Supplementary Materials for AdvDiff: Generating Unrestricted Adversarial Examples using Diffusion Models

Xuelong Dai[1], Kaisheng Liang[1], and Bin Xiao[1]

The Hong Kong Polytechnic University
{xuelong.dai, kaisheng.liang}@connect.polyu.hk, b.xiao@polyu.edu.hk

A Detailed Proof of Equation 8

We can obtain the sample x_{t-1} with condition label y, according to the sampling with the classifier-free guidance. To get the unrestricted adversarial example x^*_{t-1}, we add adversarial guidance to the conditional sampling process with Equation 8. With Bayes' theorem, we want to deduce the adversarial sampling with adversarial guidance at timestep t by:

$$p(x^*_{t-1}|y_a) = \frac{p(y_a|x^*_{t-1})p(x^*_{t-1})}{p(y_a)} \tag{11}$$

with Equation 11, we want to sample the adversarial examples with the target label y_a. Starting from x_t, the sampling of the reverse generation process with AdvDiff is:

$$p(x^*_{t-1}|x_t, y_a) = \frac{p(y_a|x^*_{t-1}, x_t)p(x^*_{t-1}|x_t)}{p(y_a|x_t)} \tag{12}$$

Noted that Equation 12 is the same as the deviation of classifier-guidance in [8]'s Section 4.1, where they treated $p(y_a|x_t)$ as a constant. Because $p(x^*_{t-1}|x_t)$ is the known sampling process by our conditional diffusion sampling, we evaluate $\frac{p(y_a|x^*_{t-1},x_t)}{p(y_a|x_t)}$ by:

$$\log p_f(y_a|x^*_{t-1}) - \log p_f(y_a|x_t) \tag{13}$$

We can approximate Equation 13 using a Taylor expansion around $x^*_{t-1} = \mu(x_t)$ as:

$$\begin{aligned}\log p_f(y_a|x^*_{t-1}) - \log p_f(y_a|x_t) \approx & \log p_f(y_a|\mu(x_t)) \\ & + (x^*_{t-1} - \mu(x_t))\nabla_{\mu(x_t)} \log p_f(y_a|\mu(x_t)) \\ & - \log p_f(y_a|x_t) + C \\ & = (x^*_{t-1} - \mu(x_t))\nabla_{\mu(x_t)} \log p_f(y_a|\mu(x_t)) + C\end{aligned} \tag{14}$$

Assume $p(x^*_{t-1}|x_t) = \mathcal{N}(x^*_{t-1}; \mu(x_t), \sigma_t^2\mathbf{I}) \propto e^{-(x^*_{t-1}-\mu(x_t))^2/2\sigma_t^2}$, we have:

$$\begin{aligned}
p(x^*_{t-1}|x_t, y_a) &\propto e^{-(x^*_{t-1}-\mu(x_t))^2/2\sigma_t^2+(x^*_{t-1}-\mu(x_t))\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t))} \\
&\propto e^{-(x^*_{t-1}-\mu(x_t)-\sigma_t^2\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)))^2/2\sigma_t^2+(\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)))^2/2\sigma_t^2} \\
&\propto e^{-(x^*_{t-1}-\mu(x_t)-\sigma_t^2\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)))^2/2\sigma_t^2+C} \\
&\approx \mathcal{N}(x^*_{t-1}; \mu(x_t)+\sigma_t^2\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)), \sigma_t^2\mathbf{I}) \qquad (15)
\end{aligned}$$

Sampling with Equation 15 should be:

$$x^*_{t-1} = \mu(x_t, y) + \sigma_t\varepsilon + \sigma_t^2 s\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)) \qquad (16)$$

where $\mu(x_t, y)$ is the conditional mean value and ε is sampled from $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. Note that $\mu(x_t, y) + \sigma_t\varepsilon$ is the normal sampling process that we will get x_{t-1}. In practice, in each diffusion step, the difference between x_{t-1} and $\mu(x_t)$ should be small enough [8, 12] for a reasonable and stable diffusion sampling. Therefore, we adopt x_{t-1} to calculate the adversarial gradient after the sampling with the conditional diffusion model, and we have:

$$x^*_{t-1} = \mu(x_t, y)+\sigma_t\varepsilon+\sigma_t^2 s\nabla_{\mu(x_t)}\log p_f(y_a|\mu(x_t)) \approx x_{t-1}+\sigma_t^2 s\nabla_{x_{t-1}}\log p_f(y_a|x_{t-1}) \qquad (17)$$

where s is the adversarial guidance scale. □

B Detailed Proof of Equation 10

The deviation of Equation 10 is similar to Equation 8, where the noise sampling guidance is added with the forward diffusion process. Similarly, we have Equation 9:

$$p(x_T|y_a) = \frac{p(y_a|x_T)p(x_T)}{p(y_a)} = \frac{p(y_a|x_T, x_0)p(x_T|x_0)}{p(y_a|x_0)} \qquad (18)$$

And Taylor expansion around $x_T = x_0$ to evaluate $\frac{p(y_a|x_T,x_0)}{p(y_a|x_0)}$.

$$\log p_f(y_a|x_T) - \log p_f(y_a|x_0) = (x_T - x_0)\nabla_{x_0}\log p_f(y_a|x_0) + C \qquad (19)$$

From x_0 to x_T, we gradually add the Gaussian noise with the predefined schedule [12]:

$$p(x_T|x_0) = \mathcal{N}(x_T; \sqrt{\bar{\alpha}_T}x_0, (1-\bar{\alpha}_T)\mathbf{I}) \qquad (20)$$

The noise sampling guidance is as follows:

$$\begin{aligned}
x_T &\approx (\bar{\mu}(x_0, y) + \bar{\sigma}_T\varepsilon) + \bar{\sigma}_T^2 a\nabla_{x_0}\log p_f(y_a|x_0) \\
&= x_T + \bar{\sigma}_T^2 a\nabla_{x_0}\log p_f(y_a|x_0) \qquad (21)
\end{aligned}$$

where $\bar{\mu}(x_0, y) + \bar{\sigma}_T\varepsilon$ is the forward diffusion process to get x_T with x_0 and a is the noise sampling guidance scale. □

Algorithm 2 DDIM Adversarial Diffusion Sampling

Require: y_a: target label for adversarial attack
Require: y: ground truth class label
Require: s, a: adversarial guidance scale
Require: w: classification guidance scale
Require: N: noise sampling guidance steps
Require: T: reverse generation process timestep
$x_T \sim \mathcal{N}(0, \mathbf{I})$
$x_{adv} = \varnothing$
for $i = 1 \dots N$ **do**
  for $t = T, \dots, 1$ **do**
    $\tilde{\epsilon}_t = (1 + w)\epsilon_\theta(x_t, y) - w\epsilon_\theta(x_t)$
    $\hat{\epsilon}_t = \tilde{\epsilon}_t - \sqrt{1 - \bar{\alpha}_t}\nabla_{x_t} \log p_f(y_a|x_t)$
    Classifier-free DDIM sampling x_{t-1} with $\hat{\epsilon}_t$
  end for
  Obtain classification result from $f(x_0)$
  Compute the gradient with $\log p_f(y_a|x_0)$
  Update x_T by $x_T = x_T + a\nabla_{x_0} \log p_f(y_a|x_0)$
  $x_{adv} \leftarrow x_0$ if $f(x_0) = y_a$
end for
return x_{adv}

C AdvDiff for DDIM

We give the derivation for AdvDiff for DDIM followed with [8]. The score function for the DDIM diffusion model is:

$$\nabla_x \log p_f(x|y) = \nabla_x \log p_f(x) + \nabla_x \log p_f(y|x) \tag{22}$$

We set y as our adversarial guidance y_a:

$$\begin{aligned} \nabla_x \log p_f(x|y_a) &= \nabla_x \log p_f(x) + \nabla_x \log p_f(y_a|x) \\ &= -\frac{1}{\sqrt{1 - \bar{\alpha}}}\epsilon_\theta(x) + \nabla_x \log p_f(y_a|x) \end{aligned} \tag{23}$$

Finally, the new epsilon prediction $\hat{\epsilon}_\theta(x_t)$ is defined as follows:

$$\hat{\epsilon}_\theta(x_t) = \epsilon_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t}\nabla_{x_t} \log p_f(y_a|x_t) \tag{24}$$

Then the DDIM with AdvDiff is Algorithm 2 over the trained classifier-free diffusion model $\epsilon_\theta(\cdot)$.

We can further deduce the DDIM with $\hat{\epsilon}_\theta(x_t)$ by:

$$\begin{aligned} x_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t}\hat{\epsilon}_\theta}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1 - \bar{\alpha}_{t-1}}\hat{\epsilon}_\theta \\ &= \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon_\theta + C \cdot \nabla_{x_t} \log p_f(y_a|x_t) \end{aligned} \tag{25}$$

where we can replace C with our adversarial guidance scale.

D Related Work

Since Szegedy et al. [30] had proved that DL models are extremely vulnerable to adversarial attacks, researchers have been digging into improving the model's adversarial robustness by proposing stronger adversarial attack methods and their counter-measurements.

Perturbation-based adversarial examples with generative models: Most related works performed adversarial attacks by perturbing a subset of clean data to fool the target classifier. These attacks [3, 16, 20] attempted to generate better perturbations with higher attack success rates and smaller perturbations. With the emergence of generative models, end-to-end adversarial attacks [1,24,33] have greatly improved the generation efficiency by pre-training the generative module. These methods integrate the advertorial loss into the training of generative models and generate adversarial examples by trained generators with clean data.

Unrestricted adversarial examples with generative models: Perturbation-based adversarial examples require insignificant norm distance to the given clean data in order to guarantee the indistinguishability, which only covers a small fraction of all possible adversarial examples [28]. To remove such restrictions, Song et al. [28] proposed an unrestricted adversarial attack method that searches over the latent space of the input noise vector with an adversarial loss function and a well-trained AC-GAN [22]. Inspired by Song's work [28], recent works [23, 32] made improvements in the generation quality and generation efficiency of UAEs. Diffusion model based adversarial attacks [4, 5, 7] also achieve satisfying attack performance against deep learning models. However, the performance of existing approaches suffers from the unstable training of GAN models as well as the lack of theoretical support for injecting PGD-based gradients. Therefore, we provide an effective and theoretically analyzed solution with the diffusion model in this paper.

D.1 Conditional Diffusion Model for Image Generation

Diffusion models have shown great generation quality and diversity in the image synthesis task since Ho et al. [12] proposed a probabilistic diffusion model for image generation that greatly improved the performance of diffusion models. Diffusion models for conditional image generation are extensively developed for more usable and flexible image synthesis. Dhariwal & Nichol [8] proposed a conditional diffusion model that adopted classifier-guidance for incorporating label information into the diffusion model. They separately trained an additional classifier and utilized the gradient of the classifier for conditional image generation. Jonathan Ho & Tim Salimans [13] performed the conditional guidance without an extra classifier to a diffusion model. They trained a conditional diffusion model together with a standard diffusion model. During sampling, they adopted the combination of these two models for image generation. Their idea is motivated by an implicit classifier with the Bayes rule. Followed by [8, 13]'s works,

many research [10, 19, 21, 25] have been proposed to achieve state-of-the-art performance on image generation, image inpainting, and text-to-image generation tasks. Despite utilizing diffusion models for image generation has been widely discussed, none of these works have discovered the adversarial examples generation method with the diffusion model. Also, it is a new challenge to defend against the adversarial examples generated by the diffusion model.

E Implementation Details

As AdvDiff supports both DDPM and DDIM sampling, we adopt LDM [1] with DDIM sampler for the experiment on ImageNet for reproducibility and poor performance of simple DDPM on ImageNet. We adopt 500 sampling steps for DDPM on MNIST and 200 sampling steps for LDM on ImageNet. For conditional sampling, we use one-hot label information for both DDPM and DDIM sampling for a fair comparison with GAN. The noise sampling step is set as $(0, 0.5]$ for the MNIST dataset and $(0, 0.2]$ for the ImageNet dataset. We follow the default settings in DiffAttack and AdvDiffuser in the experiments.

F More Experiment Results

We give more experiment results in Figure 2 to demonstrate the generation quality on the ImageNet dataset. We also provide some failure cases of our AdvDiff, which happens when we set the adversarial guidance scale s and a extremely large. Figure 3 shows that a large s (10.0) tends to generate images with noisy textures while a large a (10.0) can generate noisy images. Figure 4 shows that modifying the initial noise with a can disturb the noise distribution if we add the gradient in an irrational manner.

G AdvDiff against Adversarial Training with Diffusion Models

Table 1: Performance under AdvDiff attack against adversarial training on the ResNet18 model.

Method	Clean	PGD	AdvDiff Attack	AT-UAE	PGD	AdvDiff Attack	AT-PGD	PGD	AdvDiff Attack
Accuracy (%)	99.0	0.7	7.9	99.2	16.8	32.6	95.2	79.2	13.5

Adversarial training is an effective way to improve classification accuracy against adversarial attacks. Thus, it should be an effective way to defend against

[1] `https://github.com/CompVis/latent-diffusion`

Fig. 1: User study on MNIST datast. Flipped-label UAEs are tagged with a red box. MNIST dataset is robust against UAEs because each image only contains 28×28 pixels.

UAEs. However, UAEs are generated from random noise latents rather than fixed gradient perturbations by given input images. Therefore, adversarial training with UAEs is not as effective as it is with perturbation-based attacks. We test the AT-UAE with UAEs generated by AdvDiff on the MNIST dataset with 1000 images per class. The results are given in Table 1. The result shows that AT with UAEs improves the robust accuracy against AdvDiff, but the performance is limited as there is an infinite number of random latents to generate UAEs.

H Improving Attack Transferability

AdvDiff achieves overwhelmingly better generation quality and attack success rate against white-box target models by adversarial diffusion sampling with a given target label y. However, the attack transferability is limited due to different decision boundaries from black-box models. Normally, black-box attackers use the gradient of the original label to generate perturbations. Therefore, we adopt the same settings to improve the attack transferability of AdvDiff (denoted as AdvDiff-Untargeted), i.e., $-\nabla_{x_{t-1}} \log p_f(y|x_{t-1})$, where y is the ground truth label to generate samples. However, such sampling will decrease the generation quality as sampling from the negative distribution does not follow the benign diffusion process. Table 2 shows that the attack transferability significantly improved with a decrease in generation quality. We leave a better design of attack

Fig. 2: More unrestricted adversarial examples generated by AdvDiff on the ImageNet dataset.

Fig. 3: Failure cases when $s = 10.0$.

Fig. 4: Failure cases when $a = 10.0$.

transferability for future work. Additional experiments against transformers are also given in Table 2.

I User Study

We further perform a user study to justify the performance of AdvDiff, where we ask 20 participants to identify flipped label images on the MNIST dataset with 5 images on each class by U-GAN and AdvDiff. The results are given in Table 3. We also give the tagged examples on UAEs generated by U-GAN and AdvDiff in Figure 1, where AdvDiff's UAEs are remarkably better in generation quality and harder to identify flipped label images than U-GAN.

J Improving the Generation Quality

AdvDiff crafts adversarial examples with imperceptible perturbations, making the generation quality of our methods largely reliant on the benign diffusion model's performance. Figure 5 shows that AdvDiff produces higher-quality images when using StableDiffusion as the benign diffusion model. Moreover, we can set the adversarial guidance to a smaller value for better quality with a decrease in the generation speed. The guidance in the paper on the MNIST dataset aims at high ASR per batch for a fair comparison with previous attacks, while the visual quality can be affected by its limited 28×28 grey pixel space. We can

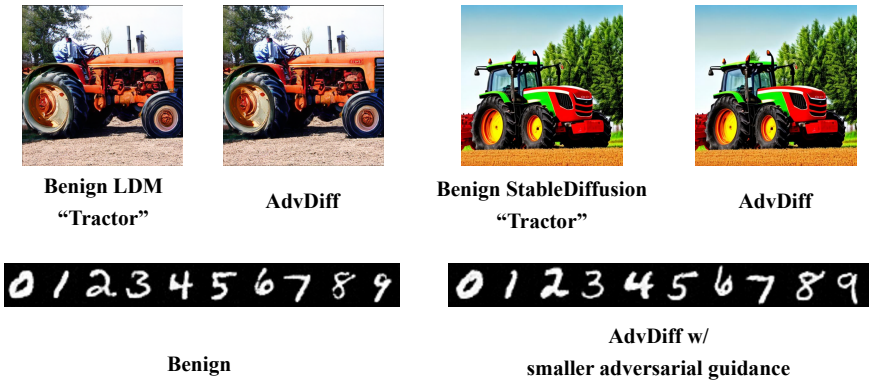

Fig. 5: The improvements for better generation quality.

also achieve stable AE generation by using latents obtained by conducting the forward diffusion process from the training dataset's clean images.

K Comparing with Existing Diffusion Model Attacks

Fig. 6: The generated adversarial examples (mushroom) from different diffusion-based attacks and corresponding perturbations.

There are several diffusion model adversarial attacks [4, 5, 7] achieve state-of-the-art performance. However, most of them did not release the official code which makes it difficult to compare with these methods. All these works adopt the optimization over given loss functions (i.e., PGD-like gradient) to generate UAEs with the diffusion models. Figure 6 provides a direct comparison of adversarial examples from different methods. Our findings indicate that PGD-like adversarial guidance perturbations significantly alter the texture of benign images from AdvDiffuser. Similarly, the perturbations from DiffAttack are also very similar to standard PGD perturbations, where the perturbations are uniformly

applied across the entire image. In contrast, our perturbations mainly target the mushroom's contour and are substantially less noticeable than those from existing attacks.

Our work can easily be combined with some exciting works by replacing the gradient with AdvDiff's adversarial guidance (especially for AdvDiffuser [5] which directly adopts the PGD gradient to conduct the adversarial attack). We hope our work can gain new insight for designing adversarial attacks using diffusion models.

L Comparing with 2021 CVPR Competition Winner

We compare with the 1st winner [18] of 2021 CVPR unrestricted adversarial attack competition [6] follows their official implementation on ImageNet. Two variants of [18]'s attacks are compared, which are GA-IFGSM and GA-FSA. The results are given in Table 4. The proposed AdvDiff outperforms [18]'s attack in terms of generation quality and attack performance. It may not be a fair comparison as [18]'s attack is not a synthetic attack.

M Discussion about perturbations, flipped-label, and diffusion adversarial examples

Perturbation-based adversarial attacks typically generate adversarial examples by iteratively adding adversarial gradients to clean images, which inevitably introduces noisy patterns. These patterns create visible defects that can be detected by humans. However, these perturbations are applied at the pixel level, leaving the content of the clean image unchanged. In contrast, GAN-based unrestricted adversarial attacks create Unrestricted Adversarial Examples (UAEs) by perturbing the latents. The generator then produces images based on these GAN latents. This method introduces perturbations at the content level, as GAN-based techniques do not directly add noise to the final images. Given the generator's sensitivity to changes in low-dimensional latents, adversarial latents can result in images with entirely different content. This can even lead to a change in the label of the adversarial images, creating what we refer to as flipped-label images.

Adversarial examples generated by diffusion models follow a diffusion generation process, which can be seen as a denoising process. As a result, the noisy gradients injected are removed during the generation process. This necessitates a larger Projected Gradient Descent (PGD) gradient in previous works to successfully generate a UAE, often resulting in a decrease in image quality. In our work, we inject the adversarial objective in an interpretable manner by increasing the conditional likelihood on the target attack label, following the diffusion process. We provide detailed proof of the effectiveness of our adversarial guidance in Appendix A and B. Consequently, our proposed AdvDiff method is more reliable in generating high-quality adversarial examples than simply conducting the PGD attack on the sampled images of the diffusion model.

N Ethics Concerns

AdvDiff can bring security problems to existing DL-based applications, and it generates visually indistinguishable adversarial examples to humans while deceiving the target DL model. This characteristic makes the AdvDiff's images hard to detect by current defense mechanisms, even with human experts. However, our unrestricted adversarial examples can be adopted for adversarial training because our adversarial examples are generated close to the decision boundary of the target classifier. Another critical reason for achieving adversarial training is that the generated adversarial examples have high fidelity and high diversity on the large-scale dataset. Therefore, AdvDiff can have positive social impacts on improving the AI model robustness.

O Limitations

Although AdvDiff shows superior performance on the unrestricted adversarial attack with large-scale datasets, the generation speed of adversarial examples with diffusion models is relatively slower than GAN-based models. This limitation makes AdvDiff hard to perform a real-time attack. However, the unrestricted adversarial attack does not have a real-time attack scenario. And we can also adopt a fast-sampling method to improve the sampling speed of the AdvDiff, which we aim to improve in future work. Another limitation is that AdvDiff is sensitive to the parameter settings of two adversarial guidance scales a and s. The reason is that AdvDiff can deploy in any conditional diffusion model, which has different sampling mechanisms in other datasets. Therefore, we should set the adversarial guidance scales accordingly, but the attack performances are not vastly changed if the scales are in an appropriate range.

References

1. Baluja, S., Fischer, I.: Learning to attack: Adversarial transformation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
4. Chen, J., Chen, H., Chen, K., Zhang, Y., Zou, Z., Shi, Z.: Diffusion models for imperceptible and transferable adversarial attack. arXiv preprint arXiv:2305.08192 (2023)
5. Chen, X., Gao, X., Zhao, J., Ye, K., Xu, C.Z.: Advdiffuser: Natural adversarial example synthesis with diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4562–4572 (2023)
6. Chen, Y., Mao, X., He, Y., Xue, H., Li, C., Dong, Y., Fu, Q.A., Yang, X., Xiang, W., Pang, T., et al.: Unrestricted adversarial attacks on imagenet competition. arXiv preprint arXiv:2110.09903 (2021)

7. Chen, Z., Li, B., Wu, S., Jiang, K., Ding, S., Zhang, W.: Content-based unrestricted adversarial attack. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. pp. 51719–51733 (2023)
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems **34**, 8780–8794 (2021)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
10. Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., Taigman, Y.: Make-a-scene: Scene-based text-to-image generation with human priors. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV. pp. 89–106. Springer (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
13. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
15. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
16. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
17. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: Proceedings of the European conference on computer vision (ECCV). pp. 19–34 (2018)
18. Liu, F., Zhang, C., Zhang, H.: Towards transferable unrestricted adversarial examples with minimum changes. In: 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML). pp. 327–338. IEEE (2023)
19. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022)
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
21. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
22. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the International conference on machine learning. pp. 2642–2651 (2017)
23. Poursaeed, O., Jiang, T., Goshu, Y., Yang, H., Belongie, S., Lim, S.N.: Fine-grained synthesis of unrestricted adversarial examples. arXiv preprint arXiv:1911.09058 (2019)

24. Poursaeed, O., Katsman, I., Gao, B., Belongie, S.: Generative adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4422–4431 (2018)
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
26. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Song, Y., Shu, R., Kushman, N., Ermon, S.: Constructing unrestricted adversarial examples with generative models. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 8322–8333 (2018)
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
30. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
31. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2820–2828 (2019)
32. Xiang, T., Liu, H., Guo, S., Gan, Y., Liao, X.: Egm: An efficient generative model for unrestricted adversarial examples. ACM Transactions on Sensor Networks **18**(4), 1–25 (2022)
33. Xiao, C., Li, B., Zhu, J.Y., He, W., Liu, M., Song, D.: Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610 (2018)
34. Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A.L., Le, Q.V.: Adversarial examples improve image recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 819–828 (2020)
35. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)

Table 2: The attack success rates (%) of ResNet50 examples for transfer attack and attack against defenses on the ImagetNet dataset.

Method	ResNet-152 [11]	Inception v3 [29]	DenseNet-121 [15]
AutoAttack	32.5	38.6	43.8
U-BigGAN	30.8	35.3	16.8
AdvDiffuser	18.3	20.0	24.8
DiffAttack	21.1	43.9	23.8
AdvDiff	20.5	14.9	35.8
AdvDiff-Untargeted	**52.0**	**42.7**	**60.9**
Method	MobileNet v2 [26]	PNASNet [17]	MNASNet [31]
AutoAttack	41.6	38.5	42.5
U-BigGAN	18.4	22.1	16.8
AdvDiffuser	30.3	15.2	26.7
DiffAttack	22.3	26.9	30.4
AdvDiff	15.4	23.2	38.9
AdvDiff-Untargeted	**49.5**	**53.0**	**47.6**
Method	VGG-19 [27]	SENet [14]	WRN [35]
AutoAttack	48.3	23.7	29.5
U-BigGAN	18.4	22.1	16.8
AdvDiffuser	28.7	18.8	22.0
DiffAttack	30.0	22.1	23.6
AdvDiff	16.8	10.0	11.8
$\text{AdvDiff}_{\text{transfer}}$	**58.5**	**51.2**	**57.4**
Method	ViT-B [9]	DeiT-B [34]	BEiT [2]
AutoAttack	9.3	8.9	45.3
U-BigGAN	30.1	27.7	69.4
AdvDiffuser	18.5	12.5	79.4
DiffAttack	17.4	17.5	38.6
AdvDiff	17.8	17.6	78.8
AdvDiff-Untargeted	**36.0**	**58.5**	**81.5**

Table 3: User Study about flipped label problem on MNIST.

Method	U-GAN	AdvDiff
User Study	425/1000	**102/1000**

Table 4: The attack performance on the ImagetNet dataset.

	ASR	PGD-AT	FID	LPIPS	SSIM	BRISQUE	TRES
AdvDiff-Untargeted	99.5	**94.5**	22.8	0.14	0.85	16.2	76.8
AdvDiff	99.8	92.4	**16.2**	**0.03**	**0.96**	**18.1**	**82.1**
GA-IFGSM	99.8	82.6	50.4	0.24	0.78	40.4	62.0
GA-FSA	**99.9**	91.4	70.6	0.32	0.56	50.8	58.4