Latent Diffusion Shield - Mitigating Malicious Use of Diffusion Models through Latent Space Adversarial Perturbations

Huy Phan[1†,2*] Boshi Huang[2] Ayush Jaiswal[2‡] Ekraam Sabir[2] Prateek Singhal[2] Bo Yuan[1]
[1]Rutgers University [2]Amazon

Abstract

Diffusion models have revolutionized the landscape of generative AI, particularly in the application of text-to-image generation. However, their powerful capability of generating high-fidelity images raises significant security concerns on the malicious use of the state-of-the-art (SOTA) text-to-image diffusion models, notably the risks of misusing personal photos and copyright infringement through the replication of human faces and art styles. Existing protection methods against such threats often suffer from lack of generalization, poor performance, and high computational demands, rendering them unsuitable for real-time or resource-constrained environments. Addressing these challenges, we introduce the Latent Diffusion Shield (LDS), a novel protection approach designed to operate within the latent space of diffusion models, thereby offering robust defense against unauthorized diffusion-based image synthesis. We validate LDS's performance through extensive experiments across multiple personalized diffusion models and datasets, establishing new benchmarks in image protection against the malicious use of diffusion models. Notably, the generative version of LDS provides SOTA protection, while being $150\times$ *faster and using* $2.6\times$ *less memory.*

1. Introduction

Diffusion models [4, 6, 12, 14, 15, 17], an innovative emerging deep generative techniques, have shown remarkable capabilities in generating high-fidelity, diverse, and visually appealing images, leading to their wide adoption in various image synthesis applications. Recently, Stable Diffusion [2], an advanced text-to-image based on latent diffusion model (LDM) [15] known for its fast generation speed, has been enthusiastically utilized by over 10 million users daily, significantly advancing and promoting the widespread application of text-to-image synthesis.

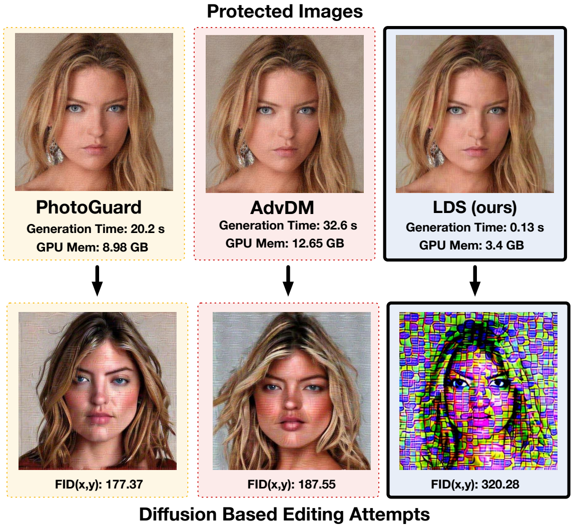

Figure 1. LDS surpasses PhotoGuard and AdvDM, while being $150\times$ faster and using $2.6\times$ less memory w. enhanced protection.

Despite this unprecedented success, the pre-trained text-to-image diffusion models typically depend on the user-provided text prompts, limiting their ability to generate more specific and personalized concepts. Aiming to address this limitation, a series of Personalized Diffusion Models (PDMs), e.g., Textual Inversion [5], DreamBooth [16], and Custom Diffusion [8], have been proposed to allow the adaption to the individualized concepts with only given few reference examples, thereby further facilitating subject- and style-driven image generation.

However, the rise of PDMs poses significant security challenges and risks. For instance, with just a few victim images, malicious actors can exploit PDMs to replicate human facial identities, posing a serious threat to user privacy and identity integrity. Additionally, PDMs can mimic artists' styles in style-driven generation tasks, raising concerns over potential copyright violations.

Recognizing the strong demand for robust protection against the malicious use of PDMs, recent works [10, 11, 18, 22, 24, 25, 27] have aimed to mitigate unwanted and malicious image manipulations. By injecting imperceptible

*Corresponding author: phanhuy@amazon.com
†Work done while at Rutgers. ‡Work done while at Amazon.

adversarial perturbations into the reference images, these methods seek to make PDMs output images with numerous visual artifacts and compromised quality, thereby ensuring the safe progression of diffusion-based image synthesis.

Though prior efforts offer some protection, existing solutions have limitations. **First**, as the SOTA PDM models are based on LDM architecture, which encodes input images into latent representations, the strategy that directly applies the adversarial noise to the pixels, naturally diminishes the effectiveness of protection. **Second**, Due to the mechanism of performing iterative operations on the large-size input images, instead of the small latent representations, the current methods have high computational intensity and large storage requirement, bringing high GPU memory usage and long processing time. **Third**, this proactive protection solution is sensitive to the change of input image. Classical image processing techniques, such as JPEG compression [19], can effectively disrupt the tiny adversarial noise, thereby decreasing the protection performance.

To address these challenges, this paper develops Latent Diffusion Shield (LDS), a protection method delivering high performance and robustness while maintaining low computational costs by working in the latent space instead of the pixel space. LDS mitigates the effects of malicious manipulations by PDMs, addressing limitations of previous pixel-based approaches and offering a higher-performance, more efficient, and robust solution for image protection (see Fig. 1). Our key contributions are summarized as follows:

- We propose Latent Diffusion Shield (LDS), a solution that applies protective perturbations in the latent space, providing superior defense against unwanted usage of PDMs with enhanced performance and robustness.
- We develop iterative (LDS-I) and generative (LDS-RT) versions for diverse scenarios. Notably, LDS-RT sets new protection benchmarks, reducing GPU memory usage by 2.6× and image generation time by 150×.
- We conduct extensive experiments on various PDMs (Textual Inversion, Dream Booth, Custom Diffusion) and compare with SOTA methods across three datasets to safeguard against misuse by diffusion models.

2. Related Works

Diffusion Models and Personalized Diffusion Models (PDMs). Diffusion models, as highlighted by [4, 6, 14, 15, 17], have significantly advanced text-to-image synthesis with their high fidelity and diversity. The Latent Diffusion Model (LDM), particularly Stable Diffusion [15], optimizes this process in a low-dimensional space, enhancing accessibility and efficiency. In many practical scenarios, the pre-trained text-to-image diffusion models face the demands for personalization to create specific and individualized concepts. To cater to this, methods like Textual Inversion [5] optimize new “word” embeddings using a few user-supplied images. DreamBooth [16] fine-tunes the entire model for high-fidelity novel concepts linked to rare word-embeddings. For quicker tuning, Custom Diffusion [8] updates only key parameters in cross-attention layers, improving performance and enabling multiple concept integration.

Adversarial Protection Against Malicious Use of PDMs. The field of adversarial vulnerability in deep learning has rapidly evolved [3, 21], with a key focus on generating inputs that cause misclassification in models without visually differing from clean inputs. Specifically for diffusion models, adversarial attacks have taken on a unique character, distinguishing themselves from traditional classifier attacks. These attacks are not solely disruptive but can be employed for protective purposes, such as safeguarding user images from misuse in diffusion models [10, 11, 18, 22, 24–28].

Recent advancements include Glaze [20], which is a targeted adversarial attack on the feature extractor of text-to-image models. Further developments in this area include PhotoGuard [18], which introduced encoder and SDEdit attacks. These attacks specifically aim to maximize the distance between the Variational Autoencoder (VAE) latent representations of adversarial and clean examples. Additionally, AdvDM [10] utilizes Monte Carlo approximation techniques; while DUAW [25] focuses on maximizing the Structural Similarity Index (SSIM) between the clean and adversarial examples within the VAE framework. Notably, Anti-Dreambooth [22] and UDP [27] both employ strategies to render adversarial examples unlearnable during the fine-tuning of LDM, emphasizing the prevention of unintended learning of sensitive data in these models.

However, these exiting protection methods face limitations, such as inadequate protection, high computational costs, and weak robustness. We aim to develop a protection method that addresses these drawbacks in Section 4.

3. Threat Model

Attacker’s Goal: As can be seen from Figure 2, an attacker gathers a collection of photos x from a user, then applies one of the PDMs discussed in Section 2 to x. The aim is to extract and synthesize the main and common characteristics of the input photos, whether they are faces, art styles, objects, into new renditions y while preserving the original appearance. A successful attack should synthesize outputs y that are high quality, with low artifacts, realistic looking (for faces), have the same art style (for paintings), or have the same characteristics (for objects).

User’s Goal: The user aims preemptively disrupt the personalized diffusion process utilized by the attacker. This is achieved by subtly embedding imperceptible adversarial noise δ_x into the photos x, resulting in x_{adv}, before they are made public. The intention is that if the attacker attempts to

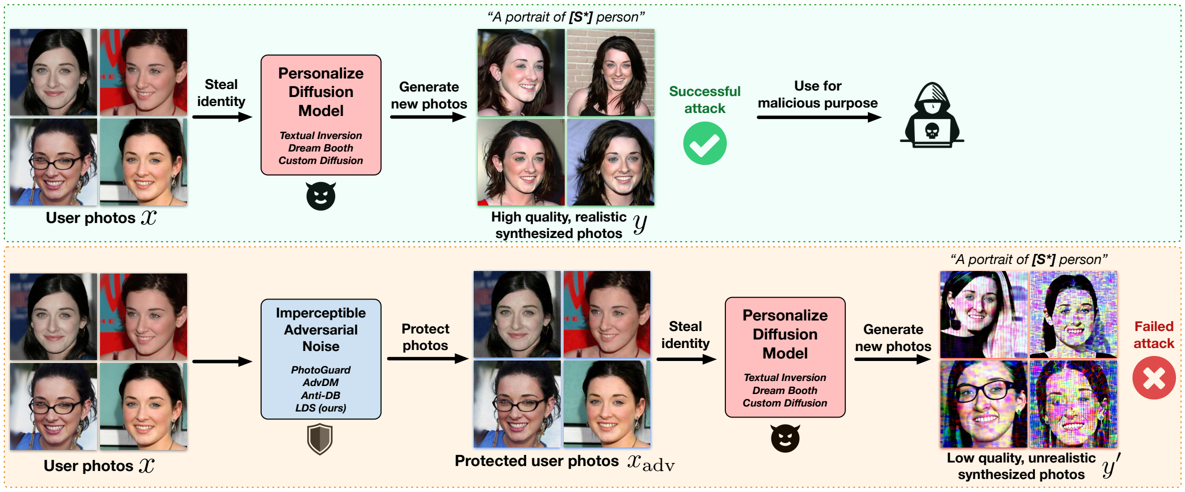
"A portrait of [S*] person"
Steal identity
Personalize Diffusion Model
Textual Inversion
Dream Booth
Custom Diffusion
Generate new photos
Successful attack
Use for malicious purpose
User photos x
High quality, realistic synthesized photos y
Imperceptible Adversarial Noise
PhotoGuard
AdvDM
Anti-DB
LDS (ours)
Protect photos
Protected user photos x_{adv}
Steal identity
Personalize Diffusion Model
Textual Inversion
Dream Booth
Custom Diffusion
Generate new photos
"A portrait of [S*] person"
Failed attack
User photos x
Low quality, unrealistic synthesized photos y'

Figure 2. Defense against misuse of personal photos by malicious actors using PDMs. Our protection method, LDS, applies imperceptible perturbations to users' images before release. This preemptive measure causes any PDMs trained on these altered images to generate distorted, unusable outputs, effectively safeguarding users' photos.

apply the PDM on these altered images x_{adv}, the quality of the resulting images y' will be significantly compromised, characterized by low quality and numerous artifacts. A successful protection needs to balance between utility and security requirements. The protected image x_{adv} should have minimal perturbations compared to x, while causing maximum disruption to the PDM process.

4. Methodology

4.1. Preliminaries

Diffusion Model. Central to diffusion models are two processes: adding noise to data and reversing it. For the training phase, the forward process introduces noise to the original image x, reaching a Gaussian distribution through a series $\{x_1, \ldots, x_T\}$. A noisy image x at time step t can be found as: $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The backward process predicts and subtracts noise, optimizing the following objective to regenerate the original image: $\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,\boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(x_{t+1}, t)\|_2^2 \right]$.

Latent Diffusion Model (LDM). LDM first uses a pretrained encoder $\mathbf{E}(\cdot)$ to map input image x to latent space z, and then forward and backward diffusion processes are applied directly on z. After that, a decoder $\mathbf{D}(\cdot)$ is used to map the denoised information in the latent space back to the pixel space. LDM is typically deployed in the conditional format to control content generation. Let z_t denotes the noisy latent at time step t, given input condition y and domain-specific encoder $\boldsymbol{\tau}$, LDM can be trained using the following optimization objective:

$$\mathcal{L}_{\text{LDM}}(\theta) = \mathbb{E}_{t,z_0,\boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(z_{t+1}, t, \boldsymbol{\tau}(y)\|_2^2 \right]. \quad (1)$$

The three PDMs discussed in Section 2, along with the various protection methods and our proposed LDS, fundamentally rely on the architecture of LDM.

4.2. Our Proposed Solution - Latent Diffusion Shield

Overview of LDS. LDS aims to interrupt the training process of LDMs described in Eq. 1. Unlike previous methods that add the adv. perturbation to the input image x, LDS introduces noise directly to the latent variable z. LDS is designed to preserve image quality, as the noise is constrained within the pixel space. We propose two variants of LDS for different scenarios. **LDS-I** is an iterative method that operates without the need for a large dataset or a training process, allowing for immediate protection of any given input. **LDS-RT (Real-Time)** is a generative method that does require an offline training phase to offer the advantage of significantly reduced GPU memory and computational time during the inference phase, facilitating real-time protection.

Optimization Objective. Let $\mathbf{E}(\cdot)$ and $\mathbf{D}(\cdot)$ represent the encoder and decoder of the LDM, respectively. The U-net is denoted as $\mathbf{U}(\cdot)$, and x is the input image. The latent representation z can be obtained as $z = \mathbf{E}(x)$, and the image can be recovered from this latent representation using $x = \mathbf{D}(z)$. The perturbation in the latent space is denoted as δ_z, and the perturbation in the pixel space can be expressed as $\delta_x = \mathbf{D}(z + \delta_z) - x$. Recall that a LDM fundamentally comprises two parts: the encoder, which maps an image from the high-dimensional pixel space to latent space, and the U-net, which denoises the image. Our adversarial attack is designed to target both of these crucial components to create effective protection. The first objective component aims to maximize the distance between the original latent z and the adversarial latent $z + \delta_z$:

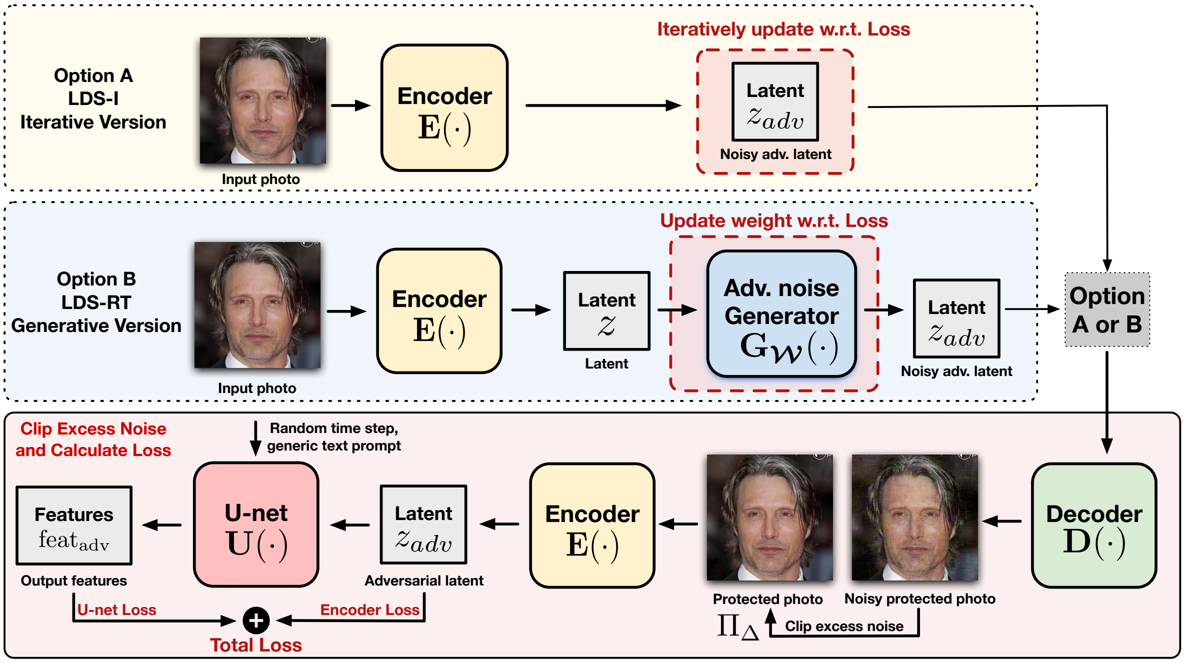

Figure 3. The proposed LDS framework. The user can select either LDS-I or LDS-RT based on the application. LDS-I employs an iterative method to optimize the adv. latent z_{adv}, whereas LDS-RT trains an Adv. Noise Generator $\mathbf{G}_{\mathcal{W}}$, boosting computational efficiency during inference and enabling real-time protection.

$$\mathcal{L}_{encoder} = \max_{\delta_z} ||\mathbf{E}(\mathbf{D}(z+\delta_z)) - z||_2 \ \textbf{s.t.}\ ||\delta_x|| \leq \Delta, \tag{2}$$

where Δ denotes the maximum allowed perturbation in the pixel space. The second component focuses on maximizing the U-net loss. We aim to find the perturbation δ_z such that the U-net cannot effectively denoise at any given time step, thereby establishing an effective protection:

$$\mathcal{L}_{unet} = \max_{\delta_z} ||\mathbf{U}(\mathbf{E}(\mathbf{D}(z+\delta_z))) - \mathbf{U}(z)||_2 \ \textbf{s.t.}\ ||\delta_x|| \leq \Delta. \tag{3}$$

Let β be a hyper-parameter controls the weight between $\mathcal{L}_{encoder}$ and $\mathcal{L}_{unet}$, the overall loss can be described as:

$$\mathcal{L}_{total} = \max_{\delta_z} \mathcal{L}_{encoder} + \beta \cdot \mathcal{L}_{unet} \ \textbf{s.t.}\ ||\delta_x|| \leq \Delta, \tag{4}$$

Solving via Iterative Method (LDS-I). To address the optimization problem outlined in Eq. 4, we propose using projected gradient ascent for iterative solving. Specifically, let $\Pi_\Delta(\cdot)$ denote a projection function that limits an image within its maximum allowable perturbation budget Δ. By recovering an image from a noisy latent representation, we the can directly constrain the noise at the pixel level to minimize quality degradation, improving the image utility:

$$x_{adv} = \Pi_\Delta(x+\delta_x) = \Pi_\Delta(\mathbf{D}(z+\delta_z)). \tag{5}$$

Therefore, Eq. 4 can now be solved directly via projected gradient ascent. The adversarial latent at step $n+1$, denoted as z_{adv}^{n+1}, is computed as follows:

Algorithm 1: Iterative version of our Latent Diffusion Shield (LDS-I)

```
Input: Encoder $\mathbf{E}(\cdot)$, Decoder $\mathbf{U}(\cdot)$, Unet $\mathbf{U}(\cdot)$,
    max. pert. $\Delta$, attack iter. $N$, input image $x$.
Output: Protected / adversarial image $x_{adv}$.
$z \leftarrow \mathbf{E}(x)$, feat $\leftarrow \mathbf{U}(z)$
$\delta_z \leftarrow \texttt{random}()$, $z_{\text{adv}} \leftarrow z + \delta_z$
for $i$ in $N$ do  ▷ iterative attack
    $x_{\text{adv}} \leftarrow \Pi_\Delta(\mathbf{D}(z_{\text{adv}}))$ ▷ via Eq. 5
    $z_{\text{temp}} \leftarrow \mathbf{E}(x_{\text{adv}})$, $\text{feat}_{\text{temp}} = \mathbf{U}(z_{\text{temp}})$
    $\mathcal{L}_{\text{total}} = ||z - z_{\text{temp}}||_2 + \beta \cdot ||\text{feat} - \text{feat}_{\text{temp}}||_2$
    $z_{\text{adv}} \leftarrow z_{\text{adv}} + \alpha \cdot \nabla \mathcal{L}_{\text{total}}$ ▷ via Eq. 6
return $x_{\text{adv}} \leftarrow \Pi_\Delta(\mathbf{D}(z_{\text{adv}}))$
```

$$z_{adv}^{n+1} = z_{adv}^{n} + \alpha \cdot \nabla \mathcal{L}_{total} = \mathbf{E}(x_{adv}^{n}) + \alpha \cdot \nabla \mathcal{L}_{total}, \tag{6}$$

where α is the learning rate. Intuitively, we start by initializing δ from a random distribution. Following [10], at each step, we select a random time step for the U-net to denoise. Then, we iteratively update the adv. noise using projected gradient ascent to maximize the training loss of the LDM. The overall process is summarized in Algo. 1.

Solving via Generative Method (LDS-RT). The iterative method, despite not needing a large dataset or extensive training, suffers from slow generation times and high computational resource demands. To enable the desired protection on resource-constrained devices and accelerate processing time for real-time inference, we developed a gener-

Algorithm 2: Generative version of Latent Diffusion Shield (LDS-RT)

```
Training Input: encoder $\mathbf{E}(\cdot)$, decoder $\mathbf{D}(\cdot)$, U-net $\mathbf{U}(\cdot)$, max. pert. $\Delta$, training iter. $N$, dataset $\mathcal{D}$, adversarial latent generator $\mathbf{G}_{\mathcal{W}}$.
Training Output: trained $\mathcal{W}$ for $\mathbf{G}_{\mathcal{W}}$.
$\mathcal{W} \leftarrow \texttt{random\_init}()$
for $i$ in $N$ do    ▷ training step
    $x \leftarrow \texttt{random\_sample}(\mathcal{D})$
    $z \leftarrow \mathbf{E}(x)$, feat $\leftarrow \mathbf{U}(z)$
    $x_{\text{temp}} \leftarrow \Pi_{\Delta}(\mathbf{G}_{\mathcal{W}}(z))$
    $z_{\text{temp}} \leftarrow \mathbf{E}(x_{\text{temp}}), \text{feat}_{\text{temp}} = \mathbf{U}(z_{\text{temp}})$
    $\mathcal{L}_{\text{total}} = ||z - z_{\text{temp}}||_2 + \beta \cdot ||\text{feat} - \text{feat}_{\text{temp}}||_2$
    $\mathcal{W} \leftarrow \mathcal{W} + \alpha \cdot \nabla \mathcal{L}_{\text{total}}$ ▷ via Eq. 7
Inference Input: trained $\mathbf{G}_{\mathcal{W}}$, encoder $\mathbf{E}(\cdot)$, max. pert. $\Delta$, input image $x$.
Inference Output: Protected adv. image $x_{adv}$.
return $x_{\text{adv}} \leftarrow \Pi_{\Delta}(\mathbf{G}_{\mathcal{W}}(\mathbf{E}(x)))$
```

ative version of our method, named LDS-RT. More specifically, let $\mathbf{G}_{\mathcal{W}}(\cdot)$ be a neural network parameterized by weights $\mathcal{W}$. The input to $\mathbf{G}_{\mathcal{W}}$ is the latent representation z, and the output is the protected latent representation $z_{adv} = \mathbf{G}_{\mathcal{W}}(z) = \mathbf{G}_{\mathcal{W}}(\mathbf{E}(x))$. The architecture of $\mathbf{G}$ is a small U-net. The network $\mathcal{W}$ is trained by solving the optimization problem:

$$\max_{\mathcal{W}} \; ||\mathbf{E}(\Pi_{\Delta}(\mathbf{D}(\mathbf{G}_{\mathcal{W}}(z)))) - z||_2 \\ + ||\mathbf{U}(\mathbf{E}(\Pi_{\Delta}(\mathbf{D}(\mathbf{G}_{\mathcal{W}}(z))))) - \mathbf{U}(z)||_2. \tag{7}$$

Once $\mathcal{W}$ is trained, an adv. example x_{adv} can be efficiently generated from original image x, bypassing time-consuming iterative process using: $x_{adv} = \Pi_{\Delta}(\mathbf{G}_{\mathcal{W}}(\mathbf{E}(x)))$. LDS-RT is summarized in Algo. 2

5. Experimental Results

5.1. Experiment Setup

Model Selection. We evaluate our methods on various personalized diffusion models including Textual Inversion [5], DreamBooth [16], and Custom Diffusion [8]. In terms of protection against diffusion model attacks, we use SOTA methods including Glaze [20], Anti-DreamBooth [22], PhotoGuard [18], and AdvDM [10] to serve as baselines. Notably, the diffusion attack used in PhotoGuard requires substantial GPU resources (approximately 50GB) and considerable computation time, making it impractical for widespread application, hence we only use the encoder attack. Regarding the foundational diffusion model, we opt for Stable Diffusion v1.5 as our primary pretrained model. Note that Glaze is provided as closed-source code, in contrast to the open-source code utilized by Anti-DreamBooth, PhotoGuard, and AdvDM. This difference prevents us from ensuring a fully equitable comparison, as we cannot control the underlying diffusion model weights, nor can we adjust various hyperparameters such as the number of iterations or the perturbation budget with Glaze. Therefore, we include Glaze in Table 1 solely as a reference baseline.

Methods	**FID**(x,y)↑	**FID**(y,y')↑	**Pre**(x,y)↓	**Pre**(y,y')↓	**CLIP**↓
		Using Textual Inversion PDM			
No Protection	123.95	n/a	0.6093	n/a	0.7239
Glaze	158.20	105.38	0.1482	0.1188	0.4517
Anti-DB	165.92	110.17	0.1385	0.1032	0.4218
PhotoGuard	177.37	115.87	0.1241	0.0889	0.4157
AdvDM	187.55	125.29	0.1296	0.0972	0.4289
LDS-I (ours)	**322.71**	**260.60**	0.0074	0.0157	**0.3353**
LDS-RT (ours)	320.28	258.48	**0.0069**	**0.0149**	0.3690
		Using DreamBooth PDM			
No Protection	113.08	n/a	0.737	n/a	0.5928
Glaze	180.45	130.07	0.0812	0.0169	0.4462
Anti-DB	202.17	165.91	0.0356	0.0061	0.3312
PhotoGuard	192.03	136.69	0.0778	0.0157	0.4158
AdvDM	189.56	133.08	0.0648	0.0148	0.3911
LDS-I (ours)	**217.69**	**170.16**	**0.0315**	**0.0065**	**0.3221**
LDS-RT (ours)	210.58	167.49	0.0328	0.0071	0.3385
		Using Custom Diffusion PDM			
No Protection	146.46	n/a	0.4241	n/a	0.6156
Glaze	189.38	101.52	0.1922	0.5697	0.4215
Anti-DB	163.28	94.18	0.1732	0.5682	0.4183
PhotoGuard	177.23	92.39	0.1648	0.5231	0.4109
AdvDM	176.70	92.82	0.1426	0.5213	0.4094
LDS-I (ours)	**239.85**	**152.38**	**0.0565**	**0.1889**	0.3744
LDS-RT (ours)	234.28	140.28	0.0602	0.2017	**0.3613**

Table 1. Comparison of various image protection methods using different PDMs on the CelebA-HQ dataset. Best results are **bolded**, second-best results are underlined.

Datasets. We evaluate our method using 3 distinct datasets. CelebA-HQ [7] comprises 30,000 celebrity faces, from which we randomly select 10 individuals, ensuring a minimum of 10 images per individual. WikiArt [1] contains 42,000 paintings; here, we choose 10 artists at random and gather at least 10 paintings per artist. The DreamBooth Data [16] is a smaller dataset of 150 photos featuring live subjects (dogs and cats) and objects, from which we select 10 subjects, each represented by 5 photos.

Metrics. 1) To evaluate the effectiveness of our protection methods, we adopt the Fréchet Inception Distance (FID) as primary metric, following [10, 18]. The $\textbf{FID}(x, y)$ metric measures the distance between the input images x (or x_{adv}), and the output images y produced by the PDMs. We propose to use the $\textbf{FID}(y, y')$ metric to assess the distance between output y (generated from x) and y' (generated from x_{adv}), to eliminate the influence of the input text prompt. We also incorporate the Precision metric, $\textbf{Pre}(x, y)$ and $\textbf{Pre}(y, y')$ [9], along with the recently developed CLIP-IQA metric [23], to evaluate the overall quality of images. **2)** For effective protection, the altered image x_{adv} should retain qualitative semantic similar to original image x. The

Methods	**FID**(x,y)↑	**FID**(y,y')↑	**Pre**(x,y)↓	**Pre**(y,y')↓	**CLIP**↓
Using Textual Inversion PDM					
No protection	274.21	n/a	0.7632	n/a	0.7406
PhotoGuard	330.02	260.29	0.4539	0.0868	0.6050
AdvDM	334.08	261.33	0.4211	0.0750	0.5829
LDS-I (ours)	348.80	282.59	**0.3658**	**0.0421**	**0.5759**
LDS-RT (ours)	**354.12**	**285.18**	0.3782	0.0457	0.5818
Using DreamBooth PDM					
No protection	245.23	n/a	0.9039	n/a	0.7716
PhotoGuard	296.74	201.32	0.6579	0.2026	0.5791
AdvDM	294.50	201.32	0.6421	0.2092	0.5987
LDS-I (ours)	**321.23**	**237.87**	**0.5303**	**0.1118**	**0.5283**
LDS-RT (ours)	318.27	232.48	0.5517	0.1284	0.5313
Using Custom Diffusion PDM					
No protection	283.95	n/a	0.6882	n/a	0.5521
PhotoGuard	310.89	171.74	0.5184	0.6118	0.4369
AdvDM	312.88	173.62	0.525	0.6013	0.4297
LDS-I (ours)	**331.83**	**193.27**	0.4724	0.5171	0.3488
LDS-RT (ours)	327.18	189.21	**0.4602**	**0.5121**	**0.3452**

Table 2. Comparison of various image protection methods using different PDMs on the WikiArt dataset. Best results are **bolded**, second-best results are underlined.

efficacy of protection is quantitatively determined by the distinctness of the generated output y' from both the original input x and the output y. Additionally, the visual quality of y' should be noticeably reduced. Optimal protection is thus indicated by high **FID**(x, y) and **FID**(y, y') scores, signifying effective alteration, while maintaining low scores in **Pre**(x, y), **Pre**(y, y'), and CLIP-IQA.

Hyper-parameters. In our experiments, the l_∞ norm is used with a maximum perturbation budget of $\Delta = 8/255$. We conduct iterative attacks using 200 iterations. For the genertic prompt, we use *"a photo"* for CelebA-HQ and DreamBooth dataset, and *"a painting"* for WikiArt dataset. For generative attacks, we train $\mathbf{G}(\cdot)$ using 200 epochs with Adam optimizer and learning rate $\alpha = 0.0003$. The parameter β is set to 1. To maintain consistency, all input images are standardized to a resolution of 512×512. The computational experiments are carried out using PyTorch with FP16 mixed precision, on NVIDIA A10 GPUs.

5.2. High Protection Performance Against Malicious use of PDMs

We utilize the five metrics proposed in Section 5.1 to quantitatively evaluate our LDS method across three different Personalized Diffusion Models (PDMs): Textual Inversion, DreamBooth, and Custom Diffusion. These evaluations are conducted on three datasets: CelebA-HQ, WikiArt, and DreamBooth. Due to the substantial computational resources required by Anti-DB compared to other protection methods, its testing is limited to the CelebA-HQ only.

As shown in Table 1, the baseline methods, PhotoGuard and AdvDM provide effective protection across all PDMs. However, Anti-DB excels only with DreamBooth and falls short in offering protection with the other two PDMs. In contrast, our LDS-I and LDS-RT demonstrate exceptional protection in all scenarios using all metrics. Notably, with Textual Inversion PDM, LDS-I achieves an **FID**(x, y) score of 322.71, significantly surpassing AdvDM score of 187.55.

Methods	**FID**(x,y)↑	**FID**(y,y')↑	**Pre**(x,y)↓	**Pre**(y,y')↓	**CLIP**↓
Textual Inversion - DreamBooth Dataset					
No protection	148.55	n/a	0.9139	n/a	0.9025
PhotoGuard	172.62	75.59	0.8037	0.1935	0.6258
AdvDM	176.10	82.04	0.7713	0.1565	0.6465
LDS-I (ours)	**229.22**	**134.07**	**0.5398**	**0.0574**	**0.6178**
DreamBooth - DreamBooth Dataset					
No protection	262.89	n/a	0.3241	n/a	0.6433
PhotoGuard	339.06	163.42	0.1667	0.1037	0.3212
AdvDM	334.00	159.89	0.1759	0.1130	0.3128
LDS-I (ours)	**350.84**	**194.37**	**0.0824**	**0.0574**	**0.3112**
Custom Diffusion - DreamBooth Dataset					
No protection	157.80	n/a	0.9046	n/a	0.8298
PhotoGuard	164.81	64.91	0.8407	0.4907	0.6483
AdvDM	165.99	66.25	0.8315	0.4704	0.6422
LDS-I (ours)	**192.78**	**97.25**	**0.6361**	**0.2407**	**0.6324**

Table 3. Comparison of various image protection methods using different PDMs on the DreamBooth dataset. Best results are **bolded**, second-best results are underlined. LDS-RT is not applicable here because of small size of DreamBooth dataset.

In Table 2, on the WikiArt dataset, LDS-I shows the highest performance, closely followed by LDS-RT. Both methods significantly outperform the previous baselines, PhotoGuard and AdvDM, in all tested scenarios. Remarkably, with the Textual Inversion PDM, our LDS-RT achieves an **FID**(x, y) score of 354.12, followed by LDS-I's 348.80, a substantial improvement over PhotoGuard's 330.02, indicating a significant performance gap.

The results from the DreamBooth dataset are presented in Table 3. Given DreamBooth's limited size, training the Adversarial Noise Generator $\mathbf{G}(\cdot)$ is not feasible on this dataset. Despite this, LDS-I version can achieve a **FID**(x, y) score of 192.78, which is a substantial improvement over PhotoGuard's score of 164.81, indicating a significant gap in performance.

5.3. Low Computational Cost

High computational costs, particularly in terms of GPU memory usage and generation time, are major drawbacks of existing protection methods. Our generative version, LDS-RT, significantly mitigates these issues, as evidenced in Table 4. Notably, LDS-RT requires only 3.4GB of GPU memory, which is substantially less than PhotoGuard's 8.98GB, marking a 2.6× reduction in memory usage. Moreover, LDS-RT impressively reduces generation time to just 0.13 seconds/image, compared to PhotoGuard's 20.2 seconds/image. This translates to a remarkable speed improvement, offering a 150× increase in efficiency while maintaining the highest level of protection performance.

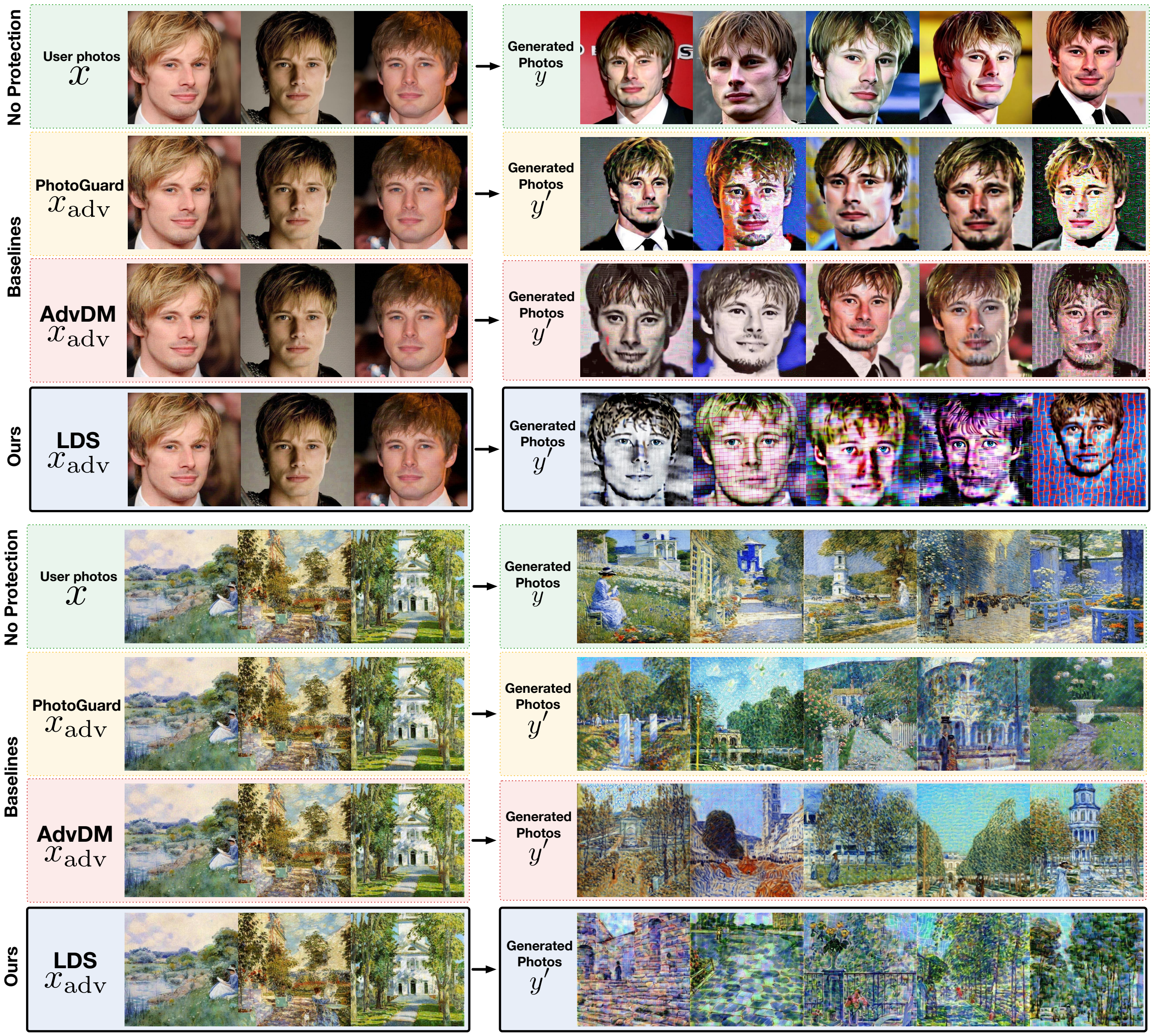

Figure 4. **Column 1:** Original images x from CelebA, and WikiArt datasets (from top to bottom) and their protected versions x_{adv} using various protection methods. Our LDS has perturbations comparable to previous techniques, preserving high visual quality in x_{adv}. **Column 2:** Outputs of the Textual Inversion on CelebA, WikiArt. For unprotected images x, outputs y are high-quality and realistic for CelebA, or maintain the art style for WikiArt. Using PhotoGuard and AdvDM, outputs y' have lower quality and increased artifacts. LDS results in the most significantly degraded outputs y', highlighting its superior visual protection.

5.4. Enhanced Robustness Against Image Pre-processing Methods

A notable limitation in previous research is the robustness of protection methods against various image pre-processing techniques, as highlighted in prior study [19]. Specifically, we assess the resilience of our LDS-I method compared to the strongest baseline AdvDM, against common pre-processing methods such as Gaussian blur, JPEG compression, and also the powerful diffusion based denoising Diff Pure [13]. The results of this comparison are detailed in Table 5. Our findings indicate that JPEG compression, Gaussian blur, and Diff Pure [13] can reduce the protection effectiveness. However, LDS-I consistently outperforms AdvDM in all scenarios. This superior performance across different pre-processing methods shows the robustness of our LDS method, demonstrating its enhanced resilience in various image processing methods.

5.5. Qualitative Results.

We conduct a visual comparison in Figure 4. Specifically, we examine the differences between using original user photos x, PhotoGuard-protected photos x_{adv}, AdvDM-protected photos x_{adv}, and our LDS-protected photos x_{adv}.

Methods	GPU Memory	GPU Time
Glaze	4.8 GB	63 s
Anti-DreamBooth	22.02 GB	>600 s
PhotoGuard	8.98 GB	20.2 s
AdvDM	12.65 GB	32.6 s
LDS-I (ours)	17.71 GB	57.1 s
LDS-RT (ours)	**3.4 GB**	**0.13 s**

Table 4. Comparison of computational resources required to generate a 512×512 image across various protection methods. Compared to the best-performing baseline, PhotoGuard, our LDS-RT offers a significant $2.6\times$ reduction in GPU memory usage and an impressive $150\times$ decrease in image generation time. This enables real-time protection even for resource-constrained devices.

	No Preproc.	Gaus. Blur	JPEG	Diff. Pure
PhotoGuard	92.39	84.21	83.05	73.14
AdvDM	92.82	85.05	79.18	77.17
LDS-I	**152.38**	**142.84**	**134.61**	**130.37**
LDS-RT	140.28	130.11	128.10	126.61

Table 5. Robustness against image preprocessing using Custom Diffusion on the CelebA-HQ dataset using **FID**(y, y') metric.

Δ	CelebA-HQ	WikiArt	DreamBooth
$\Delta = 2/255$	60.91	133.45	40.96
$\Delta = 4/255$	98.82	150.00	47.17
$\Delta = 6/255$	124.12	164.97	73.84
$\Delta = 8/255$	152.38	193.27	97.25
$\Delta = 10/255$	183.89	212.28	110.58

Table 6. Varying perturbation budget (Δ) using the **FID**(y, y') metric with Custom Diffusion PDM.

Imperceptible Adversarial Noise. The comparison focuses on the difference between original user images x and their adversarial/protected counterparts x_{adv}. An effective adversarial image x_{adv} should closely resemble x, with visually imperceptible perturbations. As shown in the first column of Figure 4, the noise introduced by LDS is comparable to previous protections like PhotoGuard or AdvDM, maintaining overall image quality.

Strong Visible Protection. In the second column of Figure 4, we assess the quality and realism of generated output photos y and y'. Without protection, diffusion models generate highly realistic images y from input x. However, when x_{adv} is protected by PhotoGuard or AdvDM, the diffusion process is disrupted, leading to y' with reduced quality and realism. When LDS is applied, the resulting images y' from diffusion models exhibit noticeable degradation in quality and realism, making them unusable. This comparative analysis highlights the superior protective efficacy of LDS against PDMs, especially when compared to PhotoGuard's y', AdvDM's y', and the unprotected y.

5.6. Ablation Study

Perturbation Budget (Δ). We examine how adjusting the perturbation budget, from a minimal $\Delta = 2$ to a more substantial level, affects the **FID**(y, y') in Table 6. As expected, reducing Δ weakens protection, while increasing Δ enhances it. We chose $\Delta = 8/255$ as our default for balancing visual quality and protection.

Iteration	CelebA-HQ	WikiArt	DreamBooth
50	48.29	140.62	28.81
100	90.17	168.17	57.13
200	152.38	193.27	97.25
300	167.20	205.18	114.02
400	180.47	211.74	121.18

Table 7. Impact of varying number of iterations on three datasets (CelebA-HQ, WikiArt, and DreamBooth Data) using the **FID**(y, y') metric with Custom Diffusion.

Model	CelebA-HQ	WikiArt	DreamBooth
Textual Inversion PDM			
v1.5 (white-box)	260.60	282.59	134.07
v1.4 (black-box)	202.17	259.38	89.19
v2.1 (black-box)	216.91	262.15	94.18
Custom Diffusion PDM			
v1.5 (white-box)	152.38	193.27	97.25
v1.4 (black-box)	110.72	176.12	72.08
v2.1 (black-box)	121.04	180.51	79.12

Table 8. Transferability of Protected Images to Blackbox PDMs on three datasets using the **FID**(y, y') metric.

Number of Iterations. We examine the effect of changing the number of iterations for LDS-I in Table 7. Fewer iterations reduce computational costs but decrease protection performance. Conversely, more iterations provide better protection. After 200 iterations, protection gains diminish. Thus, we use 200 iterations as our default.

Blackbox Performance - Transferability. We investigate the transferability of our protected images to blackbox PDMs using pretrained Stable Diffusion v1.5. We evaluate performance with v1.5, v1.4, and v2.1 using Textual Inversion and Custom Diffusion PDMs. Table 8 shows that the white box model (v1.5) performs best. Transferability from v1.5 to v2.1 is better than to v1.4. Although there's a performance drop with the blackbox model v1.4, it is not substantial, and our method remains effective.

6. Conclusion

We introduced the LDS, a novel approach aimed at enhancing image protection against malicious use of Personalized Diffusion Models (PDMs). Our extensive evaluations across various PDMs and datasets demonstrate that LDS outperforms existing methods in terms of protection efficacy, computational efficiency, and robustness.

Acknowledgement

This paper was partially supported by National Science Foundation under Grant CMMI-2152908.

References

[1] Wikiart: Visual art encyclopedia. https://www.wikiart.org, 2016. 5

[2] Stable diffusion - stability ai. https://stability.ai, 2022. 1

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 2

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 1, 2, 5

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 5

[8] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 1, 2, 5

[9] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019. 5

[10] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023. 1, 2, 4, 5

[11] Jiang Liu, Chun Pong Lau, and Rama Chellappa. Diffprotect: Generate adversarial examples with diffusion models for facial privacy protection. *arXiv preprint arXiv:2305.13625*, 2023. 1, 2

[12] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1

[13] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022. 7

[14] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 1, 2

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[16] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 1, 2, 5

[17] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2

[18] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023. 1, 2, 5

[19] Pedro Sandoval-Segura, Jonas Geiping, and Tom Goldstein. Jpeg compressed images can bypass protections against ai editing. *arXiv preprint arXiv:2304.02234*, 2023. 2, 7

[20] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023. 2, 5

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[22] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2116–2127, 2023. 1, 2, 5

[23] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2555–2563, 2023. 5

[24] Ruijia Wu, Yuhang Wang, Huafeng Shi, Zhipeng Yu, Yichao Wu, and Ding Liang. Towards prompt-robust face privacy protection via adversarial decoupling augmentation framework. *arXiv preprint arXiv:2305.03980*, 2023. 1, 2

[25] Xiaoyu Ye, Hao Huang, Jiaqi An, and Yongtao Wang. Duaw: Data-free universal adversarial watermark against stable diffusion customization. *arXiv preprint arXiv:2308.09889*, 2023. 1, 2

[26] Jianping Zhang, Zhuoer Xu, Shiwen Cui, Changhua Meng, Weibin Wu, and Michael R Lyu. On the robustness of latent diffusion models. *arXiv preprint arXiv:2306.08257*, 2023. 2

[27] Zhengyue Zhao, Jinhao Duan, Xing Hu, Kaidi Xu, Chenan Wang, Rui Zhang, Zidong Du, Qi Guo, and Yunji Chen. Unlearnable examples for diffusion models: Protect data from unauthorized exploitation. *arXiv preprint arXiv:2306.01902*, 2023. 1, 2

[28] Boyang Zheng, Chumeng Liang, Xiaoyu Wu, and Yan Liu. Understanding and improving adversarial attacks on latent diffusion model. *arXiv preprint arXiv:2310.04687*, 2023. 2