Research on Image Adversarial Example Generation Method Based on Diff-AIGAN

Hanqi Liu, Jiaming Xu, Yanbing Liang
School of Science, North China University of Technology, Tangshan, Hebei, China

Abstract: To address the issues of disturbance deviating from key areas and insufficient controllability in adversarial sample generation methods based on Generative Adversarial Networks (AIGAN), which lead to suboptimal attack effectiveness and low authenticity, this paper proposes the Diff-AIGAN adversarial sample generation method. First, a Channel-Spatial Attention Module (Convolutional Block Attention Module, CBAM) is introduced to re-calibrate the feature maps using a "channel-first, spatial-later" attention mechanism, guiding the network to focus automatically on more important channels and positions. Next, the fused feature maps are input into the generator to generate the initial disturbance, and a Stochastic Differential Guide Module (SDGM) is used to enhance the controllability of the disturbance, generating better adversarial samples. Finally, the adversarial samples are input into the discriminator and target model, and the loss value is iteratively computed and fed back to the generator to optimize the generation of more effective perturbations. Experimental results show that the Diff-AIGAN method achieves an attack success rate of over 99% on LeNetC and VGG11 in the MNIST dataset, and an attack success rate of 96.15% and 96.43% on ResNet18 and ResNet32 models in the CIFAR-10 dataset, respectively. At the same time, the generated disturbances focus on key image areas, with high sparsity and small magnitude, and outperform comparison methods across various metrics.

Keywords: Adversarial Examples; Generative Adversarial Networks (GANs); Diffusion Models; Perturbations; Image Generations

1. Introduction

In recent years, deep neural networks (DNNs) have achieved remarkable performance in image classification [1], object detection [2], semantic segmentation [3], and autonomous driving [4]. However, numerous studies have demonstrated that DNNs are highly vulnerable to carefully crafted inputs. By adding imperceptible, small-magnitude perturbations to benign images, an adversary can generate adversarial examples that induce incorrect predictions. This vulnerability poses serious risks in safety-critical applications such as surveillance, autonomous driving, and medical diagnosis, and highlights the need for a systematic study of adversarial example generation and its improvement to enhance the security and trustworthiness of AI systems.

Szegedy et al. [5] first systematically revealed the existence of adversarial examples. Subsequently, gradient-based attacks such as the Fast Gradient Sign Method (FGSM) [6], the Basic Iterative Method (BIM)[7], and Projected Gradient Descent (PGD) [8] have become standard white-box baselines. In parallel, optimization-based approaches, including DeepFool [9] and the Carlini & Wagner (C&W) attack [10], formulate adversarial example generation as a constrained optimization problem, achieving high attack success rates with small perturbation magnitudes. Nevertheless, these methods typically incur high computational cost and are difficult to deploy in efficient or real-time attack scenarios.

To improve generation efficiency, generative adversarial networks (GANs) have been introduced into adversarial example generation. AdvGAN [11] directly learns a mapping from clean images to perturbations via a generator, achieving a favorable trade-off between attack success rate and generation speed. AdvGAN++[12] further enhances the stability and generalization of the generator by refining the perturbation constraints and network architecture. Zhao et al. [13] proposed DG-GAN, a bidirectional GAN framework that jointly addresses adversarial example generation and defense, demonstrating strong performance under both white-box and black-box settings. Zhu et al. [14] presented GE-AdvGAN, which employs a gradient editing mechanism to improve the transferability and quality of adversarial examples. Meanwhile, representative GAN variants such as DCGAN, WGAN, and StyleGAN have significantly improved generation quality, training stability, and sample diversity, yet they still suffer from mode collapse. AIGAN [15] introduces an Attacker module that supplies strong external adversarial samples to the discriminator, compelling the generator to produce more deceptive perturbations.

Despite these advances, existing methods for image adversarial example generation still exhibit two major limitations: (1) most generators mainly rely on single-scale features and lack joint modeling of global semantics and local details, leading to suboptimal feature representations; (2) the generated perturbations often have limited sparsity and smoothness, which reduces their naturalness and makes them more detectable. As a result, current approaches struggle to simultaneously balance attack success rate, imperceptibility, and transferability. To address these issues, this paper proposes Diff-AIGAN, an improved adversarial example generation framework based on feature fusion and noise diffusion. Built upon the AIGAN architecture, Diff-AIGAN incorporates a Convolutional Block Attention Module (CBAM) to fuse deep and shallow feature maps, enabling the generator to focus perturbations on semantically critical regions while preserving fine details and enhancing attack effectiveness. In addition, a Stochastic Differential Guide Module (SDGM) is designed to refine and control the perturbations, improving the realism of the generated adversarial examples. Extensive experiments demonstrate that Diff-AIGAN achieves a more favorable balance between attack success rate and perturbation imperceptibility, providing a new perspective for research on adversarial example generation.

2. Design of the Diff-AIGAN Model

2.1 Architecture of the AIGAN Model

The overall architecture of AIGAN consists of a generator, a discriminator, an attacker, and a fixed target model. the adversarial example generation process is as follows. Given a clean image x and a target label t, the generator produces a class-conditional perturbation δ, which is added to the original image to obtain an adversarial example x'. The adversarial example is then fed into the discriminator, which is trained to distinguish real images from generated adversarial ones. Meanwhile, adversarial examples produced by the attacker are also provided to the discriminator to further strengthen adversarial training. A joint loss, composed of the target attack loss, the discriminator attack loss, and the discrimination loss, is backpropagated to the generator, encouraging it to learn stronger adversarial perturbations. By iteratively updating these components, AIGAN is able to generate high-quality targeted adversarial examples for arbitrary classes. In this work, we build upon this baseline architecture and introduce several improvements.

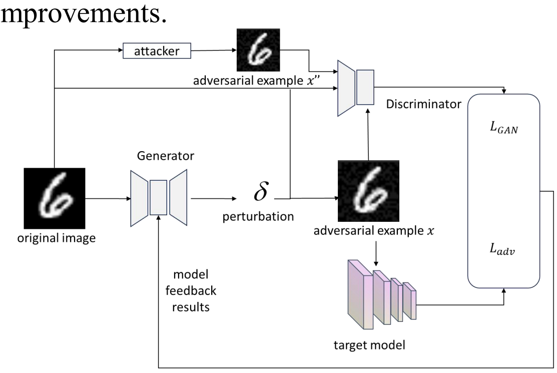


Figure 1. Architecture of the AIGAN Model

2.2 Architecture of the Diff-AIGAN Model

In this paper, we propose an improved adversarial example generation method, termed Diff-AIGAN. As shown in Fig. 2, Diff-AIGAN mainly consists of a Convolutional Block Attention Module (CBAM), a generator, a Stochastic Differential Guide Module (SDGM), a discriminator, an attacker module, and a fixed target model. CBAM is employed to adaptively fuse multi-scale feature maps, while the generator produces initial perturbations. SDGM refines these perturbations and outputs the final adversarial examples. the discriminator distinguishes adversarial examples from original images and computes the adversarial loss, whereas the attacker module further strengthens the generator's attack capability. the target model classifies the adversarial examples and provides a misclassification loss. the adversarial loss and misclassification loss are jointly backpropagated to the generator, encouraging it to produce adversarial examples that are both visually similar to the original images and highly effective in misleading the target model. capabilities.

2.3 Channel-Spatial Attention Module

Feature fusion aims to exploit feature maps from different layers to enhance the model's representation capability for input images. In the original AIGAN model, the encoder primarily extracts deep feature maps, while fine-grained details contained in shallow features tend to be suppressed or lost. To preserve local details while retaining global semantic information, we introduce a feature fusion operation after deep feature extraction. the Convolutional Block Attention Module (CBAM) [16] alleviates the inconsistency among multi-scale features by adaptively learning channel-wise and spatial weights for each Feature scale, with only minor inference overhead. As a result, CBAM enables flexible and efficient feature fusion by assigning appropriate weights to different feature maps, rather than merely adding or concatenating them. In Diff-AIGAN, CBAM is inserted before the generator takes the original image as input, dynamically reweighting the feature maps and guiding the network to focus on salient regions. This multi-scale attention mechanism allows the generator to capture richer multi-scale information and concentrate perturbations on critical areas of the image, thereby improving the attack effectiveness of the generated adversarial examples.

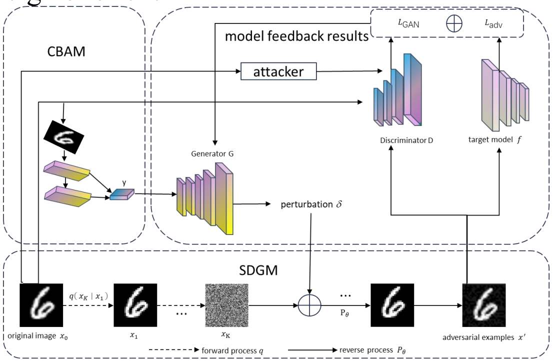


Figure 2. Architecture of the Proposed Diff-AIGAN Model

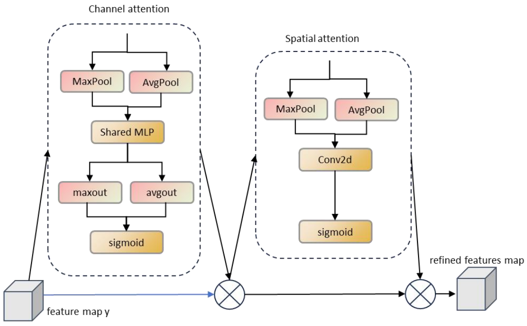


Figure 3. Feature Fusion Process

2.4 Generator Module

The generator takes the fused feature maps as input and produces the adversarial perturbation. As illustrated in Fig. 4, it consists of four residual blocks followed by three deconvolution layers. the residual blocks are employed to stabilize the training of the deep network, mitigate performance degradation, and enhance the feature representation capability. the subsequent deconvolution layers progressively transform the output feature maps of the residual blocks into the final perturbation. For the first two deconvolution layers, instance normalization is applied after the deconvolution operation to accelerate convergence, followed by a ReLU activation function to alleviate the vanishing gradient problem. the last deconvolution layer is followed by a Tanh activation function, which produces the final perturbation map.

2.5 Stochastic Differential Guide Module (SDGM)

We introduce diffusion models (DMs) [17] to construct the Stochastic Differential Guide Module (SDGM), which guides the optimization of perturbations, improves their controllability, and encourages the generated adversarial examples to remain visually similar to the original images while preserving strong attack capability. SDGM consists of a forward process q and a reverse process p_θ. Let the original image be drawn from $x_0 \sim q(x_0)$, where x_0 denotes the initial clean state. the forward process is discretized into T time steps and gradually adds Gaussian noise to x_0, producing a sequence $[x_1, x_2, \cdots, x_t, \cdots, x_T]$ that forms a Markov chain. By accumulating the single-step transitions, the conditional distribution of x_t given x_0 can be written in closed form as

$$q(x_t \mid x_0) = N\left(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I\right) \qquad (1)$$

where N represents the Gaussian distribution, β_t is a fixed value in the interval (0, 1), which increases with the increase of t, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. the noise variance is determined by β_t, and the mean value is determined by β_t and the current noisy data distribution.

The reverse process aims to recover image data from the Gaussian noise data distribution $x_T \sim N(0, I)$ through a denoising function. Using a resampling strategy, a modified U-Net network is trained as a denoiser, and the denoising process is regarded as an optimization process of parameters. the reverse process p_θ is defined as Formula (2) and Formula (3):

The reverse process aims to recover the image data from Gaussian noise $x_T \sim N(0, I)$ by progressively denoising. Following the standard diffusion framework, we adopt a modified U-Net as the denoising network and formulate the reverse dynamics as a parametric optimization problem. the reverse process p_θ is defined as

$$p(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t) \tag{2}$$

$$p_\theta(x_{t-1} \mid x_t) \sim N(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

where б denotes the learnable parameters, and $\mu_\theta(x_t, t)$ and $\sigma_t^2 I$ are the mean and variance of the reverse transition, respectively. For convenience, we denote the denoising step from x_t to $x_{t-1} = \mathrm{P}_\theta(x_t, t)$ as $x_{t-1} = \mathrm{P}_\theta(x_t, t)$ where $P\theta(\cdot)$ is the denoising function.

Within SDGM, the perturbation is refined and the adversarial example is generated as follows. First, the forward process is run for K steps to obtain a noisy sample $x_K \sim q(x_K \mid x_0)$ from the original image x_0. the perturbation δ produced by the generator is then injected at this intermediate state by forming $x_K + \delta$, which serves as the starting point of the reverse process. Finally, $x_K + \delta$ is iteratively denoised from $t = K$ down to $t = 0$:

$$x' = \mathrm{P}_\theta(\ldots(\mathrm{P}_\theta(\mathrm{P}_\theta(x_K + \delta, K), K-1)\ldots, 0)\ \#(4)$$

yielding the refined adversarial example x'.

Through this guided diffusion procedure, SDGM smooths and restructures the perturbation, producing adversarial examples that are visually close to the original images while remaining highly effective in misleading the target model.

2.6 Discriminator Module

The discriminator is designed to distinguish whether the input is a clean image or an adversarial example. As illustrated in Fig. 5, it consists of four convolutional layers for feature extraction. Each convolutional layer employs a 3×33 \times 33×3 kernel with a stride of 1. After each convolution, batch normalization is applied to accelerate training and improve generalization, followed by a LeakyReLU activation function to introduce nonlinearity. the final layer maps the extracted features to a single scalar, which is passed through a Sigmoid function to produce a confidence score: values closer to 1 indicate that the input is more likely to be a clean image, whereas values closer to 0 suggest that the input is more likely to be an adversarial example.

Attacker Module

The Attacker module receives feature maps of the target image and generates initial perturbations using the PGD algorithm. By exploiting multi-level feature information, it produces adversarial perturbations that remain highly similar to the original image in appearance. This process ensures that the perturbations can effectively mislead the target model's classification while inducing minimal perceptual distortion, thereby improving the realism of the adversarial examples.

The adversarial example maps generated by the Attacker module are then fed into the discriminator D for joint training, which effectively strengthens the discriminator and, in turn, facilitates the generation of higher-quality adversarial examples.

2.7 Loss Function

The training objective of Diff-AIGAN consists of three components: the adversarial loss L_{GAN}, the misclassification loss L_{adv} for the target model f, and the soft hinge loss L_{hinge} that constrains the perturbation magnitude. the adversarial loss L_{GAN} encourages the generator to produce samples that are difficult for the discriminator to distinguish, and is defined as

$$L_{\mathrm{GAN}} = E_{x_0}\log D(x_0) + E_{x_0}\log\left(1 - D(x')\right) \tag{4}$$

where $D(\cdot)$ denotes the output of the discriminator, x_0 is the clean image, and x' is the corresponding adversarial example.

The misclassification loss L_{adv} is computed based on the output of the target model f. It guides the generator to produce adversarial examples with strong attack capability by encouraging a significant discrepancy between the predictions on x_0 and x':

$$L_{\mathrm{adv}} = E_{x_0} l_f(x') \tag{5}$$

where l_f denotes the loss function used to train the target model f.

To limit the perturbation magnitude and preserve the realism and naturalness of adversarial examples, we adopt a norm-based soft hinge loss L_{hinge} to stabilize the training of the generative model:

$$L_{\text{hinge}} = E_{x_0} \max(0, \| \delta \|_2 - c) \tag{6}$$

where ccc is a margin hyperparameter and δ\deltaδ denotes the generated perturbation.

Therefore, the overall training objective of Diff-AIGAN is given by

$$L = L_{\text{adv}} + \alpha L_{\text{GAN}} + \beta L_{\text{hinge}} \tag{7}$$

where α and β are hyperparameters that control the relative contributions of the adversarial loss and the soft hinge loss, enabling a better trade-off between attack strength and perturbation regularization.

3 Experimental Design and Result Analysis

3.1 Datasets

To evaluate the performance of Diff-AIGAN in generating adversarial examples, experiments are conducted on the MNIST and CIFAR-10 datasets. the MNIST dataset consists of 70, 000 grayscale images of handwritten digits, including 60, 000 training samples and 10, 000 test samples, each with a resolution of 28×28. the CIFAR-10 dataset contains RGB color images from 10 classes, with 50, 000 training samples and 10, 000 test samples, and each image has a resolution of 32×32.

3.2 Evaluation Metrics

To evaluate the performance of the proposed Diff-AIGAN model, we adopt classification accuracy (ACC), attack success rate (ASR), and the L_0、L_1, and L_2 norms as evaluation metrics. ACC measures the classification capability of the target model on clean images; a higher ACC indicates a stronger classifier and thus provides a more reliable basis for assessing the effectiveness of the attack model. Let N denote the number of clean images and mmm the number of images correctly classified by the target model. ACC is defined as

$$ACC = \frac{m}{N} \times 100\% \tag{8}$$

ASR measures the ability of the attack model to mislead the target model. A higher ASR implies a stronger attack capability. Let N_{adv} denote the total number of generated adversarial examples, and n the number of adversarial examples that successfully fool the target model. ASR is defined as

$$ASR = \frac{n}{N_{adv}} \times 100\% \tag{9}$$

The L_0 norm represents the number of pixels that are modified in the adversarial example compared with the original image, constraining how many pixels can be altered but not the magnitude of each change. A smaller L0L_0L0 value indicates fewer modified pixels and thus sparser perturbations. the L_1 norm is the sum of absolute differences between the adversarial example and the original image at all pixel locations; a larger L_1 norm indicates a more pronounced overall difference. the L_2 norm is defined as the square root of the sum of squared pixel-wise differences, providing a trade-off between the number and magnitude of modifications. A smaller L_2 norm implies that the overall perturbation added to the image is of lower magnitude.

3.3 Experimental Settings

All experiments are conducted on a Linux platform running Ubuntu 18.04.6 LTS (64-bit), equipped with an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz, 240 GB of RAM, and an NVIDIA A100 GPU.

For training the target models, we use the Adam optimizer with an initial learning rate of 0.001. the models are trained for 120 epochs with a batch size of 128. the learning rate is reduced by a factor of 10 at the 50th and 80th epochs.

For training the proposed attack model Diff-AIGAN, we adopt a batch size of 64 and use Adam with the same initial learning rate of 0.001. the model is trained for 100 epochs, and the learning rate is decayed by a factor of 10 at the 50th and 80th epochs. the perturbation threshold is set to 0.3.

Due to inference-time and memory constraints, we employ a Denoising Diffusion Implicit Model (DDIM) with an accelerated sampling strategy. Specifically, the original diffusion process with T=1000 time steps is subsampled to $T_d = 40$ steps. We then set $K = 0.2T = 200$ in the original time scale, which corresponds to a subsequence length of $K_d = 8$ =8 in the subsampled schedule. SDGM is run with the reduced number of steps $K_d \ll K$ to refine the perturbations and generate high-fidelity adversarial examples.

3.4 Experimental Result Analysis

In the untargeted attack setting, the time required by different methods to generate 500 adversarial examples against the LeNetC model on the MNIST test set is reported in Table 1. During measurement, we ignore the time for loading the model and dataset, and only record the time spent on adversarial example generation.

AdvGAN and AIGAN achieve the fastest generation speed, requiring less than 0.01 s to produce 500 adversarial examples, whereas the C&W attack is the slowest, taking approximately 3 hours for the same number of samples. Diff-AIGAN is slightly slower than AIGAN but still significantly faster than C&W. This is because Diff-AIGAN introduces additional feature fusion and perturbation refinement operations during generation, which increase the model complexity and thus incur extra computational overhead.

Table 1. Attack Success Rates of Different Methods under Adversarial Training Defenses

Method	FGSM	C&W	PGD	advGAN	AIGAN	Diff-AIGAN
Time	0.06s	>3h	0.07s	<0.01s	<0.01s	0.5s

3.4. 1 Evaluation of Attacks Without Defense

In the absence of any defense mechanism, we evaluate the attack success rates of different generation algorithms on the MNIST and CIFAR-10 datasets. On MNIST, Diff-AIGAN achieves attack success rates above 99% when attacking the LeNetC and VGG11 models. On CIFAR-10, Diff-AIGAN attains attack success rates above 95% against both ResNet18 and ResNet32. As reported in Table 2, Diff-AIGAN consistently achieves the best performance across all target models compared with the other attack methods.

Among the baselines, FGSM exhibits the lowest attack success rate on all target models. For example, when attacking VGG11, its attack success rate is only 56.61%, which is substantially lower than that of the other methods. Except for the VGG11 case, Diff-AIGAN also surpasses AdvGAN and AIGAN on all models. More specifically, on MNIST, the attack success rate of Diff-AIGAN against LeNetC is improved by 1.14% and 0.64% over AdvGAN and AIGAN, respectively; on CIFAR-10, its attack success rate against ResNet18 is increased by 1.63% and 0.14% compared with these two methods. These results demonstrate that the adversarial examples generated by Diff-AIGAN are more effective in misleading the target models.

Table 2. Attack Success Rates of Different Methods on MNIST and CIFAR-10

Dataset	Target models	ACC (%)	ASR(%)					
			FGSM	C&W	DeepFool	AdvGAN	AIGAN	Diff-AIGAN
MNIST	LeNetC	99.60	94.04	96.67	95.16	97.90	98.40	**99.04**
	VGG11	99.36	56.61	87.50	86.04	**99.65**	99.58	99.28
CIFAR-10	ResNet18	94.74	90.69	93.75	93.22	94.52	96.01	**96.15**
	ResNet32	93.25	90.06	90.30	91.67	94.70	95.29	**96.43**

We also compute the norm values of perturbations generated by different attack methods to assess the magnitude of the induced changes. As representative examples, we report the results for attacks on LeNetC using MNIST and on ResNet32 using CIFAR-10, as summarized in Table 3. Among all methods, FGSM yields the worst performance in terms of perturbation norms.

Overall, Diff-AIGAN achieves loweL_0r and L_1 norms than AdvGAN and AIGAN, and shares the best results with the C&W attack on two norm-based metrics. Since C&W produces adversarial examples with very small norm values, its outputs exhibit high visual fidelity. More specifically, on MNIST, Diff-AIGAN consistently outperforms AdvGAN and AIGAN in terms of the L_0 norm and attains three best norm scores, with an L_0 value of only 150.24. On CIFAR-10, Diff-AIGAN achieves an L_1 norm as low as 12.8, which is substantially smaller than those of the other methods. These results indicate that the perturbations generated by Diff-AIGAN are sparser and of lower overall magnitude, enabling the construction of more realistic and less perceptible adversarial examples.

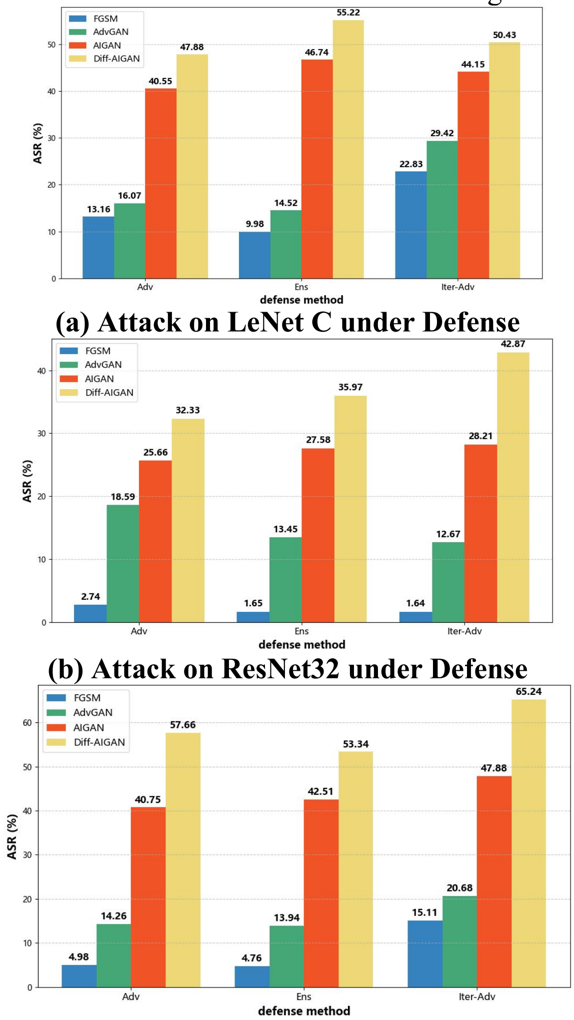
Table 3. Average Norms Between Adversarial Examples and Original Images for Different Methods

norm	Datasets	FGSM	Deepfool	C&W	AdvGAN	AIGAN	Diff-AIGAN
L_0	MNIST	780.79	463.69	784.02	454.77	389.10	150.24
	CIFAR-10	3054.4	3052.2	2951.5	3052.4	3049.8	3020.1
L_1	MNIST	373.22	32.99	38.98	83.11	58.26	22.95
	CIFAR-10	520.30	24.28	56.16	221.03	131.07	12.8
L_2	MNIST	13.72	2.21	1.88	3.95	3.41	2.79
	CIFAR-10	9.85	0.68	1.51	4.28	4.09	3.69

3.4. 2Evaluation of Attacks With Defense

In the defended setting, an attack is considered effective if the adversarial examples can still successfully mislead the target model after defense mechanisms have been applied. Adversarial training is a widely used defense strategy, in which adversarial examples generated by the attack model are combined with clean images to form an augmented training set for the target model, thereby improving its robustness.

In this work, we adopt three representative adversarial training schemes as defenses: standard FGSM-based adversarial training (Adv) [6], ensemble adversarial training (Ens) [18], and iterative adversarial training (Iter-Adv) [19]. Specifically, clean images and adversarial examples are mixed at a 1:1 ratio to construct a new training dataset for the target models. This setup is used to evaluate the effectiveness of Diff-AIGAN under various defense strategies.



(a) Attack on LeNet C under Defense

(b) Attack on ResNet32 under Defense

(c) Attack on Wide-ResNet34 under Defense

Figure 6. Attack Success Rates of Different Methods under Defense.

The Attack Success Rates of FGSM, AdvGAN, AIGAN, and Diff-AIGAN under different adversarial training strategies. As shown in the Fig, Diff-AIGAN consistently achieves the highest attack success rate when attacking defended target models. In particular, Fig. 6(c) illustrates that, on CIFAR-10 against the Wide-ResNet34 model, Diff-AIGAN exhibits the most pronounced advantage. Compared with AIGAN, the attack success rate of Diff-AIGAN is improved success rate of Diff-AIGAN is improved by 16.91% under Adv defense, by 10.83% under Ens defense, and by 17.36% under Iter-Adv defense. In all other configurations, Diff-AIGAN also outperforms FGSM, AdvGAN, and AIGAN in terms of attack success rate. These results clearly demonstrate that Diff-AIGAN remains highly effective even in the presence of adversarial training defenses.

3.4.3Ablation Experiment

To further validate the effectiveness of the key components in Diff-AIGAN, we conduct ablation studies on the MNIST dataset against the LeNetC model and on the CIFAR-10 dataset against the ResNet32 model. Starting from the AIGAN baseline, we denote the variant that only incorporates the adaptive spatial feature fusion module as "+CBAM", and the variant that only employs the Stochastic Differential Guide Module as "+SDGM".

As shown in Table 4, Diff-AIGAN achieves the best performance across all evaluation metrics, indicating that the full method, which combines feature fusion and perturbation refinement, can simultaneously improve the attack success rate, increase perturbation sparsity, and reduce perturbation magnitude. Specifically, compared with AIGAN, the "+CBAM" and "+SDGM" variants improve the attack success rate by 0.47% and 0.53% on MNIST, and by 0.53% and 0.94% on CIFAR-10, respectively. This demonstrates that the feature fusion strategy, which integrates global semantics and local details, enables the generator to concentrate perturbations on critical regions of the image, making the resulting adversarial examples more likely to mislead the target model.

Moreover, relative to "+CBAM", the "+SDGM" variant yields consistently lower L_0, L_1, and L_2 norms, indicating that SDGM effectively refines and controls the perturbations. the resulting perturbations are sparser and of smaller overall magnitude, thereby enhancing the realism and stealthiness of the generated adversarial examples.

Table 4. Ablation Study on the MNIST and CIFAR-10 Datasets.

Datasets	Method	ASR(%)	L_0	L_1	L_2
MNIST	AIGAN	98.40	389.10	58.26	3.41
	+CBAM	98.87	211.42	37.48	3.82
	+SDGM	98.93	189.26	28.77	3.11
	Diff-AIGAN	99.04	150.24	22.95	2.79
CIFAR-10	AIGAN	95.29	3049.8	131.07	4.09
	+ CBAM	95.82	3024.88	32.84	4.11
	+SDGM	96.23	3030.15	20.55	3.89
	Diff-AIGAN	96.43	3020.10	12.80	3.69

4. Conclusion

In this paper, we propose an improved image adversarial example generation method, termed Diff-AIGAN. By incorporating the CBAM module, Diff-AIGAN effectively exploits both global semantic information and local details of the input images, enabling the generator to concentrate perturbations on critical regions and thereby enhancing the attack effectiveness. In addition, a Stochastic Differential Guide Module (SDGM) is designed based on the reverse process of diffusion models to refine the perturbations. SDGM improves the controllability of the perturbations, increases their sparsity, and reduces their magnitude, leading to more realistic adversarial examples. Experimental results demonstrate that the key perturbations generated by Diff-AIGAN can successfully mislead the target models while remaining sparse and low in amplitude, and the resulting adversarial examples appear more natural and less perceptible than those produced by the baseline methods.

Although Diff-AIGAN is capable of generating high-quality adversarial examples, it still has certain limitations. Compared with AIGAN, the proposed framework has a more complex architecture, which results in higher computational and memory costs during adversarial example generation. In future work, we plan to further lighten the network structure to reduce the time and memory overhead, while preserving the high quality and strong attack capability of the generated adversarial examples.

References

[1] Qian S, Ning C, Hu Y. MobileNetV3 for image classification [C]//2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021:490-497.

[2] Xie X, Cheng G, Wang J, et al. Oriented R-CNN for object detection [C] //Proceedings of the IEEE/CVF international conference on computer vision. 2021:3520-3529.

[3] Wang Min, Li Sheng, Zhuang zhihao, et al. Ground-based Cloud Graph Segmentation Method Based on Deep Learning Semantic Segmentation Network. 2023, 23(31). (in Chinese)

[4] Chib P S, Singh P. Recent advancements in end-to-end autonomous driving using deep learning: A survey [J]. IEEE Transactions on Intelligent Vehicles, 2023, 9(1):103-118. [5] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks [J]. arXiv preprint arXiv:1312.6199, 2013.

[6] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples [J]. arXiv preprint arXiv:1412.6572, 2014.

[7] Kurakin A, Goodfellow I J, Bengio S. Adversarial examples in the physical world [M]//Artificial intelligence safety and security. Chapman and Hall/CRC, 2018:99-112.

[8] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks [J]. arXiv preprint arXiv:1706.06083, 2017.

[9] Moosavi-Dezfooli S M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:2574-2582.

[10] Carlini N, Wagner D. Towards evaluating the robustness of neural networks [C]//2017 ieee symposium on security and privacy (sp). Ieee, 2017:39-57.

[11] Xiao C, Li B, Zhu J Y, et al. Generating adversarial examples with adversarial networks [J]. arXiv preprint arXiv:1801.02610, 2018.

[12] Jandial S, Mangla P, Varshney S, et al. Advgan++: Harnessing latent layers for

adversary generation [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019:0-0.

[13] He X, Luo Z, Li Q, et al. DG-GAN: a high quality defect image generation method for defect detection [J]. Sensors, 2023, 23(13):5922.

[14] Zhu Z, Chen H, Wang X, et al. Ge-advgan: Improving the transferability of adversarial samples by gradient editing-based adversarial generative model [C]//Proceedings of the 2024 SIAM international conference on data mining (SDM). Society for Industrial and Applied Mathematics, 2024:706-714.

[15] Bai T, Zhao J, Zhu J, et al. Ai-gan: Attack-inspired generation of adversarial examples [C]//2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021:2543-2547.

[16] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module [C]//Proceedings of the European conference on computer vision (ECCV). 2018:3-19.

[17] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models [J]. Advances in neural information processing systems, 2020, 33:6840-6851.

[18] Tramèr F, Kurakin A, Papernot N, et al. Ensemble adversarial training: Attacks and defenses [J]. arXiv preprint arXiv:1705.07204, 2017.

[19] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks [J]. arXiv preprint arXiv:1706.06083, 2017