

---

# Investigating and Defending Shortcut Learning in Personalized Diffusion Models

---

**Yixin Liu**  
Lehigh University  
yila22@lehigh.edu

**Ruoxi Chen**  
Lehigh University  
ruc323@lehigh.edu

**Lichao Sun**  
Lehigh University  
lis221@lehigh.edu

## Abstract

Personalized diffusion models have gained popularity for adapting pre-trained text-to-image models to generate images of specific topics with minimal training data. However, these models are vulnerable to minor adversarial perturbations, leading to degraded performance on corrupted datasets. Such vulnerabilities are further exploited to craft protective perturbations on sensitive images like portraits that prevent unauthorized generation. In response, diffusion-based purification methods have been proposed to remove these perturbations and retain generation performance. However, existing works lack *detailed analysis of the fundamental shortcut learning vulnerability of personalized diffusion models* and also *turn to over-purifying the images, which causes information loss*. In this paper, we take a closer look at the fine-tuning process of personalized diffusion models through the lens of shortcut learning. And we propose a hypothesis explaining the manipulation mechanisms of existing perturbation methods, demonstrating that perturbed images significantly deviate from their original prompts in the CLIP-based latent space. This misalignment during fine-tuning causes models to associate noisy patterns with identifiers, resulting in performance degradation. Based on these insights, we introduce a systematic approach to maintain training performance through purification. Our method first purifies the images to realign them with their original semantic meanings in latent space. Then, we introduce contrastive learning with negative tokens to decouple the learning of clean identities from noisy patterns, which shows a strong potential capacity against adaptive perturbation. Our study uncovers shortcut learning vulnerabilities in personalized diffusion models and provides a firm evaluation framework for future protective perturbation research. Code is available at <https://github.com/liuyixin-louis/DiffShortcut>.

## 1 Introduction

The rapid advancements in text-to-image diffusion models, such as DALL-E 2 [1], Stable Diffusion [2], and MidJourney [3], have revolutionized the field of image generation. These models can generate highly realistic and diverse images based on textual descriptions, enabling a wide range of applications in creative industries, entertainment, and beyond. However, the ability to fine-tune these models for personalized generation using a small set of reference images has raised concerns about their potential misuse, such as generating misleading or harmful content targeting individuals [4, 5] or threatening the livelihood of artists by mimicking unique artistic styles without compensation [6].

To address these issues, several data-poisoning-based methods have been proposed to protect user images from unauthorized personalized synthesis. These methods aim to proactively make images resistant to AI-based manipulation by crafting adversarial perturbations [5, 7], applying subtle style-transfer cloaks [6], or crafting misleading perturbation that causes model’s overfitting [4, 8]. The model trained on perturbed data will generate images poor in quality, and thus, the unauthorized

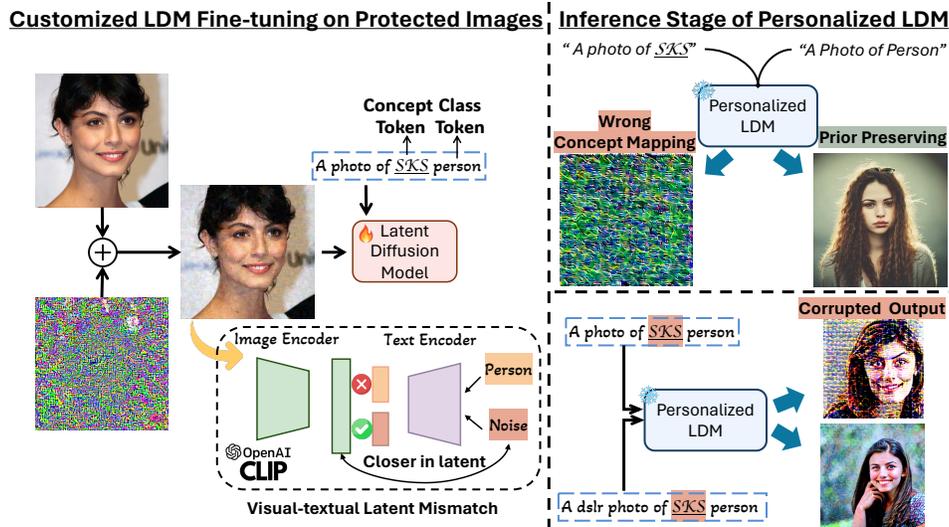


Figure 1: We observe that protective perturbation for personalized diffusion models creates a latent mismatch in the image-prompt pair. Fine-tuning on such perturbed data tricks the models, learning the wrong concept mapping. Thus, model generations suffer from high degradation in quality.

fine-training fails. Despite the protection effectiveness, different from the adversarial perturbation crafted for fixed off-the-shelf diffusion models, where the protection against unauthorized editing [7] can be well explained by the adversarial vulnerability of neural networks [9], and also the sharpness of the latent space of VAE [10, 11]. The underlying mechanism for why protective perturbation disturbs the fine-tuning of the diffusion model has not been explored yet.

Furthermore, purification studies are also purposed to further break those protections. As demonstrated in [4], most of the protection methods lack resilience against minor data transformations like Gaussian smoothing or JPEG compression. However, this simple rule-based purification causes severe data quality degradation. As demonstrated in [12, 13], diffusion-reconstructed-based purification shows a strong capacity to denoise the images and yield high-quality output by leveraging the distribution modeling ability of diffusion models. Based on the observation that clean images have better consistency upon LDMs-based reconstruction, IMPRESS [12] proposes optimization on the images to ensure reconstruction consistency with visual LPIPS similarity constraints. Despite effectiveness, such purification requires a tremendous of time due to the iterative nature of the propose optimization. On the other line, DiffPure [14] proposes to leverage off-the-shelf pixel-space diffusion models to conduct an SDEdit process that converts the perturbed images into a certain noisy state with a diffusion forward process and then denoises in the reverse process. GrIDPure [15] further divides the images into smaller grids and employs small-step DiffPure that yields better visual consistency. However, these SDEdit-based purification approaches still yield unfaithful content that causes great change in identity due to the generative nature of the diffusion model leveraged. How to design an effective, efficient, and faithful purification approach is still an open question.

Given the above two major limitations of existing works, in this paper, we first take a closer look at the reason behind this protection, where we found the latent of the perturbed images by these methods are largely shifted from its original concept. Going deeper, we discover that the diffusion models fine-tuned on these perturbed images turn to link the rarely-appeared concept token with the injected noisy perturbation instead of learning the clean identity shown in images. Based on these findings, we showcase the shortcut-learning vulnerability of personalized diffusion upon fine-tuning on poison.

Motivated by this empirical understanding, we present a systematic purification and training approach to empower robust personalized diffusion models upon perturbed data. Our approach conducts comprehensive purification from three perspectives, including input image-text pair purification, contrastive learning with the negative token, and quality-enhanced sampling. Compared to existing purification methods that are only limited to image denoising, the advantages, and benefits of our framework are three folds: i) *efficiency and faithfulness*: we conduct image purification by using off-the-shelf super-resolution and image restoration models that convert low-quality, noisy images

into high-quality, purified ones; ii) *robustness and once-for-all*: we demonstrate that contrastive learning with negative token shows strong ability in defending against adaptive attack crafted with knowledge on the purification models; iii) *systematic*: not limiting to image denoising, we conduct a systematic investigation on other potential approaches that a model trainer can leverage in training and sampling, to better facilitate the evaluate the comprehensive evaluation on the effectiveness and resistance of future protection methods. Our main contributions are summarized as follows:

- We empirically explain the underlying mechanism of the success of recent protective perturbations for safeguarding against unauthorized personalize diffusion model fine-tuning, where we find that the perturbed images no longer match the prompt pair in latent space, and thus it misleads the models on learning the wrong concept.
- Based on this understanding, we propose a systematic defensive framework that empowers robust personalized diffusion models from input image-text pair, model training, and the sampling process. Specifically, on the input denoising, we leverage off-the-shelf image restoration models to realign the image with its semantic meaning in the prompt; on model training, we propose contrastive learning with the negative token that leverages the clean class-prior dataset to decouple the learning of false relationship further; on the sampling, we design simple quality-enhanced approach with negative prompting. As a result, our method is more effective, efficient, and faithful.
- We empirically demonstrate the effectiveness and faithfulness of our framework on the facial dataset VGGFace2 and painting dataset WikiArt. Moreover, we demonstrate the robustness of our defending framework under adaptive attack when other variants fail.

## 2 Related Works

**Latent Diffusion Models and Customized Adaption.** Generative Models (GMs) aim to synthesize samples from a data distribution given a set of training examples. Diffusion Probabilistic Models [16] are now the dominant GMs with various applications such as text-to-image synthesis [17], image editing [18–20], and image inpainting [21]. To improve the training efficiency, Latent Diffusion Models (LDMs) [2] are proposed to conduct a more efficient diffusion process in a low-dimensional latent space with a pair of pre-trained image encoder and decoder (e.g., VAE [10]). Despite the general capability, model users often wish to synthesize specific concepts from their own personal lives. To meet that, customized text-to-image generation [22–25] is proposed to learn the concepts using a small set of reference images. Among them, DreamBooth [23] is one of the mainstream approaches that involve full fine-tuning of parameters, yielding more superior generation performance compared to those only fine-tune partial parameter [26] or a pseudo-word vector [27]. Besides, it uses a rarely-used token to link the concept and incorporate prior-preserving loss with class-oriented images, enabling a high-quality and diverse generation of subjects.

**Data Poisoning as Protection against Unauthorized Data Usage with LDMs.** Recent text-to-image LDMs have raised concerns about their potential misuse in generating misleading or harmful content targeting individuals [4, 5] and threatening the livelihood of professional artists by style mimicking [6]. To address these issues, several data-poisoning-based methods have been proposed to protect user images from unauthorized personalized synthesis by injecting adversarial perturbations through minimizing adversarial target loss in image encoder or UNet denoiser [6, 5], or denoising-loss maximization [7, 4, 8] or in opposite direction, denoising-loss minimization [28], or cross-attention loss maximization [29]. Despite its effectiveness, the underlying mechanism of protection against diffusion model fine-tuning has not been well explored yet. To the best of our knowledge, [13] is the only work that attempts to investigate this. However, [13] only shows some relevance of the vulnerability to the text encoder. In this work, we take a closer look at this mechanism and provide a more comprehensive explanation from a latent mismatch and shortcut learning perspective.

**Data Purification that Further Breaks Protection.** Despite promising protection performance, studies [4, 8, 13] suggest that these perturbations are brittle and can be easily removed with minor transformations. However, adaptive protection with EoT [30] indicates that protection can further bypass these simple rule-based transformations. Another direction is to leverage diffusion models as purifiers to perturb images back to their clean distributions. In the classification scenario, DiffPure [14] is a mainstream approach for adversarial purification by applying SDEdit [31] on the poison with an off-the-shelf diffusion model. For purification against protective perturbation, GridPure [32] further adapts iterative DiffPure with small steps on multi-grid spitted image to perverse the original

resolution and structure. However, due to their generative nature, these SDEdit-based purifications have limitations in yielding unfaithful content, where the purified images fail to preserve the original identity. Observing the perceptible inconsistency between the perturbed images and the diffusion-reconstructed ones, IMPRESS [12] conducts the purification via minimizing the consistency loss with constraints on the maximum LPIPS-based [33] similarity change on pixel space. While it manages to preserve similarity, IMPRESS suffers from the inefficiency issue due to its iterative process and is ineffective under stronger protections like [8].

### 3 Preliminary

**Latent Diffusion Models (LDMs).** Generative diffusion model [16] aims to model an unknown data distribution by learning an iterative denoiser to map random variables from pre-defined Gaussian distribution to those real data samples. Latent diffusion model [2] further conducts such a process in a lower-dimensional latent space to improve training and inference efficiency. Furthermore, by training on prompt-image pair, LDMs can take additional embedding as conditions to guide the denoising process, enabling it to generate or edit images with guidance from a prompt (e.g., a text). In LDMs, an image encoder  $\mathcal{E}$  is first used to map the image  $\mathbf{x}_0$  into a latent representation, i.e.,  $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ . And then LDMs craft noisy latent  $z_t$  in forward phrase with  $q_t(z_t | z_0) = \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t} z_{t-1}, (1 - \bar{\alpha}_t)\mathbf{I})$ , where  $\beta_t$  growing from 0 to 1 are pre-defined values,  $\alpha_t = 1 - \beta_t$ , and  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ . Suppose we have a textual prompt  $c$ , and a text encoder  $\tau_\theta$  that yield the embedding  $\mathbf{c} = \tau_\theta(c)$ . The goal of the LDM is to train a conditional noise estimator network  $\epsilon_\theta$ , e.g., a UNet [34], by predicting the Gaussian noise added in previous timestamp, to model the conditional distribution  $p(\mathbf{z}_0 | c)$  by gradually recovering  $\mathbf{z}_0$  from  $\mathbf{z}_T$  with additional textual information  $c$ . Suppose  $\epsilon_\theta(\mathbf{z}_t, t, c)$  is the Gaussian noise estimated in the  $t$ -th step and  $\epsilon$  is the ground-truth Gaussian noise sampled for  $\mathbf{z}_{t-1}$ . On each image-text pair  $(\mathbf{x}_0, c)$ , the denoising network  $\epsilon_\theta$  is trained based on the following loss:

$$\mathcal{L}_{\text{denoise}}(\mathbf{x}_0, c; \theta) := \mathbb{E}_{\mathbf{z}_0 \sim E(\mathbf{x}_0), \epsilon, t} [\|\epsilon - \epsilon_\theta(\mathbf{z}_0, t, c)\|_2^2]. \quad (1)$$

During the inference phrase, we first sample  $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$  and then conduct multi-steps denoising to obtain a latent  $\tilde{\mathbf{z}}_0$  with potential prompt conditioning  $c$ , and then we can obtain generated image with an image decoder  $\mathcal{D}$ , i.e.,  $\tilde{\mathbf{x}}_0 = \mathcal{D}(\tilde{\mathbf{z}}_0)$ .

**Personalized LDMs via Dreambooth Fine-tuning.** DreamBooth [23] is a fine-tuning-based method aimed at updating LDMs for generating images of specific concepts. The main idea of Dreambooth is to introduce a new identifier to link the subject concepts and use a class-specific prior-preserving loss to mitigate overfitting and language-drifting issues. Specifically, it fine-tunes a pre-trained LDM on two additional text-image sets, including an instance dataset  $\mathcal{D}_{x_0} = \{(\mathbf{x}_0^i, \mathbf{c}^{ID})\}_i$  and a class dataset  $\mathcal{D}_{\bar{x}_0} = \{(\bar{\mathbf{x}}_0^i, c)\}_i$ , where the subject image is  $\mathbf{x}_0$ , and the class image is  $\bar{\mathbf{x}}_0$ . For the prompt design, the class-specific prompt  $C$  is set as “a photo of a [class noun]” and the instance prompt  $C^{ID}$  is set as “a photo of *sks* [class noun]”, where *sks* specifies the subject and “[class noun]” is the category of object (e.g., “person”). Formally, DreamBooth first generates prior class data using the original checkpoint and then optimizes a weighted sum of instance denoising loss and prior-preservation loss,

$$\mathcal{L}_{db}(\mathbf{x}_0, \mathbf{c}^{ID}, \bar{\mathbf{x}}_0, c; \theta) = \mathcal{L}_{\text{denoise}}(\mathbf{x}_0, \mathbf{c}^{ID}) + \lambda \mathcal{L}_{\text{denoise}}(\bar{\mathbf{x}}_0, c), \quad (2)$$

where  $\lambda$  controls for the relative importance of this term. With approximately  $\sim 1000$  training steps and  $\sim 4$  subject images, it can generate vivid personalized subject images with Stable Diffusion [35].

**Protective Perturbation against Personalized LDMs.** Recent studies [4] suggest that minor adversarial perturbation to clean images can disturb the learning of customized diffusion and also prevent image editing with an off-the-shelf diffusion model by greatly degrading the quality of the generated image. Existing protective perturbation can be classified into two categories: perturbation crafted with fixed diffusion models and perturbation crafted with noise-model alternative updating. In this paper, we focus on the second category since they are more effective in the fine-tuning setting. The general framework of these protective perturbation methods is to craft noise that maximizes an adversarial loss  $\mathcal{L}_{adv}$  that is typically designed as the denoising loss  $\mathcal{L}_{denoise}$  and also alternatively update the noise generator surrogates  $\theta'$  can be a single model [4] or an ensemble of models [8]). Furthermore, the surrogates might only update part of their components [29]. Formally, at the  $j$ -th

alternative step, the noise surrogate  $\theta'_j$  and perturbation  $\delta^{(j)}$  are updated via solving,

$$\theta'_{j-1} \leftarrow \arg \min_{\theta'} \sum_x \mathcal{L}_{db}(\mathbf{x} + \delta^{(j-1)}, \mathbf{c}^{ID}, \bar{\mathbf{x}}, \mathbf{c}; \theta'_{j-1}); \delta^{(j)} \leftarrow \arg \max_{\|\delta^{(j-1)}\|_\infty \leq r} \mathcal{L}_{adv}(\mathbf{x} + \delta^{(j-1)}, \mathbf{c}; \theta'_{j-1}). \quad (3)$$

To solve this, standard Gradient Descent is performed on the model parameter while the images are updated via Project Gradient Descent (PGD) [36] to satisfy the  $\ell_\infty$ -ball perturbation budget constrain,

$$\theta_i \leftarrow \theta_{i-1} - \beta \nabla_{\theta_{i-1}} \mathcal{L}_{db}; \quad \mathbf{x}^{k+1} \leftarrow \Pi_{B_\infty(\mathbf{x}^0, \delta)} [\mathbf{x}^k + \eta \text{sign} \nabla_{\mathbf{x}^k} \mathcal{L}_{adv}(\mathbf{x}^k)], \quad (4)$$

where  $\Pi_{B_\infty(\mathbf{x}^0, r)}(\cdot)$  is a projection operator on the  $\ell_\infty$  ball that ensures  $\mathbf{x}^k \in B_p(\mathbf{x}^0, r) = \{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}^0\|_\infty \leq r\}$ ,  $\eta$  denotes the PGD step size and the total PGD step is  $K$ .

## 4 Methodology

### 4.1 Adversarial Perturbation Causes Latent-space Image-Prompt Mismatch

We first derive the formulation of learning personalized diffusion models on perturbed data. For the case of data poisoning, the instance data is perturbed by some adversarial noise  $\delta$ , and the personalized diffusion models optimize the following loss,

$$\mathcal{L}_{db}^{adv}(\mathbf{x}_0, \mathbf{c}^{ID}, \bar{\mathbf{x}}_0, \mathbf{c}; \theta) = \mathcal{L}_{\text{denoise}}(\mathbf{x}_0 + \delta, \mathbf{c}^{ID}) + \lambda \mathcal{L}_{\text{denoise}}(\bar{\mathbf{x}}_0, \mathbf{c}). \quad (5)$$

Based on the adversarial loss in Eq. 5, with annotation of  $\mathbf{c}^{ID} = \mathbf{c} \oplus ID$  where  $ID$  denotes the embedding of the unique identifier, we build the causal dependency graph in Fig 3 to represent the variable relationship during the learning of the personalized diffusion model to capture the concept of instance. From this graph, we found that there are two false correlations (colored in red) derived from the prior prompt condition  $\bar{\mathbf{c}}$  and instance condition  $ID$  to the noise  $\delta$ . These two paths hinder the personalized diffusion from capturing the true relationship between the instance condition  $\mathbf{c}^{ID}$  with its semantic identity behind  $\mathbf{x}_0$  and can cause the model to overfit to simpler relationship  $ID \rightarrow \delta$ . We empirically demonstrate this point in Figure 1 via conducting sampling from models trained on perturbed images using [4] with  $ID$  as prompting condition, where we found the model falsely links the  $ID$  with the noisy pattern. Furthermore, using CLIP [37] as a zero-shot classifier with labels of “noise” and “person” (shown in Figure 2), we identify that the perturbed images largely diverge from their original semantic textual concept and also its paired textual prompt  $\mathbf{c}^{ID}$  in terms of CLIP embedding space. We hypothesize that learning on such perturbed images will create contradiction and force the models to dump that chaotic perturbation pattern into the rarely-appeared identifier token  $ID$  instead of learning clean identity behind  $\mathbf{x}_0$ .

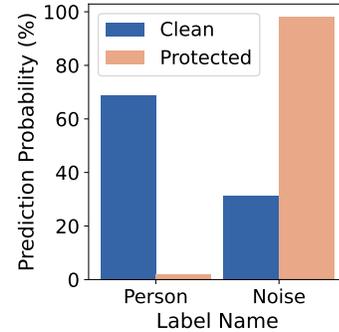


Figure 2: Zero-shot classification with CLIP-based classifier.

### 4.2 Systematic Defending with Input Purification, Decouple Learning, and Guided Sampling

Motivated by our empirical observation, we propose two directions to enhance the robustness when learning personalized diffusion on a potentially corrupted dataset: i) *weakening those spurious paths toward noisy prediction*; ii) *enhancing those toward prior data  $\bar{\mathbf{x}}_0$  and true instance data  $\mathbf{x}_0$* . For the first direction, we leverage the off-the-shelf super-resolution and image restoration models to improve the data quality and remove the noise  $\delta$ . For the second part, we introduce a novel noise-aware contrastive learning with negative tokens to make the model focus on those concept-related features and avoid being trapped in the local minimum of learning perturbation. Furthermore, during the sampling, we guide to generate high-quality images with simple negative prompting. We present our overall framework in Figure 4 and Alg. 1 and describe each of the proposed modules in detail as follows.

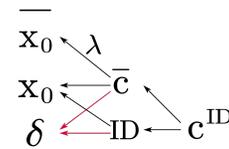


Figure 3: Variables dependency graph when learning personalized diffusion models.

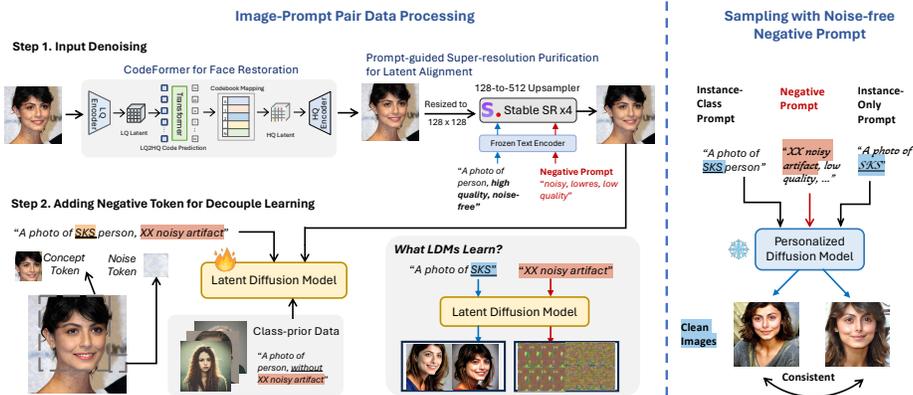


Figure 4: The framework of our method. Our method includes three steps, including *input denoising*, *decoupling training with negative tokens*, and *sampling with negative prompting*. Firstly, we leverage off-the-shelf image restoration models for purification. Secondly, we link the instance prompt and prior prompt with negative tokens in a contrastive way to guide the model in learning the correct mapping. Lastly, we insert a noise-free negative prompt to further boost the quality.

**Image Purification as Image Restoration.** The first intuitive and effective approach is to directly conduct denoising on the input image and seek to remove the adversarial perturbation. To achieve that, we view the perturbed images as degraded images in the image restoration domain and first leverage off-the-shelf image restoration models to convert low-quality, noisy images into high-quality, purified ones. Specifically, we use a face-oriented model named CodeFormer [38], which is trained on facial data to restore the images based on the latent code discretization. Besides, we leverage an additional diffusion-based super-resolution model to further enhance the purification on non-face region. Compared to the previous SoTA optimization-based purification method [12] and diffusion-based purification methods [13], this simple yet effective strategy yields faithful purified images with better efficiency since it only requires a single inference.

**Contrastive Decouple Learning with Noise Tokens.** Furthermore, in the prompt design, we add an additional tag related to the noise pattern, such as “*with XX noisy pattern*” aiming to decouple the learning of the concept and the unwanted noise structure. We also leverage the prior dataset that is prone to have a cleaner quality to help the model differentiate the noise pattern by adding a suffix, “*without XX noisy pattern*” into their textual prompt. During the sampling process, we also added this suffix in prompt input to guide the model in dropping those unwanted noisy patterns.

**Noise-free Sampling Process.** By using classifier-free guidance (CFG) [39], with a negative prompt  $c_{\text{neg}} = \text{'noisy, abstract, pattern, low quality'}$ , we further seek to guide the model generates high-quality images related to the learned concept. Specifically, given timestamp  $t$ , we perform sampling using the linear combination of the good-quality and bad-quality conditional noise estimates with  $w^{\text{neg}} = 7.5$ :

$$\tilde{\epsilon}_{\theta}(\mathbf{z}_t, \mathbf{c}) = (1 + w^{\text{neg}})\epsilon_{\theta}(\mathbf{z}_t, \mathbf{c}^{\text{ID}}) - w^{\text{neg}}\epsilon_{\theta}(\mathbf{z}_t, \tau_{\theta}(c_{\text{neg}})) \quad (6)$$

## 5 Experiments

In this section, we first perform evaluations on the effectiveness, efficiency and faithfulness of the proposed purification techniques. Then, we investigate the resilience of our framework under adaptive perturbation against the purification models. Lastly, we perform systematic ablation studies and analysis to provide an in-depth exploration of each module in our framework.

### 5.1 Experimental Setup

**Datasets and Metrics.** Our experiments are mainly performed on the VGGFace2 [40] face dataset following [4, 8]. Four identities are selected from each dataset, and we randomly pick eight images from each individual and split those images into two subsets for image protection and reference.

Table 1: Results of different purification methods under different protective perturbations. The best performances are in **bold**, and second runners are shaded in gray. \* denotes significant improvement that passes the Wilcoxon signed-rank significance test with  $p \leq 0.01$ .

Methods	FSMG		ASPL		EASPL		MetaCloak		AdvDM		PhotoGuard		Glaze	
	IMS $\uparrow$	Q $\uparrow$	IMS	Q	IMS	Q								
Clean	0.27 ± 0.10	0.21 ± 0.11	0.27 ± 0.10	0.21 ± 0.10	0.27 ± 0.10	0.21 ± 0.10	0.27 ± 0.10	0.21 ± 0.12	0.27 ± 0.10	0.21 ± 0.11	0.27 ± 0.10	0.21 ± 0.11	0.27 ± 0.10	0.21 ± 0.11
Perturbed	-0.02 ± 0.09	-0.61 ± 0.07	-0.04 ± 0.08	-0.62 ± 0.12	-0.04 ± 0.08	-0.60 ± 0.12	0.10 ± 0.10	-0.60 ± 0.08	0.01 ± 0.09	-0.29 ± 0.09	0.17 ± 0.09	-0.13 ± 0.08	0.25 ± 0.10	-0.19 ± 0.10
Gaussian F.	0.14 ± 0.10	-0.32 ± 0.08	0.15 ± 0.10	-0.34 ± 0.06	0.14 ± 0.10	-0.33 ± 0.07	0.15 ± 0.09	-0.44 ± 0.04	<b>0.15 ± 0.09</b>	-0.35 ± 0.07	0.14 ± 0.10	-0.28 ± 0.08	0.12 ± 0.09	-0.24 ± 0.07
JPEG	0.11 ± 0.10	-0.46 ± 0.10	0.14 ± 0.11	-0.43 ± 0.07	0.13 ± 0.11	-0.43 ± 0.07	0.06 ± 0.11	-0.47 ± 0.12	0.01 ± 0.11	-0.01 ± 0.11	0.12 ± 0.11	<b>0.15 ± 0.11</b>	0.16 ± 0.10	<b>0.24 ± 0.14</b>
TVM	0.03 ± 0.08	-0.45 ± 0.06	0.05 ± 0.09	-0.47 ± 0.07	0.09 ± 0.09	-0.47 ± 0.06	0.06 ± 0.09	-0.50 ± 0.04	0.07 ± 0.09	-0.47 ± 0.06	0.04 ± 0.09	-0.50 ± 0.06	-0.09 ± 0.08	-0.48 ± 0.06
PixelDiffPure	-0.04 ± 0.08	-0.37 ± 0.05	0.01 ± 0.08	-0.42 ± 0.06	-0.05 ± 0.09	-0.49 ± 0.08	-0.04 ± 0.08	-0.44 ± 0.06	0.00 ± 0.09	-0.42 ± 0.06	0.04 ± 0.09	-0.44 ± 0.04	-0.09 ± 0.08	-0.57 ± 0.08
L.DiffPure	0.06 ± 0.09	-0.45 ± 0.08	-0.03 ± 0.09	-0.50 ± 0.06	0.07 ± 0.10	-0.42 ± 0.08	-0.07 ± 0.08	-0.28 ± 0.08	-0.13 ± 0.08	-0.18 ± 0.11	0.04 ± 0.08	-0.28 ± 0.08	-0.09 ± 0.08	-0.21 ± 0.08
L.DiffPure	0.02 ± 0.08	-0.36 ± 0.09	-0.02 ± 0.09	-0.52 ± 0.06	0.02 ± 0.09	-0.41 ± 0.09	-0.05 ± 0.08	-0.26 ± 0.06	-0.09 ± 0.08	-0.18 ± 0.10	-0.02 ± 0.08	-0.26 ± 0.06	-0.13 ± 0.08	-0.22 ± 0.08
DDSPure	0.17 ± 0.11	-0.23 ± 0.07	0.20 ± 0.10	-0.28 ± 0.06	0.23 ± 0.09	-0.26 ± 0.06	0.16 ± 0.10	-0.48 ± 0.07	0.13 ± 0.10	-0.30 ± 0.07	<b>0.17 ± 0.10</b>	-0.25 ± 0.06	0.13 ± 0.10	-0.25 ± 0.07
GrIDPure	0.17 ± 0.10	-0.11 ± 0.07	0.18 ± 0.10	-0.09 ± 0.07	0.20 ± 0.10	-0.16 ± 0.07	0.12 ± 0.09	-0.17 ± 0.07	0.02 ± 0.09	-0.07 ± 0.08	0.13 ± 0.10	-0.10 ± 0.08	<b>0.28 ± 0.10</b>	-0.08 ± 0.08
IMPRESS	0.12 ± 0.10	-0.55 ± 0.07	0.11 ± 0.10	-0.54 ± 0.09	0.08 ± 0.10	-0.56 ± 0.04	0.11 ± 0.10	-0.60 ± 0.09	0.06 ± 0.10	-0.35 ± 0.09	0.11 ± 0.10	-0.10 ± 0.08	<b>0.28 ± 0.10</b>	-0.45 ± 0.07
Ours	<b>0.27*</b> ± 0.08	<b>0.45*</b> ± 0.06	<b>0.27*</b> ± 0.08	<b>0.34*</b> ± 0.04	<b>0.28*</b> ± 0.09	<b>0.35*</b> ± 0.06	<b>0.29*</b> ± 0.07	<b>0.37*</b> ± 0.08	<b>0.16*</b> ± 0.08	<b>0.49*</b> ± 0.06	<b>0.19*</b> ± 0.09	<b>0.47*</b> ± 0.06	<b>0.23 ± 0.08</b>	<b>0.48*</b> ± 0.07

Moreover, we also visually demonstrate the purification ability of our approach on the artwork painting dataset, WikiArt [41], and the CelebA [42]. For the metric, we evaluate the generated images in terms of their *semantic-related quality* and *graphical aesthetic quality*. For the semantic-related score, we compute the cosine similarity between the embedding of generated images and reference images, which we term the Identity Matching Similarity (IMS) score. We reported the weighted averaged IMS score by employing two face embedding extractors, including *antelopev2* model from InsightFace library [43] following IP-adapter [44] and *VGG-Net* [45] from Deepface library [46] following [4]. The IMS score is computed via a weighted sum:  $IMS = \lambda IMS_{IP} + (1 - \lambda) IMS_{VGG}$ , where  $\lambda$  is set as 0.7. For the graphical quality  $Q$ , we report the average of two metrics: i) *LIQE* [47], a SoTA general image quality assessment model (with re-normalization to  $[-1, +1]$ ); ii) *CLIP-IQAC* following [8], which is based on CLIP-IQA [48] with class label. See App. A.2 for details.

**Purification Baselines and Perturbation Methods.** For *purification baselines*, we consider both model-free and diffusion-based approaches. The model-free methods include ❶ Gaussian Filtering, which reduces noise and detail using a Gaussian kernel; ❷ Total Variation Minimization (TVM), which reconstructs images by minimizing the difference between original and reconstructed images while enforcing smoothness; and ❸ JPEG Compression, which reduces image file size by transforming images into a compressed format. The diffusion-based methods include ❹ (Pixel)DiffPure [14], which leverages pre-trained pixel-space diffusion models to smooth adversarial noise with small-step SDEdit process [31]; ❺ LatentDiffPure, which is developed in the paper similar as DiffPure but with LDM as a purifier (two variants w/ and w/o prompting); ❻ DDSPure [49], which finds an optimal timestamp for adversarial purification with SDEdit process; ❼ GrIDPure [15], which further conducts iterative DiffPure with small steps with grid-based splitting to improve structure similarity; and ❽ IMPRESS [12], which purifies by optimizing latent consistency with visual similarity constraints. For *protective perturbation*, we consider six of existing SoTA approaches, including perturbation crafted with bi-level optimization, such as *FSMG*, *ASPL*, *EASPL* [4], *MetaCloak* [8], and perturbations crafted with adversarial perturbation with fixed models, such as *AdvDM* [7], *PhotoGuard* [5], and *Glaze* [6]. For each setting, we set the perturbation to be ASPL by default. We set the  $\ell_\infty$  radius to 11/255 with a six-step PGD step size of 1/255 by default following [4]. See App. A for more details.

## 5.2 Effectiveness, Efficiency, and Faithfulness

**Effectiveness Comparison.** We present the effectiveness of different purification across six perturbation methods in Table 1. From the table, we can see that compared to the clean case, training on perturbing data causes serve model degradation from both identity similarity and image quality. Across all perturbations, ASPL causes the most severe degradation under the setting without purification, while MetaCloak performs more robustly under rule-based purification. Compared to rule-based purification, diffusion-based approaches achieve better performance in improving both identity similarity and image quality in most settings. Among them, GrIDPure yields relatively better purification performance since it considers the structure consistency, which suppresses the generative nature during the purification. However, there are still gaps in the IMS score compared to the clean case, and most of the Q scores after conducting GrIDPure purification are still negative. Compared to these baselines, our method closes the gap by further improving the IMS and Q scores, which are even higher than the clean training case for most of the settings. The reasons are two folds: first, we use image-restoration-based approaches, which well perverse the structure; and furthermore, our negative token learning and quality-enhanced sampling contribute to the quality improvement.

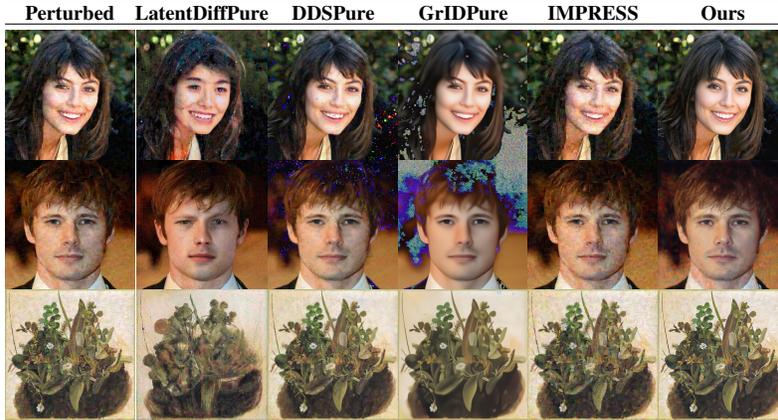


Figure 5: Visualization of purified images that were originally protected by MetaCloak. Our method shows high faithfulness and high quality, while others fail to effectively purify the perturbation.

**Efficiency and Faithfulness of Purification.** We present the evaluation of time cost and faithfulness compared to all diffusion-based purification in Table 2. The time cost is measured in seconds per sample with consideration of model loading. Compared to other methods, our purification has the lowest time cost and is  $10\times$  faster than the previous SoTA, IMPRESS. The reason behind this is that we leverage the super-resolution module, which empowers the usage of skip-step sampling to boost the generation time. Moreover, we test the purification faithfulness of each method in terms of LPIPS loss [33], a common metric measuring the visual perception distance of two images. From Table 2, we can see that our method achieves the lowest LPIPS loss. To visually validate this, we additionally present the purified images in Figure 5. From the figure, we can see that other diffusion-based approaches have limitations in hallucinating the content, introducing severe artifacts, or not having enough purification strength. In particular, we observed that LatentDiffPure causes a great change in identity during the purification while GrIDPure brings some artifacts. In comparison, our purification greatly improves faithfulness with off-the-shelf image restoration models that are prone to have nearly the same structure input and output.

Table 2: Faithfulness and time cost of different diffusion-based purifications.

Methods	LPIPS ↓	Time Cost↓(s)
IMPRESS	0.451	675
PixelDiffPure	0.495	102
DDSPure	0.384	122.5
GrIDPure	0.429	92.75
LatentDiffPure	0.453	63.75
LatentDiffPure- $\emptyset$	0.450	63.25
<b>Ours</b>	<b>0.271</b>	<b>51</b>

### 5.3 Resilience Against Adaptive Perturbations

DNN-based purification is prone to further adaptive attacks due to the non-smoothness in terms of latent representation space [11] and also the vulnerability by exploiting adversarial examples [9]. To validate whether our framework can still work upon further adversarial perturbation with new knowledge of our pipeline, we additionally conduct experiments on evaluations of different variants of our approach before and after the adaptive perturbation crafted against the image purification part. The adversarial perturbation is crafted following AdvDM [7] with consideration of the CFG [39] sampling trajectory with a large perturbation budget of  $r = 16/255$ . For the model variants, we consider the full variant with both modules turned on, and also the ablated versions with one of them turned off. From Table 3, we can see that the full variant with Neg. Token is the most effective and robust one under adaptive attack across other variants. Furthermore, we notice that the variant with only SR is also robust to the adaptive attack since it yields the lowest change in average score after the attack. However, only using SR is not enough, since when we

Table 3: Different variants under Adaptive Attack (AA).

Method	Neg. Token	Before AA			After AA			
		IMS	Q	Avg.	IMS	Q	Avg.	$\Delta$
Ours-CodeSR	x	0.18	0.41	<b>0.29</b>	-0.06	0.03	-0.01	↓0.30
	✓	<b>0.23</b>	-0.03	0.10	-0.09	<b>0.24</b>	<b>0.08</b>	↓0.02
Ours-Code	x	0.14	<b>0.42</b>	0.28	-0.09	-0.17	-0.13	↓0.41
	✓	0.05	0.11	0.08	-0.16	-0.47	-0.32	↓0.40
Ours-SR	x	0.07	-0.02	0.03	<b>0.20</b>	-0.39	-0.09	↓0.12
	✓	0.07	-0.02	0.03	-0.03	0.08	0.02	↓ <b>0.01</b>

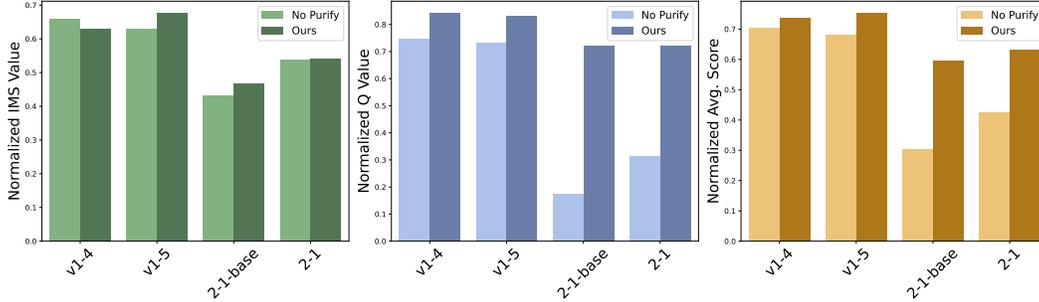


Figure 6: Evaluation of generation performance of models across different stable diffusion versions.

remove the CodeFormer, the purification is sub-optimal. These results conclude that our method is robust to further adaptive perturbation and potentially a once-for-all defense approach.

## 5.4 Ablation Study

**Contribution of Modules.** We present full ablations on the three modules in our method in Table 4. From the table, our method works best under the full setting. When turning off any of the modules, the average performance degrades, with turning off Neg. Token suffers the most. Surprisingly, when only enabling the SR module, the IMS score is relatively good but with bad quality. For the settings that only enable one of the modules, we find that enabling Neg. Token also yields the best performance. These indicate that Neg. Token is quite important when training on purified images. Furthermore, for the settings that only allow two modules enabled, we found that the SR+Neg.Token yields the best performance, with just some gap between the IMS scores. In conclusion, the results suggest that all the modules contribute to the learning performance gain in IMS and quality.

**Different Models and Radii.** We additionally validate the effectiveness of our method under different LDM versions. From Figure 6, we can see that across different stable diffusion versions, models trained on our purified data yield better performance in terms of IMS, Q, and average scores under most of the settings. And the major improvement is in the quality part. We also demonstrate the effectiveness of our method across different perturbation radii at App. B. These results conclude the generalizability of our method across settings.

Table 4: Ablation study.

Settings			Metrics		
CodeF.	SR	Neg. Token	IMS $\uparrow$	Q $\uparrow$	Avg. $\uparrow$
✓	✓	✓	<b>0.266</b>	0.374	<b>0.320</b>
✓		✓	0.199	<b>0.433</b>	0.316
	✓	✓	0.153	0.209	0.181
		✓	0.207	0.105	0.156
✓	✓		0.147	0.039	0.093
✓			0.184	-0.042	0.071
	✓		0.234	-0.378	-0.072
			0.096	-0.634	-0.269

## 6 Conclusion, Limitations, and Future Works

In this paper, we dive into the underlying mechanism behind the effectiveness of existing error-maximizing protective perturbation approaches against the unauthorized fine-tuning of personalized diffusion models. Motivated by the latent mismatch observation, we propose to use super-resolution and image restoration models for latent realignment. Furthermore, we propose contrastive learning with negative tokens based on the analysis from the shortcut learning perspective. Extensive experiments demonstrate the effectiveness, efficiency, and faithfulness of our method.

Despite being mainly evaluated on the facial dataset, our framework is generally designed and can be adapted for various image domains, like objects and artwork. This indicates that it might be used to further invade those already “protected” data with previous perturbations, which might create some negative societal impact but also promote the advancement of this domain. Furthermore, in Section 5.3, we find that CodeFormer might be more vulnerable under adaptive attacks, but it does better in purification than the SR. Thus, there might still be room to improve in determining the optimal layer combination of the SR and CodeFormer modules to maximize the robustness-effectiveness trade-off.

## References

- [1] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [2] 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.
- [3] Midjourney. <https://www.midjourney.com>.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Yixin Liu, Chenrui Fan, Yutong Dai, Xun Chen, Pan Zhou, and Lichao Sun. Toward robust imperceptible perturbation against unauthorized text-to-image diffusion-based synthesis. *arXiv preprint arXiv:2311.13127*, 3, 2023.
- [9] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.
- [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [11] Jiayi Guo, Xingqian Xu, Yifan Pu, Zanlin Ni, Chaofei Wang, Manushree Vasu, Shiji Song, Gao Huang, and Humphrey Shi. Smooth diffusion: Crafting smooth latent spaces in diffusion models. *arXiv preprint arXiv:2312.04410*, 2023.
- [12] Bochuan Cao, Changjiang Li, Ting Wang, Jinyuan Jia, Bo Li, and Jinghui Chen. Impress: Evaluating the resilience of imperceptible perturbations against unauthorized data usage in diffusion-based generative ai. *Advances in Neural Information Processing Systems*, 36, 2024.
- [13] Zhengyue Zhao, Jinhao Duan, Kaidi Xu, Chenan Wang, Rui Zhang, Zidong Dup Qi Guo, and Xing Hu. Can protective perturbation safeguard personal data from being exploited by stable diffusion? *arXiv preprint arXiv:2312.00084*, 2023.
- [14] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.
- [15] Boyang Zheng, Chumeng Liang, Xiaoyu Wu, and Yan Liu. Understanding and improving adversarial attacks on latent diffusion model. *arXiv preprint arXiv:2310.04687*, 2023.
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, 2020.
- [17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.
- [18] Jooyoung Choi, Yunjey Choi, Yunji Kim, Junho Kim, and Sungroh Yoon. Custom-edit: Text-guided image editing with customized diffusion models, 2023.

- [19] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation, 2022.
- [20] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing, 2023.
- [21] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35: 25005–25017, 2022.
- [22] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *CVPR*, 2023.
- [23] 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.
- [24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. *arXiv preprint arXiv:2307.06949*, 2023.
- [25] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023.
- [26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [27] 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.
- [28] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *The Twelfth International Conference on Learning Representations*, 2023.
- [29] Jingyao Xu, Yuetong Lu, Yandong Li, Siyang Lu, Dongdong Wang, and Xiang Wei. Perturbing attention gives you more bang for the buck: Subtle imaging perturbations that efficiently fool customized diffusion models. *arXiv preprint arXiv:2404.15081*, 2024.
- [30] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples, 2018.
- [31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [32] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022.
- [33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [35] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.

- [36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [38] Guangming Liu, Xin Zhou, Jianmin Pang, Feng Yue, Wenfu Liu, and Junchao Wang. Codeformer: A gnn-nested transformer model for binary code similarity detection. *Electronics*, 12(7):1722, 2023.
- [39] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [40] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018. doi:[10.1109/FG.2018.00020](https://doi.org/10.1109/FG.2018.00020).
- [41] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015.
- [42] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.
- [43] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.
- [44] Hu Ye, Jun Zhang, Sibio Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023.
- [45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [46] Yaniv Taigman, Ming Yang, Marc’ Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [47] Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14071–14081, 2023.
- [48] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023.
- [49] Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.
- [50] 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 (CVPR)*, pages 10684–10695, June 2022.
- [51] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [52] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4. IEEE, 2021. doi:[10.1109/ICEET53442.2021.9659697](https://doi.org/10.1109/ICEET53442.2021.9659697). URL <https://doi.org/10.1109/ICEET53442.2021.9659697>.

- [53] W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoi-son: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.
- [54] Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. Black-box backdoor defense via zero-shot image purification. *Advances in Neural Information Processing Systems*, 36, 2024.
- [55] Bao Wang, Alex T. Lin, Wei Zhu, Penghang Yin, Andrea L. Bertozzi, and Stanley J. Osher. Adversarial defense via data dependent activation function and total variation minimization, 2020.
- [56] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=rJz1BfZAb>.
- [57] Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023.
- [58] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *Proceedings of the 29th USENIX Security Symposium*, 2020.
- [59] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 53–62, 2020.
- [60] Qidong Huang, Jie Zhang, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. Initiative defense against facial manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1619–1627, 2021.
- [61] Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John P Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [62] Chaofei Yang, Leah Ding, Yiran Chen, and Hai Li. Defending against gan-based deepfake attacks via transformation-aware adversarial faces. In *2021 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2021.

## A Experiment Details

### A.1 Hardware and Training Details

**Hardware Details.** All the experiments are conducted on an Ubuntu 20.04.6 LTS (focal) environment with 503GB RAM, 10 GPUs (NVIDIA® RTX® A5000 24GB), and 64 CPU cores (Intel® Xeon® Silver 4314 CPU @ 2.40GHz). Python 3.9.18 and Pytorch 1.13.1 are used for all the implementations.

**Training and Inference Settings.** The Stable Diffusion (SD) v2-1-base [50] is used as the model backbone. For Dreambooth training, we conduct full fine-tuning, which includes both the text-encoder and U-Net model with a constant learning rate of  $5 \times 10^{-7}$  and batch size of 2 for 1000 iterations in mixed-precision training mode. We use the 8-bit Adam optimizer [51] with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  under bfloat16-mixed precision and enable the xformers for memory-efficient training. For calculating prior loss, we use 200 images generated from Stable Diffusion v2-1-base with the class prompt “a photo of a [class norm]”. The weight for prior loss is set to 1. For the evaluation phase, we set the inferring steps as 100 with prompts “a photo of sks person” and “a smiling photo of sks person” during inference to generate 16 images per prompt. For all the settings, the Classifier-free Guidance (CFG) is turned on by default with a guidance scale of 7.5.

### A.2 Metrics

In this section, we describe the evaluation metrics used in our experiments in more detail. Following [8], we use CLIP-IQAC, which calculates the CLIP score difference between “a good photo of [class]” and “a bad photo of [class]”. For calculating IMS-VGGNet, we leverage the VGGNet in the DeepFace library for face recognition and face embedding extraction [52]. For IMS-IP, we leverage *antelopev2* model from InsightFace library [43] following IP-adapter [44]. We report the weighted average of them with a weighting factor on IMS-IP as 70% since we find it yields a more stable evaluation with IMS-VGG as 30%. We compute all the mean scores over all generated images and all instances. For the instance  $i$  and its  $j$ -th metric, its  $k$ -th observation value is defined as  $m_{i,j,k}$ . For the  $j$ -th metric, the mean value is obtained with  $\sum_{i,k} m_{i,j,k} / (N_i N_k)$ , where  $N_i$  is the instance number for that particular dataset, and  $N_k$  is the image generation number.

## B More Experimental Results

**Different Radii.** We present the evaluation of the models trained on perturbed data and purified data under different perturbation radii in Table 8. As the results shown in the Table, we found that our method consistently yields better models in terms of IMS and Q score. This demonstrates that our method is insensitive to different perturbation radii.

**Different Negative Tokens.** To investigate the effect of using different negative tokens, we additionally present results in Table 5 and Figure 7. As we can see, setting the right negative tokens is crucial for the performance. In our four tested prompts, we found that “UNKNOWN artifact masked” yielded the best performance in both IMS and Q scores. Future works can be conducted using automatic negative prompt searching. Another direction is to study visualization of the learned pattern for each negative prompt setting for a deeper understanding of personalized diffusion models.

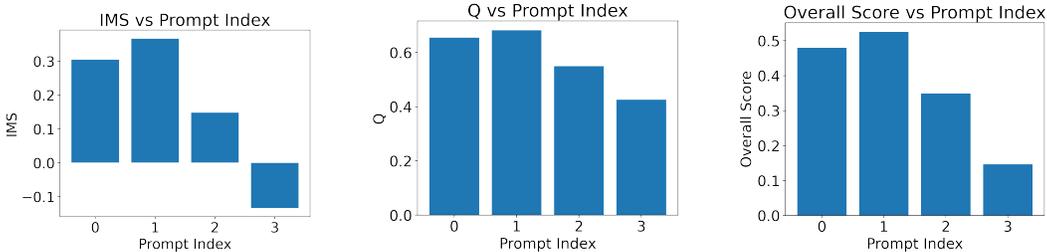


Figure 7: Scores vs Prompt Index.

---

**Algorithm 1 Training clean LDM on adversarial perturbed data with systematic defending (Summary of Sec 4.2)**

---

**Input:** Corrupted training set  $X'_0$ , LDM  $\theta_0$ , CodeFormer  $\phi = \{\mathcal{E}_\phi, \mathcal{D}_\phi, \mathcal{T}_\phi, \mathcal{C}\}$ , SR model  $\psi$ , Prior data  $\bar{X}_0$ , Negative Token  $\mathcal{V}^*_{*N}$ , Instance prompt  $c^{ID}$ , Class prompt  $c$ , #Generation  $N_{gen}$ , Quality negative prompt  $\bar{\mathcal{P}}_Q$ , Personalized identifier  $\mathcal{V}^*$

**Output:** Personalized diffusion model with clean-level generation performances  $\theta_T$ , and clean personalized generation  $X_{gen}$

- 1: **// Step 1: Input Purification**
  - 2: *CodeFormer*: Low-quality latent  $Z_l = \mathcal{E}_\phi(X'_0) \Rightarrow$  Code prediction  $\tilde{Z}_c = \mathcal{T}(Z_l, \mathcal{C}; \phi) \Rightarrow$  High-quality restoration  $\tilde{X}_0 = \mathcal{D}_\phi(\tilde{Z}_c)$
  - 3: *Super-resolution*: Resize  $\tilde{X}_0$  to  $128^2 \Rightarrow$  Applying SR model  $\psi$  with quality-guided prompt and obtain  $\tilde{X}_0^{\text{purified}}$  in  $512^2$
  - 4: **// Step 2: Training Personalized LDM with Contrastive Decouple Learning**
  - 5: **for**  $i$  **in**  $1, \dots, T$  training steps **do**
  - 6:     Sample an instance data  $x_i$  from  $\tilde{X}_0^{\text{purified}}$ , and an class-prior data  $\bar{x}_0$  from  $\bar{X}_0$
  - 7:     Crafting decoupled instance prompt  $c_{dec}^{ID} = \text{concat}(c^{ID}, \mathcal{V}^*_{*N})$  and class-prior prompt  $c_{dec} = \text{concat}(c, \text{"without"}, \mathcal{V}^*_{*N})$ .
  - 8:     Optimize the LDM  $\theta_i$  with standard DreamBooth loss  $\mathcal{L}_{db}$  ▷ Following Eq. 2
  - 9:      $\mathcal{L}_{db}(x_i, c_{dec}^{ID}, \bar{x}_0, c_{dec}; \theta_i) = \mathcal{L}_{\text{denoise}}(x_i, c_{dec}^{ID}) + \lambda \mathcal{L}_{\text{denoise}}(\bar{x}_0, c_{dec})$ ,
  - 10:    Update LDM weight  $\theta_i$  with  $\nabla_{\theta_i} \mathcal{L}_{db}$  using AdamW optimizer on UNet Denoiser and Text Encoder
  - 11: **end for**
  - 12: **// Step 3: Noise-free Sampling Process during Inference**
  - 13: **for**  $j$  **in**  $1, \dots, N_{gen}$  **do** ▷ Following Eq. 6 with Classifier-free Guidance
  - 14:     Sampling  $X_{gen}^j$  with personalized identifier  $\mathcal{V}^*$ , and negative prompt of  $\text{concat}(\mathcal{V}^*_{*N}, \bar{\mathcal{P}}_Q)$  with personalized LDM  $\theta_T$
  - 15: **end for**
  - 16: **return**  $\theta_T, X_{gen}$
- 

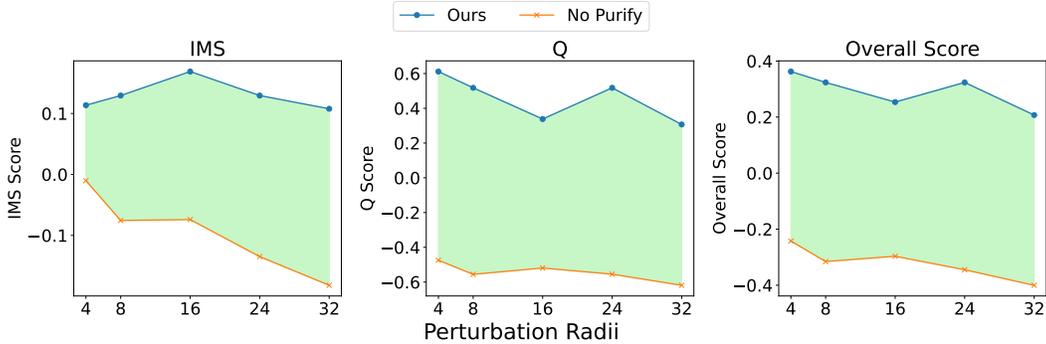


Figure 8: Evaluation of trained models on perturbed and purified images across perturbation radii.

Table 5: Evaluation on our method with different negative tokens.

IMS	Q	Overall Score	#Prompt Index	Negative Tokens
0.305	0.655	0.480	0	UNKNOWN degraded quality
0.367	0.682	0.525	1	UNKNOWN artifact masked
0.148	0.550	0.349	2	UNKNOWN noise corruption
-0.134	0.426	0.146	3	BHI noise corruption



Figure 9: More visualization of perturbed and purified images.

## C Implementation of Baselines

### C.1 Purification Methods

We implement two classes of purification approaches; the first ones are model-free and operate with certain image processing algorithms, such as Gaussian Filtering, total variation minimization (TVM), and JPEG compression. Despite the simplicity, researchers found that these approaches can achieve non-trivial defense performance against adversarial attacks [7], availability attacks [8, 4], and more general data poisoning attacks [53]. Another line of approach is based on powerful diffusion probabilistic models, which have a strong ability to model real-world data distribution

and also show potential in being leveraged for zero-shot purifiers [54, 13, 49, 12]. We include a wide range of SoTA diffusion-based purification approaches that are designed for the protective perturbation specifically, including GrIDPure [15], IMPRESS [12], or those are proposed for more general adversarial perturbation [14, 49], including DiffPure [14] (with pixel-space diffusion models or latent-space diffusion models), DDS-based purification (DDSPure) [49].

**1. Gaussian Filtering.** Gaussian Filtering is a well-known image-processing technique used to reduce image noise and detail by applying a Gaussian kernel. The high-frequency part in adversarial perturbation can be smoothed after filtering. We set the kernel size as 5 following [4].

**2. Total Variation Minimization (TVM) [55]** The main idea of TVM is to conduct image reconstruction based on the observation that the benign images should have low total variation. We implemented the TVM defense in the following steps: we first resized the instance image to 64x64 pixels, applied a random dropout mask with a 2% pixel dropout rate, and solved a TVM optimization problem. The optimization aims to reconstruct the image by minimizing the difference between the original and reconstructed images while enforcing smoothness through the total variation term:  $\min_Z \|(1 - X) \odot (Z - x)\|_2 + \lambda_{TV} TV_2(Z)$ . After optimization, the reconstructed image is reshaped back to 64x64 and then upsampled to 512x512 through two SR steps with a middle resizing process.

**3. JPEG Compression.** It involves transforming an image into a format that uses less storage space and reduces the image file’s size. We set the JPEG quality to 75 following [8].

**4. DiffPure [14].** Diffusion Purification (DiffPure) first diffuses the adversarial example with a small amount of noise given a pre-defined timestep  $t$  following a forward diffusion process, where the adversarial noise is smoothed and then recovers the clean image through the reverse generative process. Depending on the type of diffusion model used, this simple yet effective approach can be adapted into two versions: PDM-based DiffPure and LDM-based DiffPure. In our implementation, we term the PDM-based DiffPure as *PixelDiffPure* for short and leverage 256x256\_diffusion\_uncond pre-trained on ImageNet released in the guided-diffusion following common practice. For the LDM-based DiffPure, we term it as LatentDiffPure since the diffusion process is conducted in latent space and leverage Stable Diffusion v1-4 [50] for its superior performance. Since the SD model has the ability to input additional text prompts during the purification process, we investigate two variants with and without the usage of purified text prompting. For LatentDiffPure- $\emptyset$ , we set the text to null, while for LatentDiffPure, we set it as “a photo of [class\_name], high quality, highres”.

**5. DDSPure [49].** Similar to DiffPure [14], the main idea behind Diffusion Denoised Smoothing (DDS) is to find an optimal timestamp that can maximally remove the adversarial perturbation via the SDEdit process [31]. Given smoothing noise level  $\delta$ , the optimal timestamp  $t^*$  is computed via,  $t^* = \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} = \sigma^2$ . Following common practice, we leverage the pre-trained diffusion model on ImageNet released in the guided-diffusion. Specifically, the 256x256\_diffusion\_uncond is used as a denoiser. To resolve the size mismatch, we resize the images to fit the model input and resize the image size back after purification. And we clip  $t^*$  when it falls outside the sampling step range of [0, 1000].

**4. GrIDPure [15].** GrIDPure notices that for purification in defending protective perturbation, conducting iterative DiffPure with small steps can outperform one-shot DiffPure with larger steps. Furthermore, it suppresses the generative nature during diffusion purification by additionally splitting the image into multiple small grids that are separately processed with a final merging process. This allows the model to focus more on purifying those perturbed textures and curves in the image without mistakenly affecting the overall structure, thus preserving the faithfulness of purification.

---

**Algorithm 2** GrIDPure

---

**Input:** Input image  $x_0$ , number of iterations  $N$ , time-stamp  $t$ , grid size  $g$ , stride  $s$ , merging weight  $\gamma$

**Output:** Purified image  $x_N$

```
1: for  $i = 0$  to  $N - 1$  do
2:   Split  $x_i$  into grids of size  $g \times g$  with stride  $s$ 
3:   for each grid  $x_{i,j}$  do
4:     Apply DiffPure with time-stamp  $t$  to obtain  $\tilde{x}_{i,j}$ 
5:   end for
6:   Merge all  $\tilde{x}_{i,j}$  to obtain  $\tilde{x}_i$ , averaging pixel values in overlapping regions
7:    $x_{i+1} = (1 - \gamma) \cdot \tilde{x}_i + \gamma \cdot x_i$ 
8: end for
9: return  $x_N$ 
```

---

Given an input image size of  $512 \times 512$ , we implement the GrIDPure algorithm as follows with the hyper-parameter recommended in the original paper. We first obtain multiple grids using a sliding window approach. The window size is  $256 \times 256$ , and the stride is 128. For each  $256 \times 256$  grid, we apply DiffPure with a time-stamp of  $t = 10$ . After all the grids are denoised, they are merged back into a single image. In the overlapping regions, the pixel values are averaged. Given  $\gamma$  as 0.1, the purified image is then obtained via a moving average with the original image,

$$x_{i+1} = (1 - \gamma) \cdot \tilde{x}_i + \gamma \cdot x_i. \quad (7)$$

These steps constitute one iteration, and the algorithm is repeated for a total of 10 iterations.

**6. IMPRESS [12]** The key idea of IMPRESS is to conduct purification that ensures *latent consistency with visual similarity constraints*: (1) the purified image should be visually similar to the perturbed image, and (2) the purified image should be consistent upon a LDM-based reconstruction. To quantify the similarity condition, IMPRESS uses the LPIPS metric [33], which measures the human-perceived image distortion between the purified image  $x_{\text{pur}}$  and the perturbed image  $x_{\text{ptb}}$ . The loss is defined as  $\max(\text{LPIPS}(x_{\text{pur}}, x_{\text{ptb}}) - \Delta_L, 0)$ , where  $\Delta_L$  is the perceptual perturbation budget. For the consistency condition, IMPRESS simplifies the loss by removing the diffusion process and defines it as  $\|x_{\text{pur}} - \mathcal{D}(\mathcal{E}(x_{\text{pur}}))\|_2^2$ , where  $\mathcal{E}$  and  $\mathcal{D}$  are the image encoder and decoder in the LDM, respectively. The final optimization problem combines the two losses:

$$\min_{x_{\text{pur}}} \|x_{\text{pur}} - \mathcal{D}(\mathcal{E}(x_{\text{pur}}))\|_2^2 + \alpha \cdot \max(\text{LPIPS}(x_{\text{pur}}, x_{\text{ptb}}) - \Delta_L, 0), \quad (8)$$

where  $\alpha$  is a hyperparameter to balance the two losses, which is set as 0.1. The optimization is solved with PGD [56] with Adam optimizer with lr of 0.001, and the total iteration is set as 3000.

## C.2 Protective Perturbation Methods

We test a wide range of protective perturbation approaches, including those that craft noise against fixed LDMs by exploiting the out-of-distribution adversarial vulnerability of DNNs [7, 57, 28, 5, 6], and those that jointly and alternatively learn the noise generator and perturbation [4, 8, 29], which show better protection capacity for the LDM fine-tuning settings [22, 23].

**Fully-trained Surrogate Model Guidance (FSMG).** Following [58, 58, 59], FSMG employs a surrogate DreamBooth model with original parameters  $\theta_{\text{clean}}$  fully finetuned on a small subset of clean samples  $\mathcal{X}_A \subset \mathcal{X}$ . We implement the subset with the same identity to maximize the protection capability. Using  $\theta_{\text{clean}}$  as guidance, we find the optimal noise for each target image:  $\delta^{*(i)} = \text{argmax}_{\delta^{(i)}} \mathcal{L}_{\text{cond}}(\theta_{\text{clean}}, x^{(i)} + \delta^{(i)})$ , where  $\mathcal{L}_{\text{cond}}$  is the conditional denoising loss. This encourages any DreamBooth model finetuned on the perturbed samples to deviate from  $\theta_{\text{clean}}$  and generate low-quality images.

**Alternating Surrogate and Perturbation Learning (ASPL).** Since FSMG fails to effectively solve the underlying bi-level optimization, inspired by [60], ASPL further alternates the training of the surrogate DreamBooth model with perturbation learning. The surrogate model  $\epsilon_\theta$  is initialized with pre-trained weights. In each iteration, a clone  $\epsilon'_\theta$  is finetuned on clean reference data to simulate the learning trajectory on potential leaked clean data. This model is then used to expedite learning adversarial noises  $\delta^{(i)}$  with denoising-error-maximization in the current loop. Finally, ASPL updates

the actual surrogate model  $\epsilon_\theta$  on the updated adversarial samples with gradient descent and proceeds to the next iteration. This procedure allows the surrogate model to mimic better the models trained by malicious DreamBooth users, as it is only trained on perturbed data.

**Ensemble-based ASPL (EASPL).** Since the model trainer’s pre-trained text-to-image generator is often unknown, an improved approach is to use an ensemble [61, 62] of surrogate models finetuned from different pre-trained generators, which can lead to better transferability. We implement this approach with three surrogates. Besides, we follow the practice of a single model at a time in an interleaving manner to produce optimal perturbed data due to GPU memory constraints.

**MetaCloak.** Despite the effectiveness of perturbation crafted from noise-surrogate joint learning, studies find that these approaches lack robustness against simple data transformations such as minor Gaussian filtering. To address this issue, MetaCloak [8] solves the underlying bi-level poisoning problem using a meta-learning framework with an additional transformation sampling process to craft transferable and robust perturbations. Incorporating an additional transformation process and a denoising-error maximization loss brings severe performance degradation in generation.

**PhotoGuard.** PhotoGuard [5] mainly focuses on the setting of malicious editing where the diffusion models are fixed. It introduces two target-adversarial-perturbation-based (TAP-based) approaches: encoder attack and diffusion attack. The encoder attack adds a perturbation  $\delta_{\text{enc}}$  to an image  $\mathbf{x}$  such that the image encoder  $\mathcal{E}$  produces a closer latent representation for  $\mathbf{x} + \delta_{\text{enc}}$  and a target image  $\mathbf{x}_{\text{target}}$ . The diffusion attack crafts a perturbation  $\delta_{\text{diff}}$  such that the LDM-reconstructed images based on the input are closer to some  $\mathbf{x}_{\text{target}}$ . The diffusion attack considers the whole LDM model with prompts, achieving better empirical performance but being less efficient compared to the encoder attack.

**GLAZE.** GLAZE [6] mainly focuses on artwork protection and aims to add perturbations to an artist’s artworks such that LDMs cannot learn the correct style from the perturbed artworks. Similar to the TAP-based encoder attack in PhotoGuard, it first chooses a target style  $T$  sufficiently different from the style of the original image  $\mathbf{x}$ . Then, it transfers  $\mathbf{x}$  to the target style using a pre-trained style transfer model  $\Omega$ . Given the style-transferred image  $\Omega(\mathbf{x}, T)$ , GLAZE crafts the perturbation  $\delta_{\text{GLAZE}}$  by minimizing the distance between the encodings of  $\Omega(\mathbf{x}, T)$  and  $\mathbf{x} + \delta$  while regularizing the perceptual distortion using LPIPS. This encourages LDMs to generate samples with the target style instead of the original style when learning from the perturbed images.

**AdvDM.** Different from the above targeting attack, AdvDM [7] is proposed to optimize the adversarial perturbation in an untargeted and denoising-error-maximizing way. In detail, instead of learning a perturbation over one single reverse process, AdvDM learns the Monte-Carlo estimation of adversarial perturbation by sampling across all timestamps to maximize the denoising loss at each step.