Adversarial Example Does Good: Preventing Painting Imitation from Diffusion Models via Adversarial Examples

Chumeng Liang [*1,2] **Xiaoyu Wu** [*1,2] **Yang Hua** [3] **Jiaru Zhang** [1] **Yiming Xue** [4,2] **Tao Song** [1] **Zhengui Xue** [1] **Ruhui Ma** [1] **Haibing Guan** [1]

Abstract

Recently, Diffusion Models (DMs) boost a wave in AI for Art yet raise new copyright concerns, where infringers benefit from using unauthorized paintings to train DMs to generate novel paintings in a similar style. To address these emerging copyright violations, in this paper, we are the first to explore and propose to utilize adversarial examples for DMs to protect human-created artworks. Specifically, we first build a theoretical framework to define and evaluate the adversarial examples for DMs. Then, based on this framework, we design a novel algorithm, named AdvDM, which exploits a Monte-Carlo estimation of adversarial examples for DMs by optimizing upon different latent variables sampled from the reverse process of DMs. Extensive experiments show that the generated adversarial examples can effectively hinder DMs from extracting their features. Therefore, our method can be a powerful tool for human artists to protect their copyright against infringers equipped with DM-based AI-for-Art applications. The code of our method is available on GitHub: https://github.com/mist-project/mist.git.

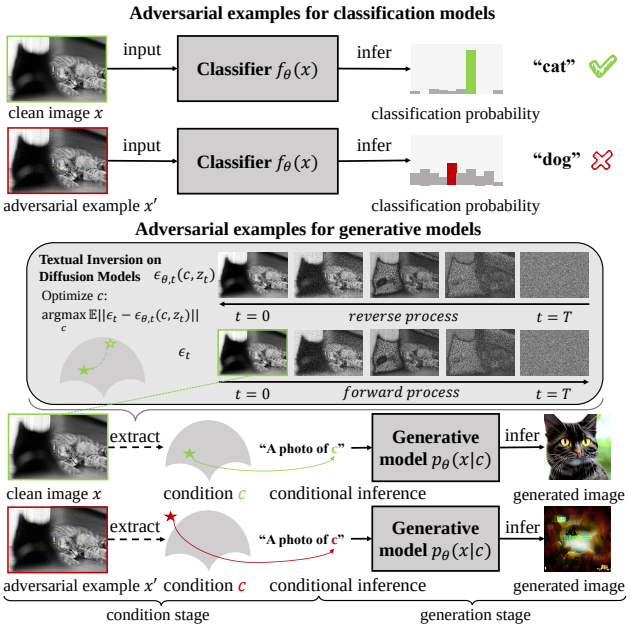


Figure 1. Comparison of workflows for adversarial examples in classification models and diffusion models. Adversarial examples in diffusion models prevent diffusion models from extracting image features as conditions by inducing out-of-distribution features. The feature extracting shown in the figure is textual inversion (Gal et al., 2022) in DMs, which has raised copyright concerns in several cases (MT, 2022; Deck, 2022).

1. Introduction

Recent years have witnessed a boom of deep diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) in computer vision. With solid theoretical foundations (Song et al., 2020b;a; Bao et al., 2022) and highly applicable techniques (Gal et al., 2022; Lu et al., 2022), diffusion models have proven to be effective in generative tasks, including image synthesis (Ruiz et al., 2022), video synthesis (Yang et al., 2022), image editing (Kawar et al., 2022), and text-to-3D synthesis (Poole et al., 2022). Among these, the latent diffusion model (Rombach et al., 2022) shows great power in artwork creation, sparking a commercialization flurry of AI for Art.

Despite the success of diffusion models in commercialization, it has been a public concern that these models empower some copyright violations. For example, *textual inversion* (Gal et al., 2022), a novel function implemented in most AI-for-Art applications based on the latent diffusion model, can imitate the art style of human-created paintings with several samples. Cases have taken place that copyright infringers easily fetch paintings created by artists online

[*]Equal contribution [1]Shanghai Jiao Tong University, China [2]cheer4creativity.ai [3]Queen's University Belfast, UK [4]NYU, USA. Correspondence to: Tao Song <songt333@sjtu.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

and illegally use them to train models with the help of textual inversion (MT, 2022; Deck, 2022). Although artists have the right to declare prohibition for their artworks to be used for training AI-for-Art models, there is no existing technology to prevent or track this illegal use, leading to an even lower crime cost and difficulty in proof generating. Moreover, artists suffer from a lack of resources to start legal challenges against the infringers (Vincent, 2022) (See Appendix A for more discussion about ethical issues of AI for Art). Hence, the art society is calling for off-the-shelf techniques to protect the copyright of paintings against AI for Art (Vincent, 2022).

Inspired by the adversarial examples in image classification (Goodfellow et al., 2014b; Madry et al., 2018; Carlini & Wagner, 2017), an idea for this protection is to add some tailored and tiny perturbations to images and make them *unrecognizable* for the diffusion model in AI-for-Art applications. Here, *unrecognizable* means the image cannot be recognized as a normal image by the diffusion model and hence restrains the model from extracting image features or imitating the art style. We consider these perturbed images as *adversarial examples for diffusion models*. By transferring paintings into adversarial examples without losing the image semantics, the lack of techniques in artwork copyright protection can be resolved.

However, generating adversarial examples for diffusion models is non-trivial. Unlike classification models, diffusion models exploit input images by generating new images conditioned on the inputs rather than conducting an end-to-end inference on them. An adversarial example must then prevent its feature (e.g., styles, contents, *etc.*) from being extracted in some identifiable conditions by the diffusion model. Furthermore, the training objective of diffusion models is optimized indirectly through a variational bound and thus is not applicable in the optimization of the adversarial example. For these reasons, existing research only exploits diffusion models to improve the robustness of classifiers (Nie et al., 2022), leaving a blank in the formulation of adversarial examples for diffusion models.

In this paper, we build a theoretical framework to define and evaluate the adversarial example for diffusion models. Specifically, adversarial examples work on protecting their own feature from being extracted in the inference workflow of diffusion models (See in Figure 1). This workflow consists of two stages: (1) the condition stage that extracts the feature from input images as conditions, and (2) the generation stage that generates images based on these conditions. In the case shown in Figure 1, the condition stage is empowered by textual inversion (Gal et al., 2022). Our adversarial examples work by misleading the feature extracting in the condition stage and resulting in an out-of-distribution condition. To this end, we define the adversarial example with an optimization target to minimize the probability that the image is recognized as a real image by the diffusion model. We optimize the target by adding tiny perturbations to the image. We then formulate the evaluation for the adversarial example according to the workflow shown in Figure 1. A good adversarial example would result in a bad quality of conditional generated images, by which we can evaluate the quality of adversarial examples.

Under the proposed framework, we propose an algorithm to generate the adversarial example for diffusion models. We conduct a Monte Carlo method to estimate the objective function given by our definition in the context of diffusion models. We also evaluate our adversarial examples with real copyright violation scenarios (MT, 2022; Deck, 2022). Extensive experiments show that our adversarial examples can efficiently hinder the latent diffusion model used by commercialization applications from extracting their features and imitating their styles or contents.

Our contributions are summarized in the following aspects.

- We construct a novel framework to define and evaluate the adversarial examples for diffusion models. To the best of our knowledge, we are the first to systematically investigate this topic.
- Under the above framework, we propose an end-to-end algorithm AdvDM to generate the adversarial examples for diffusion models.
- We conduct extensive experiments on several datasets, covering single-category and art-style ones, to validate that our method can effectively protect images from being learned, imitated, and copied by diffusion models.

2. Background

2.1. Generative Modeling and Diffusion Models

A generative model learns from data $x \sim q(x)$ and holds a distribution $p_\theta(x)$ where generated data can be sampled. Generative models based on latent variables have proven effective in generative tasks, including VAEs (Kingma & Welling, 2014; Razavi et al., 2019) and GANs (Goodfellow et al., 2014a; Brock et al., 2018). These models match data with a *latent variable* z in low-dimensional space and model the joint distribution $p_\theta(x, z)$.

An intuitive idea to train a generative model is to maximize $p_\theta(x)$ for real data $x \sim q(x)$. However, $p_\theta(x)$ is difficult to optimize directly thus requiring transformation, where the variational bound (Higgins et al., 2017; Gregor et al., 2016) given by $-\log p_\theta(x) \leq -\log \frac{p_\theta(x,z)}{q(z|x)}$ is selected to be optimized instead.

An important paradigm in generative models is conditional

generative modeling (Mirza & Osindero, 2014). Generally, conditional generative models use different forms of conditions to do image generation, including categories (Mirza & Osindero, 2014), base images (Zhu et al., 2017), characteristics (Karras et al., 2019; 2020), and condition prompting in natural languages (Rombach et al., 2022). Denoting the condition by c, they model $p_\theta(x|c)$ with parameter θ and support sampling images x from this distribution.

As the generative model defining the state-of-the-art, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) construct a series of latent variables $x_{1:T}$ by a Markov Chain $q(x_{1:T}|x_0)$. A reverse Markov Chain $p_\theta(x_{0:T})$ is then used to revert the latent variables to the data x_0. $p_\theta(x)$ is optimized with the variational bound of p_θ,

$$-\log p_\theta(x) \leq -\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} := \mathcal{L}_{DM}. \tag{1}$$

Intuitively, DMs generate images by learning to recover an image from noise by the denoising reverse process $p_\theta(x_{0:T})$. A recent breakthrough has taken place when researchers deploy the denoising reverse process in a latent space (Rombach et al., 2022). This *latent diffusion model* (LDM) has achieved state-of-the-art performance in both image quality in artwork generation and sampling efficiency, being the mainstream model used in AI-for-Art applications.

2.2. Adversarial Examples

Let $p_{data}(y, x)$ denote the joint distribution between data x and label y. A classification model with parameter θ is expected to estimate $p_{data}(y|x)$ with $p_\theta(y|x)$. The difference between the two distributions can be quantitated by KL Divergence. The optimization goal of the classification model can be then formulated by

$$\arg\min_\theta KL(p_{data}(y|x)||p_\theta(y|x)). \tag{2}$$

Various terms of loss function are exploited as alternative optimization targets to this goal in classification modeling. The maximum log-likelihood, a widely-used loss function, proves to be equivalent to the goal in Eq. (2) (Shlens, 2014). To simplify the notation, we use $\mathcal{L}_\theta(x, y)$ to denote the loss term and minimize it in the optimization.

The neural network used in the classification model is vulnerable to *adversarial examples*: given an input x and its label y, it is possible to find a new input x' not classified to y (Goodfellow et al., 2014b; Carlini & Wagner, 2017). The adversarial example x' for classification models can be formulated by

$$\begin{aligned} x' &:= \arg\max_{x'} \quad \mathcal{L}_\theta(x'), \\ s.t. &\quad ||x - x'|| \leq \epsilon. \end{aligned} \tag{3}$$

Various research investigates adversarial examples related to generative models. They mainly consider adversarial examples generated by generative models and misleading classification models (Kos et al., 2018). A discussion of adversarial attacks on flow-based generative models (Pope et al., 2020) aims at finding examples similar to real images but with low likelihood scores in flow-based models. However, its theoretical analysis only considers cases in flow-based models and applies a strong assumption that the data x is normally distributed. Moreover, it does not yield a general formulation of adversarial examples for generative models.

3. Adversarial Examples for Diffusion Models

In this section, we discuss how to generate and evaluate adversarial examples for Diffusion Models (DMs). We first formulate the objective function, which minimizes the probability that the example is a real image and sampled by the model. Then, we propose AdvDM, an algorithm to approximately generate adversarial examples for DMs. Finally, we discuss the concrete evaluation for these adversarial examples. Following the notation of DMs (Ho et al., 2020), the image x is denoted by x_0 and the latent variable z is denoted by $x_{1:T}$ in this section.

3.1. Adversarial Examples for Diffusion Models

We consider adversarial examples that cannot be recognized as real images by diffusion models but are visibly similar to real images. For a diffusion model θ, an adversarial example is out-of-distribution for the generated distribution $p_\theta(x)$. To generate such adversarial examples, an idea is to minimize $p_\theta(x + \delta)$ by adding one or several perturbations δ whose scale is strictly constrained. The constraint of the perturbation scale ensures the perturbation is human-invisible and does not hurt the image semantics. Based on this idea, we define the adversarial example for a diffusion model parameterized by θ.

Definition 3.1 (**Adversarial Example for Diffusion Models**). Given a diffusion model parameterized by θ and the distribution of real data $q(x)$, the adversarial example x' is formulated by $x' = x + \delta$, where $x \sim q(x)$ and δ is given by the following equation:

$$\begin{aligned} \delta &:= \arg\min_\delta p_\theta(x + \delta), \\ &\text{where } x \sim q(x), \|\delta\| \leq \epsilon, \\ &\epsilon \text{ is a constant and usually small.} \end{aligned} \tag{4}$$

However, p_θ is not practically computable in diffusion models. With the help of the latent variable $p_\theta(x)$, we can estimate $p_\theta(x + \delta)$ by Monte Carlo. To this end, we expand

$p_\theta(x)$ over the latent variable $x_{1:T}$,

$$p_\theta(x) = \int p_\theta(x_{0:T})\mathrm{d}x_{1:T}. \tag{5}$$

We denote the adversarial example $x+\delta$ by x', with $\|\delta\| \leq \epsilon$. Eq. (5) suggests it is possible to minimize $p_\theta(x)$ by Monte Carlo: By minimizing $p_\theta(x_{0:T})$ with different sampling processes of $x_{1:T}$, we are approximately minimizing $p_\theta(x')$. Let $u(x'_{1:T})$ denote the distribution of $x'_{1:T}$. We can alter our optimization goal to the following form:

$$\begin{aligned} &\delta := \arg\min_\delta \mathbb{E}_{x'_{1:T}\sim u(x'_{1:T})} p_\theta(x'_{0:T}), \\ &\text{where } x \sim q(x), x' = x + \delta. \end{aligned} \tag{6}$$

An advantage in DMs is that the posterior $q(x'_{1:T}|x'_0)$ is a Gaussian distribution with fixed parameters exactly independent from x'_0. Therefore, it is possible to regularize $p_\theta(x'_{0:T})$ with $q(x'_{1:T}|x'_0)$. We use the negative log term of $p_\theta(x'_{0:T})$ as usual. The final form of our objective function can be then inferred.

$$\begin{aligned} &\min_\delta \mathbb{E}_{x'_{1:T}\sim u(x'_{1:T})} p_\theta(x'_{0:T}) \\ =&\max_\delta \mathbb{E}_{x'_{1:T}\sim u(x_{1:T})} - \log p_\theta(x'_{0:T}) \\ =&\max_\delta \mathbb{E}_{x'_{1:T}\sim u(x'_{1:T})} - \log \frac{p_\theta(x'_{0:T})}{q(x'_{1:T}|x'_0)} \\ =&\max_\delta \mathbb{E}_{x'_{1:T}\sim u(x'_{1:T})} \mathcal{L}_{DM}(x', \theta). \end{aligned} \tag{7}$$

Intuitively, Eq. (7) generates adversarial example x' by maximizing the loss used for training DMs with different latent variables sampled from $u(x'_{1:T})$.

3.2. AdvDM: Generating Adversarial Examples by Monte Carlo

In this subsection, we propose AdvDM, the algorithm to generate adversarial examples for DMs. Inspired by existing methods of adversarial attack on classification tasks (Goodfellow et al., 2014b; Madry et al., 2018; Carlini & Wagner, 2017), we exploit the gradient of our optimization goal. A difference is that we cannot analytically compute the gradient of the objective function $\mathbb{E}_{x_{1:T}\sim u(x_{1:T})}\mathcal{L}_{DM}(\theta)$ since it is the gradient of an expectation. As mentioned in Section 3.1, we estimate it by the expected gradient with Monte Carlo. For each iteration, we sample a $x'_{1:T} \sim u(x'_{1:T})$ and compute a gradient of $\mathcal{L}_{DM}(\theta)$ accordingly. We then do one step of gradient ascent with this gradient. The estimation is summarized in Eq. (8):

$$\nabla_{x_0}\mathbb{E}_{x_{1:T}\sim u(x_{1:T})}\mathcal{L}_{DM}(\theta) \approx \mathbb{E}_{x_{1:T}\sim u(x_{1:T})}\nabla_{x_0}\mathcal{L}_{DM}(\theta). \tag{8}$$

We follow existing methods of adversarial attack (Goodfellow et al., 2014b; Madry et al., 2018) and apply a sign function to constrain the scale of the estimated gradient. Let $x_0^{(i)}$ denote the adversarial example of the ith step in optimization. The adversarial example of the $(i+1)$th step is generated by a signed gradient ascent with step length α,

$$x_0^{(i+1)} = x_0^{(i)} + \alpha \mathrm{sgn}(\nabla_{x_0^{(i)}}\mathcal{L}_{DM}(\theta)|_{x_{1:T}^{(i)}\sim u(x_{1:T}^{(i)})}), \tag{9}$$

where sgn refers to the sign function.

Intuitively, AdvDM samples different latent variables and iteratively conducts one step of gradient ascent on the loss of DMs with different for each sampling. In practice, we let $u(x_{1:T})$ be the posterior $q(x_{1:T}|x_0)$, for it induces a good performance empirically in the experiment. We summarize AdvDM in Algorithm 1. The implementation details are shown in Appendix D.2.

Algorithm 1 AdvDM: Adversarial Example for DMs

Input: Data x_0, parameter θ, number of Monte Carlo N, step length α
Output: Adversarial example x'_0
Initialize $x_0^{(0)} \leftarrow x_0$.
for $i = 1$ **to** N **do**
  Sample $x_{1:T}^{(i)} \sim q(x_{1:T}^{(i)}|x_0^{(i)})$
  $\delta^{(i)} \leftarrow \alpha\mathrm{sgn}(\nabla_{x_0^{(i)}}\mathcal{L}_{DM}(\theta)|_{x_{1:T}^{(i)}})$
  $x_0^{(i)} \leftarrow x_0^{(i-1)} + \delta^{(i)}$
end for
$x'_0 \leftarrow x_0^{(N)}$

3.3. Evaluating the Quality of Adversarial Examples

The diffusion model θ is evaluated by the quality of images sampled from $p_\theta(x)$ (Goodfellow et al., 2014a; Ho et al., 2020). This sampling is called the *inference* of the diffusion model. Unlike classification models, diffusion models do not take images as input directly but exploit them by extracting features from them and generating images conditioned on these features. We mainly focus our evaluation scenario on this conditional inference, where copyright violations have taken place. For unconditional inference, the model samples a noise and generates images. This process has no input images and does not raise copyright concerns, thus not included in our evaluation.

Following the existing research in adversarial examples (Goodfellow et al., 2014b; Dai et al., 2018; Jia & Liang, 2017), we evaluate the adversarial example for diffusion models in inference by measuring how much it would hurt the performance of image generation. As shown in Figure 1, the inference is divided into two stages. In the condition stage, the diffusion model extracts features from the input image. In the generation stage, the model exploits these features as conditions to generate new images. We denote the

condition by c and the feature-extracting process by $p_\theta(c|x)$. In practice, c can be a prompting in natural language (Gal et al., 2022; Ruiz et al., 2022) that abstracts the image semantics or a latent variable (Rombach et al., 2022) related to the image.

The diffusion model θ can then generate an image x_g with a condition c_g sampled from $p_\theta(c|x)$. We model this process by $p_\theta(x|c_g)$, with $x_g \sim p_\theta(x|c_g)$. Note that $p_\theta(x|c_g) = p_\theta(x)\frac{p_\theta(c_g|x)}{p_\theta(c_g)}$. We assume a dependency between c_g and image sample x.

Assumption 3.2 (**Dependency between c_g and x**). c_g is a condition sampled from $p_\theta(c|x)$. We have $\frac{p_\theta(c_g|x)}{p_\theta(c_g)} \geq 1$.

Assumption 3.2 is plausible for most cases since θ is a trained diffusion model and promises a strong relationship between samples and conditions semantically. With this assumption, $p_\theta(x|c)$ can be a higher bound of $p_\theta(x)$ and an alternative distribution for sampling. As the normal evaluation of diffusion models, we also evaluate $p_\theta(x|c)$ by applying a quality metric $\mathbf{D}(\cdot)$ to the image sampled from $p_\theta(x|c)$. The evaluation is summarized in Algorithm 2.

Algorithm 2 Evaluating Adversarial Example for diffusion models

Input: Adversarial example(s) x_{adv}, diffusion model θ, sample quality metric $\mathbf{D}(\cdot)$
Output: the sample quality $\mathcal{Q}$
Initialize the dataset $x_r \leftarrow x_{adv}$
Sample $c_g \sim p_\theta(c|x_r)$
Generate images by sampling $x_g \sim p_\theta(x|c_g)$
$\mathcal{Q} \leftarrow \mathbf{D}(x_g, x_r)$

A good adversarial example prevents $p_\theta(c|x)$ from extracting c accurately and results in a bad sample quality of x_g, which can be measured by the sample quality metric $\mathbf{D}(\cdot)$. In practice, we select Fréchet Inception Distance (FID) (Heusel et al., 2017) and Precision ($prec.$) (Kynkäänniemi et al., 2019) as $\mathbf{D}(\cdot)$.

Three scenarios of $p_\theta(c|x)$ and $p_\theta(x|c)$ are considered in the evaluation of adversarial examples for diffusion models as follows. They either have been (MT, 2022; Deck, 2022) or can be the scenario of copyright violations with AI for Art. Details of three scenarios are given in Appendix D.1.

1. **Text-to-image generation based on textual inversion**: Given a small batch of images x depicting objects of the same category, $p_\theta(c|x)$ abstracts the object in images with a word S^* in natural language. Let the condition c_g be S^*. This is often implemented by the language model embedded in the diffusion model. $p_\theta(x|c_g)$ then generates images x_g conditioned on S^* with the diffusion model. This scenario is shown in Figure 1.
2. **Style transfer based on textual inversion**: Given a small batch of images x depicting objects of the same art style, $p_\theta(c|x)$ abstracts the common art style of images with a word S^* in natural language. Let c_g be S^*. $p_\theta(x|c_g)$ then generates images x_g conditioned on S^* with the diffusion model. In practice, we start generation from step t based on the latent variable $z_{s,t}$ at step t from another source image x_s for better visualization. The generation can be exactly formulated by $p_\theta(x|c_g, z_{s,t})$.
3. **Image-to-image synthesis**: Given an image x, $p_\theta(c|x)$ samples a latent variable z_t at the denoising step t. Let c_g be z_t. We start the generation $p_\theta(x|c_g)$ from step t based on z_t.

4. Experiment

In this section, we evaluate our proposed AdvDM to generate adversarial examples for DMs. Since our motivation is to help protect paintings against being illegally used by AI-for-Art applications, we choose the Latent Diffusion Model (Rombach et al., 2022) (LDM) backbone [1], which is the mainstream model used in AI-for-Art applications (see Appendix E). The implementation details for AdvDM on Latent Diffusion Model are shown in Appendix D.2. We fix l_∞ norm as the constraint for generating all the adversarial examples. Following existing research in adversarial examples, we set the sampling step as 40, the per-step perturbation budget as 1/255, the total budget as 8/255, and the batch size as 4. We conduct experiments on categories of LSUN (Yu et al., 2015) and WikiArt (Nichol, 2016). We use 8 NVIDIA RTX A4000 GPUs for all experiments. Visualization of all experiments is shown in Appendix B. Additionally, we evaluate AdvDM on more conditional generation tools based on the Latent Diffusion Model and demonstrate the results in Appendix F. We also discuss other potential methods to generate adversarial examples for DMs in comparison with AdvDM. The results are listed in Appendix C.

4.1. Text-to-image generation based on textual inversion

We first evaluate our adversarial examples on the text-to-image generation with textual inversion, as mentioned in Section 3.3. To evaluate AdvDM quantitatively, we randomly select 1,000 images from LSUN-cat, LSUN-sheep, and LSUN-airplane. For all experimental settings, we follow the paper of textual inversion (Gal et al., 2022). The images are separated into 5-image groups and we optimize a condition prompting c, i.e., pseudo-word S^* in (Gal et al., 2022) for each group. S^* is a word vector in the semantic space of the language model embedded in LDM (Radford

[1] https://ommer-lab.com/files/latent-diffusion/nitro/txt2img-f8-large/model.ckpt

Table 1. Text-to-image generation based on textual inversion

DATASET	LSUN-CAT			LSUN-SHEEP			LSUN-AIRPLANE		
METRIC	FID ↑	*prec.* ↓	*recall.*	FID ↑	*prec.* ↓	*recall.*	FID ↑	*prec.* ↓	*recall.*
NO ATTACK	34.94	0.5643	0.1531	32.81	0.6378	0.1228	39.22	0.5016	0.2765
ADVDM	127.04	0.1708	0.061	203.5	0.0058	0.378	169.67	0.0263	0.3235

et al., 2021), expected to capture the object in 5 images, e.g., cat for images in LSUN-cat. For each 5-image group and S^*, we set the iteration steps in optimization as 5,000 as default. We then use each pseudo word S^* to generate 50 images conditionally, leveraging the text-to-image function of LDM, which results in a total of 10,000 generated images for each dataset. All the images are resized to 256×256 as default. This generation process is conducted both on clean images and adversarial examples generated by AdvDM.

We evaluate the sample quality of generated images by two metrics: Fréchet Inception Distance (FID) and Precision ($prec.$), which both measure the similarity between generated images and training images. For images generated based on adversarial examples, a high FID and a low Precision show that these images cannot capture the object in the adversarial examples. As a reference, Recall ($recall$) is also calculated in our experiment. However, the difference between Recall of images generated based on clean images and of those on adversarial examples is unpredictable. This is because Recall measures the diversity of generated images rather than the similarity between generated and training images, which is out of the concern raised in our motivation. Implementation details of the metrics mentioned above are shown in Appendix D.1.

The results are shown in Table 1. Our adversarial examples significantly increase FID and decrease Precision of the conditionally-generated images. Meanwhile, Recall does not vary consistently. This suggests our adversarial examples are powerful in protecting its contents from being extracted as generation conditions.

4.2. Qualitative Results on Style Transferring

An important evaluation scenario for our adversarial examples is the style transfer with LDM, where several copyright violations have taken place (Deck, 2022; MT, 2022). In these cases, infringers first used several image samples to train a pseudo word S^* by textual inversion (Gal et al., 2022), as mentioned in Section 4.1. Then, they exploited S^* in the image-to-image conditional generation and generated images that imitated the art style of the sample images.

To evaluate the performance of our adversarial examples in resisting this style transfer, we follow this scenario and compare the sample quality of conditionally-generated images based on clean images and adversarial examples. We select 20 paintings of 10 artists respectively from the WikiArt dataset and train an S^* for each artist. Other settings are the same as the setting in Section 4.1. As displayed in Figure 2, the results demonstrate that the style of the conditionally-generated images is significantly different from the input images when conditioning on S^* training on adversarial examples. This suggests that AdvDM can be effectively used for copyright protection against illegal style transfer. We further conduct experiments to investigate if our adversarial examples work in Stable Diffusion, a commercialized AI-for-Art application. The results are demonstrated in Appendix E.

4.3. Qualitative results on image-to-image synthesis

As mentioned in Section 3.3, image-to-image generation is another scenario that measures the quality for adversarial examples. We first apply AdvDM on several open-source photos from Pexels[2] to generate adversarial examples. Then we generate images based on both these adversarial examples and clean images with the image-to-image pipeline provided by Stable Diffusion, a large-scale commercialized LDM[3]. We compare the quality of generated images in Figure 3. The generated images based on adversarial examples are unrealistic in comparison with those based on clean images.

4.4. Ablation Study

Sampling steps. The number of sampling steps in Monte Carlo is crucial for the accuracy of estimation and thus has a significant impact on the adversarial example generated by AdvDM theoretically. To investigate the effect of this hyper-parameter, we conduct an experiment on the LSUN-airplane dataset, where we pick 100 random images and generate 1,000 images in the setting described in Section 4.1 except for the sampling steps. The number of sampling steps varies from 10 to 1,000. The results are shown in Figure 4. With the increase of the sampling steps, the FID increases, and the Precision decreases roughly. It shows that the quality of adversarial examples grows better with more sampling steps.

[2]https://www.pexels.com/

[3]https://github.com/huggingface/diffusers

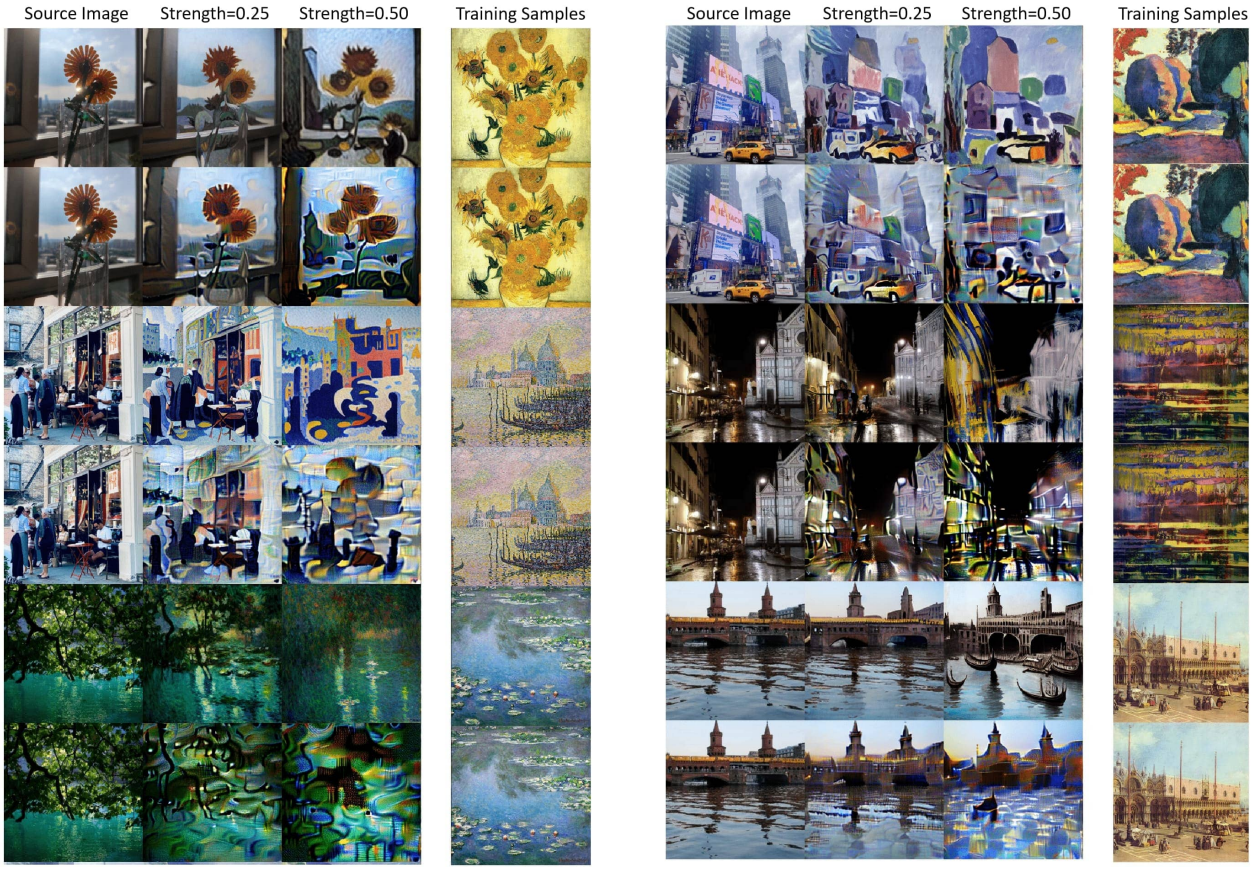


Figure 2. Comparison of generated image quality in style transfer for categories of WikiArt (Nichol, 2016). Images shown in each group share the same source image. We use textual inversion (Gal et al., 2022) to extract the style of training samples from WikiArt, shown in a separate column. For each group, the top row shows the generated images based on the style extracted from the clean examples. The bottom row shows the generated images based on the style extracted from the adversarial examples. Strength is a hyper-parameter that indicates how much the style of the source image is covered by the target style. LDM fails to capture the style from adversarial examples, compared to clean images.

Table 2. Comparison for AdvDM under different sampling steps. The inference time is the average time to generate an adversarial example over 1,000 images on an NVIDIA RTX A4000 GPU. The unit is second.

SAMPLING STEPS	FID↑	*prec.* ↓	INFERENCE TIME
10	122.9	0.05	1.803
40	186.05	0.037	6.342
1000	211.88	0.011	166.6

It appears that a larger number of sampling steps results in stronger effects on the attack, but also induces inflation in the inference time, as demonstrated in Table 2. To balance the tradeoff between performance and inference time, we fix the default sampling step to 40 in our main experiments.

Perturbation budget. We also study the impact of the perturbation budget on the quality of the adversarial example generated by AdvDM. We also follow the setting in Section 4.1 except for the perturbation budget. The perturbation budget ϵ is varied from 2/255 to 32/255. The results are shown in Table 3. We observe that with a small perturbation budget (4/255), AdvDM can already significantly affect the quality of generated images. The visualization results are shown in the Appendix B.1.

Table 3. The effects of AdvDM under different perturbation budgets in text-to-image generation scenario

	METRIC		
LSUN AIRPLANE	FID↑	*prec.* ↓	*recall*
NO ATTACK	54.03	0.659	0.242
ϵ=2	54.49	0.295	0.276
ϵ=4	116.79	0.09	0.342
ϵ=8	186.05	0.037	0.464
ϵ=16	217.09	0.015	0.569
ϵ=32	240.30	0.001	0.801

4.5. AdvDM vs. Preprocessing Adversarial Defenses

There is no existing research that specifically discussed the issue of adversarial defense for diffusion models. One poten-

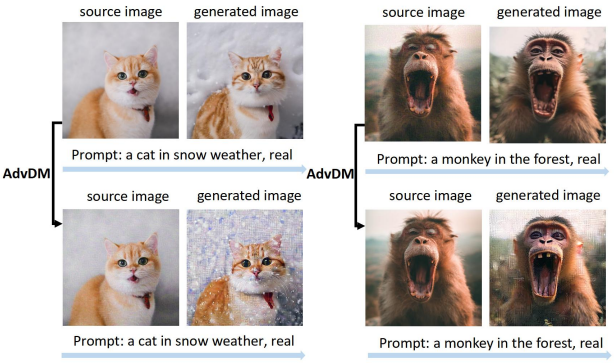
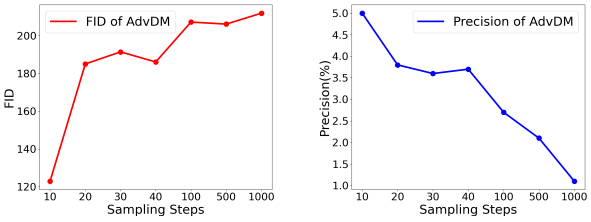


Figure 3. Comparison of images conditionally generated in the image-to-image generation. With conditions extracted from our adversarial examples, LDM generates unrealistic images.



(a) FID-Sampling Steps (b) Prec-Sampling Steps

Figure 4. (a) The FID and sampling steps for AdvDM. (b) The Precision and sampling steps for AdvDM.

tial approach to defending against AdvDM is by exploiting the use of preprocessing adversarial defenses, which focus on eliminating the adversarial perturbations. This is because they do not ask to retrain the generative model or change the architecture of the model. In light of this, we apply JPEG compression (Das et al., 2018), TVM (Guo et al., 2017), and SR (Mustafa et al., 2019) on adversarial examples generated by AdvDM. The experimental setting about AdvDM follows the same in Section 4.1.

The results of AdvDM under preprocessing adversarial defenses are summarized in Table 4. It can be observed that both JPEG and TVM have limited effectiveness against the AdvDM attack. SR shows stronger performance in defending, particularly reflected in FID. However, for the Precision, the effectiveness is not significant. This suggests that while preprocessing defenses can partially defend against AdvDM, they are disabled from fully restoring the semantic information of the original images. Furthermore, the differences between images generated from adversarial examples and clean examples are significant, as shown in Figure 5.

Despite the above results, we also apply DiffPure, a state-of-the-art purification-based adversarial defense, to evaluate the robustness of AdvDM. The experiment is demonstrated in Appendix F.4.

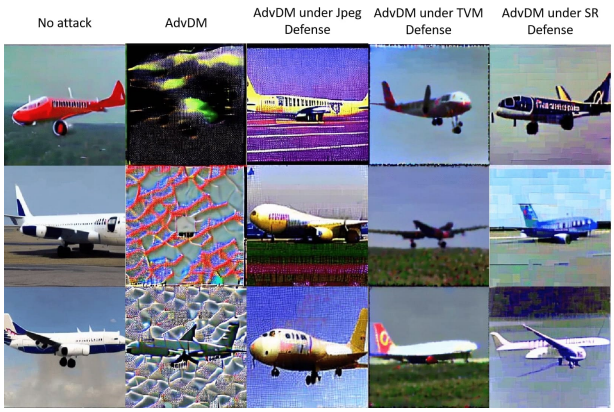


Figure 5. Visualization of conditionally-generated images based on different training images. All defenses cannot perfectly maintain the image quality under AdvDM.

Table 4. Text-to-image generation based on textual inversion with pre-processing-based adversarial defense

DEFENSE	NO DEFENSE			JPEG		
METRIC	FID ↑	*prec.* ↓	*recall*	FID ↑	*prec.* ↓	*recall*
NO ATTACK	39.22	0.5016	0.2765	39.19	0.5098	0.2639
ADVDM	169.67	0.0263	0.3235	61.67	0.1046	0.3208
DEFENSE	TVM			SR		
METRIC	FID ↑	*prec.* ↓	*recall*	FID ↑	*prec.* ↓	*recall*
NO ATTACK	44.21	0.2513	0.1766	32.67	0.3397	0.2332
ADVDM	50.95	0.1744	0.2065	40.88	0.1673	0.2360

5. Related Work

Adversarial examples have long been an essential topic in different scenarios, including the classification of images (Goodfellow et al., 2014b) and graphs (Dai et al., 2018; Zügner et al., 2018), text comprehension (Jia & Liang, 2017), and decision making (Lin et al., 2017). Our definition of the adversarial example for generative models is inspired by that in image classification (Carlini & Wagner, 2017).

Existing research has explored the adversarial example for different generative models yet no proper frameworks have been formulated. Diffusion models are used to improve the adversarial robustness of classifiers (Nie et al., 2022). Kos *et. al.* studied how to make generative models generate images that would be wrongly classified (Kos et al., 2018). A theory of adversarial examples for linear flow-based models (Dinh et al., 2014; 2016; Kingma & Dhariwal, 2018) has been proposed yet held based on a strong assumption that the data distributes normally, which is not realistic (Pope et al., 2020). Another study exploited a surrogate attack on classifiers (Fetaya et al., 2019), which is compared with our method in Appendix C.

6. Conclusion

In this paper, we are the first to explore and present a theoretical framework to define adversarial examples in diffusion models in order to protect human-created artworks. Based on the framework, we propose an algorithm to generate adversarial examples for diffusion models. Extensive experiments demonstrate that our work provides a paradigm for copyright protection against generative AI and a powerful tool for human artists to protect their artworks from being used without authorization by Diffusion Models-based AI-for-Art applications.

Acknowledgements

This research was partly supported by the National NSF of China (NO. 61872234, 61732010), the Shanghai Key Laboratory of Scalable Computing and Systems, and Intel Corporation (UFunding 12679). We extend our heartfelt gratitude to Yichuan Mo and Qingsi Lai from Peking University for their invaluable review and feedback.

Contribution: Chumeng Liang and Xiaoyu Wu are both co-first authors and have made equal contributions to this article. The problem in this paper was initially proposed by Chumeng Liang and Yiming Xue, and refined by Xiaoyu Wu. The algorithm was designed by Xiaoyu and Chumeng. Based on the algorithm, Chumeng formulated the theoretical framework. Xiaoyu then designed and conducted the experiments to evaluate the algorithm.

References

Aufderheide, P. and Jaszi, P. *Reclaiming Fair Use: How to Put Balance Back in Copyright*. University of Chicago Press, 2018.

Baio, A. Invasive Diffusion: How One Unwilling Illustrator Found Herself Turned Into an AI Model. https://waxy.org/2022/11/invasive-diffusion-how-one-unwilling-illustrator-found-herself-turned-into-an-AI-model/, 2022.

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models. *arXiv preprint arXiv:2201.06503*, 2022.

Brock, A., Donahue, J., and Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Carlini, N. and Wagner, D. Towards Evaluating the Robustness of Neural Networks. In *S&P*, 2017.

Carr, S. and Jeffrey, N. Class Action Complaint. *Sarah Anderson, et al., v. Stability AI LTD., et al*, 2022.

Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial Attack on Graph Structured Data. In *ICML*, 2018.

Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Li, S., Chen, L., Kounavis, M. E., and Chau, D. H. Shield: Fast, Practical Defense and Vaccination for Deep Learning Using JPEG Compression. In *SIGKDD*, 2018.

Deck, A. AI-Generated Art Sparks Furious Backlash from Japan's Anime Community. https://restofworld.org/2022/ai-backlash-anime-artists/, 2022.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis. In *NeurIPS*, 2021.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-Linear Independent Components Estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density Estimation Using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

Fetaya, E., Jacobsen, J.-H., Grathwohl, W., and Zemel, R. Understanding the Limitations of Conditional Generative Models. *arXiv preprint arXiv:1906.01171*, 2019.

Fisher III, W. W. Reconstructing the Fair Use Doctrine. *Harvard Law Review*, 101:1659, 1987.

Franceschelli, G. and Musolesi, M. Copyright in Generative Deep Learning. *Data & Policy*, 4:e17, 2022.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An Image Is Worth One Word: Personalizing Text-to-Image Generation Using Textual Inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *NeurIPS*, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Gregor, K., Besse, F., Jimenez Rezende, D., Danihelka, I., and Wierstra, D. Towards Conceptual Compression. In *NeurIPS*, 2016.

Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L. Countering Adversarial Images Using Input Transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans Trained By a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020.

Jia, R. and Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. *arXiv preprint arXiv:1707.07328*, 2017.

Karras, T., Laine, S., and Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of Stylegan. In *CVPR*, 2020.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. Imagic: Text-Based Real Image Editing With Diffusion Models. *arXiv preprint arXiv:2210.09276*, 2022.

Kingma, D. P. and Dhariwal, P. Glow: Generative Flow With Invertible 1x1 Convolutions. In *NeurIPS*, 2018.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

Kos, J., Fischer, I., and Song, D. Adversarial Examples for Generative Models. In *S&P Workshop*, 2018.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved Precision and Recall Metric for Assessing Generative Models. In *NeurIPS*, 2019.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. *arXiv preprint arXiv:1703.06748*, 2017.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *arXiv preprint arXiv:2206.00927*, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, 2014.

MT, D. How AI Art Can Free Artists, Not Replace Them. https://medium.com/thesequence/how-ai-art-can-free-artists-not-replace-them-a23a5cb0461e, 2022.

Mustafa, A., Khan, S. H., Hayat, M., Shen, J., and Shao, L. Image Super-Resolution as a Defense Against Adversarial Attacks. *IEEE Transactions on Image Processing*, 29: 1711–1724, 2019.

Nichol, K. Painter by Numbers, WikiArt. https://www.kaggle.com/c/painter-by-numbers, 2016.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion Models for Adversarial Purification. *arXiv preprint arXiv:2205.07460*, 2022.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in Machine Learning: From Phenomena to Black-Box Attacks Using Adversarial Samples. *arXiv preprint arXiv:1605.07277*, 2016.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3D Using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Pope, P., Balaji, Y., and Feizi, S. Adversarial Robustness of Flow-Based Generative Models. In *AISTATS*, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 2021.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *NeurIPS*, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. *arXiv preprint arXiv:2208.12242*, 2022.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning Internal Representations by Error Propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. Laion-400m: Open Dataset of Clip-Filtered 400 Million Image-Text Pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. *arXiv preprint arXiv:2210.08402*, 2022.

Shlens, J. Notes on Kullback-Leibler Divergence and Likelihood. *arXiv preprint arXiv:1404.2000*, 2014.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. In *ICML*, 2015.

Song, J., Meng, C., and Ermon, S. Denoising Diffusion Implicit Models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-Based Generative Modeling Through Stochastic Differential Equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Sullivan, J. E. Copyright for Visual Art in the Digital Age: A Modern Adventure in Wonderland. *The Journal of Arts Management, Law, and Society*, 26(1):41–59, 1996.

Vincent, J. The Scary Truth About AI Copyright Is Nobody Knows What Will Happen Next. https://www.theverge.com/23444685/generative-ai-copyright-infringement-legal-fair-use-training-data, 2022.

Xia, X., Xu, C., and Nan, B. Inception-V3 for Flower Classification. In *ICIVC*, 2017.

Yang, R., Srivastava, P., and Mandt, S. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *ICCV*, 2017.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial Attacks on Neural Networks for Graph Data. In *SIGKDD*, 2018.

A. Ethical Issues

In this section, we would like to discuss some ethical issues about state-of-the-art AI-for-Art applications based on generative AI and what role our work is expected to play in these issues.

AI-for-Art applications powered by diffusion models have reshaped the art market by significantly lowering the threshold for artistic creation. However, hidden behind such progress are unresolved copyright issues.

Using copyright-protected training data without the consent of image owners may constitute unauthorized reproduction and distribution, thereby giving rise to copyright infringement liability. One primary source of training data is LAION (Schuhmann et al., 2021; 2022), a large-scale dataset of training images with text captions. A large portion of images in LAION was scraped from commercial image-hosting websites without the consent of the image owners (Carr & Jeffrey, 2022). The same issues exist in other generating processes involving unlicensed artworks, for example, learning paintings of a particular artist based on functions of AI-for-Art on a smaller scale without authorization.

Copyright law protects authors' exclusive rights to reproduce, distribute, perform and display the artworks (Franceschelli & Musolesi, 2022). This legal structure makes it highly possible to constitute infringement by using others' artwork without a copyright license in the digital age (Sullivan, 1996). Throughout the AI-for-Art process, the transfer of unauthorized artworks from the platform on which it was originally published to AI's database along with the sale or distribution of the program including such database may constitute reproduction and distribution of the original artwork. This is related to the mechanism AI-for-Art applications created artworks. AI-for-Art applications work by fitting the training images and in turn recombining the learned data to generate new images, which may be understood as a special kind of *reproduction*. For some artworks with distinct well-known features, for example, cartoon figures owned by Disney, this reproduction is easy to detect (Baio, 2022). For this reason, the plaintiff lawyer representing artists whose works were used to train these generative AI tools referred to Diffusion Models as "21st-century collage tools" in the recent lawsuit against several companies profiting from Stable Diffusion (Carr & Jeffrey, 2022).

A possible justification for AI-for-Art applications on these issues is the Fair Use Doctrine (Fisher III, 1987). Examples of fair use include criticism, comment, news reporting, teaching, scholarship, and research. It is very likely that training AI with copyright-protected images constitutes fair use for scientific research purposes. However, the generat-

ing part is not. It is for commercial purposes and has created millions of dollars for those companies. More than that, AI-for-Art applications compete directly with the artists as a substitute, from whom it obtained its training data. All these facts are disadvantageous to the recognition of fair use.

Copyright law is about a balance between the interests of different participants (Aufderheide & Jaszi, 2018), as well as the prospect of human creativity. On one side, researchers have made great efforts to develop AI for Art. Such technology revolutionized the method of artistic expression. On the other side, artists are falling behind for lower speed for production and a far higher cost. It takes time for the law to react to new issues brought about by the development of technology, and we look forward to the court's answers on how the balance will be achieved. But before that, the reality is now severely one-sided – the tech companies make huge money at no cost by appropriating others' intellectual property, while the artists are left to witness the skills they rely on to make a living being significantly devalued by their own works. The method of protection that this paper proposes aims to arm artists with a weapon to legally protect their statutory rights under copyright law. After all, AI needs to be fair for everyone.

B. More Visualization

B.1. Ablation Study

We provide visualization of the generated images based on adversarial examples under different perturbation budgets in Figure 6. With a greater perturbation budget, the figure of airplanes grows vaguer.

B.2. Text-to-image generation based on textual inversion

We compare the adversarial examples with the clean images they are generated from in Figure 7. There are almost no human-visible differences between adversarial examples and clean images. We then generate images with these adversarial examples and clean images by text-to-image generation. The results are shown in Figure 8 and indicates that our adversarial examples severely decrease the quality of generated images.

C. Comparison with Other Potential Adversarial Examples

There is no previous research on adversarial examples for diffusion models. Therefore, in addition to AdvDM, we also investigate several other potential adversarial examples for diffusion models for a complete understanding of AdvDM. Note that there are no formulated methods to generate adversarial examples for diffusion models. We then explore more potential methods inspired by existing research.

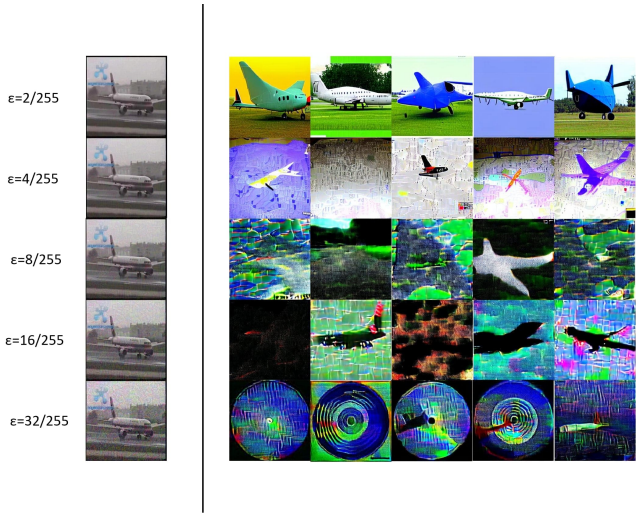


Figure 6. Visualization of ablation study in perturbation budgets. First column: adversarial examples. The second to the sixth column: the image generated conditioned on the pseudo-word derived from adversarial examples for different samplings.

For the experimental setting, the steps and perturbation budgets for these adversarial examples are constrained under the same settings as outlined in Section 4.4.

C.1. PGD on Classifiers

It is shown that some conditional generative models are vulnerable even to some adversarial examples generated for classification models (Fetaya et al., 2019). Also, the transferability of adversarial examples between neural networks is widely validated and exploited (Papernot et al., 2016; Zügner et al., 2018). Following this idea, we generate adversarial examples by Projected Gradient Descent (Madry et al., 2018) (PGD) on an InceptionV3 classifier (Xia et al., 2017). We then consider these adversarial examples to be transferable adversarial examples for LDM. The method is denoted by **PGD (InceptionV3)**.

C.2. Attacking the Embedding Layer

Note that LDM includes an embedding layer that projects images to a representation in the latent space. This can be regarded as an encoder-decoder structure in AutoEncoder (Rumelhart et al., 1985). It is shown by existing research that the encoder-decoder structure can be exploited to generate adversarial examples (Kos et al., 2018). Inspired by this idea, we apply PGD (Madry et al., 2018) to the embedding layer. We compute a new term of loss by comparing the latent representation of the clean image and that of the adversarial example, which is obtained by adding a tiny perturbation δ to the clean image. The optimization goal is

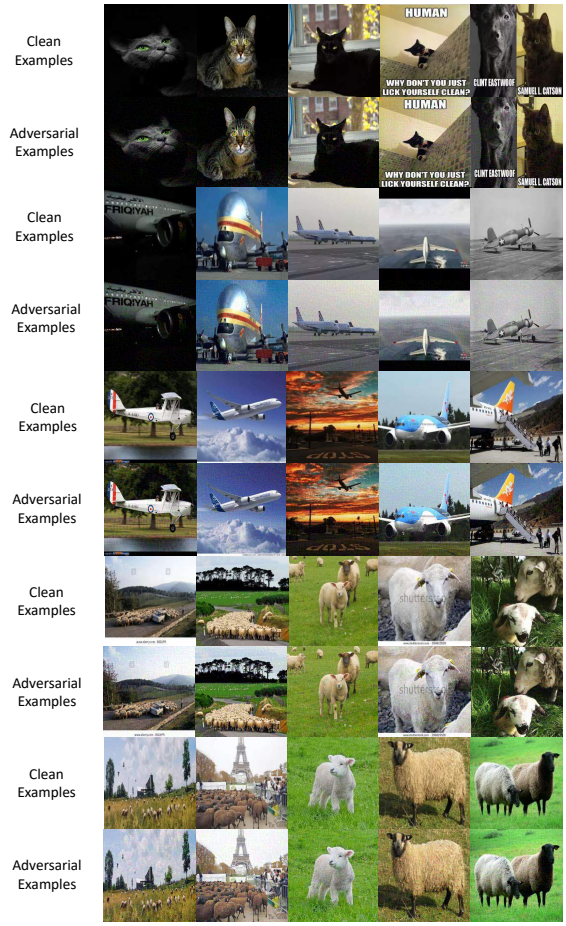


Figure 7. One group of the clean images and adversarial images in Lsun-cat, Lsun-sheep, Lsun-airplane dataset. The top row shows the clean examples while the bottom row shows the adversarial examples under AdvDM.

to maximize the loss by the perturbation. We denote this method by **Embedding Attack**, for it generates adversarial examples by applying an adversarial attack against the embedding layer in LDM.

Definition C.1 (Adversarial Example for Diffusion Models (with Embedding Attack)). Denote the encoder in the LDM by $\mathcal{E}$. x is the input image and δ is the perturbation under a certain budget. The adversarial example generated by Embedding Attack is formulated as $x' := x + \delta$, where

$$\begin{aligned}\delta &:= \arg\max_{\delta} \mathcal{L}_{embedding}(x, \delta) \\ &= \arg\max_{\delta} \|\mathcal{E}(x) - \mathcal{E}(x+\delta)\|_2.\end{aligned} \tag{10}$$

We denote the adversarial example in the optimization step i by $x^{(i)}$. For implementation, we follow the default setting of PGD (Madry et al., 2018) and randomly initialize the perturbation at the beginning of the optimization by $x^{(0)} = x + \epsilon z$, where $z \in \mathcal{N}(0, 1)$ and ϵ is the perturbation budget of the attacks. The adversarial examples are crafted by an iterative multi-step signed gradient ascent with step length α. The number of iteration steps is set to 40. The optimization process is summarized as

$$x^{(i+1)} = x^{(i)} + \alpha \text{sgn}(\nabla_{x^{(i)}} \mathcal{L}_{embedding}(x, x^{(i)} - x)), \tag{11}$$

where sgn refers to the sign function.

C.3. PGD

Another method to generate adversarial examples is to apply PGD to the loss of LDM. This is equivalent to our method when the number of sampling steps N is 1. The method is denoted by **PGD (LDM)**.

Table 5. Text-to-image generation based on textual inversion using adversarial examples under different possible attacks

	METRIC		
	FID↑	*prec.* ↓	*recall.*
NO ATTACK	55.19	0.547	0.231
PGD (INCEPTIONV3)	56.89	0.306	0.153
EMBEDDING ATTACK	175.34	**0.023**	0.352
PGD (LDM)	164.38	0.042	0.438
ADVDM	**186.05**	0.037	0.464

The results of these experiments are presented in Table 5. As can be observed from the table, AdvDM achieves the best results among all the methods benchmarked by FID. Embedding attacks also show relatively promising results, especially in Precision. On the other hand, PGD on DMs, which lacks the sampling process, fails to effectively decrease the probability p_θ, leading to poorer performance. Classifier attacks, which involve transferring an attack on a classifier to the generation model, do not show much effect, indicating that this method is not directly effective in this setting.

We also provide visualization for the generation under different attacks in Figure 9. From the visualization, we observe that under embedding attacks, while noise is created in the background of the generated images, the semantic information of the images is not largely destroyed. However, under AdvDM, the semantic information (such as the shape or the color) of the images is largely affected, indicating a stronger attacking effect.

D. Implementation Details

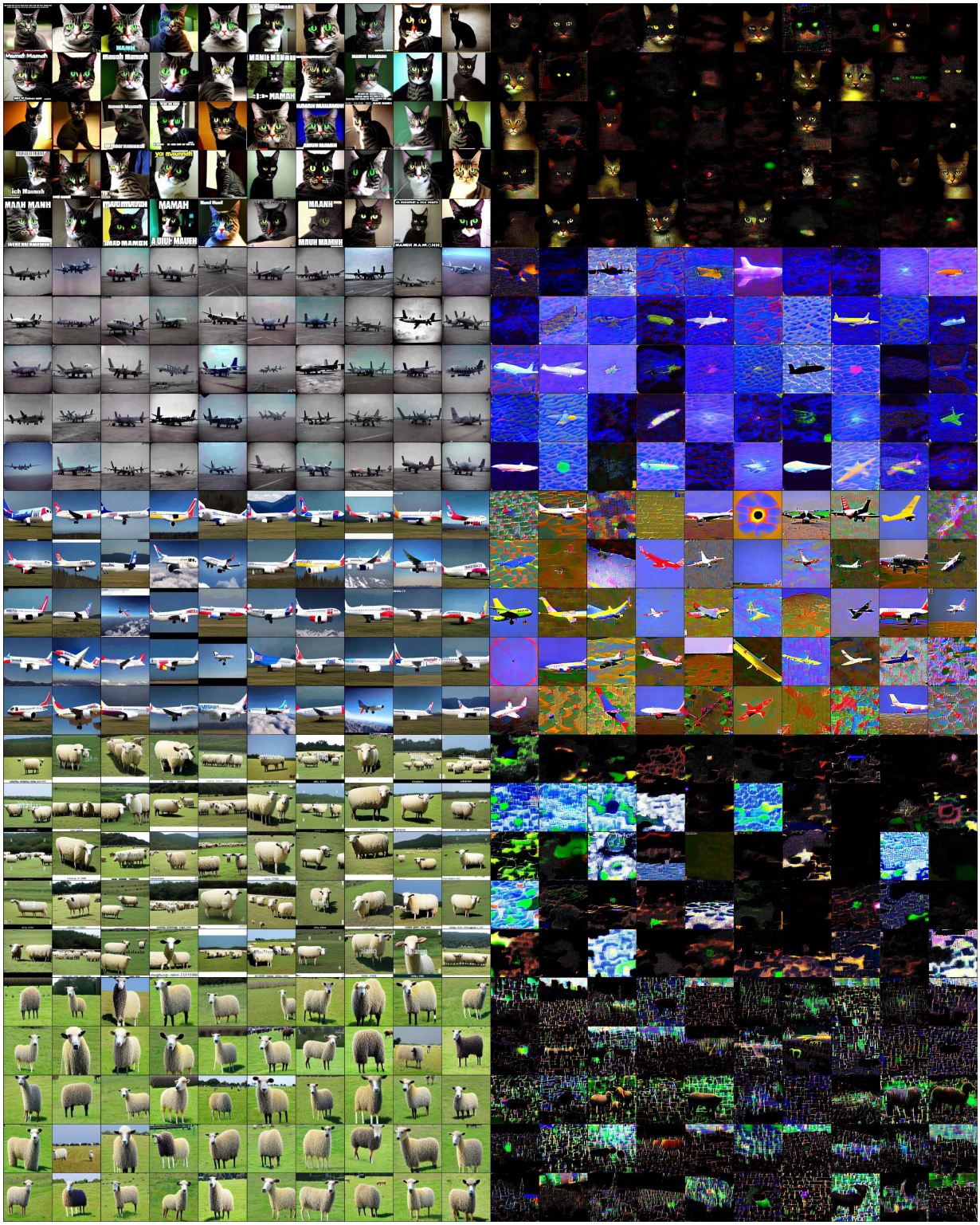


Figure 8. Left: Text-to-image generation based on textual inversion for clean examples. Right: Text-to-image generation based on textual inversion for adversarial examples.

Algorithm 3 Text-to-image generation based on textual inversion

Input: Dataset $\mathcal{D}$, latent diffusion model θ with latent encoder ε_θ and embedded language model $\mathcal{F}_\theta$, optimization steps N, step length α
Randomly pick 1,000 images X from $\mathcal{D}$
Separate X into 5-image groups $X := \{x^0, x^1, x^2, ...\}$
Initialize $X_g \leftarrow \emptyset$
for x^i **in** X **do**
  Initialize S^* with a random word vector in the semantic space of $\mathcal{F}_\theta$
  Optimizing a pseudo word S^*:
  for $j = 1$ **to** N **do**
    Randomly pick an image x from x^i
    $c \leftarrow$ "A photo of S^*"
    Compute $\mathcal{L}_j = \mathbb{E}_{z\sim\varepsilon_\theta(x^i),c,t}||\epsilon_t - \epsilon_\theta(z, t, \mathcal{F}_\theta(c))||_2^2$
    $S^* \leftarrow S^* - \alpha\nabla_{S^*}\mathcal{L}_j$
  end for
  $c_g \leftarrow$ "A photo of S^*"
  Use c_g to generate 50 images x_g^i by the text-to-image function of θ
  $X_g \leftarrow X_g \cup x_g^i$
end for
Compute FID, $prec$, and $recall$ between X_g and $\mathcal{D}$

Algorithm 4 Style transfer based on textual inversion

Input: Dataset $\mathcal{D}$, latent diffusion model θ with latent encoder ε_θ and embedded language model $\mathcal{F}_\theta$, optimization steps N, step length α
Separate D based on different styles: $D^{'} := \{x^0, x^1, x^2, ..., x^k\}$.
Initialize $X_g \leftarrow \emptyset$
for x^i **in** $D^{'}$ **do**
  Initialize S^* with a random word vector in the semantic space of $\mathcal{F}_\theta$
  Optimizing a pseudo word S^*:
  for $j = 1$ **to** N **do**
    Randomly pick an image x from g^i
    $c \leftarrow$ "A painting in the style of S^*"
    Compute $\mathcal{L}_j = \mathbb{E}_{z\sim\varepsilon_\theta(x^i),c,t}||\epsilon_t - \epsilon_\theta(z, t, \mathcal{F}_\theta(c))||_2^2$
    $S^* \leftarrow S^* - \alpha\nabla_{S^*}\mathcal{L}_j$
  end for
  $c_g \leftarrow$ A target prompt containing "in the style of S^*"
  $\overline{x} \leftarrow$ A photo or script
  Use $c_g, \overline{x}$ to generate images x_g^i by the image-to-image function of θ
  $X_g \leftarrow X_g \cup x_g^i$
end for
Return X_g

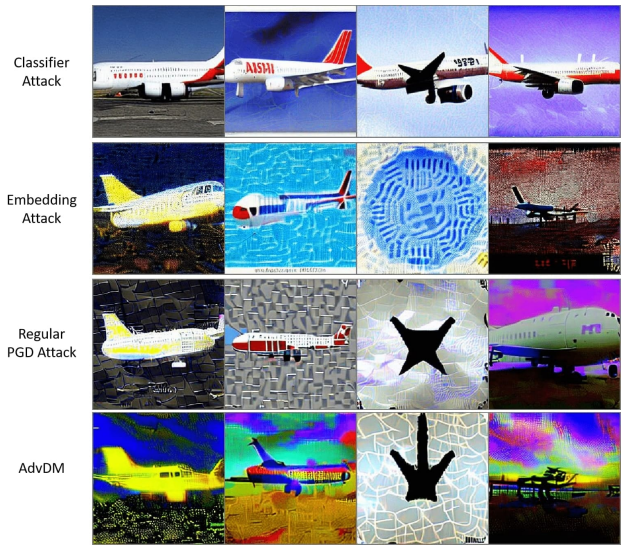


Figure 9. Visualization of generated images under different attacks. The pseudo-word used for generation is derived from Lsun-airplane dataset, where we can observe that AdvDM can more effectively influence the semantic information of the images.

Algorithm 5 Image-to-image synthesis

Input: Source Images X, latent diffusion model θ with latent encoder ε_θ and embedded language model $\mathcal{F}_\theta$
Initialize $X_g \leftarrow \emptyset$
for x **in** X **do**
  $c \leftarrow$ A target prompt for image x
  Use c, x to generate images x_g by the image-to-image function of θ
  $X_g \leftarrow X_g \cup x_g$
end for
Return X_g

D.1. Details of Evaluation

We describe the detailed procedure of the three evaluation scenarios in Algorithm 3, Algorithm 4, and Algorithm 5, respectively. We use the pre-trained model provided by the author of the latent diffusion model (Rombach et al., 2022). For text-to-image generation and style transfer, the procedure follows the setting recommended by the textual inversion paper (Gal et al., 2022). For style transfer, we fix the strength to 0.5. We also follow the paper to choose N as 5000. For image-to-image, we follow the default setting in (Rombach et al., 2022).

For FID scores, we use an open-source package [4]. For Precision- and Recall- scores, we use the script provided by Dhariwal and Nichol (Dhariwal & Nichol, 2021). Three metrics are calculated over the whole category dataset.

[4]https://github.com/w86763777/pytorch-gan-metrics

D.2. Implementation of AdvDM

Algorithm 6 Implementation of AdvDM on Latent Diffusion Models

Input: Data x_0, parameter θ, denoising autoencoder ε_θ, encoder $\mathcal{E}$, number of Monte Carlo N, step-wise perturbation budget α, overall perturbation budget ϵ
Output: Adversarial example x_0'
Initialize $x_0^{(0)} \leftarrow x_0$.
for $i = 1$ **to** N **do**
  Sample $x_{1:T}^{(i)} \sim q(\mathcal{E}(x_{1:T}^{(i)})|\mathcal{E}(x_0^{(i)}))$
  Sample $t \sim U(1, T)$
  $\delta^{(i)} \leftarrow \alpha \text{sgn}(\nabla_{x_0^{(i)}} \|\mathcal{E}(x_T^{(i)}) - \varepsilon_\theta(\mathcal{E}(x_t^{(i)}), t)\|_2)$
  Clip $\delta^{(i)}$ s.t. $\|x_0^{(i-1)} + \delta^{(i)} - x_0^{(0)}\|_\infty \leq \epsilon$
  $x_0^{(i)} \leftarrow x_0^{(i-1)} + \delta^{(i)}$
end for
$x_0' \leftarrow x_0^{(N)}$

We implement AdvDM on Latent Diffusion Models. As shown in Algorithm 6, adversarial perturbation is added to the original image under a sampling series $x_{1:T}^{(i)}$ and a random timestamp t for each step.

E. Protection Effectiveness against Stable Diffusion

Our motivation is to protect paintings created by human artists from being imitated by AI-for-Art applications. Note that various mainstream AI-for-Art applications [5] [6] [7] [8] use the model with the architecture of LDM. Hence, we expect satisfying protection effectiveness of our adversarial examples against these AI-for-Art applications. Here, we conduct an experiment to evaluate the protection effectiveness against Stable Diffusion, a famous AI-for-Art application. Note that the model [9] used by Stable Diffusion has a similar architecture as LDM [10] but it has a larger scale with more parameters.

We evaluate our adversarial examples on the WikiArt dataset (Nichol, 2016). We select 20 paintings from three artists respectively: Vincent Van Gogh, Pablo Picasso, and Henri Matisse. We then generate adversarial examples based on these paintings. The number of sampling steps N is set to 100. The perturbation budget ϵ is 8/255 and the step length α is 1/255. Then, we do style transfer with textual inversion on Stable Diffusion. The procedure is very similar to that described by Algorithm 4. For the optimization of the pseudo word S^*, the optimization step is 8000 with a step length of 0.005. The reconstruction strength is set to 0.5. We first compare the clean paintings used for optimizing S^* with the adversarial examples in Figure 10, Figure 11, and Figure 12. Then, we visualize the results of generated images on clean paintings and adversarial examples in Figure 13, Figure 14, and Figure 15.

[5] Text2Dream: https://deepdreamgenerator.com/#tools
[6] Night Cafe: https://creator.nightcafe.studio/stable-diffusion-image-generator
[7] Hotpot: https://hotpot.ai/stable-diffusion
[8] NovelAI: https://novelai.net/
[9] https://huggingface.co/CompVis/stable-diffusion-v-1-4-original
[10] https://ommer-lab.com/files/latent-diffusion/nitro/txt2img-f8-large/model.ckpt

Figure 10. Clean examples and adversarial examples of Henri Matisse's paintings.

Figure 11. Clean examples and adversarial examples of Pablo Picasso's paintings.

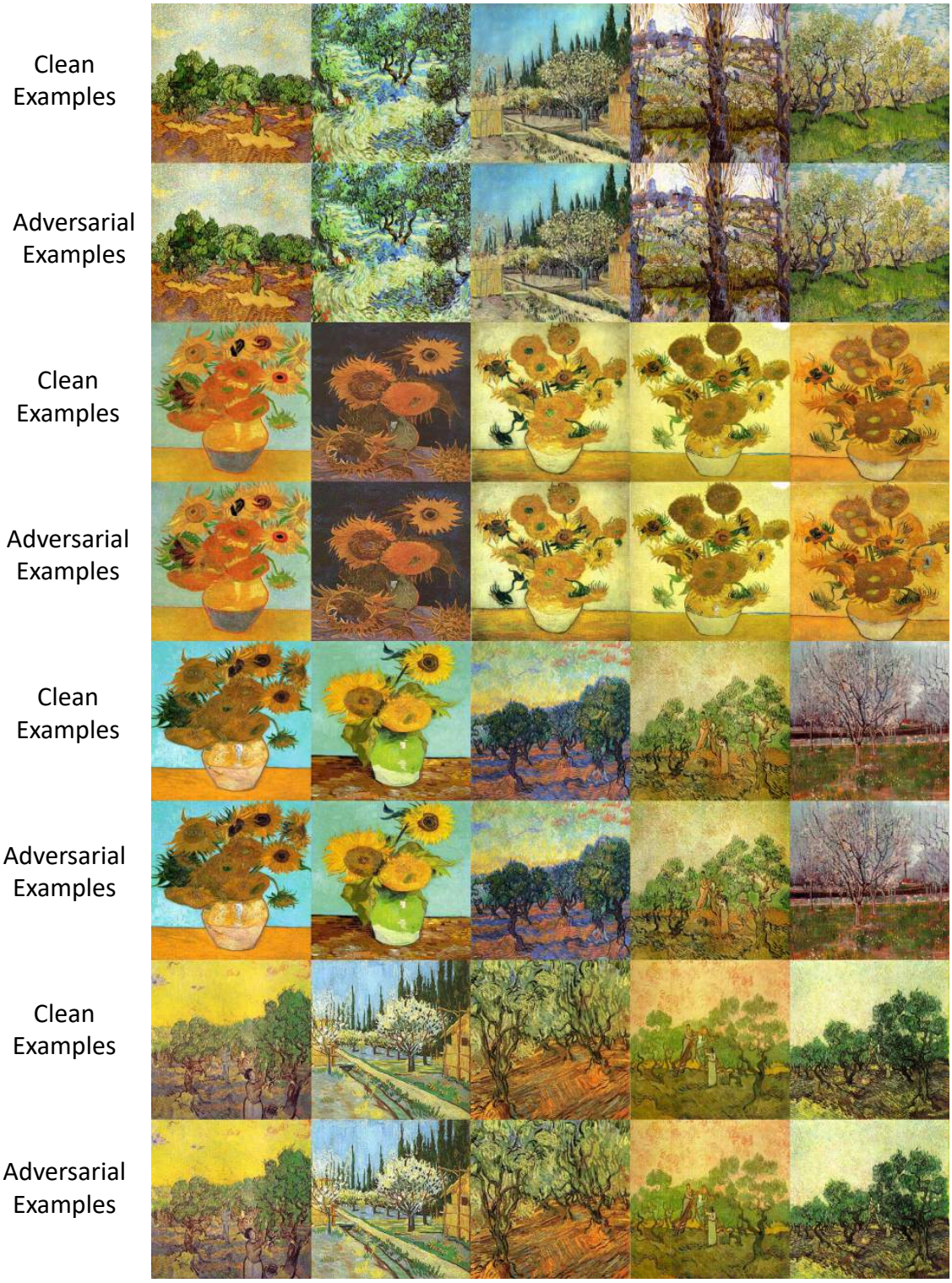


Figure 12. Clean examples and adversarial examples of Vincent Van Gogh's paintings.

Figure 13. **The first row:** Generated images by Stable Diffusion based on clean examples of Henri Matisse's paintings. **The second row:** Generated images by Stable Diffusion based on adversarial examples of Henri Matisse's paintings.

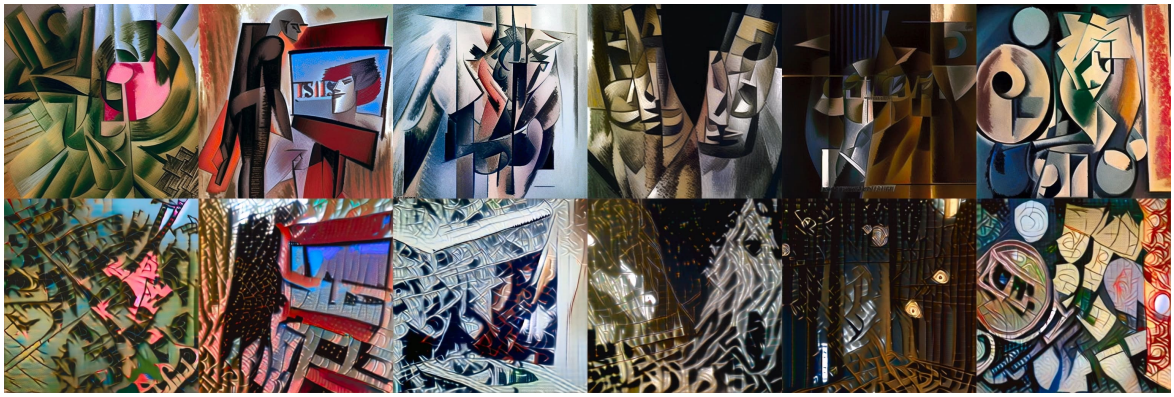


Figure 14. **The first row:** Generated images by Stable Diffusion based on clean examples of Pablo Picasso's paintings. **The second row:** Generated images by Stable Diffusion based on adversarial examples of Pablo Picasso's paintings.

Figure 15. **The first row:** Generated images by Stable Diffusion based on clean examples of Van Gogh's paintings. **The second row:** Generated images by Stable Diffusion based on adversarial examples of Vincent Van Gogh's paintings.

F. Additional Experiments

F.1. AdvDM on other image editing tasks

In this part, we investigate how AdvDM works on image editing tasks other than textual inversion and image-to-image synthesis. We consider two tasks: inpainting, and pose-guided synthesizing. To show the performance of AdvDM in commercial AI-for-Art applications, we select Stable Diffusion 1.5 [11] as the backbone to generate and evaluate adversarial examples. Other experimental setups stay consistent with the setup stated in Section 4.

Figure 16 and Figure 17 visualize a case of inpainting and pose-guided synthesis, respectively. The visualization shows that when images are generated based on our adversarial examples, they suffer a bad image quality which makes them not usable. Specifically, the content of the generated image would lose basic structure, show strange artifacts, or be oversimplified.

F.2. AdvDM on Dreambooth

In this part, we investigate the performance of AdvDM on Dreambooth, another subject-driven generation method that can be used for art style transfer. We generate adversarial examples with AdvDM on Stable Diffusion 1.5 and evaluate these adversarial examples by conducting style transfer over them with Dreambooth. We use the implementation by the Python library *diffuser* [12]. We pick the learning rate as 5×10^{-6} and the number of steps as 4000. Other experimental setups stay consistent with the setup stated in Section 4.

Figure 18 shows a comparison case that tried to mimic the art style of Van Gogh with 20 paintings, the same as the setup stated in Section 4.2. We conduct the mimicry on two groups of images: one group consists of clean paintings and the other consists of adversarial examples based on these clean paintings. The results show that our adversarial examples add chaotic textures to the generated images and thus make the generated images not usable.

F.3. AdvDM's transferability on scenario.gg

As a commercial AI-for-Art application that supports art style transfer other than Stable Diffusion, scenario.gg [13] also raises concerns of copyright violation. We conduct experiments to explore whether our adversarial examples can be transferable to scenario.gg. Since scenario.gg is driven by closed-source diffusion models, this experiment aims to investigate the transferability of adversarial examples generated by AdvDM in a black-box adversarial attack setting. For the generation of adversarial examples, the experimental setups stay consistent with the setup stated in Section 4.

The results are shown in Figure 19 and Figure 20. Compared to the generated images based on clean images, AdvDM adds chaotic textures to the generated images based on the adversarial examples, though the effect is not as strong as the experiment in the white-box setting. However, the chaotic textures can still make the generated images not usable, which achieves the practical goal of our method.

F.4. AdvDM against defense: SR and DiffPure

One main concern of AdvDM is that its strength may be greatly reduced by the preprocessing-based adversarial defense. In this part, we conduct experiments to illustrate the effectiveness of AdvDM under two state-of-the-art adversarial defenses: SR and DiffPure. To fit the real black box scenario where our method would be applied more, we choose scenario.gg, a commercial AI-for-Art application specific for art style transfer, as the backbone to evaluate the performance of adversarial examples. This can better validate the performance of AdvDM since it is exactly a transfer-learning scenario, as aforementioned in F.3. For SR, we follow the setup stated in Section 4.5. For DiffPure, we utilize the original implementation of DiffPure provided by the author [14]. Note that DiffPure is a model-based noise purification and its effect therefore highly depends on the used model. In the official implementation, the author provides three models, which are trained on Cifar-10, ImageNet, and CelebA-HQ with the image resolution of 32×32, 224×224, and 224×224, respectively. In this experiment, we choose the model trained on ImageNet, for it has a high resolution and the content of the dataset is relatively similar to the content of paintings used in our experiments. All the setups stay as the default setting of the official implementation of DiffPure.

Figure 21 visualizes the results. Both SR and DiffPure are not able to prevent our adversarial examples from adding chaotic textures to the generated images. Specifically, the generated images based on clean examples with DiffPure are also of low quality. This is because the resolution of output images in DiffPure is limited and output images suffer from a reduction in image quality during the process of noise purification.

[11] https://huggingface.co/runwayml/stable-diffusion-v1-5/tree/main

[12] https://github.com/huggingface/diffusers/

[13] https://app.scenario.gg/

[14] https://github.com/NVlabs/DiffPure

Figure 16. **Left:** The source image and the mask used for inpainting. **Right top:** The generated images based on clean images. **Right down:** The generated images based on adversarial examples. The inpainting district loses some basic structure.

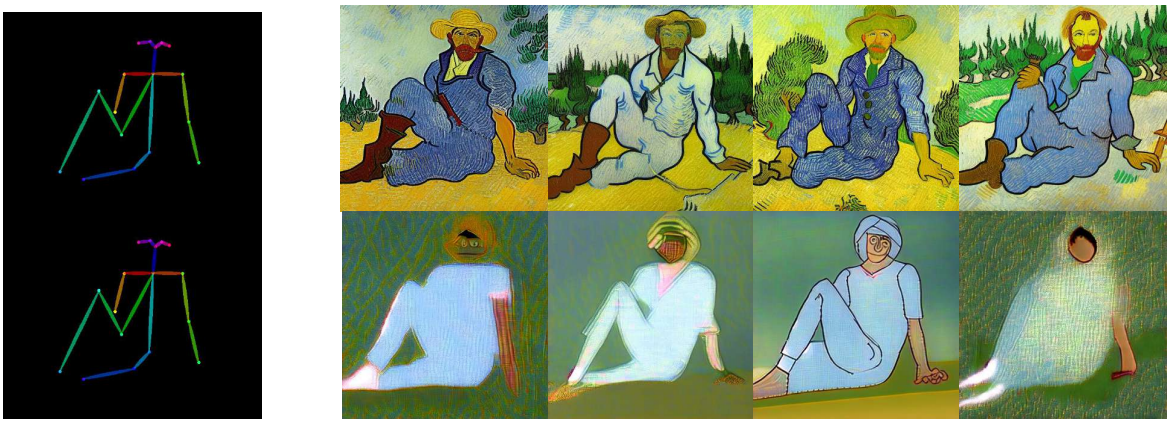


Figure 17. **Left:** The pose used for generation. **Right top:** The generated images based on clean images. **Right down:** The generated images based on adversarial examples. Generated images based on adversarial examples lose the feature of the art style.

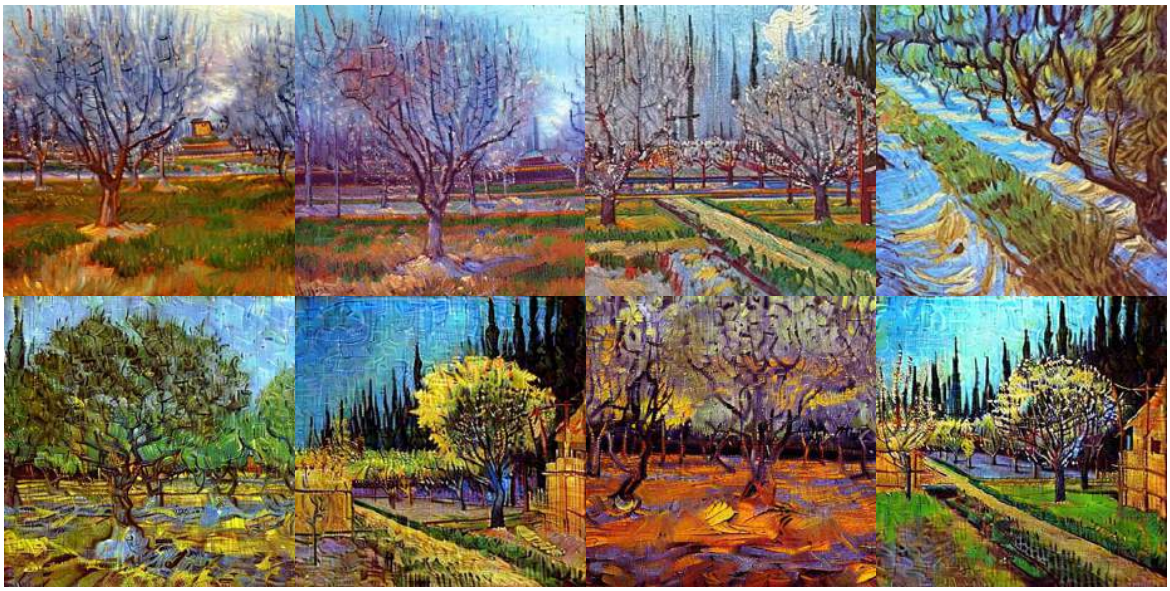


Figure 18. **Top:** The generated images based on Stable Diffusion trained by DreamBooth using clean images. **Down:** The generated images based on Stable Diffusion trained by DreamBooth using adversarial examples with chaotic textures.

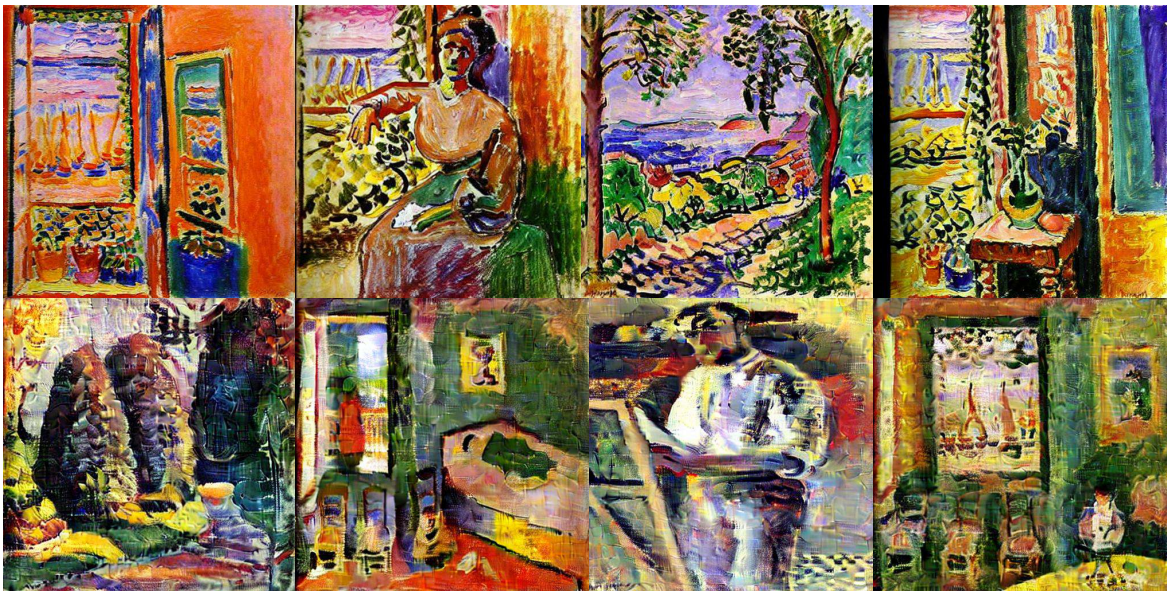


Figure 19. **The first row:** Generated images based on clean examples of Henri Matisse's paintings on commercial AI-for-art websites scenario.gg. **The second row:** Generated images based on adversarial examples of Henri Matisse's paintings on scenario.gg. There are chaotic textures on the generated images based on adversarial examples.

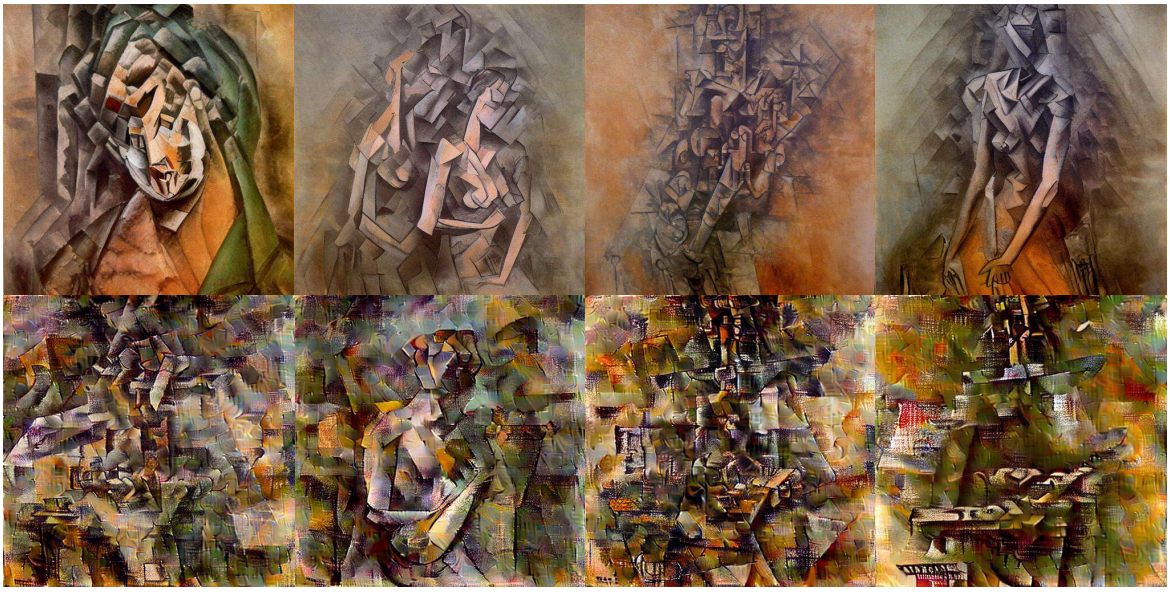


Figure 20. **The first row:** Generated images based on clean examples of Pablo Picasso's paintings on commercial AI-for-art websites scenario.gg. **The second row:** Generated images based on adversarial examples of Pablo Picasso's paintings on scenario.gg. There are chaotic textures on the generated images based on adversarial examples.

Figure 21. Generated images based on clean and adversarial examples of Van Gogh's paintings with SR and DiffPure defenses by scenario.gg. **The first row:** clean-example based with no defense. **The second row:** adversarial-example based with no defense. **The third row:** adversarial-example based with SR. **The fourth row:** clean-example based with SR. **The fifth row:** adversarial-example based with DiffPure. Both defenses are not able to purify enough adversarial perturbation so that the generated images are still of low quality.