Towards Robust Defense against Customization via Protective Perturbation Resistant to Diffusion-based Purification

Supplementary Material

A. Details on Preliminaries

Here, we briefly review and supplement the preliminary knowledge from Sec. 3 to help our readers better understand the various tasks involved in this paper.

Fine-tuning Customization (Personalization). Fine-tuning methods aim to inject specific concepts into the pre-trained SD for customization. Among them, DreamBooth (DB) [14] is widely studied for anti-customization. This approach not only minimizes $\mathcal{L}_{ldm}$ in few-shot scenarios but also incorporates a prior-preservation term to retain beneficial prior knowledge, thus mitigating forgetting. Its training objective can be formalized as:

$$\mathcal{L}_{db}(x_0;\theta_c) = \mathcal{L}_{ldm}(x_0;\theta_c) + \lambda \underbrace{\mathbb{E}_{\epsilon',t'}\|\epsilon' - \epsilon_{\theta_c}(z^{pr}_{t'}, t', \tau_{\theta_c}(y^{pr}))\|^2_2}_{\text{Class-Specific Prior Preservation Loss}}, \quad (14)$$

where the class prior image x^{pr} is generated by the pre-trained model with class prompt y^{pr}, and $z^{pr}_0 = \mathcal{E}(x^{pr})$ diffuses at timestep t' to form $z^{pr}_{t'}$. In $\mathcal{L}_{db}$, the loss term $\mathcal{L}_{ldm}$ employs instance prompts y of the form "a photo of [V][class noun]," where [V] acts as an identifier describing the target concept.

LoRA [4] is proposed to accelerate the optimization of large-scale pretrained models. It freezes the pretrained weights and injects trainable rank decomposition matrices into each layer, greatly reducing the number of trainable parameters for downstream tasks:

$$W' = W_0 + A \cdot B, \quad (15)$$

where W' and W_0 are fine-tuned and original weights, respectively, while $A \in \mathbb{R}^{m\times r}$ and $B \in \mathbb{R}^{r\times n}$ are low-rank matrices with rank $r \ll \min(m,n)$. LoRA can be used in conjunction with DB for efficient SD fine-tuning.

Adversarial Attack. In attacks against classifiers, white-box methods like I-FGSM [6, 7] or PGD [10] are commonly used, which can be formalized as:

$$x^{adv}_{t+1} = \Pi_{x_0,\eta}\left(x^{adv}_t + \alpha \cdot \text{sgn}\left(\nabla_{x^{adv}_t}\mathcal{L}(x^{adv}_t, y;\theta)\right)\right), \quad (16)$$

where $\Pi_{x_0,\eta}(\cdot)$ restricts inputs within the l_∞-ball of radius η around x_0, $\text{sgn}(\cdot)$ represents the sign function, and α is the learning rate. $\mathcal{L}(x^{adv}_t, y;\theta)$ is the loss used by the classifier with parameters θ, where x^{adv}_t is the adversarial sample at the t-th PGD step and y is the corresponding ground truth label. In brief, PGD iteratively finds the most adversarial noises for the model with parameters θ by maximizing the loss via gradient ascent.

Anti-customization. We explain the intuition behind the simplification that transforms our original maximizing objective from $\min_{\theta_c}\mathbb{E}_x\mathcal{L}_{ldm}(x_0+\delta;\theta_c)$ to $\mathcal{L}_{ldm}(x_0+\delta;\theta_c)$ as mentioned above. The key lies in the relationship between the model's training data x (of $\mathbb{E}_x$) and the adversarial data $x_0+\delta^{adv}$. For optimal performance, the training set should encompass adequately trained adversarial samples. However, this creates a bootstrap paradox: fine-tuned θ_c is needed for optimal δ^{adv} while δ^{adv} is needed for optimal θ_c, which is why surrogate models fine-tuned on clean data are frequently employed for simplification.

In the context of fine-tuning methods like Textual Inversion [3], which make no change to the internal parameters of SD, such an issue exists no more in practice. For DB (full fine-tuning) and LoRA (PEFT), simply using a model fine-tuned on clean images as a surrogate leads to Fully-trained Surrogate Model Guidance (FSMG) formalized in Anti-DB [19]. A more promising alternative, also proposed by Anti-DB, is to iteratively introduce insufficient adversarial samples, generated at different PGD steps, into the surrogate model alongside clean images. This approach is referred to as Alternating Surrogate and Perturbation Learning (ASPL).

Purification. In its original paper, DiffPure [12] is introduced via Stochastic Differential Equation (SDE). Since we use the specialized DDPM-based purification model, and considering that SD is commonly implemented discretely, the introduction of diffusion-based purification in this paper is also written in the DDPM form. We present the more generalized SDE form from the original DiffPure here for both quick reference and rigor. For a Variance Preserving SDE (VP-SDE) where drift and diffusion coefficient are respectively defined as $f(x,t) := -\frac{\beta(t)}{2}x$ and $g(t) := \sqrt{\beta(t)}$, we first diffuse adversarial x^{adv} with a fixed timestep $t^p \in [0,1]$ via:

$$x(t^p) = \sqrt{\alpha(t^p)}x^{adv} + \sqrt{1-\alpha(t^p)}\epsilon, \quad (17)$$

where $\alpha(t) := e^{-\int_0^t \beta(s)ds}$, then we solve the reverse-time SDE to get the purified sample with an SDE solver sdeint:

$$\text{Pure}(x^{adv}) = \text{sdeint}(x(t^p), f_{\text{rev}}, g_{\text{rev}}, \overline{w}, t^p, 0;\theta_p), \quad (18)$$

where sdeint takes in six inputs: initial value $x(t^p)$, drift coefficient $f_{\text{rev}}(x,t) := -\frac{\beta(t)}{2}[x + 2s_{\theta_p}(x,t)]$, diffusion coefficient $g_{\text{rev}}(t) := \sqrt{\beta(t)}$, Wiener process $\overline{w}$, initial time t^p, and end time 0. In the discrete case, this whole purification process corresponds to the specialized DDPMs.

B. Details on Analysis

B.1. More Explanation on Overall Formulation

Due to the deepening of the computational graph during iterative purification denoising, full-gradient adaptive attacks lead to $\mathcal{O}(N)$ memory cost and may cause vanishing/exploding gradients. For a 2GB 256×256 unconditional DDPM purification model, fully tracking its training loss after only 5 consecutive denoising samplings requires up to 25GB memory overhead. For differentiability, DiffPure proposes the *adjoint method* to calculate full gradients of the reverse SDE with $\mathcal{O}(1)$ memory cost. However, this method of solving the augmented SDE does not reduce the time complexity. Backward Path Differentiable Approximation (BPDA) [1] is also a common approach, but the truly effective surrogate is hard to find.

Is it entirely infeasible to use full-gradient adaptive attacks? [22] mentions such a method for anti-customization, where DDIM [17] sampling strategy is utilized to ensure memory usage remains within an acceptable range. However, they report that this adaptive attack is not effective. To demonstrate the instability of purification diffusion models as probabilistic models, we set $\alpha = 0.005$, $\eta = \frac{16}{255}$, and perform a 100-step PGD attack on $\mathcal{L}_{ddpm}$. The resolution of the input images is 256×256. Subsequently, both the clean and the adversarial sample obtained from the attack are purified using DiffPure with $t^p = 50$ and $t^p = 100$, generating four sets of images, each containing 100 samples. The distributions of these sets are visualized using t-SNE [18] with perplexity set to 10, and the results are presented in Fig. 2. The convergence of the purified clean and adversarial samples motivates us to turn to the alternative by Eq. (6).

B.2. Experimental Details on the Reason Analysis

Reason 1: Lack of Vulnerable Components. Firstly, we modify Anti-DB's ASPL method to conduct PGD attacks directly in the latent space. We provide a more comprehensive experimental result here in Fig. 10, with the CLIP text encoder [13] taken into consideration. We set $\alpha = 0.005$, $\eta_z = \frac{16}{255}$, and perform a (20×5)-step PGD attack on $\mathcal{L}_{ldm}$. Fine-tuning steps per 5 PGD steps are set to 3. In the ASPL attack, we employ two configurations: one with a trainable text encoder (Latent-ASPL, trainable text encoder) and one with a frozen text encoder (Latent-ASPL, frozen text encoder), and the adversarial examples shown in Fig. 3 are obtained via the former. Actually, these two different configurations do not result in significant differences, whether in the generated adversarial samples or in the outputs obtained after fine-tuning SD on the adversarial samples.

To avoid introducing additional noise during VAE decoding and to maintain consistency in the number of channels, we directly save the adversarial latents in ".pt" format and use them to replace the corresponding instance inputs in the DreamBooth training process.

During customization, we fine-tune SD v2.1 via DreamBooth on these two kinds of adversarial samples. We also choose to either train or freeze the text encoder during fine-tuning. When jointly training the text encoder, we set the learning rate to $5e-7$, and when freezing the text encoder, we set it to $5e-6$ to ensure the capture of the target concept, with 500 steps of training for both fine-tuning configurations. The training instance prompt is "a photo of *sks* person", the class prompt is "a photo of person", and the inference prompt is "a photo of *sks* person". The training batch size is 2, with a prior loss weight of 1.0. The results are shown in Fig. 10.

Also, we directly attack $\mathcal{L}_{ddpm}$ using 200 randomly selected images in our datasets, resized to 256×256. We set $\alpha = 0.005$, $\eta = \frac{16}{255}$, and perform a 150-step PGD attack, with Monte-Carlo sampled timesteps limited in [1, 100]. Subsequently, the adversarial images and clean images are both fed into the UNet with time condition inputs ranging from 1 to 100. The purification model we use consists of 18 downsampling blocks, 1 middle block, and 18 upsampling blocks. We record MSE between the intermediate outputs of adversarial and clean images block by block under different time conditions. The average values across 200 images are computed, and the final results are presented in Fig. 4. It can be observed that, due to the increasing coefficient $\sqrt{1-\overline{\alpha}_t}$, the differences between clean and adversarial images grow with longer timesteps.

Reason 2: Frozen Parameters with Benign Priors. The adversarial samples in Fig. 5 are generated using the $\mathcal{L}_{ldm}$-attack against SD v1.5. The configuration for generating protective perturbation is largely consistent with the setup used in the experiments above, conducted in the latent space. We set $\alpha = 0.005$, $\eta = \frac{16}{255}$, and perform a 100-step PGD attack. During editing, we employ MasaCtrl [2] combined with a pretrained T2I-Adapter [11], with the condition type set to "sketch." The significant attenuation of artifacts demonstrates that this direct attack on training objectives is not fully applicable to training-free tasks.

Reason 3: Fixed High Timestep Denoising. Perhaps the statement in Sec. 4.2.3, "the purification process can be viewed as a generation process where high-timestep denoising is fixed," is not sufficiently direct. To offer a more intuitive illustration of this process, we present a simple diagram. As depicted in Fig. 11, attacks on generation span the

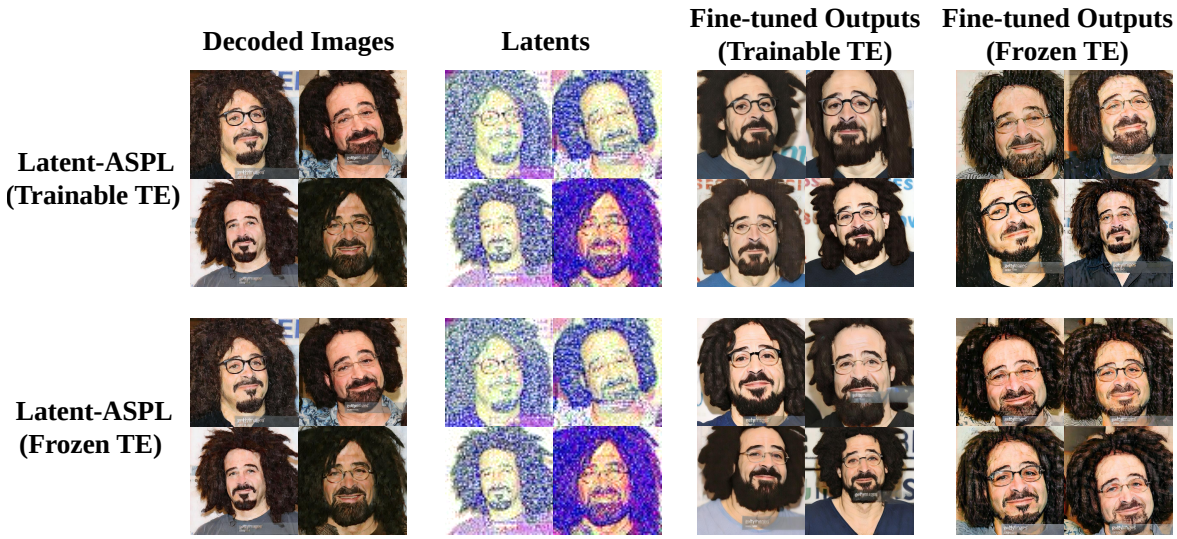

Figure 10. ASPL attacks [19] against SD in the latent space. In the ASPL attack, two configurations are used: trainable/frozen text encoder, corresponding to the two rows in the figure. Similarly, in the DreamBooth fine-tuning, the trainable/frozen text encoder configurations are also employed, corresponding to the last two columns.

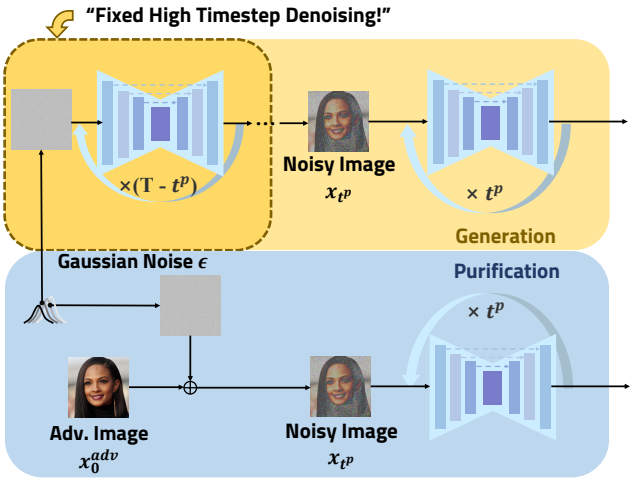

Figure 11. Why "the purification process can be viewed as a generation process where high-timestep denoising is fixed."

entire range from T (typically set to 1000) to 0. In contrast, attacks on purification are limited to a much smaller range, from t^p to 0, where the low-frequency structural information fixed during the "Fixed Higher Timestep Denoising" stage cannot be effectively altered.

C. Experimental Details

Here, we provide more configuration details used for the quantitative experimental evaluation. In Sec. 6.2 and Sec. 6.3, we use the same setup for all perturbation methods. In PGD attacks, we normalize images to $[-1, 1]$. Within this range, the noise budget η is set to $16/255$, the learning rate α is set to $5e^{-3}$, and the total PGD steps are set to 100 (20 × 5 steps for Anti-DB and SimAC). In Sec. 6.5, we evaluate the perceptual consistency of different perturbation methods with pretrained AlexNet[5]/VGG[16].

The purification and fine-tuning settings are also kept consistent. In Sec. 6.2 and Sec. 6.3, we use GrIDPure [22] for purification, applying 2 rounds of 20 iterations with $t^p = 10$, $\gamma = 0.1$ to approximate convergence of the purification effect. In Sec. 6.4, we use a finer-grained purification configuration to explore when the purification effect approximately reaches convergence. Specifically, we apply 4 rounds of 20 iterations with $t^p = 10$, $\gamma = 0.1$, on the adversarial images obtained from Anti-DB and AntiPure. Totally, Tab. 3 uses 4 rounds × 10 iters, where *P(Iter=30)-C* equals 3×10, and so on. GrIDPure mitigates image degradation caused by purification via residual connections, allowing 4×10 to rely more on intermediate results while maintaining the same computational cost as 2×20. This leads to inconsistency between the results in Tab. 1 and the results of P(Iter=40)-C in Tab. 3. But overall, the computational overhead incurred by these two settings during the purification process is the same. Here, 20-iter is the default setting of GrIDPure.

For customization, in Sec. 6.2, we fine-tune the UNet and the text encoder jointly by DreamBooth [14] with batch size of 2 and learning rate of $5e^{-7}$ for 500 training steps. The training instance prompt is "a photo of *sks* person", the class prompt is "a photo of person", and the inference prompt is "a photo of *sks* person". We also set the prior loss weight to 1.0. In Sec. 6.3, we apply LoRA [4] with the same DreamBooth settings but set the learning rate to $5e^{-5}$. The rank is set to 4. For evaluation, 30 PNG images per ID are generated, which is also consistent with the configurations

Dataset	Objective	FID↑	ISM↓ (FDFR)	BRISQUE↑
CelebA-HQ	$\mathcal{L}_{ddpm}$	69.06	0.6293 (0.09)	42.45
	$\mathcal{L}_{ddpm} + \mathcal{L}_{fre}$	65.69	0.6253 (0.08)	42.84
	$\mathcal{L}_{ddpm} + \mathcal{L}_{err-t}$	74.42	0.6489 (0.10)	37.01
	AntiPure (Ours)	**81.15**	**0.6112** (0.10)	**43.60**
VGGFace2	$\mathcal{L}_{ddpm}$	76.32	0.5958 (0.07)	39.42
	$\mathcal{L}_{ddpm} + \mathcal{L}_{fre}$	74.90	0.5644 (0.07)	45.57
	$\mathcal{L}_{ddpm} + \mathcal{L}_{err-t}$	76.75	0.5901 (0.06)	40.75
	AntiPure (Ours)	**90.77**	**0.5475** (0.05)	**46.01**

Table 6. Ablation Study on DreamBooth's [14] output quality for different AntiPure guidance following the Purification-Customization (P-C) workflow.

Dataset	Objective	FID↑	ISM↓ (FDFR)	BRISQUE↑
CelebA-HQ	$\mathcal{L}_{ddpm}$	93.79	0.6176 (0.05)	42.19
	$\mathcal{L}_{ddpm} + \mathcal{L}_{fre}$	81.32	0.5848 (0.05)	42.24
	$\mathcal{L}_{ddpm} + \mathcal{L}_{err-t}$	92.63	0.6177 (0.09)	**43.22**
	AntiPure (Ours)	**109.63**	**0.5839** (0.07)	40.01
VGGFace2	$\mathcal{L}_{ddpm}$	93.10	0.5859 (0.08)	61.79
	$\mathcal{L}_{ddpm} + \mathcal{L}_{fre}$	110.87	0.5556 (0.06)	66.01
	$\mathcal{L}_{ddpm} + \mathcal{L}_{err-t}$	102.24	0.5717 (0.06)	61.10
	AntiPure (Ours)	**127.67**	**0.5428** (0.04)	**69.97**

Table 7. Ablation Study on LoRA's [4] output quality for different AntiPure guidance following the Purification-Customization (P-C) workflow.

Dataset	Transformation	FID↑	ISM↓ (FDFR)	BRISQUE↑
VGGFace2	Crop-Scale	152.47	0.4805 (0.34)	53.60
	Rotation	92.00	0.5550 (0.05)	45.14
	None (Ours)	**90.77**	**0.5475** (0.05)	**46.01**

Table 8. Comparison of DreamBooth's [14] output quality on VGGFace2 for different transformations on AntiPure's outputs.

used in Anti-DB and SimAC.

D. More Experimental Results

D.1. Ablation Study

Our proposed AntiPure incorporates two kinds of additional guidance to address the inherent challenges of the anti-purification task: 1) Patch-wise Frequency Guidance and 2) Erroneous Timestep Guidance. In the ablation study, we gradually remove this guidance to validate the effectiveness of our method.

We use the same attack/purification/customization experimental configurations in Sec. 6.2 and Sec. 6.3 to perform the corresponding DreamBooth and LoRA fine-tuning on CelebA-HQ and VGGFace2, but with different attack targets. Specifically, our attack targets include: 1) $\mathcal{L}_{ddpm}$, 2) $\mathcal{L}_{ddpm} + \mathcal{L}_{fre}$, 3) $\mathcal{L}_{ddpm} + \mathcal{L}_{err-t}$, and we compare these results with the full AntiPure, i.e., 4) $\mathcal{L}_{ddpm} + \mathcal{L}_{fre} + \mathcal{L}_{err-t}$. The DreamBooth fine-tuning results are shown in Tab. 6, and the LoRA results are shown in Tab. 7.

It is evident that AntiPure, which combines both types of guidance, achieves the best overall performance across various metrics, datasets, and fine-tuning methods, resulting in the most significant output distortion. This represents a clear improvement over the original $\mathcal{L}_{ddpm}$-based attack. Additionally, it can be observed that among the single guidance methods, $\mathcal{L}_{fre}$ is more effective than $\mathcal{L}_{err-t}$. In fact, using $\mathcal{L}_{err-t}$ for extra guidance alone shows limited impact. However, it helps confuse the model across different time steps, thereby disrupting the frequency characteristics of the predicted noise, providing a better foundation for $\mathcal{L}_{fre}$ guidance. This is particularly evident in FID, where AntiPure sees an obvious improvement when both types of guidance are combined.

In other words, the combination of these two guidance types is not merely an additive process but achieves a synergistic "$1 + 1 > 2$" effect. Actually, the timestep inputs of the diffusion model's UNet can affect the frequency representation of the predicted noise, allowing $\mathcal{L}_{err-t}$ to be interpreted on the frequency domain like $\mathcal{L}_{fre}$. With both involved, the high-frequency components intensified by $\mathcal{L}_{fre}$ are primarily induced by erroneous high timesteps rather than real ones. Thus, the introduction of $\mathcal{L}_{err-t}$ can indirectly enhance $\mathcal{L}_{fre}$ itself, and vice versa.

D.2. Transformation Robustness

As suggested by the reviewer, we apply *Crop-Scale* (Center-Crop $\times 3/4$ side length) and *Rotation* (randomly $[0°, 15°]$) to anti-purification samples created by AntiPure, ensuring that the same transformations are applied to the original ones for fair evaluation. As shown in Tab. 8, AntiPure demonstrates robustness to *rotation*, while *crop-scale* amplifies the artifacts, leading to significantly improved performance.

D.3. More Baselines

As suggested by the reviewer, we include PhotoGuard [15] and CAAT [21] for additional comparison. We adopt the *img2img* attack pipeline for PhotoGuard, as it resembles purification more than the *inpainting* pipeline. However, as shown in Tab. 9, PhotoGuard's perturbations tend to be easily purified due to their blurred boundaries. In contrast, CAAT's perturbation closely resembles that of Anti-DB, leading to comparable robust performance.

Dataset	Perturbation	FID↑	ISM↓ (FDFR)	BRISQUE↑
VGGFace2	PhotoGuard [15]	72.25	0.6061 (0.07)	43.07
	CAAT [21]	89.07	0.5854 (0.07)	38.21
	AntiPure (Ours)	**90.77**	**0.5475** (0.05)	**46.01**

Table 9. Comparison with additional baselines on VGGFace2.

Dataset	λ_1	λ_2	t_{err}	FID↑	ISM↓ (FDFR)	BRISQUE↑
VGGFace2	0.25	0.75	999	96.33	0.5431 (0.06)	41.50
	0.50	0.50	700	90.50	0.5490 (0.04)	48.34
	0.75	0.25	999	87.81	0.5586 (0.05)	43.02
	0.50	0.50	999	**90.77**	**0.5475** (0.05)	**46.01**

Table 10. DreamBooth's [14] output quality on VGGFace2 for different hyperparameter configurations.

Fine-tuning	Perturbation	FID↑	ISM↓ (FDFR)	BRISQUE↑
DreamBooth	AdvDM [9]	82.10	0.5798 (0.06)	26.99
	Mist [8]	77.33	0.5797 (0.04)	32.58
	Anti-DB [19]	83.95	0.5686 (0.06)	27.68
	SimAC [20]	76.73	0.5762 (0.05)	26.44
	AntiPure (Ours)	**89.33**	**0.5165** (0.03)	**62.88**
LoRA	AdvDM [9]	106.48	0.5697 (0.05)	44.96
	Mist [8]	91.33	0.5731 (0.06)	55.27
	Anti-DB [19]	**115.34**	0.5591 (0.05)	46.11
	SimAC [20]	92.57	0.5622 (0.05)	45.41
	AntiPure (Ours)	112.90	**0.5101** (0.05)	**74.82**

Table 11. Comparison of DreamBooth/LoRA's [14] Stable Diffusion v1.5 output quality on VGGFace2 for different perturbation methods following the P-C workflow.

D.4. Hyperparameter Sensitivity

Originally, the selection of λ_1 and λ_2 was based on balancing the magnitude of loss components, while t_{err} was chosen to be as large as possible to maximize its effect. Here, as the reviewer suggested, we conduct a simple grid search over these three hyperparameters. As shown in Tab. 10, different metrics exhibit varying degrees of sensitivity to each parameter. Notably, the impact of t_{err} is relatively smaller compared to those of λs, while the ISM—the primary metric for identity preservation—remains largely stable across all settings. This suggests that AntiPure exhibits a certain degree of robustness with respect to its hyperparameter configurations.

D.5. Black-Box Performance

All previous experiments are conducted on SD v2.1, as recommended by Anti-DB and SimAC. However, AdvDM and Mist only support SD v1.x. We note that after sufficient purification, the effects of these perturbation methods almost completely disappear, making the distinction between SD versions insignificant.

To evaluate the performance of perturbation methods under a black-box scenario with mismatched models, and to ensure an absolutely fair SD version for all methods, we fine-tune SD v1.5 on the purified adversarial images from VGGFace2. The results are shown in Tab. 11. The similar performance observed preliminarily supports our hypothesis that "SD versions have negligible influence." Also, AntiPure still demonstrates the best overall performance.

D.6. More visualization

We provide more visualization results in Figs. 12 to 15 for qualitative evaluation. Please refer to the captions of each figure for detailed explanations. **We strongly recommend zooming in** on the following visualizations to better identify these artifacts.

It can be observed that the effects of other protective perturbation methods almost entirely vanish after sufficient purification. However, AntiPure ensures the presence of detectable artifacts, which are concentrated in the human facial regions (excluding the eyes). At lower levels of semantic distortion, these artifacts appear as unnatural high-frequency speckled regions, while more prominent artifacts manifest as patches of abnormal textures.

Furthermore, the effects of different perturbation methods on human visual perception (*iter = 0*, i.e., no purification) in Figs. 14 and 15 are also consistent with the LPIPS comparison in Tab. 4. Even under the same noise budget, Anti-DB and CAAT perturbations are more noticeable, often exhibiting blocky color artifacts. AntiPure, however, relies on frequency-domain modulation and generates samples visually closer to the original image.

Figure 12. Comparison of DreamBooth's outputs on CelebA-HQ for different perturbation methods following the Purification-Customization (P-C) workflow.

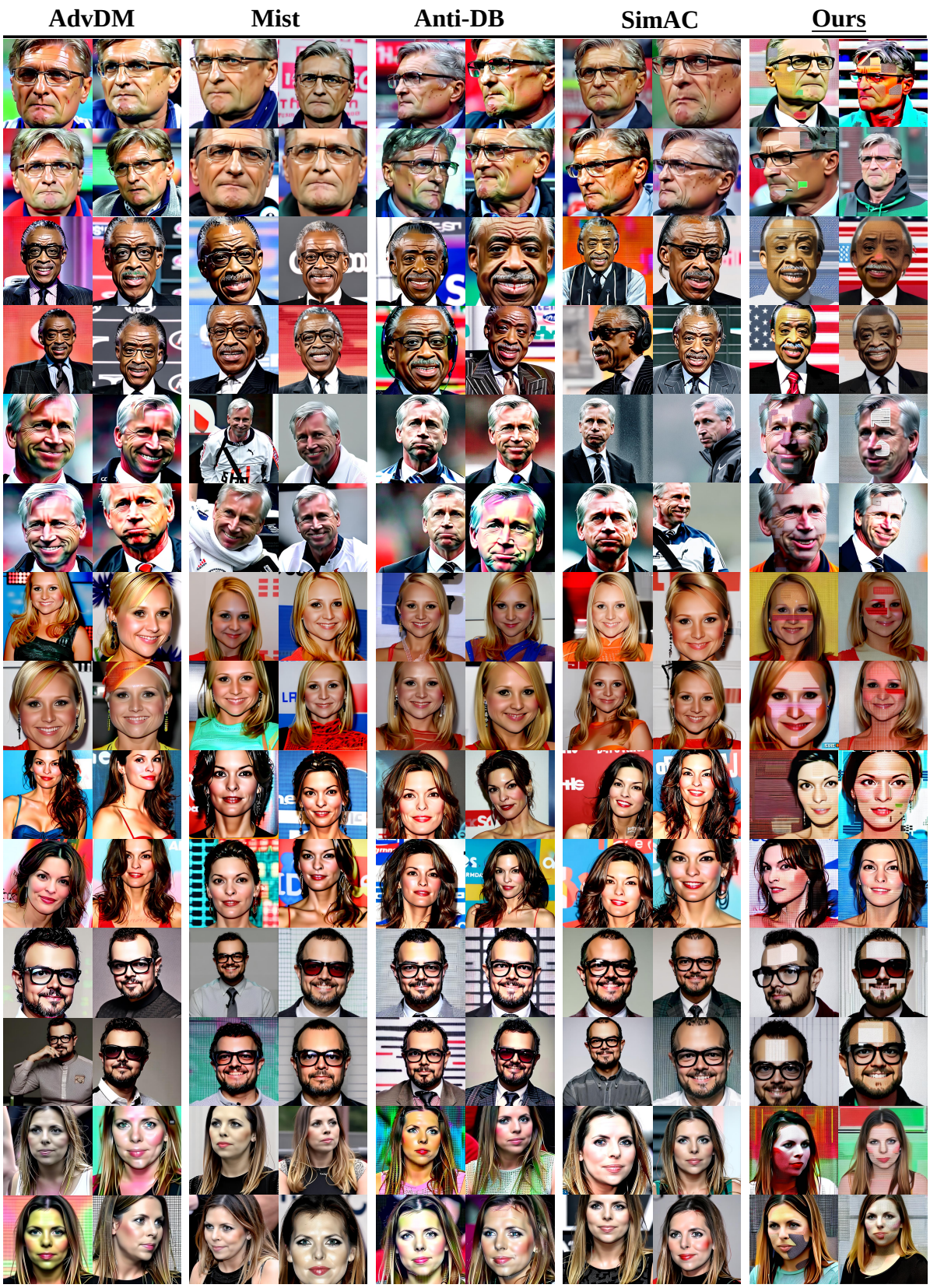

Figure 13. Comparison of DreamBooth's outputs on VGGFace2 for different perturbation methods following the Purification-Customization (P-C) workflow.

Figure 14. Comparison of GrIDPure's outputs at different iterations on VGGFace2 for different perturbation methods. Here *Iter=0* means no purification is adopted after adversarial samples are generated.

Figure 15. Comparison of GrIDPure's outputs at different iterations on VGGFace2 for different perturbation methods. Here *Iter=0* means no purification is adopted after adversarial samples are generated.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Int. Conf. Mach. Learn.*, pages 274–283. PMLR, 2018. 2

[2] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *IEEE/CVF Int. Conf. Comput. Vis.*, pages 22560–22570, 2023. 2

[3] 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. In *Int. Conf. Learn. Represent.*, 2023. 1

[4] 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. In *Int. Conf. Learn. Represent.*, 2022. 1, 3, 4

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, pages 1097–1105, 2012. 3

[6] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *Int. Conf. Learn. Represent.*, 2017. 1

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2017. 1

[8] Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023. 5

[9] 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. In *Int. Conf. Mach. Learn.* PMLR, 2023. 5

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Int. Conf. Learn. Represent.*, 2018. 1

[11] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI Conf. Artif. Intell.*, pages 4296–4304, 2024. 2

[12] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *Int. Conf. Mach. Learn.* PMLR, 2022. 1

[13] 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 *Int. Conf. Mach. Learn.*, pages 8748–8763. PMLR, 2021. 2

[14] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 22500–22510, 2023. 1, 3, 4, 5

[15] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. In *Int. Conf. Mach. Learn.* PMLR, 2023. 4, 5

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015. 3

[17] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Int. Conf. Learn. Represent.*, 2021. 2

[18] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(11), 2008. 2

[19] 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 *IEEE/CVF Int. Conf. Comput. Vis.*, pages 2116–2127, 2023. 1, 3, 5

[20] Feifei Wang, Zhentao Tan, Tianyi Wei, Yue Wu, and Qidong Huang. Simac: A simple anti-customization method for protecting face privacy against text-to-image synthesis of diffusion models. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 12047–12056, 2024. 5

[21] 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. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 24534–24543, 2024. 4, 5

[22] Zhengyue Zhao, Jinhao Duan, Kaidi Xu, Chenan Wang, Rui Zhang, Zidong Du, Qi Guo, and Xing Hu. Can protective perturbation safeguard personal data from being exploited by stable diffusion? In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 24398–24407, 2024. 2, 3