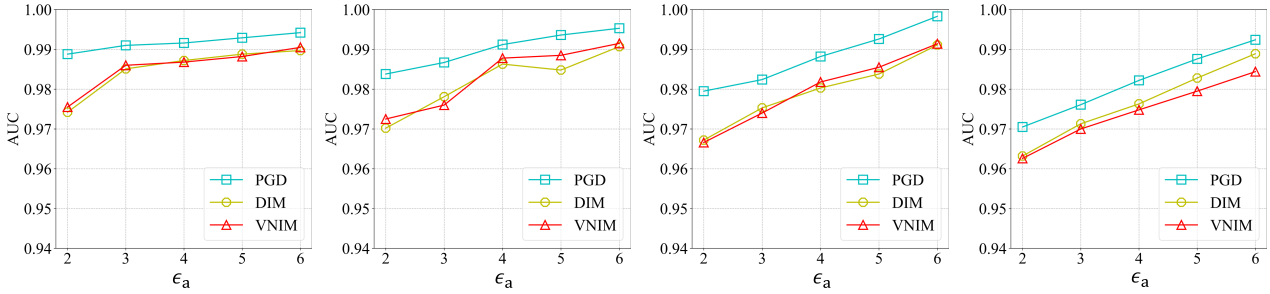


Figure 1. The influence of parameters on the detection performance on ImageNet100.



(a) $\epsilon_i = 2$ (b) $\epsilon_i = 4$ (c) $\epsilon_i = 6$ (d) $\epsilon_i = 8$

Figure 2. The influence of attack intensity on the detection performance on ImageNet100.

Synthetic image detection Since existing adversarial example detection methods do not address adversarial examples based on generative models, we compared our method with various synthetic image detection methods to effectively validate its effectiveness. We conducted detection experiments on adversarial examples based on generative models using the following methods: CNN-Detection[21][3], LGrad[17][4], Universal-Detector[15][5], and DIRE[23][6]. We conducted experiments using the pre-trained models and authors' implementation code of the above methods.

Baselines for face adversarial data detection We follow SPAD [20][7] to compare against two face forgery detection methods, Luo et al.[13][8] and He et al.[7][9], as well as ODIN [11][10] on detecting GAN-based face attacks.

2.4. Implementation Details

Experiments are implemented using 2 Nvidia Deforce RTX 3090 GPUs. For training detectors, we use an Adam optimizer with a learning rate of 2e-4, momentum of 0.9, and weight decay of 5e-6.

Data pre-processing We resize images to 32×32 for CIFAR10, 256×256 for ImageNet100, Makeup and CelebA-HQ, and 112×112 for LFW. For ImageNet100, after resizing, we center-crop the images to 224×224. After that, all of the images are resized to 256×256 followed by random horizontal flip and normalization.

Noise distribution As related in the main paper, we flatten the generated noise to a vector and estimate its distribution. However, for a noise with a shape of $256 \times 256 \times 3$, the length of the corresponding flattened noise is $196,608$, making it impossible to run the algorithm on a single GPU. Therefore, we fix the size of noise to $32 \times 32 \times 3$ and concatenate several sampled pseudo noises to fit the larger image.

3. More Results of Adversarial Detection

To further evaluate the effectiveness of our proposed method, we conduct more experiments on detecting adversarial attacks.

3.1. Detection Against More Attacks

We conduct experiments to detect more gradient-based attacks including RFGSM [19], VNIFGSM [22], and

[3] https://github.com/peterwang512/CNNDetection
[4] https://github.com/chuangchuangtan/LGrad
[5] https://github.com/Yuheng-Li/UniversalFakeDetect?tab=readme-ov-file#weights
[6] https://github.com/ZhendongWang6/DIRE
[7] https://github.com/cc13qq/SPAD
[8] https://github.com/crywang/face-forgery-detection
[9] https://github.com/SSAW14/BeyondtheSpectrum
[10] https://github.com/Jingkang50/OpenOOD

SNIFGSM [12] on CIFAR10 and ImageNet100. As reported in Table 1 and Table 2, our method demonstrates effective detection (achieving an AUROC score over 0.99) across a variety of attack methods, validating our assumptions of the noise distribution proximity.

3.2. Impact of Parameters

We select typical attacks and present curves in Figure 1 to depict the impact of parameters on the detection performance. The parameters $\epsilon_{\boldsymbol{\mu}}$ and ϵ_{σ} have little effect on detecting PGD, while γ_c and γ_l have significant effects on detecting M3D and Diff-PGD. This might be because, regardless of whether $\epsilon_{\boldsymbol{\mu}}$ and ϵ_{σ} are large or small, the accuracy of forming the open coverage remains unaffected as long as the data is sufficient. However, thresholds that are too small lead to the addition of pseudo-noise to insensitive and low-frequency regions, while thresholds that are too large result in extremely sparse pseudo-noise that is far from the adversarial noise distribution.

To explore the impact of perturbation intensity of attacks, we select typical attacks and present the curves in Figure 2, where ϵ_{a} denotes the intensity of unseen attacks and ϵ_{i} denotes the intensity of the initial attack for Perturbation Forgery. We can see that the detector trained with a minor ϵ_{i} is able to detect adversarial data generated with a larger ϵ_{a}. When ϵ_{i} is larger, adversarial data generated with a minor ϵ_{a} is not easy to detect. This might be because adversarial noises generated with a higher ϵ_{a} tend to be more obvious to detect. When ϵ_{a} is too small than ϵ_{i}, the distance between the initial attack noise and the unseen attack noise is too far away, reducing the detection performance.

3.3. More Evaluation Metrics

We further report detection performance using TPR@FPR (True Positive Rate at a fixed False Positive Rate) and accuracy (ACC) metrics on CIFAR-10, including detection accuracy on clean data. As shown in Table 3, with FPR fixed at 0.01, our method achieves an average TPR above 0.98 and an average ACC above 0.97. These results demonstrate the strong detection performance of our approach.

3.4. Another Distribution Modeling

We use a premium distribution, the von Mises-Fisher (vMF) distribution, as an alternative model for the noise representation distribution following [4]. We can model the embedding space as a mixture of class-conditional vMF distributions, one for each class $c \in \{1, 2, \ldots, C\}$:

$$p_d^c\left(\mathbf{r}; \boldsymbol{\mu}_c, \kappa_c\right) = Z_d\left(\kappa_c\right) \exp\left(\kappa_c \boldsymbol{\mu}_c^\top \mathbf{r}\right), \tag{4}$$

where κ_c and $\boldsymbol{\mu}_c$ are class-conditional parameters. Under this probability model, an noise vector $\boldsymbol{r}$ is assigned to class c with the following normalized probability:

$$p\left(y = c \mid \mathbf{r}; \left\{\kappa_j, \boldsymbol{\mu}_j\right\}_{j=1}^C\right) = \frac{Z_d\left(\kappa_c\right) \exp\left(\kappa_c \boldsymbol{\mu}_c^\top \mathbf{r}\right)}{\sum_{j=1}^C Z_d\left(\kappa_j\right) \exp\left(\kappa_j \boldsymbol{\mu}_j^\top \mathbf{r}\right)}. \tag{5}$$

where I_v is the first kind of modified Bessel function with order v. $Z_d(\kappa)$ can be calculated in closed form based on κ and the dimensionality d. Importantly, the vMF distribution is characterized by two parameters: the mean vector $\boldsymbol{\mu}$ and concentration parameter κ. Samples that are more aligned with the center $\boldsymbol{\mu}$ have a higher probability density, and vice versa. Here κ indicates the tightness of the distribution around the mean direction $\boldsymbol{\mu}$. The larger the value of κ, the stronger the distribution is concentrated in the mean direction. In the extreme case of $\kappa = 0$, the sample points are distributed uniformly on the hypersphere.

We conduct experiments on CIFAR-10 using the vMF distribution. As shown in Table 4, compared with Gaussian modeling, using the vMF distribution does not improve detection performance. Modeling the noise distribution as a multivariate Gaussian is sufficient to construct the open covering of adversarial noise and train a robust detector.

Table 3. More detection metrics on CIFAR-10($\epsilon = 4/255$). "thr" denotes the classification threshold and "Clean" denotes clean data without being attacked.

Attack	TPR@FPR=0.01	ACC@thr=0.5
BIM	0.9800	0.9650
PGD	0.9800	0.9700
RFGSM	0.9840	0.9710
DIM	0.9760	0.9530
MIM	0.9870	0.9910
NIM	0.9860	0.9920
VNIM	0.9850	0.9850
SNIM	0.9840	0.9860
AA	0.9810	0.9880
Clean	-	0.9860

Table 4. Comparison of AUROC scores against adversarial attacks on CIFAR-10 with different noise distributions.

Attack	Gaussian	vMF
BIM	0.9965	0.9977
PGD	0.9965	0.9961
RFGSM	0.9968	0.9965
DIM	0.9947	0.9952
MIM	0.9980	0.9979
NIM	0.9980	0.9983
VNIM	0.9976	0.9952
SNIM	0.9964	0.9965
AA	0.9988	0.9974

3.5. Time Overhead of Inference

We conducted experiments to calculate the inference time overhead for 100 samples on ImageNet100. As shown in Table 5, the time cost of our model is slightly higher than that of LID, LiBRe, and SPAD, and it only takes 0.0485 seconds to process one image. For actual use, a minor increase in time is perfectly acceptable in exchange for a significant increase in detection performance.

Table 5. Time overhead of inference (second) on ImageNet100

Detector	LID	LiBRe	EPSAD	SPAD	ours
Time (second)	1.80	2.56	396.81	4.56	4.85

3.6. Detect Attacks with Various Attack Intensities

To evaluate detection performance against attacks of varying intensities ϵ, we train a detector using the FGSM attack with $\epsilon = 2$, $\epsilon = 4$, and $\epsilon = \{2, 4\}$, and test it against attacks with $\epsilon = 2, 4, 6, 8$. As shown in Table 6, the detection performance experiences a minor decrease when encountering lower-intensity attacks, such as $\epsilon = 2$. However, for higher-intensity attacks, the detector maintains strong detection performance. To mitigate performance degradation, training the detector across a range of attack intensities could be beneficial, achieving an AUROC of 0.9924 and 0.9910 against PGD at $\epsilon = 2$ on CIFAR-10 and ImageNet100, respectively.

Table 6. Cross-Intensity Detection (AUROC) against PGD, where the detectors are trained with the initial attack FGSM under $\epsilon = 2$, $\epsilon = 4$, and $\epsilon = \{2, 4\}$.

Dataset	Detector	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$
CIFAR10	$\epsilon = 2$	0.9920	0.9944	0.9997	0.9999
	$\epsilon = 4$	0.9554	0.9965	0.9993	0.9998
	$\epsilon = \{2, 4\}$	0.9924	0.9961	0.9995	0.9999
ImageNet100	$\epsilon = 2$	0.9911	0.9914	0.9920	0.9964
	$\epsilon = 4$	0.9513	0.9912	0.9957	0.9994
	$\epsilon = \{2, 4\}$	0.9910	0.9907	0.9968	0.9994

3.7. Cross-Model Detection

To validate detection ability in a cross-model scenario, we train a detector using adversarial data generated on WRN-28 and test it with adversarial data crafted on ResNet50 and ResNet101. As shown in Table 7, the detector maintains high detection performance in this setting, achieving AUROC scores above 0.99, demonstrating the stability of our method. The results also demonstrate that a smaller backbone, such as WRN-28, can still achieve strong detection performance.

Table 7. Cross-Model Detection (AUROC) on CIFAR-10 under $\epsilon = 4/255$, where the initial attack is executed on WRN-28 but detect the adversarial data crafted with other models.

Attack	ResNet50	ResNet101
BIM	0.9966	0.9965
PGD	0.9965	0.9965
RFGSM	0.9962	0.9958
DIM	0.9928	0.9950
MIM	0.9974	0.9963
NIM	0.9983	0.9966
VNIM	0.9974	0.9962
SNIM	0.9968	0.9970
AA	0.9990	0.9988

3.8. Standard Deviation

We run our method on CIFAR-10 and ImageNet100 against attacks under $\epsilon = 4/255$ with 5 different random seeds and report the standard deviation of AUROC in Table 8. From the results, the detector trained with perturbation forgery has a very small standard error, indicating the consistency and repeatability of our method.

Table 8. Standard deviation of AUROC with 5 different random seeds against adversarial attacks under $\epsilon = 4/255$

Attack	CIFAR-10	ImageNet100
BIM	0.9965±0.0013	0.9911±0.0024
PGD	0.9965±0.0011	0.9912±0.0032
RFGSM	0.99368±0.0012	0.9911±0.0030
DIM	0.9947±0.0024	0.9863±0.0018
MIM	0.9980±0.0009	0.9931±0.0033
NIM	0.9980±0.0015	0.9934±0.0027
VNIM	0.9976±0.0017	0.9878±0.0022
SNIM	0.9964±0.0013	0.9914±0.0021
AA	0.9988±0.0008	0.9941±0.0016

3.9. Cross-Dataset Detection

To evaluate detection ability in a cross-dataset scenario, we train detectors on CIFAR-10, CIFAR-10 + ImageNet100, and ImageNet100 with distributions estimated on CIFAR-10, and test them on ImageNet100. As shown in Table 9, detectors trained on low-resolution datasets, such as CIFAR-10, struggle to perform well on higher-resolution datasets like ImageNet100. However, this limitation can be addressed by jointly training the detector on both low- and high-resolution datasets. Additionally, distribution estimation is not restricted by the dataset, allowing us to use a well-estimated distribution to optimize detector training on a specific dataset.

Figure 3. Natural images and corresponding pseudo-adversarial images generated by Perturbation Forgery.

Table 9. Cross-Dataset Detection against attacks under $\epsilon = 4/255$, where the detectors are trained on CIFAR-10, CIFAR-10 + ImageNet100, and ImageNet100 with distribution estimated on CIFAR10, and tested on ImageNet100 dataset. "Joint" denotes the detector jointly trained on CIFAR-10 + ImageNet100, and "ImageNet100†" denotes the detector trained on ImageNet100 with distribution estimated on CIFAR10.

Attack	CIFAR-10	Joint	ImageNet100†
BIM	0.4176	0.9911	0.9911
PGD	0.4152	0.9910	0.9918
RFGSM	0.4155	0.9912	0.9911
DIM	0.4275	0.9904	0.9871
MIM	0.3983	0.9935	0.9933
NIM	0.4087	0.9926	0.9921
VNIM	0.4052	0.9903	0.991
SNIM	0.4120	0.9918	0.9921
AA	0.4263	0.9954	0.9936

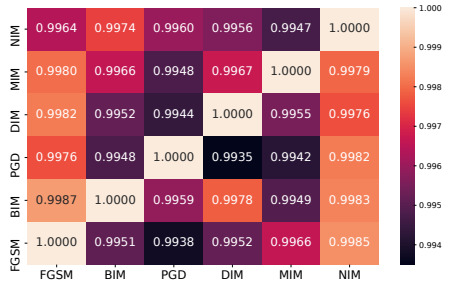


Figure 4. Impact of the initial attacks (AUROC) on CIFAR-10 under $\epsilon = 4/255$. Y-axis: initial attack. X-axis: testing attack.

3.9.1. Impact of the Initial Attack

Our proposed perturbation forgery requires a commonly used attack to construct the open covering. To assess the impact of the initial attack choice, we train detectors using distributions estimated from various initial attacks and test them against other attacks on CIFAR-10. As shown in Figure 4, the detectors consistently achieve a high AUROC score above 0.99, indicating that our method does not depend on a specific attack.

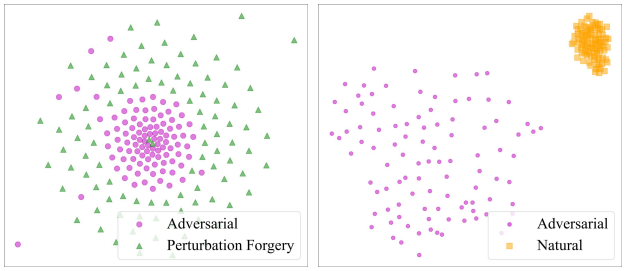


Figure 5. 2D T-SNE visualizations. (a) CIFAR-10 flattened noises of adversarial data and Perturbation Forgery. (b) CIFAR-10 features extracted by the model trained with Perturbation Forgery.

4. Limitations

The major limitation of our method is that finding the optimal scale parameters for distribution perturbation is challenging. We must rely on continuous experimentation to determine these parameters. Specifically, we initiate parameters of uniform distribution as $\epsilon_{\boldsymbol{\mu}} = 1$ and $\epsilon_\sigma = 0.001$, and adjust them using a small step factor. Finally, we find the experiment results when $\epsilon_{\boldsymbol{\mu}} = 3$ and $\epsilon_\sigma = 0.005$ are close to optimal. However, even if these parameters are not optimal, our method still maintains consistent detection performance across various attacks on multiple general and facial datasets, with a satisfactory inference time cost.

The second limitation of our method is that while detectors trained with a smaller attack intensity can perform well against attacks of higher intensity, detectors trained with a larger attack intensity struggle to detect attacks of lower intensity. However, this issue can be addressed by jointly training the detector across a range of attack intensities, as verified in Section 3.6.

Another limitation is the lack of cross-dataset general-

ization. A detector trained on a low-resolution dataset, such as CIFAR-10, often struggles to generalize to higher-resolution datasets like ImageNet. One approach to address this limitation is to jointly train the detector on datasets with varying resolutions, as verified in Section 3.9.

5. More Visualization Results

5.1. Visualization on CIFAR-10

We extract features and flattened noises from adversarial data and Perturbation Forgery from CIFAR-10 and visualize them using 2D t-SNE projection in Figure 5.

5.2. Visualization Examples

Some examples of Perturbation Forgery at $\epsilon = 4/255$ are shown in Figure 3.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4685–4694, 2019. 2

[3] Zhijie Deng, Xiao Yang, Shizhen Xu, Hang Su, and Jun Zhu. Libre: A practical bayesian approach to adversarial detection. In *CVPR*, pages 972–982, 2021. 2

[4] Xuefeng Du, Gabriel Gozum, Yifei Ming, and Yixuan Li. Siren: Shaping representations for detecting out-of-distribution objects. *NeurIPS*, 35:20434–20449, 2022. 4

[5] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[7] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. In *IJCAI*, pages 2534–2541. IJCAI, 2021. Main Track. 3

[8] Shengshan Hu, Xiaogeng Liu, Yechao Zhang, Minghui Li, Leo Yu Zhang, Hai Jin, and Libing Wu. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In *CVPR*, pages 15014–15023, 2022. 2

[9] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 2

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. 2

[11] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. 3

[12] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2019. 4

[13] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *CVPR*, pages 16312–16321, 2021. 3

[14] 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, 2018. 2

[15] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023. 3

[16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *SIGSAC*, pages 1528–1540, 2016. 2

[17] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*, pages 12105–12114, 2023. 3

[18] Liang Tong, Zhengzhang Chen, Jingchao Ni, Wei Cheng, Dongjin Song, Haifeng Chen, and Yevgeniy Vorobeychik. Facesec: A fine-grained robustness evaluation framework for face recognition systems. In *CVPR*, pages 13254–13263, 2021. 2

[19] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 3

[20] Qian Wang, Yongqin Xian, Hefei Ling, Jinyuan Zhang, Xiaorui Lin, Ping Li, Jiazhong Chen, and Ning Yu. Detecting adversarial faces using only real face self-perturbations. In *IJCAI*, pages 1488–1496, 2023. Main Track. 2, 3

[21] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020. 3

[22] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, pages 1924–1933, 2021. 3

[23] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *ICCV*, pages 22445–22455, 2023. 3

[24] Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, Jun Zhu, Yuefeng Chen, and Hui Xue. Towards face encryption by generating adversarial identity masks. In *ICCV*, pages 3877–3887, 2021. 2

[25] Bangjie Yin, Wenxuan Wang, Taiping Yao, Junfeng Guo, Zelun Kong, Shouhong Ding, Jilin Li, and Cong Liu. Adv-makeup: A new imperceptible and transferable attack on face recognition. In *IJCAI*, pages 1252–1258, 2021. 2

[26] Shuhai Zhang, Feng Liu, Jiahao Yang, Yifan Yang, Changsheng Li, Bo Han, and Mingkui Tan. Detecting adversarial data by probing multiple perturbations using expected perturbation score. In *ICML*, pages 41429–41451. PMLR, 2023. 2