Anti-DreamBooth: Protecting users from personalized text-to-image synthesis

Thanh Van Le∗1, Hao Phung∗1, Thuan Hoang Nguyen∗1, Quan Dao∗1, Ngoc N. Tran†2, Anh Tran1

1VinAI Research 2Vanderbilt University

v.{thanhlv19, haopt12, thuannah5, quandm7, anht152}@vinai.io, ngoc.n.tran@vanderbilt.edu

Abstract

Text-to-image diffusion models are nothing but a revolution, allowing anyone, even without design skills, to create realistic images from simple text inputs. With powerful personalization tools like DreamBooth, they can generate images of a specific person just by learning from his/her few reference images. However, when misused, such a powerful and convenient tool can produce fake news or disturbing content targeting any individual victim, posing a severe negative social impact. In this paper, we explore a defense system called Anti-DreamBooth against such malicious use of DreamBooth. The system aims to add subtle noise perturbation to each user’s image before publishing in order to disrupt the generation quality of any DreamBooth model trained on these perturbed images. We investigate a wide range of algorithms for perturbation optimization and extensively evaluate them on two facial datasets over various text-to-image model versions. Despite the complicated formulation of DreamBooth and Diffusion-based text-to-image models, our methods effectively defend users from the malicious use of those models. Their effectiveness withstands even adverse conditions, such as model or prompt/term mismatching between training and testing. Our code will be available at https://github.com/VinAIResearch/Anti-DreamBooth.git.

1. Introduction

Within a few years, denoising diffusion models [23, 54, 43] have revolutionized image generation studies, allowing producing images with realistic quality and diverse content [19]. They especially succeed when being combined with language [40] or vision-language models [39] for text-to-image generation. Large models [41, 3, 43, 47, 9] can produce photo-realistic or artistic images just from simple text description inputs. A user can now generate art within a few seconds, and a generated drawing even beat professional artists in an art competition [44]. Photo-realistic synthetic images can be hard to distinguish from real photos [27]. Besides, ControlNet [62] offers extra options to control the generation outputs, further boosting the power of the text-to-image models and bringing them closer to mass users.

One extremely useful feature for image generation models is personalization, which allows the models to generate images of a specific subject, given a few reference examples. For instance, one can create images of himself/herself in a fantasy world for fun, or create images of his/her family members as a gift. Textual Inversion [20] and DreamBooth [46] are two prominent techniques that offer that impressive ability. While Textual Inversion only optimizes the text embedding inputs representing the target subject, DreamBooth finetunes the text-to-image model itself for better personalization quality. Hence, DreamBooth is particularly popular and has become the core technique in many applications.

While the mentioned techniques provide a powerful and
convenient tool for producing desirable images at will, they also pose a severe risk of being misused. A malicious user can propagate fake news with photo-realistic images of a celebrity generated by DreamBooth. This can be classified as DeepFakes [28], one of the most serious AI crime threats that has drawn an enormous attention from the media and community in recent years. Besides creating fake news, DreamBooth can be used to issue harmful images targeting specific persons, disrupting their lives and reputations. While the threat of GAN-based DeepFakes techniques is well-known and has drawn much research interest, the danger from DreamBooth has yet to be aware by the community, making its damage, when happening, more dreadful.

This paper discusses how to protect users from malicious personalized text-to-image synthesis. Inspired by DeepFakes’s prevention studies [60, 45, 59, 26, 58], we propose to pro-actively defend each user from the DreamBooth threat by injecting subtle adversarial noise into their images before publishing. The noise is designed so that any DreamBooth model trained on these perturbed images fails to produce reasonable-quality images of the target subject. While the proposed mechanism, called Anti-DreamBooth, shares the same goal and objective as the techniques to disrupt GAN-based DeepFakes, it has a different nature due to the complex formulation of diffusion-based text-to-image models and DreamBooth:

- In GAN-based disruption techniques, the defender optimizes the adversarial noise of a single image, targeting a fixed DeepFakes generator. In Anti-DreamBooth, we have to optimize the perturbation noise to disrupt a dynamic, unknown generator that is finetuned from the perturbed images themselves.

- GAN-based DeepFakes generator produces each fake image via a single forward step; hence, adversarial noise can be easily learned based on the model’s gradient. In contrast, a diffusion-based generator produces each output image via a series of non-deterministic denoising steps, making it impossible to compute the end-to-end gradient for optimization.

- Anti-DreamBooth has a more complex setting by considering many distinctive factors, such as the prompt used in training and inference, the text-to-image model structure and pre-trained weights, and more.

Despite the complexity mentioned above, we show that the DreamBooth threat can be effectively prevented. Instead of targeting the end-to-end image generation process, we can adapt the adversarial learning process to break each diffusion sampling step. We design different algorithms for adversarial noise generation, and verify their effectiveness in defending DreamBooth attack on two facial benchmarks. Our proposed algorithms successfully break all DreamBooth attempts in the controlled settings, causing the generated images to have prominent visual artifacts. Our proposed defense shows consistent effect when using different text-to-image models and different training text prompts. More impressively, Anti-DreamBooth maintains its efficiency even under adverse conditions, such as model or prompt/term mismatching between training and testing.

In summary, our contributions include: (1) We discuss the potential negative impact of personalized text-to-image synthesis, particularly with DreamBooth, and define a new task of defending users from this critical risk. (2) We propose proactively protecting users from the threat by adding adversarial noise to their images before publishing. (3) We design different algorithms for adversarial noise generation, adapting to the step-based diffusion process and finetuning-based DreamBooth procedure, (4) We extensively evaluate our proposed methods on two facial benchmarks and under different configurations. Our best defense works effectively in both convenient and adverse settings.

2. Related work

2.1. Text-to-image generation models

Due to the advent of new large-scale training datasets such as LAION5B [49], text-to-image generative models are advancing rapidly, opening new doors in many visual-based applications and attracting attention from the public. These models can be grouped into four main categories: auto-regressive [61], mask-prediction [13], GAN-based [48] and diffusion-based approaches, all of which show astounding qualitative and quantitative results. Among these methods, diffusion-based models [43, 47, 9, 38, 42] have exhibited an exceptional capacity for generating high-quality and easily modifiable images, leading to their widespread adoption in text-to-image synthesis. GLIDE [38] is arguably the first to combine a diffusion model with classifier guidance for text-to-image generation. DALL-E 2 [42] then improves the quality further using the CLIP text encoder and diffusion-based prior. For better trade-off between efficiency and fidelity, following-up works either introduce coarse-to-fine generation process like Imagen [47] and eDiff-I [9] or work on latent space like LDM [43]. StableDiffusion [4], primarily based on LDM, is the first open-source large model of this type, further boosting the widespread applications of text-to-image synthesis.

2.2. Personalization

Customizing the model’s outputs for a particular person or object has been a significant aim in the machine-learning community for a long time. Generally, personalized models are commonly observed in recommendation systems [5] or federated learning [50, 22]. Within the context of diffusion models, previous research has focused on adapting a pre-
trained model to create fresh images based on a particular target idea using natural language cues. Existing methods for personalizing the model either involve adjusting a collection of text embeddings to describe the idea [20] or fine-tuning the denoising network to connect a less commonly used word-embedding to the novel concept [46]. For better customization, [30] propose a novel approach to not only model a new concept of an individual subject by optimizing a small set of parameters of cross-attention layers but also to combine multiple concepts of objects via joint training. Among these tools, DreamBooth [46] is particularly popular due to its exceptional quality and has become the core technique in many applications. Hence, we focus on defending the malign use of this technique.

2.3. Adversarial attacks

With the introduction of the Fast Gradient Sign Method (FGSM) attack [21], adversarial vulnerability has become an active field of research in machine learning. The goal of adversarial attacks is to generate a model input that can induce a misclassification while remaining visually indistinguishable from a clean one. Following this foundational work, different attacks with different approaches started to emerge, with more notable ones including: [31, 35] being FGSM’s iterative versions, [12] limiting the adversarial perturbation’s magnitude implicitly using regularization instead of projection, [37] searching for a close-by decision boundary to cross, etc. For black-box attacks, where the adversary does not have full access to the model weights and gradients, [56] estimates the gradient using sampling methods, while [10, 14, 6] aim to synthesize a close-by example by searching for the classification boundary, then finding the direction to traverse towards a good adversarial example. Combining them, [16] is an ensemble of various attacks that are commonly used as a benchmark metric, being able to break through gradient obfuscation [7] with the expectation-over-transformation technique [8].

2.4. User protection with image cloaking

With the rapid development of AI models, their misuse risk has emerged and become critical. Particularly, many models exploit the public images of each individual for malicious purposes. Instead of passively detecting and mitigating these malign actions, many studies propose proactively preventing them from succeeding. The idea is to add subtle noise into users’ images before publishing to disrupt any attempt to exploit those images. This approach is called “image cloaking”, which our proposed methods belong to.

One application of image cloaking is to prevent privacy violations caused by unauthorized face recognition systems. Fawkes [52] applies targeted attacks to shift the user’s identity towards a different reference person in the embedding space. Although it learns the adversarial noise using a surrogate face recognition model, the noise successfully transfers to break other black-box recognizers. Lowkey [15] further improves the transferability by using an ensemble of surrogate models. It also considers a Gaussian smoothed version of the perturbed image in optimization, improving robustness against different image transformations. AMT-GAN [25] crafts a natural-looking cloak as makeup, while OPOM [63] optimizes person-specific universal privacy masks.

Another important application of image cloaking is to disrupt GAN-based image manipulation for DeepFakes. Yang et al. [59] exploits differentiable image transformations for robust image cloaking. Yeh et al. [60] defines new effective objective functions to nullify or distort the image manipulation. Huang et al. [26] addresses personalized DeepFakes techniques by alternating the training of the surrogate model and a perturbation generator. Anti-Forgery [58] crafts the perturbation for channels a and b in the Lab color space, aiming for natural-looking and robust cloaking. Lately, UnGANable [32] prevents StyleGAN-based image manipulation by breaking its inversion process.

Similar to our goals, two concurrent works, GLAZE [51] and AdvDM [33], aim to protect against personalized text-to-image diffusion models exploited without consent using image cloaking. However, our work differs from theirs in several key aspects. First, GLAZE and AdvDM focus on disrupting artistic mimicry, while Anti-DreamBooth concentrates on preventing the negative impacts of generating fake or harmful personalized images. Second, GLAZE utilizes complex style-transferred guidance during noise optimization, which is difficult to adapt to the user protection setting. AdvDM only focuses on a simpler technique Textual Inversion [20] where the generator is fixed, unlike the challenging finetuning of DreamBooth in our setting.

3. Problem

3.1. Background

Adversarial attacks. The goal of adversarial attacks is to find an imperceptible perturbation of an input image to mislead the behavior of given models. Typical works have been developed for classification problems where for a model \( f \), an adversarial example \( x' \) of an input image \( x \) is generated to stay visually undetectable while inducing a misclassification \( y_{\text{true}} \neq f(x') \) (untargeted attack), or making the model predict a predefined target label \( y_{\text{target}} = f(x_{\text{adv}}) \neq y_{\text{true}} \) (targeted attack). The minimal visual difference is usually enforced by bounding the perturbation to be within an \( \eta \)-ball w.r.t. an \( \ell_p \) metrics, that is \( \| x' - x \|_p < \eta \). To achieve this objective, denoting \( \Delta = \{ \delta : \| \delta \|_p \leq \eta \} \), we find the optimal perturbation \( \delta \) to maximize the classification loss in the untargeted version:

\[
\delta_{\text{adv}} = \arg \max_{\delta \in \Delta} \mathcal{L}(f(x + \delta), y_{\text{true}}), \tag{1}
\]
or to minimize the loss for the targeted variant:

$$\delta_{adv} = \arg \min_{\delta \in \Delta} \mathcal{L}(f(x + \delta), y_{target}).$$

Projected Gradient Descent (PGD) [35] is a commonly used attack based on an iterative optimization process. The updating pipeline to predict $x'$ for untargeted attack is:

$$\begin{align*}
x'_0 &= x \\
x'_k &= \Pi_{x, \eta}(x'_k - \alpha \cdot \text{sgn}(\nabla_x \mathcal{L}(f(x'_k - \eta, y_{true})))) \tag{3}
\end{align*}$$

where $\Pi_{x, \eta}(z)$ restraints pixel values of $z$ within an $\eta$-ball around the original values in $x$. We acquire the adversarial example $x'$ after a pre-defined number of iterations.

**Diffusion models** are a type of generative models [53, 23] that decouple the role of generation into two opposing procedures: a forward process and a backward process. While the forward process gradually adds noise to an input image until data distribution becomes pure Gaussian noise, the latter learns to reverse that process to obtain the desired data from random noise. Given input image $x_0 \sim q(x)$, the diffusion process perturbs the data distribution with a noise scheduler $\{\beta_t : \beta_t \in (0, 1)\}_{t=0}^{T}$ producing increasing levels of noise addition through $T$ steps to obtain a sequence of noisy variables: $\{x_1, x_2, \ldots, x_T\}$. Each variable $x_t$ is constructed via injecting noise at corresponding timestep $t$:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon \tag{4}$$

where $\alpha_t = 1 - \beta_t$, $\alpha_t = \prod_{s=1}^{t} \alpha_s$ and $\epsilon \sim \mathcal{N}(0, 1)$.

The backward process learns to denoise from noisy variable $x_{t+1}$ to less-noisy variable $x_t$ via simply estimating the injected noise $\epsilon$ with a parametric neural network $\epsilon_\theta(x_{t+1}, t)$. The denoising process is trained to minimize $\ell_2$ distance between estimated noise and true noise:

$$\mathcal{L}_{\text{unc}}(\theta, x_0) = \mathbb{E}_{x_0, t, \epsilon \in \mathcal{N}(0,1)} \|\epsilon - \epsilon_\theta(x_{t+1}, t)\|_2^2 \tag{5}$$

where $t$ is uniformly samples within $\{1, \ldots, T\}$.

**Prompt-based Diffusion Models.** Unlike unconditional diffusion models, prompt-based diffusion models control the sampling process with an additional prompt $c$ to generate photo-realistic outputs which are well-aligned with the text description. The objective is formulated as follows:

$$\mathcal{L}_{\text{cond}}(\theta, x_0) = \mathbb{E}_{x_0, t, c \in \mathcal{N}(0,1)} \|\epsilon - \epsilon_c(x_{t+1}, t, c)\|_2^2 \tag{6}$$

Thanks to prompt condition, the model can produce more excellent performance in terms of visual quality than its unconditional counterparts. However, the implementation of most prominent methods [47, 9] are not publicly available. Alternatively, Stable Diffusion has released the pre-trained weights based on Hugging Face implementation [57] to facilitate research in the community. Hence, we mainly perform experiments on different versions of Stable Diffusion.

**DreamBooth** is a finetuning technique to personalize text-to-image diffusion models for instance of interest. This technique has two aims. First, it enforces the model to learn to reconstruct the user’s images, with a generic prompt $c$ such as “a photo of $\text{sk}_{\text{class noun}}$”, with $\text{sk}$ is a special term denoting the target user, and “$\text{class noun}$” is the object type, which can be “person” for human subject. To train this, DreamBooth employs the base loss of diffusion models in Eq. (6), with $x_0$ is each user’s reference image. Second, it further introduces a prior preservation loss to prevent overfitting and text-shifting problems when only a small set of instance examples is used. More precisely, it uses a generic prior prompt $c_{pr}$, e.g., “a photo of [class noun]”, and enforces the model to reproduce instance examples randomly generated from that prior prompt using the original weights $\theta_{ori}$. The training loss combines two objectives:

$$\begin{align*}
\mathcal{L}_{db}(\theta, x_0) &= \mathbb{E}_{x_0, t, t'} \|\epsilon - \epsilon_\theta(x_{t+1}, t, c)\|_2^2 + \lambda \|\epsilon' - \epsilon_\theta(x'_{t'+1}, t', c_{pr})\|_2^2 \tag{7}
\end{align*}$$

where $c, c'$ are both sampled from $\mathcal{N}(0,1)$. $x'_{t+1}$ is noisy variable of class example $x'$ which is generated from original stable diffusion $\theta_{ori}$, with prior prompt $c_{pr}$, and $\lambda$ emphasizes the importance of the prior term. While DreamBooth was originally designed for Imagen [47], it was quickly adopted for any text-to-image generator.

**3.2. Problem definition**

As a powerful tool to generate photo-realistic outputs of a target instance, DreamBooth can be a double-edged sword. When misused, it can generate harmful images toward the target individual. To mitigate this phenomenon, we propose to craft an imperceptible perturbation added to each user’s image that can disrupt the finetuned DreamBooth models to generate distorted images with noticeable artifacts. We define the problem formally below.

Denote $X'$ as the set of images of the person to protect. For each image $x \in X'$, we add an adversarial perturbation $\delta$ and publish the modified image $x' = x + \delta$, while keeping the original one private. The published image set is called $X'$. An adversary can collect a small image set of that person $X'_{db} = \{x'_{(i)} + \delta_{(i)}\}_{i=1}^{N_{db}} \subset X'$. He then uses that set as a reference to finetune a text-to-image generator $\epsilon_\theta$, following the DreamBooth algorithm, to get the optimal hyper-parameters $\theta^*$. The general objective is to optimize the adversarial noise $\Delta_{db} = \{\delta_{(i)}\}_{i=1}^{N_{db}}$ that minimizes the personalized generation ability of that DreamBooth model:

$$\begin{align*}
\Delta_{db}^* &= \arg \min_{\Delta_{db}} A(\epsilon_{\theta^*}, X') \\
\text{s.t.} \quad \theta^* &= \arg \min_{\theta} \sum_{i=1}^{N_{db}} \mathcal{L}_{db}(\theta, x_{(i)} + \delta_{(i)}), \tag{8}
\end{align*}$$

and $\|\delta_{(i)}\|_p \leq \eta \quad \forall i \in \{1, 2, \ldots, N_{db}\}$.
Fully-trained Surrogate Model Guidance (FSMG)

Alternating Surrogate and Perturbation Learning (ASPL)

Figure 2: We present here two variants of Anti-DreamBooth, namely Fully-trained Surrogate Model Guidance (FSMG) and Alternating Surrogate and Perturbation Learning (ASPL). Both methods craft the adversarial noise $\delta$ using Projected Gradient Descent (PGD) to maximize the reconstruction loss $L_{\text{cond}}$ of the surrogate model. Left: FSMG uses a fixed surrogate model $\theta$ fully finetuned on a small clean image set $X_A$ to guide the PGD optimization. Right: ASPL alternates between (i) finetuning a clone surrogate model $\theta'$ on clean images $X_A$, and (ii) using this clone model to craft $\delta$ for the current image set $X_{db}^i$ via PGD. The actual surrogate model $\theta$ is then finetuned on the perturbed images $X_{db}^{i+1}$ before the next iteration.

where $L_{db}$ is defined in Eq. 7 and $A(\epsilon_{\theta^*}, X)$ is some personalization evaluation function that assesses the quality of images generated by the DreamBooth model $\epsilon_{\theta^*}$ and the identity correctness based on the reference image set $X$.

However, it is hard to define a unified evaluation function $A$. A defense succeeds when the DreamBooth-generated images satisfy one of the criteria: (1) awful quality due to extreme noise, blur, distortion, or noticeable artifacts, (2) none or unrecognizable human subjects, (3) mismatched subject identity. Even with the first criteria, there is no all-in-one image quality assessment metric. Instead, we can use simpler objective functions disrupting the DreamBooth training to achieve the same goal.

We further divide the defense settings into categories, from easy to hard: convenient, adverse, and uncontrolled.

Convenient setting. In this setting, we have prior knowledge about the pretrained text-to-image generator, training term (e.g., “skts”), and training prompt $c$ the attacker will use. While sounding restricted, it is practical. First, the pretrained generator has to be high-quality and open-source. So far, only Stable Diffusion has been made publicly available with several versions released. Second, people often use the default training term and prompt provided in DreamBooth’s code. This setting is considered as “white-box”.

Adverse settings. In these settings, the pretrained text-to-image generator, training term, or training prompt used by the adversary is unknown. The defense method, if needed, can use a surrogate component that potentially mismatches the actual one to craft the adversarial noises. These settings can be considered as “gray-box”.

Uncontrolled setting. This is an extra, advanced setting in which some of the user’s clean images are leaked to the public without our control. The adversary, therefore, can collect a mix of perturbed and clean images $X' = X_{adv}' \cup X_{cl}$, with $X_{adv}' \subset X'_{db}$ and $X_{cl} \subset X$. This setting is pretty challenging since the DreamBooth model can learn from unperturbed photos to generate reasonable personalized images.

4. Proposed defense methods

4.1. Overall direction

As discussed, we can aim to attack the learning process of DreamBooth. As the DreamBooth model overfits the adversarial images, we can trick it into performing worse in reconstructing clean images:

$$\delta^{*}(i) = \arg \max_{\delta(i)} L_{\text{cond}}(\theta^*, x^{(i)}), \forall i \in \{1, \ldots, N_{db}\},$$

subject to

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^{N_{db}} L_{db}(\theta, x^{(i)} + \delta^{(i)}),$$

and

$$||\delta^{(i)}||_p \leq \eta \quad \forall i \in \{1, \ldots, N_{db}\},$$

where $L_{\text{cond}}$ and $L_{db}$ are defined in Eq. 6 and 7. Note that, unlike traditional adversarial attacks, our loss functions are
computed only at a randomly-chosen timestep in the denoising sequence during training. Still, this scheme is effective in breaking the generation output (Sec. 5).

4.2. Algorithms

The problem in Eq. 9 is still a challenging bi-level optimization. We define different methods to approximate its solution based on prominent techniques used in literature.

**Fully-trained Surrogate Model Guidance (FMSG).** Most previous image cloaking approaches [52, 52, 60] employ a model trained on clean data as a surrogate to guide the adversarial attack. We can naively follow that direction by using a surrogate DreamBooth model with hyper-parameters \( \theta_{\text{clean}} \) fully finetuned from a small subset of samples \( \mathcal{X}_A \subset \mathcal{X} \). This set does not need to cover the target images \( \mathcal{X}_{db} = \{x(i)\}_{i=1}^{N_{db}} \); it can be fixed, allowing the surrogate model to be learned once regardless of the constant change of \( \mathcal{X} \) and \( \mathcal{X}_{db} \). After getting \( \theta_{\text{clean}} \), we can use it as guidance to find optimal noise for each target image \( \delta(i) = \arg \max_{\delta(i)} \mathcal{L}_{\text{cond}}(\theta_{\text{clean}}, x(i) + \delta(i)) \). By doing so, we expect any DreamBooth model finetuned from the perturbed samples to stay away from \( \theta_{\text{clean}} \).

**Alternating Surrogate and Perturbation Learning (ASPL).** Using a surrogate model full-trained on clean data may not be a good approximation to solve the problem in Eq. 9. Inspired by [26], we propose to incorporate the training of the surrogate DreamBooth model with the perturbation learning in an alternating manner. The surrogate model \( \epsilon_\theta \) is first initiated with the pretrained weights. In each iteration, a clone version \( \epsilon'_\theta \) is finetuned on the reference clean data \( \mathcal{X}_A \), following Eq. (7). This model is then utilized to expedite the learning of adversarial noises \( \delta(i) \) in the current loop. Finally, we update the actual surrogate model \( \epsilon_\theta \) on the updated adversarial samples, and move to the next training iteration. We provide a snippet for one training iteration in Eq. 10. With such a procedure, the surrogate model better mimics the true models trained by the malicious DreamBooth users since it is only trained on perturbed data.

\[
\begin{align*}
\theta' &\leftarrow \theta_{\text{clone}}() \\
\theta' &\leftarrow \arg \min_{\theta'} \sum_{x \in \mathcal{X}_{db}} \mathcal{L}_{db}(\theta', x) \\
\delta(i) &\leftarrow \arg \max_{\delta(i)} \mathcal{L}_{\text{cond}}(\theta', x(i) + \delta(i)) \\
\theta &\leftarrow \arg \min_{\theta} \sum_{i=1}^{N_{db}} \mathcal{L}_{db}(\theta, x(i) + \delta(i)).
\end{align*}
\]

**Targeted approaches.** The proposed algorithms above are untargeted; each perturbation noise is learned to maximize the reconstruction loss \( \mathcal{L}_{\text{cond}} \). Therefore, the adversarial examples \( x(i) + \delta(i) \) may guide the target DreamBooth model to learn different adversarial directions, potentially canceling out their effects. Inspired by the success of targeted attacks in [52], we can select a single target \( x^{tar} \) to learn optimal \( \delta \) such that the output of model is pulled closer to \( x^{tar} \) when trained with \((x + \delta)\). This targeted attack scheme can be plugged into all previous methods, and we denote new algorithms with the prefix “T-”, e.g., T-FSMG and T-ASPL.

**Ensemble approaches.** In adverse settings, the pretrained text-to-image generator used by the attacker is unknown. While we can pick one to train the perturbation and hope it transfers well to the target generator, one better approach is to use an ensemble [15, 59] of surrogate models finetuned from different pretrained generators. This approach can be an easy plug-in for the previous approaches. Due to memory constraints, instead of using these surrogate models all at once, we only used a single model at a time, in an interleaving manner, to produce optimal perturbed data.

5. Experiments

5.1. Experimental setup

**Datasets.** To evaluate the effectiveness of the proposed methods, we look for facial benchmarks that satisfy the following criteria: (1) each dataset covers a large number of different subjects with identity annotated, (2) each subject must have enough images to form two image sets for reference (\( \mathcal{X}_A \)) and protection (\( \mathcal{X}_{db} \)), (3) the images should have mid- to high-resolution, (4) the images should be diverse and in-the-wild. Based on those criteria, we select two famous face datasets CelebA-HQ [29] and VGGFace2 [11].

CelebA-HQ is a high-quality version of CelebA [34] that consists of 30,000 images at 1024 \( \times \) 1024 resolution. We use the annotated subset [2] that filters and groups images into 307 subjects with at least 15 images for each subject.

VGGFace2 [11] contains around 3.31 million images of 9131 person identities. We filter the dataset to pick subjects that have at least 15 images of resolution above 500 \( \times \) 500.

For fast but comprehensive evaluations, we choose 50 identities for each dataset. For each subject in these datasets, we use the first 12 images and divide them into three subsets, including the reference clean image set, the target protecting set, and an extra clean image set for uncontrolled setting experiments (Sec. 5.5). Each mentioned subset has 4 images with diverse conditions. We then center-crop and resize images to resolution 512 \( \times \) 512.

**Training configurations.** We train each DreamBooth model, both text-encoder and UNet model, with batch size of 2 and learning rate of \( 5 \times 10^{-7} \) for 1000 training steps. By default, we use the latest Stable Diffusion (v2.1) as the pretrained generator. Unless specified otherwise, the training instance prompt and prior prompt are “a photo of sks person” and “a photo of person”, respectively. It takes 15 minutes to train a model on an NVIDIA A100 GPU 40GB.

We optimize the adversarial noise \( \delta(i) \) in each step of FSMG and ASPL using the untargeted PGD scheme (Eq.
### Dataset "a photo of sks person" "a dslr portrait of sks person"

<table>
<thead>
<tr>
<th>Method</th>
<th>FDFR↑</th>
<th>ISM↓</th>
<th>SER-FQA↓</th>
<th>BRISQUE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Defense</td>
<td>0.07</td>
<td>0.63</td>
<td>0.73</td>
<td>15.61</td>
</tr>
<tr>
<td>FSMG</td>
<td>0.56</td>
<td><strong>0.33</strong></td>
<td><strong>0.31</strong></td>
<td><strong>36.61</strong></td>
</tr>
<tr>
<td>ASPL</td>
<td><strong>0.63</strong></td>
<td><strong>0.33</strong></td>
<td><strong>0.31</strong></td>
<td>36.42</td>
</tr>
<tr>
<td>T-FSMG</td>
<td>0.07</td>
<td>0.58</td>
<td>0.74</td>
<td>15.49</td>
</tr>
<tr>
<td>T-ASPL</td>
<td>0.07</td>
<td>0.57</td>
<td>0.72</td>
<td>15.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>FDFR↑</th>
<th>ISM↓</th>
<th>SER-FQA↓</th>
<th>BRISQUE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Defense</td>
<td>0.10</td>
<td>0.68</td>
<td>0.72</td>
<td>17.06</td>
</tr>
<tr>
<td>FSMG</td>
<td><strong>0.34</strong></td>
<td><strong>0.48</strong></td>
<td>0.56</td>
<td>36.13</td>
</tr>
<tr>
<td>ASPL</td>
<td>0.31</td>
<td>0.50</td>
<td><strong>0.55</strong></td>
<td><strong>38.57</strong></td>
</tr>
<tr>
<td>T-FSMG</td>
<td>0.06</td>
<td>0.64</td>
<td>0.73</td>
<td>25.75</td>
</tr>
<tr>
<td>T-ASPL</td>
<td>0.06</td>
<td>0.64</td>
<td>0.73</td>
<td>20.58</td>
</tr>
</tbody>
</table>

**Table 1:** Comparing the defense performance of the proposed methods in a convenient setting on different datasets.

3). We use 100 PGD iterations for FSMG and 50 iterations for ASPL. Both methods use $\alpha = 0.005$ and the default noise budget $\eta = 0.05$. It takes 2 and 5 minutes for FSMG and ASPL to complete on an NVIDIA A100 GPU 40GB.

**Evaluation metrics.** Our methods aim to disrupt the target DreamBooth models, making them produce poor images of the target user. To measure the defense’s effectiveness, for each trained DreamBooth model and each testing prompt, we generate 30 images. We then use a series of metrics to evaluate these generated images comprehensively.

Images generated from successfully disrupted Dream-Booth models may have no detectable face, and we measure that rate, called **Face Detection Failure Rate (FDFR)**, using RetinaFace detector [17]. If a face is detected, we extract its face recognition embedding, using ArcFace recognizer [18], and compute its cosine distance to the average face embedding of the entire user’s clean image set. This metric is called **Identity Score Matching (ISM)**. Finally, we use two extra image quality assessment metrics. One is **SER-FQA** [55], which is an advanced, recent metric dedicated to facial images. The other is **BRISQUE** [36], which is classical and popular for assessing images in general.

5.2. Convenient setting

We first evaluate the proposed defense methods, including FSMG, ASPL, T-FSMG, and T-ASPL, in a convenient setting on the two datasets. We try two image generation prompts, one used in training ("a photo of sks person") and one novel, unseen prompt ("a dslr portrait of sks person"). The average scores over DreamBooth-generated images with each defense are reported in Table 1. As can be seen, the untargeted defenses significantly increase the face detection failure rates and decrease the identity matching scores, implying their success in countering the Dream-Booth threat. We provide some qualitative images in Fig. 3a. As expected, ASPL defends better than FSMG since it mimics better the DreamBooth model training at test time. Targeted methods perform poorly, suggesting that the noise generated by these methods, while providing more consistent adversarial guidance in DreamBooth training, is suboptimal and ineffective. Since ASPL performs the best, we will try only this method in all follow-up experiments.

**Qualitative Evaluation.** We further evaluate ASPL via a user study with 40 participants. For each of 50 identities in VGGFace2, we train two DreamBooth models on (1) origi-


<table>
<thead>
<tr>
<th>Version</th>
<th>Defense?</th>
<th>“a photo of sks person”</th>
<th>“a dslr portrait of sks person”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FDFR↑</td>
<td>ISM↑</td>
</tr>
<tr>
<td>v1.4</td>
<td>✗</td>
<td>0.05</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>0.80</td>
<td>0.18</td>
</tr>
<tr>
<td>v1.5</td>
<td>✗</td>
<td>0.07</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>0.71</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 2: Defense performance of ASPL with different generator versions on VGGFace2 in a convenient setting.

<table>
<thead>
<tr>
<th>η</th>
<th>Quality</th>
<th>“a photo of sks person”</th>
<th>“a dslr portrait of sks person”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR↑</td>
<td>LPIPS↓</td>
<td>FDFR↑</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0.07</td>
</tr>
<tr>
<td>0.01</td>
<td>48.74</td>
<td>0.01</td>
<td>0.08</td>
</tr>
<tr>
<td>0.03</td>
<td>38.42</td>
<td>0.12</td>
<td>0.44</td>
</tr>
<tr>
<td>0.05*</td>
<td>34.56</td>
<td>0.21</td>
<td>0.63</td>
</tr>
<tr>
<td>0.10</td>
<td>28.77</td>
<td>0.40</td>
<td>0.76</td>
</tr>
<tr>
<td>0.15</td>
<td>25.97</td>
<td>0.50</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 3: Quality of protected images and defense performance of ASPL with different noise budgets on VGGFace2 in a convenient setting. “*” is default.

5.3. Ablation studies

Text-to-image generator version. In previous experiments, we used Stable Diffusion (SD) v2.1 as the pretrained text-to-image generator. In this section, we examine if our proposed defense (ASPL) is still effective when using different pretrained generators. Since SD is the only open-source large text-to-image model series, we try two of its versions, including v1.4 and v1.5. Note that while belonging to the same model family, these models, including v2.1, are slightly different in networks and behaviors. As reported in Table 2, ASPL shows consistent defense effectiveness.

Noise budget. Next, we examine the impact of noise budget η on ASPL attack using SD v2.1. As illustrated in Table 3, our defense is already effective with η = 0.03. The larger the noise budget is, the better defense performance we get, at the cost of the perturbation’s stealthiness.

Inference text prompt. As indicated in Table 1, ASPL well-disturbs images generated with an unseen text prompt (“a dslr portrait of sks person”). We further test the ASPL-disturbed DreamBooth models with different inference text prompts and get similar results. Fig. 3b provides some examples generated with these extra prompts.

Textual Inversion [20] and DreamBooth with LoRA [24]. We additionally test ASPL against other personalized text-to-image techniques. Textual Inversion learns new concepts by optimizing a word vector instead of finetuning the full model. LoRA uses low-rank weight updates to improve memory efficiency and is commonly used by the community. While being weaker than DreamBooth, they can be used to verify the robustness of our defense. As shown in Table 7, ASPL successfully defends against both methods, further demonstrating our effectiveness against other personalization techniques.

5.4. Adverse settings

In this section, we investigate if our proposed defense still succeeds when some component is unknown, leading to a mismatch between the perturbation learning and the target DreamBooth model finetuning processes. Unless otherwise specified, all experiments will be conducted using ASPL with SD V2.1 and on the VGGFace2 dataset.

Model mismatching. The first scenario is when the pretrained generators are mismatched. We provide an example of transferring adversarial noise trained on SD v1.4 to defend DreamBooth models trained from v2.1 and v2.0 in the first and third rows in Table 4. ASPL still provides good scores as in Table 1. We also examine the ensemble solution suggested in the literature, as discussed in Sec. 4.2. We combine that ensemble idea with ASPL, called E-ASPL, using SD v1.4, 1.5, and 2.1. It further improves the defense in both cases, as illustrated in the upper half of Table 4.

Term mismatching. The malicious user can change the term representing the target from the default value (“sks”) to another, e.g., “t@t”. As reported in the first row in the lower half of Table 4, this term mismatch has only a moderate effect on our results; key scores, like ISM, are still good.

Prompt mismatching. The malicious user can also use a different DreamBooth training prompt. ASPL still provides low ISM scores, as reported in the last row of Table 4.
Table 4: Defense performance of ASPL on VGGFace2 when the model, term, or prompt used to train the target DreamBooth model is different from the one used to generate defense noise and when the ensemble technique is applied. Here, $S_\star$ is “t@t” for the first row and “sks” for second row.

<table>
<thead>
<tr>
<th>Perturbed</th>
<th>Clean</th>
<th>“a photo of sks person”</th>
<th>“a dslr portrait of sks person”</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>0.63 0.33 0.31 0.31 36.42</td>
<td>0.76 0.28 0.30 39.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.50 0.43 0.41 35.53</td>
<td>0.52 0.35 0.51 34.01</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.29 0.53 0.61 28.99</td>
<td>0.40 0.37 0.62 26.13</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.08 0.61 0.73 18.92</td>
<td>0.27 0.45 0.70 15.55</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>0.07 0.63 0.73 15.61</td>
<td>0.21 0.48 0.71 9.64</td>
</tr>
</tbody>
</table>

Table 5: Defense performance of ASPL on VGGFace2 in uncontrolled settings. We include two extra results with 0 clean image (convenient setting) and 0 perturbed image (no defense) for comparison.

<table>
<thead>
<tr>
<th>Model mismatch</th>
<th>Train</th>
<th>Test</th>
<th>“a photo of sks person”</th>
<th>“a dslr portrait of sks person”</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.4</td>
<td>v2.1</td>
<td>0.62 0.31 0.28 36.00</td>
<td>0.70 0.31 0.35 38.39</td>
<td></td>
</tr>
<tr>
<td>v1.4</td>
<td>v2.0</td>
<td>0.70 0.27 0.23 36.83</td>
<td>0.61 0.26 0.31 37.28</td>
<td></td>
</tr>
<tr>
<td>Ensemble v1.4, 1.5, 2.1</td>
<td>v2.0</td>
<td>0.70 0.27 0.28 36.71</td>
<td>0.75 0.29 0.33 39.23</td>
<td></td>
</tr>
<tr>
<td>Ensemble v1.4, 1.5, 2.1</td>
<td>v2.1</td>
<td>0.70 0.27 0.28 36.71</td>
<td>0.75 0.29 0.33 39.23</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: ASPL’s performance on VGGFace2 using the prompt “A photo of sks person”.

<table>
<thead>
<tr>
<th>Term/ Prompt mismatch</th>
<th>DreamBooth prompt</th>
<th>“a photo of $S_\star$ person”</th>
<th>“a dslr portrait of $S_\star$ person”</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.4</td>
<td>v2.1</td>
<td>0.62 0.31 0.28 36.00</td>
<td>0.70 0.31 0.35 38.39</td>
</tr>
<tr>
<td>v1.4</td>
<td>v2.0</td>
<td>0.70 0.27 0.23 36.83</td>
<td>0.61 0.26 0.31 37.28</td>
</tr>
</tbody>
</table>

Table 7: ASPL’s performance against Textual Inversion and LoRA DreamBooth, the prompt is “A photo of sks person”.

<table>
<thead>
<tr>
<th>Def.?</th>
<th>FDFR↑</th>
<th>ISM↓</th>
<th>SER-FQA↓</th>
<th>BRISQUE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>0.06</td>
<td>0.50</td>
<td>0.67</td>
<td>7.79</td>
</tr>
<tr>
<td>TI</td>
<td>0.43</td>
<td>0.52</td>
<td>0.59</td>
<td>36.79</td>
</tr>
<tr>
<td>LoRA</td>
<td>0.06</td>
<td>0.52</td>
<td>0.69</td>
<td>17.25</td>
</tr>
<tr>
<td>LoRA</td>
<td>0.64</td>
<td>0.23</td>
<td>0.27</td>
<td>42.07</td>
</tr>
</tbody>
</table>

Image preprocessing. We also evaluate the robustness of our defense under common image preprocessing techniques. As shown in Table 6, applying Gaussian blur or JPEG compression on protected images slightly weakens the defense. However, the impact on generated image quality remains significant, as evidenced by the high BRISQUE scores across different settings. Hence, our defense demonstrates reasonable robustness against these techniques.

Real-world test. Our method successfully disrupts personalized images generated by Astria [1], a black-box commercial service (see the Supplementary).

5.5. Uncontrolled settings

Anti-DreamBooth is designed for controlled settings, in which all images have protection noises added. In this section, we examine when the assumption does not hold, i.e., the malicious user gets some clean images of the target subject and mixes them with the perturbed images for DreamBooth training. Assuming the number of images for DreamBooth finetuning is fixed as 4, we examine three configurations with the number of clean images increasing from 1 to 3 (Table 5). Our defense is still effective when half of the images are perturbed, but its effectiveness reduces when more clean images are introduced. Still, these uncontrolled settings can be prevented if our system becomes popular and used by all social media with lawmakers’ support.

6. Conclusions

This paper reveals a potential threat of misused DreamBooth models and proposes a framework to counter it. Our solution is to perturb users’ images with subtle adversarial noise so that any DreamBooth model trained on those images will produce poor personalized images. We designed several algorithms and extensively evaluated them. Our defense is effective, even in adverse conditions. In the future, we aim to improve the perturbation’s imperceptibility and robustness [15, 59, 58] and conquer uncontrolled settings.
References


