PURISHIELD: ROBUST DEFENSE AGAINST UNAUTHORIZED DIFFUSION CUSTOMIZATION VIA STRUCTURED PROTECTIVE PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

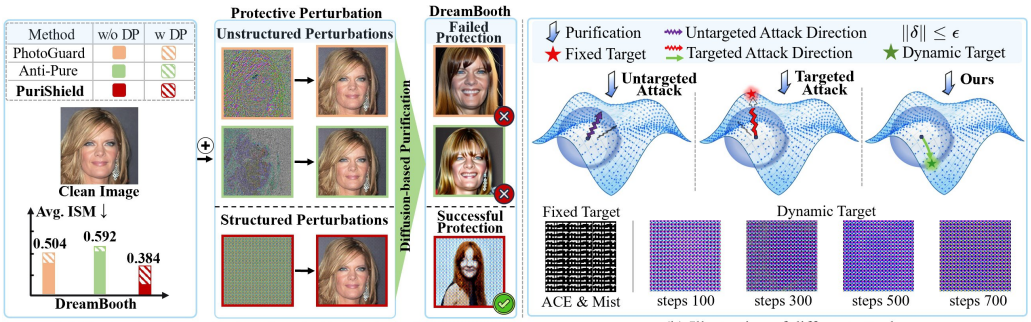
Latent diffusion models have raised severe security concerns regarding unauthorized customization. Existing anti-customization defenses protect images by injecting subtle protective perturbations, yet such unstructured perturbations suffer from poor robustness against Diffusion-based Purification (DP). To tackle this limitation, this paper proposes PuriShield, a novel anti-customization method that trains a Robust Protective Perturbation Generator (RPPG) to adaptively produce structured protective perturbations. Specifically, PuriShield incorporates three core designs: pattern density-guided generation to yield structured perturbations, latent distribution contrastive loss to refine protective perturbations, and embedded dual-head modulation to enable high-quality perturbation synthesis. Extensive experimental results demonstrate the advantages of PuriShield: ① Excellent effectiveness. It effectively defends against 5 unauthorized customization methods prior to purification. ② Strong robustness. Across 13 purification methods, it attains optimal robustness and significantly surpasses baselines against UDAP, a DP approach tailored to counter anti-customization defenses. ③ Great generalizability. It performs well in cross-task and cross-domain scenarios. ④ High efficiency. Training the RPPG takes only 31.6 minutes, with a mere 5.8 ms inference latency.

1 INTRODUCTION

Latent Diffusion Models (LDMs) (Yellapragada et al., 2025), represented by Stable Diffusion (Rombach et al., 2022), have emerged as the leading paradigm for high-fidelity text-to-image synthesis. Personalized fine-tuning methods such as DreamBooth (Ruiz et al., 2023), Textual Inversion (Gal et al., 2023), and LoRA (Hu et al., 2022) further enhance their customized generation capabilities with only a small number of reference images. Despite the promising progress, unauthorized diffusion customization poses serious security threats (Carlini et al., 2023; Wang et al., 2025; Xu et al., 2025). To mitigate such risks, recent **anti-customization defenses** (Van Le et al., 2023; Liang et al., 2023) embed imperceptible **protective perturbations** into original images, rendering protected images *unlearnable* for LDM-based customization. Typically, such methods adopt L_∞-norm constrained adversarial attacks, such as PGD (Madry et al., 2018), to iteratively optimize protective perturbations. However, existing protective perturbations remain fragile, exhibiting weak robustness against Diffusion-based Purification (DP) (Nie et al., 2022). The main reason is that existing methods tend to generate unstructured protective perturbations, which diffusion models perceive as residing off the natural data manifold. As shown in Fig. 1a, such unstructured perturbations can be easily removed by DP, leading to protection failure. This motivates us to exploit **structured protective perturbations** lying on the natural data manifold to enhance robustness against DP.

In this paper, we propose a novel anti-customization method PuriShield, which devises a Robust Protective Perturbation Generator (RPPG) to adaptively learn structured protective perturbations. Specifically, PuriShield constructs the RPPG based on the following three core techniques: ① **Pattern density-guided generation.** RPPG leverages pattern-density embeddings, with frequency and spatial density as structural prior conditions. It adaptively learns remaining parameters such as amplitude and mixing coefficients to generate structured perturbations lying on the natural data manifold. ② **Latent distribution contrastive loss.** This loss encourages protected images to be far from original images yet close to dynamically updated perturbations in the latent space, enabling

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Unstructured vs. structured protective perturbations (b) Illustration of different attacks

Figure 1: (a) Robustness of anti-customization methods against Diffusion-based Purification (DP). Perturbations are amplified 10× for clarity. (b) Comparison of different attacks. Unlike vanilla targeted attacks using fixed targets, PuriShield adopts a dynamically updated target that evolves over timesteps. DP projects out-of-manifold data onto the natural data manifold (3D curved surface).

the RPPG to seek the optimal protective perturbations within perturbation budget. ③ **Dual-Head Modulation (DHM).** The RPPG is implemented using a ResUnet architecture (Ronneberger et al., 2015) with an embedded DHM module, which serves to synergistically integrate structured perturbations and protective perturbations. Considering that the data manifold undergoes local shifts after fine-tuning, we design a fine-tuning loss to adapt structured protective perturbations to the updated data manifold. We further adopt a UNet-based perceptual feature loss for better imperceptibility and a dynamic block mask optimization module to strengthen protection robustness.

At a high level, our PuriShield belongs to targeted attacks, which regards the structured protective perturbations as the target to guide the protected image toward them. Unlike existing targeted attacks (Salman et al., 2023; Zheng et al., 2025a) that predefine target images, PuriShield adaptively learn the optimal target. As shown in Fig. 1b, untargeted attacks maximize the distance between original and protected images via unstructured perturbations, pushing protected images off the natural data manifold. Targeted attacks minimize the distance between the protected image and a predefined target. Though such targets lie on the natural data manifold, they exceed the perturbation budget, preventing valid structured perturbations. In contrast, PuriShield adaptively searches for the optimal target within the perturbation budget, while minimizing the distance between the protected image and dynamically updated target, enabling structured protective perturbations. Off-manifold deviations of different protected images before and after purification are compared in **Appx. G**.

Notably, while Anti-Pure (Yang et al., 2025b) and BlurGuard (Kim et al., 2025) are specially tailored to counter DP, they suffered from limited protection efficacy prior to adversarial purification. To the best of our knowledge, **PuriShield is the first work that employs structured protective perturbation to defend against diverse purification approaches, achieving protection effectiveness and robustness simultaneously.** Our main contributions are summarized as follows:

- We identify that structured protective perturbations yield stronger resistance to adversarial purification, and propose PuriShield, a novel anti-customization method that achieves adaptive learning of structured protective perturbations for the first time.
- PuriShield innovatively builds a robust protective perturbation generator via pattern density-guided generation, latent distribution contrastive loss, and dual-head modulation.
- Extensive experiments show that PuriShield attains excellent effectiveness and strong robustness, with great generalizability and high efficiency: The RPPG requires only 31.6 mins of training and generates protective perturbations in a plug-and-play fashion at merely 5.8 ms per image. We further implement PuriShield as a browser extension (see Supplementary Material) that supports fully on-device inference for local image protection.

2 Related Work

A branch of research relies on untargeted attacks to craft protective perturbations. ASPL (Van Le et al., 2023), AdvDM (Liang et al., 2023), SimAC (Wang et al., 2024), Anti-Diff (Zheng et al.,

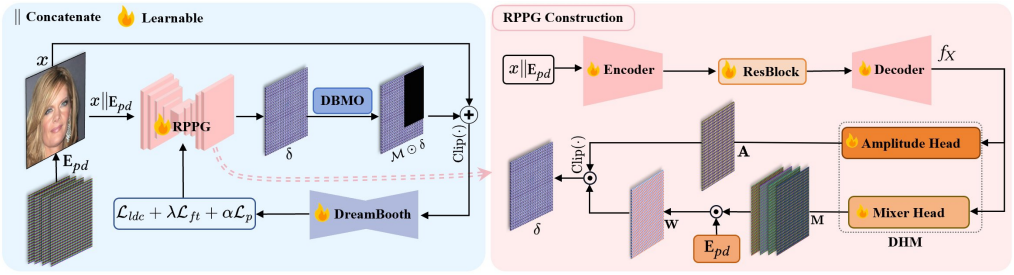


Figure 2: Overview of PuriShield. RPPG G_ϕ generates structured protective perturbation δ for image x conditioned on patten-density embeddings $\mathbf{E}_{pd}$. (a) Loss $\mathcal{L}_{ldc}$ compels the protected image $\hat{x} = \text{Clip}(x + \mathcal{M} \odot \delta, 0, 1)$ to be far from x yet close to δ in latent space, $\mathcal{L}_{ft}$ fine-tunes δ, and $\mathcal{L}_p$ improves imperceptibility. (b) The DHM module fuses structured perturbations (from Mixer Head) and protective perturbations (from Amplitude Head). The DBMO module improves robustness.

2025b), and DisDiff (Liu et al., 2024b) focus on UNets, explicitly maximizing the denoising error. Similarly for attacking UNet, CAAT (Xu et al., 2024) maximizes cross-attention loss, while SDS (Xue et al., 2024) minimizes denoising loss instead. In contrast, PID (Li et al., 2024), VaC (Yang et al., 2025a), and VOID (Qiu et al., 2026) operate on the VAE encoder, by maximizing the divergence between the latent distributions of protected and original images. Another research branch explores targeted attacks against diffusion models. Glaze (Shan et al., 2023), PhotoGuard (Salman et al., 2023), and Mist (Liang & Wu, 2023) attack VAE encoders by minimizing the latent distance between protected and target images. Among UNet attack methods, ACE (Zheng et al., 2025a) minimizes the discrepancy between the denoising noises of protected images and fixed targets, and TAFAP (Lee et al., 2026) aligns the denoising trajectories with those of target images.

Recently, researchers have begun exploring how to enhance the robustness of protective perturbations. MetaCloak (Liu et al., 2024c) maximizes denoising errors via surrogate-based meta-learning to resist transform-based purification (e.g., JPEG and super-resolution). To better defend against DP, Anti-Pure (Yang et al., 2025b) introduces patch-wise frequency guidance and erroneous timestep guidance, while BlurGuard (Kim et al., 2025) aligns adversarial noise with natural frequencies via adaptive blurring and spectrum regularization. Although these two methods provide certain robustness against DP, they suffer from insufficient protection efficacy before purification and weak robustness against transform-based purification. Additional related work is provided in **Appx. A**.

3 Methodology

As shown in Fig. 2, PuriShield comprises an RPPG G_ϕ and a customized Stable Diffusion (e.g., DreamBooth). Specifically, the RPPG consists of an encoder, a ResBlock, a decoder, and a DHM module, which is implemented by embedding module DHM at the last layer of a ResUnet. Inspired by (Liu et al., 2024a), we adopt a mini-batch-wise alternating optimization strategy. For every j mini-batches, we first freeze DreamBooth and update the RPPG using loss $\mathcal{L}_{total}$. We then freeze the RPPG, generate protective perturbations, and fine-tune DreamBooth with fine-tuning loss $\mathcal{L}_{ft}$.

3.1 Core Techniques

Pattern density-guided condition. Inspired by SIREN (Sitzmann et al., 2020), we utilize sine and cosine functions to generate the pattern-density embeddings $\mathbf{E}_{pd}$. Let $\mathrm{P} \in [-1, 1]^{2\times H\times W}$ be the normalized spatial grid coordinates, where H and W denote the height and width of input images. For input images with C color channels, we define the density control parameter Ω. The pattern density embeddings at single color channel $\mathbf{E}_i \in \mathbb{R}^{4\times H\times W}$ are generated as: $\mathbf{E}_i = [\sin(\Omega\pi\mathrm{P})||\cos(\Omega\pi\mathrm{P})]$, where $||$ denotes feature-channel-wise concatenation. Concatenating C embeddings along feature channels yields the final pattern-density condition $\mathbf{E}_{pd} = [\mathbf{E}_1||\mathbf{E}_2||\ldots||\mathbf{E}_C] \in \mathbb{R}^{4C\times H\times W}$. The $\mathbf{E}_{pd}$ guides the generated protective perturbations toward structured patterns, encouraging the protected images to remain close to the natural data manifold. Note that $\mathbf{E}_{pd}$ is independent of the contents of input images, and is determined entirely by the predefined density control parameters Ω.

Latent distribution contrastive loss. Given a frozen VAE $\mathcal{E}$, let $(\mu_k, \sigma_k^2) = \mathcal{E}(k)$ denote the latent mean and variance of input k and let $v_k = \log \sigma_k^2$ be the log-variance. The input k corresponds to the original image x, the protective perturbation δ, or the protected image $\hat{x}$. We treat the dynamically updated $\hat{x}$ as the anchor, the dynamically updated δ as the positive sample (i.e., dynamic target), and the fixed original image x as the negative sample, while calculating the following distances:

$$D_{pos}^{\mu} = \mathrm{MSE}(\mu_{\hat{x}}, \mathrm{sg}[\mu_{\delta}]), D_{neg}^{\mu} = \mathrm{MSE}(\mu_{\hat{x}}, \mu_x), D_{pos}^{v} = \mathrm{MSE}(v_{\hat{x}}, \mathrm{sg}[v_{\delta}]), D_{neg}^{v} = \mathrm{MSE}(v_{\hat{x}}, v_x), \tag{1}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation and MSE denotes Mean Squared Error. While δ undergoes dynamic updates, the current μ_δ and v_δ are detached, and gradients of $\mathcal{L}_{ldc}$ are backpropagated only through the anchor $\hat{x}$. Then, we define loss $\mathcal{L}_{ldc}$ as:

$$\mathcal{L}_{ldc} = \max(0, D_{pos}^{\mu} - D_{neg}^{\mu} + m_1) + \max(0, D_{pos}^{v} - D_{neg}^{v} + m_2), \tag{2}$$

where m_1 and m_2 are margin parameters. Thus, loss $\mathcal{L}_{ldc}$ pulls the protected image $\hat{x}$ towards the structured protective perturbation δ and pushes it away from the original image x in the latent space.

Dual-head modulation. Let $X = x||\mathbf{E}_{pd} \in \mathbb{R}^{5C\times H\times W}$ denote the input of encoder, and let $f_X \in \mathbb{R}^{C'\times H\times W}$ denote the features of input X output by the decoder. The DHM module takes the decoded features f_X as input and branches into two specific functional heads: ① **Amplitude Head (**AH**):** This head is composed of a convolutional layer coupled with a Tanh activation function, which projects f_X into an amplitude map $\mathbf{A} = \mathrm{AH}(f_X) \in [-1, 1]^{C\times H\times W}$. ② **Mixer Head (**MH**):** This head consists of a convolutional layer followed by a channel-wise Softmax normalization, which maps f_X into a mixer map $\mathbf{M} = \mathrm{MH}(f_X) \in (0, 1)^{4C\times H\times W}$.

Besides using $\mathbf{E}_{pd}$ as input-stage conditioning, we further adopt it as output-stage conditioning to regularize generated perturbations to lie close to the data manifold. The pattern-density embeddings $\mathbf{E}_{pd}$ and the mixer map $\mathbf{M}$ can be viewed as 4 structural images and 4 coefficient maps, respectively. Let $\mathbf{E}_{pd}^k \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{M}^k \in [0, 1]^{C\times H\times W}$ be the k-th image and map ($k = 1, \ldots, 4$). We first calculate a weight map $\mathbf{W} \in \mathbb{R}^{C\times H\times W}$ as: $\mathbf{W} = \sum_{k=1}^{4} \mathbf{E}_{pd}^k \odot \mathbf{M}^k$, where $\odot$ denotes the Hadamard product. Then, we compute the final protective perturbation δ as: $\delta = \mathrm{Clip}(\mathbf{A} \odot \mathbf{W}, -\epsilon, \epsilon)$, where $\mathrm{Clip}(\cdot)$ operator restricts the perturbation within $[-\epsilon, \epsilon]$. The DHM module synergistically combines protective and structured perturbations to enhance protection effectiveness and robustness.

3.2 Total Loss

The data manifold will undergo local shifts after fine-tuning. To adapt to the local shifts of the data manifold during unauthorized customization, we design the fine-tuning loss $\mathcal{L}_{ft}$ as follows:

$$\mathcal{L}_{ft}(\theta, c, z_0^{\delta}) = \mathbb{E}_{z_0^{\delta}\sim\mathcal{E}(\delta),\varepsilon\sim\mathcal{N}(0,I),t} \|\varepsilon - \varepsilon_\theta(z_t^{\delta}, t, c)\|_2^2, \tag{3}$$

where z_0^{δ} is the latent representation of protective perturbation δ, and is sampled from $\mathcal{N}(\mu_\delta, \sigma_\delta^2) := \mathcal{E}(\delta)$ via reparameterization. Minimizing loss $\mathcal{L}_{ft}$ is equivalent to minimizing the denoising error, making the protective perturbation close to the updated data manifold. To improve imperceptibility, we introduce a UNet-based perceptual feature loss $\mathcal{L}_p$ as follows:

$$\mathcal{L}_p = \mathbb{E}_{t,l} \left\| f_l^t(\hat{x}) - f_l^t(x) \right\|_2^2, \tag{4}$$

where $f_l^t(\cdot)$ denotes the intermediate UNet feature of input sample extracted at layer l and timestep t. This loss constrains feature discrepancies between the original and protected images to render perturbations imperceptible. The overall training loss of RPPG is formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{ldc} + \lambda\mathcal{L}_{ft} + \alpha\mathcal{L}_p, \tag{5}$$

where λ ($= 1000$) and α ($= 20$) are balancing weights for the two objectives.

By minimizing loss $\mathcal{L}_{total}$, the RPPG learns to generate optimal structured protective perturbations. Once trained, the RPPG can be used to as a portable defense functionality to protect images against unauthorized customization. The algorithmic description of PuriShield is provided in **Appx. B.3**.

3.3 Dynamic Block Mask Optimization

To further enhance the robustness of protective perturbation against adversarial purification, we design a DBMO module, which randomly blocks partial perturbations during training. This component

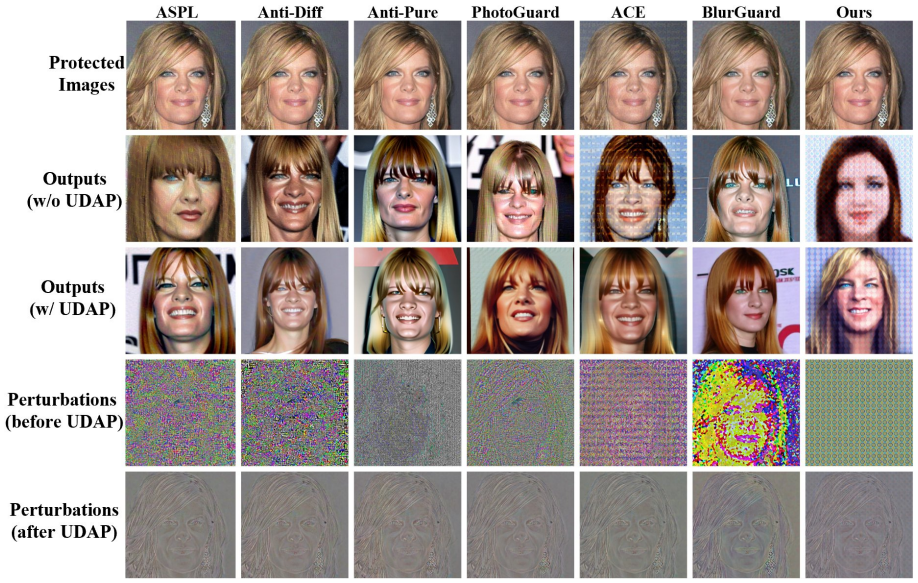


Figure 3: Visualization of protection efficacy pre- and post-UDAP. Perturbations amplified 10×.

can be flexibly integrated into the PuriShield framework as an optimization module. Specifically, we first partition the perturbation δ into an $P \times P$ grid, and then apply a random binary mask $\mathcal{M} \in \{0,1\}^{C\times H\times W}$ to drop selected blocks with a predefined probability p. Next, we calculate the protected image as: $\hat{x} = \text{Clip}(x + \mathcal{M} \odot \delta, 0, 1)$, where $\text{Clip}(\cdot)$ restricts the pixel values of protected images to $[0, 1]$. In a sense, the DBMO module mitigates the influence of local perturbations, ensuring the robust protection effectiveness even when partial perturbations are erased.

4 Experiments

We assess our method under 8 metrics across 5 benchmarks, 5 personalization methods, 13 purification methods and 6 black-box diffusion models, comparing against 10 baselines. This section reports representative experimental results; supplementary results are available in the Appendix.

4.1 Experimental Setup

Datasets and models. Experiments are conducted on two widely adopted facial datasets, CelebA-HQ (Liu et al., 2015) and VGGFace2 (Cao et al., 2018). Besides, WikiArt (Saleh & Elgammal, 2015) dataset is employed to validate the cross-domain generalization. Following (Yang et al., 2025a), we randomly select 50 identities from each dataset, utilizing 4 images per identity for protection and subsequent fine-tuning. We employ Stable Diffusion v1.5 (SD v1.5) (Rombach et al., 2022) as the white-box model, with DreamBooth serving as the parameter fine-tuning technique. The text encoder is kept frozen during fine-tuning unless otherwise specified. The details of datasets and fine-tuning hyperparameters are provided in **Appx. B.1**. More results on different fine-tune methods are included in **Appx. C**, and those on black-box transferability are included in **Appx. D**.

Baselines. We compare PuriShield against 6 anti-customization methods: ASPL (Van Le et al., 2023), Anti-Diff (Zheng et al., 2025b), Anti-Pure (Yang et al., 2025b), PhotoGuard (Salman et al., 2023), ACE (Zheng et al., 2025a), and BlurGuard (Kim et al., 2025). To ensure a fair comparison, all the baselines employ PGD to generate perturbations, using the optimal hyperparameters configurations reported in the respective literature. The perturbation budget is constrained by L_∞ norm with $\epsilon = 8/255$. For personalized generation, we select 50 identities and generate 20 images per identity using the prompt "a photo of *sks* person", where *sks* denotes the identity-specific identifier. The configuration details of all anti-customization methods are provided in **Appx. B.2**.

Purification methods. ① Transformation-based purification: JPEG (Guo et al., 2018), Gaussian (Zantedeschi et al., 2017), Noisy Upscaling (Hönig et al., 2025), and Super-Resolution

Table 2: Quantitative comparison of robustness against transformation-based purification methods.

Methods	JPEG				Gaussian				Noisy Upscaling				SR			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	193.940	0.485	-13.063	2.155	135.193	0.593	-3.510	2.051	185.468	0.490	3.723	3.077	189.653	0.457	-11.602	2.677
Anti-Diff	178.582	0.463	-13.234	1.961	142.001	0.474	-5.677	2.046	147.246	0.594	3.746	3.409	165.868	0.517	-12.406	2.507
Anti-Pure	121.857	0.605	3.273	2.790	123.000	0.588	-3.258	1.881	125.234	0.669	7.676	3.336	104.338	0.674	5.461	2.794
PhotoGuard	172.737	0.555	-9.922	2.001	143.179	0.519	-6.148	2.010	145.659	0.521	1.435	3.160	170.451	0.417	-8.164	2.577
ACE	143.473	0.500	-5.539	3.002	121.380	0.589	-4.371	2.519	143.310	0.560	10.133	3.949	211.315	0.489	-6.242	2.736
BlurGuard	121.284	0.696	-2.641	3.572	111.023	0.631	-5.578	1.989	172.009	0.585	6.461	4.168	113.713	0.640	2.861	3.333
Ours	**257.933**	**0.405**	**-15.961**	**1.703**	**171.830**	**0.303**	**-11.313**	**1.466**	**217.711**	**0.414**	**-3.846**	**2.566**	**307.689**	**0.384**	**-15.203**	**2.102**

Table 3: Quantitative comparison of robustness against diffusion-based purification methods.

Methods	DiffPure				GrIDPure				Impress				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	156.584	0.574	-5.203	1.667	189.964	0.542	-5.731	2.107	193.554	0.454	-10.047	2.599	183.145	0.469	-7.461	1.443
Anti-Diff	189.481	0.487	-2.106	1.888	154.829	0.538	0.473	2.949	193.183	0.460	-9.570	2.117	167.595	0.562	-6.445	1.544
Anti-Pure	172.475	0.453	5.852	2.813	146.454	0.569	1.023	2.612	125.327	0.637	5.539	3.095	177.945	0.544	-1.701	1.525
PhotoGuard	183.727	0.511	-4.805	1.924	133.873	0.579	1.839	2.864	156.679	0.474	-5.266	2.124	190.772	0.453	-3.844	1.879
ACE	174.604	0.485	-4.500	2.481	163.978	0.487	-2.563	2.472	210.565	0.485	-7.844	2.494	175.472	0.477	-5.668	1.618
BlurGuard	160.066	0.549	3.098	2.464	168.490	0.510	-6.844	2.891	158.340	0.598	-1.793	2.684	195.247	0.414	-4.270	1.473
Ours	**226.388**	**0.412**	**-6.981**	**1.345**	**214.095**	**0.375**	**-9.602**	**1.825**	**319.230**	**0.416**	**-13.016**	**1.832**	**216.215**	**0.365**	**-12.172**	**1.284**

(SR) (Mustafa et al., 2019). ② DP-based methods: DiffPure (Nie et al., 2022), GrIDPure (Zhao et al., 2024), Impress (Cao et al., 2023a), and UDAP (Zheng et al., 2026).

Evaluation metrics. ① Fréchet Inception Distance (FID) (Heusel et al., 2017); ② Identity Similarity Metric (ISM) (Deng et al., 2019); ③ Image Quality Score (IQS) (Radford et al., 2021);④ Language-Image Quality Evaluator (LIQE) (Zhang et al., 2023). Detailed descriptions of these metrics are provided in **Appx. B.4**, and more results on other metrics are presented in **Appx. E**.

4.2 Main Results

Effectiveness. Table 1 compares the protection efficacy across two datasets before purification. Compared to unprotected data, all anti-customization methods exhibit favorable defensive capabilities against standard direct fine-tuning; PuriShield achieves the strongest performance across all metrics. The 2nd row of Fig. 3 shows that most baseline methods only partially degrade identity consistency and offer limited protection. By contrast, PuriShield introduces noticeable artifacts in the generated images to achieve stronger protection. As shown in the 4th row of Fig. 3, the perturbations generated by baseline methods appear irregular and unstructured, whereas those generated by PuriShield exhibit a well-defined structured pattern. The supplementary results presented in **Appx. F** show that PuriShield consistently achieves the best protection effectiveness *across different prompts*, *with an unfrozen text encoder*, and *across various perturbation magnitudes*. Perturbation imperceptibility results in **Appx. E.2** show that while inferior to BlurGuard, PuriShield, achieves visual quality comparable to ACE, PhotoGuard, and Anti-Diff.

Table 1: Quantitative comparison of protection efficacy. **Bold**: best; underlined: second-best.

Datasets	Methods	FID ↑	ISM ↓	IQS ↓	LIQE ↓
CelebA-HQ	Clean	97.917	0.678	5.826	4.335
	ASPL	201.215	0.343	-8.562	2.106
	Anti-Diff	163.455	0.402	-7.562	2.256
	Anti-Pure	155.198	0.553	0.641	2.579
	PhotoGuard	190.306	0.449	-8.688	2.538
	ACE	289.216	0.197	-10.867	2.484
	BlurGuard	161.507	0.554	-2.998	2.836
	Ours	**338.980**	**0.098**	**-17.547**	**1.902**
VGG-Face2	Clean	121.600	0.623	16.094	4.169
	ASPL	330.828	0.218	-13.578	2.315
	Anti-Diff	286.826	0.299	-16.672	2.318
	Anti-Pure	265.745	0.385	4.082	2.601
	PhotoGuard	265.105	0.290	-16.641	2.271
	ACE	336.469	0.173	-11.461	2.215
	BlurGuard	253.498	0.365	8.141	2.767
	Ours	**389.917**	**0.087**	**-18.719**	**2.086**

Robustness. Tables 2-3 report the quantitative results on CelebA-HQ dataset under diverse purification methods. None of baselines can simultaneously defend against two types of purification except for PuriShield. The 3rd and 5th rows of Fig. 3 show that UDAP substantially eliminates unstructured protective perturbations, allowing baseline-generated images to recover clear identity features. In contrast, structured protective perturbations are well retained, enabling PuriShield to sustain effective protection after purification. The supplementary results in **Appx. G** provide

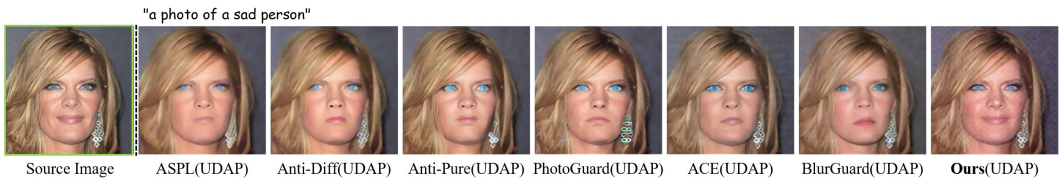


Figure 4: Visualized protection robustness of anti-customization methods against DiffEdit.



Figure 5: Visualized protection efficacy of PuriShield against style mimicry.

quantitative evidence of robustness, and show that PuriShield attains optimal robustness on VGGFace2 dataset, maintaining stable performance under *different prompts*, *an unfrozen text encoder*, *varied perturbation budgets*, *extended baselines*, *extra purification methods*, and *adaptive attacks*.

Cross-task generalizability. Based on the CelebA-HQ dataset and DiffEdit (Couairon et al., 2023), we evaluate the protection robustness against unauthorized diffusion-based editing tasks. The protected images are first preprocessed by UDAP and then edited using prompts such as "a photo of a *sad* person". Fig. 4 shows that the perturbations produced by baselines are largely neutralized after UDAP, enabling successful manipulation of the target attributes. In contrast, images protected by PuriShield remain difficult to edit after purification. Correspondingly, quantitative results in Table 4 show that after UDAP purification, PuriShield achieves the best performance across all evaluated metrics. More results on cross-task generalizability can be found in **Appx. H.1**.

Table 4: Protection robustness against DiffEdit customization after UDAP.

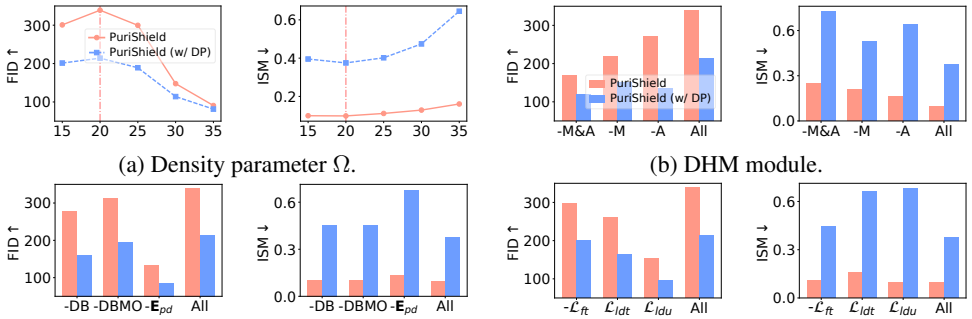
Methods	IQS ↓	LIQE ↓
ASPL	2.926	2.825
Anti-Diff	3.065	3.153
Anti-Pure	1.492	2.918
PhotoGuard	0.836	2.622
ACE	1.257	2.556
BlurGuard	2.381	2.458
Ours	**-1.966**	**2.292**

Cross-domain generalizability. We apply the RPPG trained on CelebA-HQ dataset to directly craft protective perturbations for WikiArt images. The protected images are used to fine-tune Textual Inversion for mimicking diverse artistic styles. We replace the facial-data metric ISM with Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) and Multi-Scale Structural Similarity (MS) (Wang et al., 2003), which are suited for assessing artistic style works. As shown in Table 5, compared to clean images, the protected images can significantly degrade the quality of style-transferred images, thus effectively defending against style mimicry. Fig. 5 visualizes the effect of style learning from Van Gogh-style protected images. More results on cross-domain generalizability can be found in **Appx. H.2**.

Table 5: Protection efficacy against style mimicry. ↑ and ↓ indicate changes in protection efficacy compared to learning from clean images.

Metrics	Monet	Dalí	Van Gogh
FID ↑	327.853$_{\uparrow 28.900}$	413.375$_{\uparrow 13.341}$	325.652$_{\uparrow 29.102}$
IQS ↓	2.834$_{\downarrow 4.232}$	2.381$_{\downarrow 18.135}$	-1.604$_{\downarrow 16.432}$
LIQE ↓	2.705$_{\downarrow 0.289}$	2.631$_{\downarrow 0.270}$	2.853$_{\downarrow 0.862}$
LPIPS ↑	0.690	0.706	0.686
MS ↓	0.332	0.262	0.291

Efficiency. Experiments are implemented in PyTorch and executed on a server equipped with an NVIDIA L40 GPU. Table 6 compares the computational and memory costs of different methods. Although PuriShield requires a moderate training cost, it offers significant advantages at inference time. Once trained, RPPG can generate protective perturbations in a plug-and-play manner in only 5.8 ms per image. By contrast, all baselines require substantially longer inference ($\geq$42s per image). This efficiency makes our generator-based design practical for large-scale or real-time protection. RPPG converges rapidly mainly due to the pattern density-guided condition, which provides explicit structural priors. Thus, we use early stopping once the loss stabilizes within a small tolerance.



(a) Density parameter Ω. (b) DHM module.

(c) Different components. (d) Loss design.

Figure 6: Ablation studies of our method.

4.3 Ablation Studies

We adopt the FID and ISM metrics on CelebA-HQ for ablation studies, and employ GrIDPure as the default DP method. Further ablation results appear in **Appx. I**.

Table 6: Comparison of computational and memory cost.

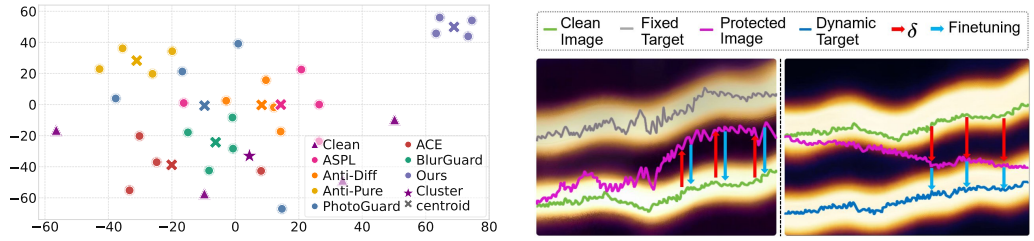
Method	Memory/GB	Training	Per-Image
ASPL	21.4	-	66.0 s
Anti-Diff	24.7	-	148.5 s
Anti-Pure	14.2	-	244.5 s
PhotoGuard	13.2	-	63.0 s
ACE	13.8	-	75.0 s
BlurGuard	18.2	-	42.0 s
Ours	30.1	31.6 mins	**5.8 ms**

Density control parameter Ω. Parameter Ω determines the density of patterns. The higher the value of Ω, the denser the patterns. As shown in Fig. 6a, both the protection effectiveness and robustness show an upward trend as Ω increases initially, and begin to decline once Ω reaches the threshold value. This indicates that excessively dense patterns (which degenerate into random noise) or excessively sparse patterns (which lack sufficient structural complexity) are detrimental to the generation of structured protective perturbations. PuriShield achieves optimal performance when $\Omega = 20$.

The role of DHM. We design three ablation settings: **-M** removes the Mixer Head, **-A** removes the Amplitude Head, and **-M&A** removes both heads. As shown in Fig. 6b, removing either component degrades performance, while removing both causes the largest drop; with both enabled (**All**), PuriShield performs best across all metrics. Specifically, **-M** weakens protective strength, whereas **-A** disrupts structural consistency and protective robustness. This validates that the DHM module effectively integrates protective strength and structural priors for robust defense.

The effect of different components. We analyze the contribution of key components in PuriShield, including the fine-tuning strategy (**DB**), the **DBMO** module, and the pattern density-guided condition $\mathbf{E}_{pd}$. To this end, we construct three variants by removing each component individually (**-DB**, **-DBMO**, $-\mathbf{E}_{pd}$). As shown in Fig. 6c, removing any component leads to a clear performance degradation. In particular, removing **DB** results in reduced adaptability to the updated data manifold, while removing **DBMO** weakens robustness against purification. Furthermore, removing $\mathbf{E}_{pd}$ disrupts the structured nature of perturbations, leading to inferior overall performance. When all components are enabled (**All**), PuriShield achieves the optimal performance across all metrics. These results indicate that each component contributes to the final performance, and their combination is essential for achieving robust and effective protection.

The impact of loss design. We design the following ablation scenarios: ① $-\mathcal{L}_{ft}$ removes the fine-tuning loss; ② $\mathcal{L}_{ldt}$ replaces the latent distribution contrastive loss $\mathcal{L}_{ldc}$ with a targeted loss $\mathcal{L}_{ldt} = \mathrm{MSE}(\mu_{\hat{x}}, \mu_{\delta}) + \mathrm{MSE}(v_{\hat{x}}, v_{\delta})$, where the perturbation δ is regarded as the dynamically updated target; ③ $\mathcal{L}_{ldu}$ replaces loss $\mathcal{L}_{ldc}$ with an untargeted loss $\mathcal{L}_{ldu} = -\mathrm{MSE}(\mu_{\hat{x}}, \mu_{x}) - \mathrm{MSE}(v_{\hat{x}}, v_{x})$. As shown in Fig. 6d, removing loss $\mathcal{L}_{ft}$ leads to a noticeable performance drop, indicating its importance in adapting perturbations to the updated data manifold. Moreover, replacing loss $\mathcal{L}_{ldc}$ with either targeted or untargeted alternatives results in significant performance degradation. This is because targeted loss cannot enlarge distances between protected and original images, preventing the learning of semantically meaningful protective perturbations. Untargeted loss, without narrowing distances between protected images and perturbations, obstructs structured robust perturbation gen-



(a) t-SNE visualization

(b) Left: Traditional targeted attack. Right: Ours.

Figure 7: (a) t-SNE visualization of protective perturbation features. Tighter clusters indicate higher latent similarity and more consistent perturbation patterns; greater distance from the clean centroid indicates stronger protective efficacy. (b) Fixed target in baselines v.s. dynamic target in PuriShield.

eration. By contrast, contrastive loss pulls protected images toward structured perturbations while pushing them away from originals, enhancing protective efficacy and robustness. Ablation results in **Appx. I** further show performance drop significantly when replacing $\mathcal{L}_{ldc}$ with typical untargeted or targeted loss, underscoring its importance for generating robust, structured protective perturbations.

4.4 Discussion

Why do structured protective perturbations yield better protection effectiveness and robustness? We visualize feature embeddings of protective perturbations from PuriShield and all baselines. As shown in Fig. 7a, most baseline perturbations scatter broadly in feature space, indicating low feature consistency and irregular perturbation patterns vulnerable to DP. While Anti-Pure and BlurGuard yield relatively compact perturbations, they lie near the clean-image cluster centroid. Such perturbations thus exhibit limited protective efficacy compared with structured protective perturbations and can be readily projected back onto the clean data manifold by DP. In contrast, PuriShield produces highly coherent latent features, yielding consistent and structured perturbation patterns that remain well separated from baseline counterparts and clean image references. This enables these perturbations to provide effective protection and defends against advanced purification.

Why do dynamic targets facilitate the search for optimally protective perturbations? Unlike traditional targeted attacks that optimize toward fixed targets, PuriShield continuously guides protected images toward dynamically updated perturbation patterns. As conceptually illustrated in Fig. 7b, the predefined fixed targets often lie outside the perturbation budget, causing the perturbation optimization process to truncate at the boundary. This results in protected images closer to the original images than to targets. Thus, fine-tuning direction is opposite to the optimization direction, aiming to pull the protected image back to the original distribution. In contrast, our PuriShield searches for the optimal targets that lie within perturbation budget while remaining distant from clean images. After optimization, the protected image is close to the target but far from the original image. Consequently, the fine-tuning process tends to learn the structured protective perturbations rather than eliminating them. The dynamic target strategy thus helps to create a smoother optimization landscape, allowing the perturbations to converge into readily learnable structural patterns.

5 Conclusion

This paper investigates the vulnerability of existing anti-customization methods against purification, and explores structured protective perturbations for boosting protection robustness. We propose PuriShield, a novel method for adaptive learning of structured protective perturbations. Experimental results show that PuriShield surpasses state-of-the-art baselines in protection effectiveness and robustness, while achieving great generalizability and efficiency. These encouraging results validate structured protective perturbations for effective robustness gains, shedding light on future anti-customization designs. Our current construction employs pattern-density embeddings with fixed frequency and spatial density as structural priors, which may limit the adaptability and diversity of structured perturbations. Future work will leverage reinforcement learning to adaptively select these parameters, enabling perturbation structures to dynamically adapt to image characteristics.

AI USE STATEMENT

In this work, we did not use generative AI tools for research methodology or experimental design, hypothesis formulation, mathematical claims or proofs, method implementation, data generation or cleaning, or result interpretation. We used generative AI tools to assist with language polishing, improving the readability of the manuscript. All AI-assisted edits were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. of ICML*, pp. 274–283. PMLR, 2018.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proc. of CVPR*, pp. 22563–22575, 2023.

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. In *Proc. of NeurIPS*, 2023a.

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 *Proc. of ICCV*, pp. 22560–22570, 2023b.

Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Proc. of FG*, 2018.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *Proc. of USENIX Security*, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proc. of CVPR*, pp. 2818–2829. IEEE, 2023.

CompVis. Stable diffusion v1-4 model card. `https://huggingface.co/CompVis/stable-diffusion-v1-4`, 2022.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *Proc. of ICLR*, 2023.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. of CVPR*, 2019.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proc. of CVPR*, pp. 5203–5212, 2020.

Hanna Foerster, Sasha Behrouzi, Phillip Rieger, Murtuza Jadliwala, and Ahmad-Reza Sadeghi. {LightShed}: Defeating perturbation-based image copyright protections. In *Proc. of USENIX Security*, pp. 7271–7290, 2025.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *Proc. of ICLR*, 2023.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *Proc. of ICLR*, 2018.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. of NeurIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proc. of NeurIPS*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Proc. of NeurIPS*, pp. 8633–8646, 2022.

Robert Hönig, Javier Rando, Nicholas Carlini, and Florian Tramèr. Adversarial perturbations cannot reliably protect artists from generative ai. In *Proc. of ICLR*, 2025.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *Proc. of ICPR*, pp. 2366–2369. IEEE, 2010.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proc. of ICLR*, 2022.

Kaggle. Painter by number. `https://www.kaggle.com/c/painter-by-numbers/data`, 2017. Accessed: 2017.

Jinsu Kim, Yunhun Nam, Minseon Kim, Sangpil Kim, and Jongheon Jeong. Blurguard: A simple approach for robustifying image protection against ai-powered editing. In *Proc. of NeurIPS*, 2025.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proc. of CVPR*, pp. 1931–1941, 2023.

Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *Proc. of ECCV*, pp. 679–692. Springer, 2012.

Hojun Lee, Mijin Koo, Yeji Song, and Nojun Kwak. Targeted data protection for diffusion model by matching training trajectory. In *Proc. of AAAI*, pp. 5854–5862, 2026.

Ang Li, Yichuan Mo, Mingjie Li, and Yisen Wang. Pid: Prompt-independent data protection against latent diffusion models. In *Proc. of ICML*, 2024.

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

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 *Proc. of ICML*, 2023.

Hanwen Liu, Zhicheng Sun, and Yadong Mu. Countering personalized text-to-image generation with influence watermarks. In *Proc. of CVPR*, pp. 12257–12267, 2024a.

Yisu Liu, Jinyang An, Wanqian Zhang, Dayan Wu, Jingzi Gu, Zheng Lin, and Weiping Wang. Disrupting diffusion: Token-level attention erasure attack against diffusion-based customization. In *Proc. ACM MM*, 2024b.

Yixin Liu, Chenrui Fan, Yutong Dai, Xun Chen, Pan Zhou, and Lichao Sun. Metacloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning. In *Proc. of CVPR*, 2024c.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *Proc. of CVPR*, pp. 860–868, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. of ICCV*, 2015.

Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. In *Proc. of NeurIPS*, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of ICLR*, 2019.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proc. of CVPR*, pp. 2837–2845, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of ICLR*, 2018.

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. In *Proc. of ICLR*, 2022.

Microsoft. Onnx runtime. `https://onnxruntime.ai/`, 2026. Accessed: 2026-08-22.

Aamir Mustafa, Salman H Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. Image super-resolution as a defense against adversarial attacks. *IEEE Transactions on Image Processing*, 29: 1711–1724, 2019.

Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proc. of CVPR*, pp. 262–271, 2020.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *Proc. of ICML*, 2022.

Gaozheng Pei, Ke Ma, Yingfei Sun, Qianqian Xu, and Qingming Huang. Diffusion-based adversarial purification from the perspective of the frequency domain. In *Proc. of ICML*, pp. 48679–48698. PMLR, 2025.

Sen Peng, Mingyue Wang, Jianfei He, Jijia Yang, and Xiaohua Jia. Cat: Contrastive adversarial training for evaluating the robustness of protective perturbations in latent diffusion models. In *Proc. of ICML*, pp. 48872–48885, 2025.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proc. of ICLR*, 2023.

Chunlin Qiu, Ang Li, Tianxiao Huang, Ruilin Gan, Yunjie Ge, Shenyi Zhang, Huayi Duan, Lingchen Zhao, Chao Shen, and Qian Wang. Void: Defeating unauthorized mimicry in latent diffusion models. In *Proc. of USENIX Security*, pp. 6505–6524, 2026.

Vu Duc Quang. Stable diffusion v2.1-base model card. `https://huggingface.co/VuDucQuang/Stable_Diffusion_2.1_base`, 2023.

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 *Proc. of ICML*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. of the CVPR*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of MICCAI*, 2015.

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 *Proc. of CVPR*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Proc. of NeurIPS*, 35:36479–36494, 2022.

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.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Mądry. Raising the cost of malicious ai-powered image editing. In *Proc. of ICML*, 2023.

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. In *Proc. of USENIX Security*, 2023.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. of NeurIPS*, 2020.

Kaiyu Song, Hanjiang Lai, Yan Pan, and Jian Yin. Mimicdiffusion: Purifying adversarial perturbation via mimicking clean diffusion model. In *Proc. of CVPR*, pp. 24665–24674, 2024.

Stable Diffusion v1.5 Community. Stable diffusion inpainting. `https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-inpainting`, 2024.

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 *Proc. of ICCV*, 2023.

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 *Proc. of CVPR*, 2024.

Hanhui Wang, Yihua Zhang, Ruizheng Bai, Yue Zhao, Sijia Liu, and Zhengzhong Tu. Edit away and my face will not stay: Personal biometric defense against malicious generative editing. In *Proc. of CVPR*, 2025.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Proc. of ACSSC*, 2003.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *Proc. of ICLR*, 2018.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proc. of CVPR*, pp. 2730–2739, 2019.

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 *Proc. of CVPR*, 2024.

Yifang Xu, Benxiang Zhai, Yunzhuo Sun, Ming Li, Yang Li, and Sidan Du. Hifi-portrait: Zero-shot identity-preserved portrait generation with high-fidelity multi-face fusion. In *Proc. of CVPR*, 2025.

Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *Proc. of ICLR*, 2024.

Jiachen Yang, Yusong Wang, Yanmei Fang, Yunshu Dai, and Fangjun Huang. Variance as a catalyst: Efficient and transferable semantic erasure adversarial attack for customized diffusion models. In *Proc. of ICML*, 2025a.

Wenkui Yang, Jie Cao, Junxian Duan, and Ran He. Towards robust defense against customization via protective perturbation resistant to diffusion-based purification. In *Proc. of ICCV*, 2025b.

Srikar Yellapragada, Alexandros Graikos, Kostas Triaridis, Prateek Prasanna, Rajarsi Gupta, Joel Saltz, and Dimitris Samaras. Zoomldm: Latent diffusion model for multi-scale image generation. In *Proc. of CVPR*, 2025.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proc. of workshop AIS*, 2017.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of CVPR*, 2018.

Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proc. of CVPR*, pp. 14071–14081, 2023.

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 *Proc. of CVPR*, 2024.

Boyang Zheng, Chumeng Liang, and Xiaoyu Wu. Targeted attack improves protection against unauthorized diffusion customization. In *Proc. of ICLR*, 2025a.

Li Zheng, Liangbin Xie, Jiantao Zhou, Xintao Wang, Haiwei Wu, and Jinyu Tian. Anti-diffusion: Preventing abuse of modifications of diffusion-based models. In *Proc. of AAAI*, 2025b.

Li Zheng, Liangbin Xie, Jiantao Zhou, and He YiMin. Universal adversarial purification with ddim metric loss for stable diffusion. In *Proc. of AAAI*, 2026.

APPENDIX

The Appendix includes related work and implementation details (**Secs. A– B**), extended evaluations on customization methods, black-box transferability and extra metrics (**Secs. C–E**), supplementary quantitative results for effectiveness, robustness, generalization and ablations (**Secs. F–I**), and additional visualizations (**Sec. J**). Unless otherwise specified, our experiments employ Stable Diffusion v1.5 (SD v1.5) as the backbone model, use DreamBooth as the default fine-tuning approach, and keep the text encoder frozen during fine-tuning. The default perturbation budget is set to $\epsilon = 8/255$ under L_∞ norm, and the default prompt is "a photo of *sks* person".

A MORE RELATED WORK

This section provides additional background on diffusion models and adversarial purification.

A.1 DIFFUSION MODELS

Diffusion models have demonstrated strong generative performance across a wide range of applications, including text-to-image synthesis (Rombach et al., 2022; Saharia et al., 2022), video generation (Blattmann et al., 2023; Ho et al., 2022), and 3D modeling (Luo & Hu, 2021; Poole et al., 2023). Among them, Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) are likelihood-based generative models inspired by non-equilibrium thermodynamics. A DDPM consists of a forward diffusion process and a reverse denoising process. Given a data sample $\mathbf{x}_0 \sim q(\mathbf{x})$, the forward process progressively adds Gaussian noise over T timesteps according to a predefined variance schedule $\beta_1, \ldots, \beta_T$. The transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{6}$$

Using the reparameterization property of Gaussian distributions, $\mathbf{x}_t$ can also be sampled directly from $\mathbf{x}_0$ at an arbitrary timestep t:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{7}$$

where $\alpha_t = 1-\beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. As t approaches T, $\bar{\alpha}_t$ decreases toward zero, causing $q(\mathbf{x}_T)$ to approach an isotropic Gaussian distribution. The reverse process then gradually removes noise to recover the original data distribution. Since the exact reverse transition $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable, a parameterized neural network is used to approximate it:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)). \tag{8}$$

In practice, the model typically predicts the added noise ϵ through a noise-prediction network $\varepsilon_\theta(\mathbf{x}_t, t)$, rather than directly estimating the mean μ_θ.

We mainly consider Latent Diffusion Models (LDMs) (Rombach et al., 2022; Yellapragada et al., 2025), which perform diffusion in a learned latent space instead of the pixel space. Given an image x, a pretrained Variational Autoencoder (VAE) (Lopez et al., 2018) $\mathcal{E}$ maps it to a latent distribution $\mathcal{N}(\mu_x, \sigma_x^2) := E(x)$, from which the latent representation z_0 is sampled. A conditional U-Net (Ronneberger et al., 2015) ε_θ is trained to predict the noise added to z_t at timestep t by minimizing:

$$\mathcal{L}_{ldm}(\theta, c, z_0) = \mathbb{E}_{z_0\sim\mathcal{E}(x),\varepsilon\sim\mathcal{N}(0,I),t}\|\varepsilon - \varepsilon_\theta(z_t, t, c)\|_2^2, \tag{9}$$

where c denotes the conditioning input, and $\emptyset$ represents the unconditional setting.

A.2 ADVERSARIAL PURIFICATION

Adversarial purification removes adversarial perturbations from inputs before downstream processing, providing a plug-in defense without retraining the target model. It can be broadly divided into transformation-based and diffusion-based approaches.

Transformation-based purification. These methods suppress adversarial perturbations through input transformations such as JPEG compression (Guo et al., 2018), Gaussian blurring (Zantedeschi et al., 2017), or GAN-based reconstruction (Naseer et al., 2020) to disrupt adversarial noise. For instance, Resizing (Xie et al., 2018) applies stochastic geometric transformations to disrupt adversarial

structures, while Feature Distillation (FD) (Liu et al., 2019) filters perturbations using frequency-aware JPEG compression. Noisy Upscaling (Hönig et al., 2025) first injects Gaussian noise and then restores the image with the Stable Diffusion Upscaler (Rombach et al., 2022). Super-Resolution (SR) (Mustafa et al., 2019) combines super-resolution with wavelet denoising to project adversarial examples toward the natural image manifold. However, deterministic or weakly stochastic transformations remain vulnerable to adaptive attacks such as BPDA (Athalye et al., 2018).

Diffusion-based purification. Recent methods exploit the denoising capability of diffusion models to remove adversarial perturbations. DiffPure (Nie et al., 2022) adds Gaussian noise through forward diffusion and then reconstructs the image using reverse denoising. GrIDPure (Zhao et al., 2024) adopts an iterative grid-based diffusion procedure to remove protective perturbations. MimicDiffusion (Song et al., 2024) removes adversarial noise by approximating the diffusion trajectory of clean images, while CAT (Peng et al., 2025) realigns latent representations through contrastive adversarial training. FreqPure (Pei et al., 2025) performs purification in the frequency domain by selectively preserving amplitude and phase information at each diffusion timestep. LightShed (Foerster et al., 2025) reconstructs protection perturbations with an autoencoder and detects and removes them based on perturbation entropy. Among existing methods, UDAP (Zheng et al., 2026) as a dedicated DP scheme designed to defeat anti-customization methods adaptively removes protective perturbations by minimizing a novel DDIM metric loss.

B Implementation and Experiment Details

In this section, we first describe the datasets and hyperparameters used in customization methods, followed by the implementation details for our RPPG and all baselines. After presenting algorithmic description of PuriShield, we elaborate on all evaluation metrics.

B.1 Datasets and Hyperparameters

To comprehensively evaluate the protection performance (effectiveness, robustness, and generalization) of our PuriShield and baselines, we conduct experiments on five standard datasets:

- **CelebA-HQ** (Liu et al., 2015): A widely used high-resolution facial image dataset containing 30,000 images. It serves as the primary benchmark for evaluating the protection performance of all methods against unauthorized identity-based personalization.
- **VGGFACE2** (Cao et al., 2018): A human face recognition dataset that contains 9,131 identities, corresponding to about 3.31 million images, with substantial variations in pose, age, lighting, and ethnicity. It is also used to assess the protection performance of all methods against unauthorized identity-based personalization.
- **WikiArt** (Saleh & Elgammal, 2015): A comprehensive visual art collection, which comprises 80,020 paintings, covering 1,119 artists and 27 artistic styles. This dataset is adopted to evaluate the cross-domain generalization of our PuriShield in defending against unauthorized style imitation.
- **Paintings** (Kaggle, 2017): A curated collection of 103,093 high-quality artwork images. This dataset is employed to further validate the cross-domain generalization of our PuriShield in defending against style-oriented unauthorized customization.
- **Helen** (Le et al., 2012): A high-resolution facial dataset comprising 2,330 images, renowned for its rich structural details and fine-grained facial variations. It is employed to further evaluate the cross-task generalization of our PuriShield in defending against unauthorized inpainting tasks.

Besides DreamBooth (Meng et al., 2022), we evaluate the protection performance of all anti-customization methods against including SDEdit (Meng et al., 2022), Textual Inversion (Gal et al., 2023), LoRA (Hu et al., 2022), and Custom Diffusion (Kumari et al., 2023). As for SDEdit, we set the strength of SDEdit to 0.3 because this setting provides a good balance between modifying the input image and preserving its original content. Additionally, we use a null prompt to mitigate the influence of textual prompts during the generation process. The detailed hyperparameters of other customization methods are provided in Table 7.

Table 7: Fine-tuning hyper-parameters.

Version	Freeze-TE	LR	Steps	Batch Size	Grad. Accu.	Output Res.
1.4	Yes	6e-7	1000	2	1	512×512
1.4	No	6e-7	1000	2	1	512×512
1.5	Yes	6e-7	1000	2	1	512×512
1.5	No	6e-7	1000	2	1	512×512
2.1base	Yes	6e-7	1000	2	1	512×512
2.1base	No	6e-7	1000	2	1	512×512
SDXL 1.0	Yes	1e-4	500	1	1	1024×1024
3	Yes	1e-5	500	1	1	1024×1024

(a) DreamBooth.

Version	LR	Steps	Batch Size	Grad. Accu.	Output Res.
1.5	5e-4	2000	1	1	512×512
2.1base	5e-4	2000	1	1	512×512

(b) Textual Inversion.

Version	Rank	LR	Steps	Batch Size	Grad. Accu.	Output Res.
1.5	32	1e-4	800	1	1	512×512
2.1base	32	1e-4	800	1	1	512×512

(c) LoRA.

LR	Steps	Batch Size	Grad. Accu.	Output Res.
1e-5	250	1	1	512×512

(d) Custom Diffusion.

B.2 Implementation Details

RPPG. For the training of the Robust Protective Perturbation Generator (RPPG), we employ the AdamW (Loshchilov & Hutter, 2019) optimizer with a learning rate of $2e{-}5$ and a weight decay of $1e{-}4$. The ResUNet component is similarly optimized using AdamW, but with a reduced learning rate of $5e{-}7$. The model is trained using a batch size of 8 for $N = 5$ epochs, and we set the alternating update interval to $j = 3$. Furthermore, we utilize fp16 mixed-precision training to enhance computational efficiency. Regarding the training data, we construct a subset based on the standard CelebA-HQ training set (which originally consists of 27,000 images). The actual subset utilized for training comprises 16,929 facial images, accounting for approximately 62.7% of the complete training set. As for the VGGFace2 dataset, the official training split contains approximately 3.14 million images. To balance computational efficiency and data diversity, we apply a 20% sampling ratio, randomly constructing a dedicated training subset of roughly 628,000 images of 500 identities.

PuriShield extension. To make PuriShield easily usable in practice, we package the trained RPPG as a plug-and-play browser extension. The model is exported to ONNX and runs locally via ONNX Runtime Web (Microsoft, 2026), with WebGPU acceleration and a WebAssembly fallback. Users can locally protect images within the browser using the extension. All inference is performed on-device, and original images are never uploaded to external servers. The compressed extension is only about 11 MB. This lightweight implementation enables convenient and privacy-preserving image protection in the browser. For example, when running directly in the browser on a laptop equipped with an AMD Ryzen 5 5600H CPU and 16 GB of RAM, PuriShield takes only about 0.63 s on average to construct a protected image of resolution 512×512.

Baselines. We compare PuriShield against 10 anti-customization methods: ASPL (Van Le et al., 2023), Anti-Diff (Zheng et al., 2025b), Anti-Pure (Yang et al., 2025b), PhotoGuard (Salman et al., 2023), ACE (Zheng et al., 2025a), BlurGuard (Kim et al., 2025), AdvDM (Liang et al., 2023), Mist (Liang & Wu, 2023), VaC (Yang et al., 2025a), and TAFAP (Lee et al., 2026). All the baselines

Algorithm 1 The working process of PuriShield

Require: Training images $\mathbb{X}$, condition $\mathbf{E}_{pd}$, DreamBooth model DB_θ, epoch number N, batch size B, update interval j, and perturbation budget ϵ
Ensure: Trained RPPG G_ϕ

Randomly initialize generator G_ϕ
repeat
  for $i = 1$ to N **do**
    Shuffle $\mathbb{X}$ and initialize the mini-batch iterator
    while unprocessed mini-batches remain **do**
      for $r = 1$ to j **do**
        Get the next mini-batch of B samples x from $\mathbb{X}$
        $\delta = G_\phi(x||\mathbf{E}_{pd})$ // Clipping is performed before generator output to restrict the perturbation within $[-\epsilon, \epsilon]$, i.e., $\delta = \text{Clip}(\mathbf{A} \odot \mathbf{W}, -\epsilon, \epsilon)$
        Dynamic block mask optimization: $\hat{x} = \text{Clip}(x + \mathcal{M} \odot \delta, 0, 1)$
        Calculate the latent distribution contrastive loss using Eq. (2) of the main text:

$$\mathcal{L}_{ldc} = \max(0, D^{\mu}_{pos} - D^{\mu}_{neg} + m_1) + \max(0, D^{v}_{pos} - D^{v}_{neg} + m_2)$$

        Calculate the fine-tuning loss using Eq. (3) of the main text:

$$\mathcal{L}_{ft} = \mathbb{E}_{z_0^\delta \sim \mathcal{E}(\delta), \varepsilon \sim \mathcal{N}(0,I), t} \|\varepsilon - \varepsilon_\theta(z_t^\delta, t, c)\|_2^2$$

        Calculate the perceptual feature loss using Eq. (4) of the main text:

$$\mathcal{L}_p = \mathbb{E}_{t,l} \left\| f_l^t(\hat{x}) - f_l^t(x) \right\|_2^2$$

        Freeze DB_θ and optimize G_ϕ by minimizing $\mathcal{L}_{total} = \mathcal{L}_{ldc} + \lambda \mathcal{L}_{ft} + \alpha \mathcal{L}_p$
      end for
      for $r = 1$ to j **do**
        Get the next mini-batch of B samples x from $\mathbb{X}$
        $\delta = G_\phi(x||\mathbf{E}_{pd})$ // Clipping is performed before generator output
        Calculate the fine-tuning loss $\mathcal{L}_{ft}$ using Eq. (3) of the main text
        Freeze G_ϕ and optimize DB_θ by minimizing $\mathcal{L}_{ft}$
      end for
    end while
  end for
until generator G_ϕ converges
return trained RPPG G_ϕ

employ PGD to generate perturbations, where perturbation budget is constrained by L_∞ norm with $\epsilon = 8/255$. For a fair and comprehensive comparison, all baseline methods are implemented utilizing their official open-source codes, while we reimplement TAFAP from its original paper due to the absence of public implementations. We adopt their default hyperparameter settings as reported in the original literature unless otherwise specified. For ASPL and Anti-Diff, the defense procedure is run for 50 iterations. Within each iteration, the surrogate model is updated for 3 steps, while the perturbations are updated for 6 steps, where step size is set to $\epsilon/10$. TAFAP uses 2000 iterations, VaC uses 30 iterations, and BlurGuard uses 150 iterations. All other baseline methods adopt 100 steps of PGD updates, with the step size fixed at 1/255.

B.3 Algorithmic Description of PuriShield

The overall working process of PuriShield is described in Alg. 1. Specifically, PuriShield adopts the alternating training strategy: DreamBooth is frozen when RPPG is updated for j mini-batches under the total loss $\mathcal{L}_{total} = \mathcal{L}_{ldc} + \lambda \mathcal{L}_{ft} + \alpha \mathcal{L}_p$, where λ and α are balanced hyper-parameters; subsequently, RPPG is frozen, and the perturbations generated by the current RPPG are used to fine-tune DreamBooth under loss $\mathcal{L}_{ft}$ for the next j mini-batches. This alternating procedure is repeated throughout training. Once trained, the RPPG can be used as a portable defense functionality to protect images against unauthorized customization.

Table 8: Quantitative comparison of pre-purification protection efficacy against **SDEdit**.

Methods	CelebA-HQ				VGGFace2			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
Clean	72.159	0.537	8.685	4.268	102.410	0.572	15.437	4.385
ASPL	197.243	0.323	-7.094	2.040	254.109	0.345	-6.641	2.194
Anti-Diff	140.445	0.319	-5.703	2.356	264.965	0.366	-3.463	2.587
Anti-Pure	110.336	0.430	5.293	2.261	232.283	0.389	5.812	3.098
PhotoGuard	156.814	0.271	-4.195	2.192	258.266	0.283	2.578	2.363
ACE	210.292	0.259	-8.258	2.132	267.407	0.253	-6.867	2.311
BlurGuard	118.550	0.368	4.754	3.890	234.266	0.392	6.469	3.838
Ours	**223.526**	**0.235**	**-10.3047**	**1.825**	**291.058**	**0.231**	**-8.883**	**1.830**

B.4 Metric Definitions

To assess the visual quality degradation of unauthorized generated outputs, we adopt several evaluation metrics spanning overall image quality, identity preservation, and stylistic similarity.

- **Fréchet Inception Distance (FID) (Heusel et al., 2017):** FID measures the distributional discrepancy between the synthesized images and the real images. In the context of anti-customization, a higher FID score indicates that the generated images deviate significantly from the original natural data manifold, implying stronger protective efficacy.
- **Identity Similarity Metric (ISM) (Deng et al., 2019):** For facial datasets, we employ ISM to measure how well facial identity features are preserved during customized generation. It computes the feature similarity between the original source images and the synthesized outputs. A lower ISM value signifies that the identity features are successfully erased or distorted, indicating effective protection.
- **Image Quality Score (IQS) (Radford et al., 2021):** IQS is defined to assess the quality of the generated images with the powerful visual-language model CLIP (Cherti et al., 2023). A lower IQS value indicates more severe perceptual degradation, demonstrating that the protective perturbation successfully disrupted the synthesis process of diffusion models.
- **Language-Image Quality Evaluator (LIQE) (Zhang et al., 2023):** LIQE is a no-reference vision-language metric that evaluates perceptual image quality using image-text correspondence. It converts predicted probabilities over five quality levels into a continuous score, where a lower LIQE score indicates worse visual quality and stronger defensive performance.
- **Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018):** For evaluating cross-domain generalizability, such as defense against artistic style mimicry, LPIPS is used instead of ISM to measure the perceptual distance between images. A higher LPIPS score indicates a greater perceptual divergence from the targeted artistic style, confirming successful protection.
- **Multi-Scale Structural Similarity (MS) (Wang et al., 2003):** MS can be applied in conjunction with LPIPS to evaluate the multi-scale structural integrity and stylistic degradation in artistic style transfer tasks. A lower MS value suggests that the structural patterns and stylistic characteristics of the original artwork are effectively disrupted by the defense.
- **Face Detection False Rate (FDFR) (Deng et al., 2020):** For facial datasets, FDFR reflects if the generated images preserve recognizable facial structure. Given an off-the-shelf face detector (Retinaface), FDFR is calculated as $\mathrm{FDFR} = 1 - \frac{\mathrm{N_{det}}}{\mathrm{N}}$, where N is the total number of generated images and $\mathrm{N_{det}}$ is the number of images in which a valid face is successfully detected.
- **Face Detection Score (FDS) (Zhang et al., 2016):** For facial datasets, FDS further measures the visual quality of the detected faces. Given a face analysis model (MTCNN), FDS is defined as the average face-quality score over the generated samples.

Table 9: Quantitative comparison of pre-purification protection efficacy against **Textual Inversion**.

Methods	CelebA-HQ				VGGFace2			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
Clean	103.729	0.599	8.022	4.197	112.689	0.663	16.305	4.353
ASPL	346.283	0.229	-12.055	2.184	363.384	0.211	-7.813	2.107
Anti-Diff	302.708	0.275	-12.188	2.265	313.094	0.253	-13.547	2.290
Anti-Pure	123.011	0.478	1.227	2.876	207.243	0.502	11.313	3.965
PhotoGuard	358.550	0.244	-10.625	1.936	325.726	0.229	-9.093	2.137
ACE	232.895	0.192	-3.033	3.132	278.000	0.190	8.320	3.276
BlurGuard	170.636	0.468	-0.989	2.164	223.620	0.419	11.547	3.684
Ours	**410.910**	**0.172**	**-15.539**	**1.627**	**435.607**	**0.153**	**-17.047**	**1.585**

Table 10: Quantitative comparison of pre-purification protection efficacy against **LoRA**.

Methods	CelebA-HQ				VGGFace2			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
Clean	87.652	0.723	7.685	3.972	107.124	0.766	18.438	4.122
ASPL	263.719	0.363	-8.206	2.347	305.414	0.283	-10.703	2.352
Anti-Diff	252.831	0.387	-4.266	2.606	284.595	0.295	-5.344	2.538
Anti-Pure	160.693	0.568	3.338	3.172	204.443	0.621	15.093	2.717
PhotoGuard	242.412	0.361	-8.500	2.418	271.352	0.310	-9.750	2.107
ACE	256.223	0.343	-4.547	3.152	316.698	0.281	-9.055	2.334
BlurGuard	130.117	0.628	0.440	2.506	222.017	0.471	8.945	2.988
Ours	**306.182**	**0.303**	**-15.180**	**1.262**	**359.433**	**0.255**	**-20.484**	**2.033**

Table 11: Quantitative comparison of pre-purification protection efficacy against **Custom Diffusion**.

Methods	CelebA-HQ				VGGFace2			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
Clean	112.389	0.536	7.239	4.349	118.036	0.542	11.682	4.386
ASPL	192.617	0.290	-6.957	1.412	234.153	0.269	-2.777	2.768
Anti-Diff	209.834	0.392	-10.219	1.600	304.168	0.305	-9.555	1.971
Anti-Pure	123.794	0.454	-0.726	3.534	224.581	0.475	9.773	4.380
PhotoGuard	187.882	0.297	-11.352	1.528	226.666	0.362	-4.450	2.914
ACE	291.437	0.288	-0.610	2.797	267.937	0.276	1.243	3.538
BlurGuard	129.383	0.446	-3.867	4.074	213.201	0.436	9.602	3.300
Ours	**340.268**	**0.157**	**-24.188**	**1.262**	**390.766**	**0.121**	**-27.953**	**1.727**

Table 12: Post-purification protection robustness against **SDEdit** on CelebA-HQ.

Methods	Gaussian				Noisy Upscaling				GrIDPure				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	133.265	0.329	2.793	1.755	124.415	0.397	6.664	4.282	116.166	0.443	6.027	2.649	154.932	0.363	0.572	1.627
Anti-Diff	130.415	0.341	-5.090	1.924	113.952	0.412	6.203	3.945	117.628	0.415	6.211	2.861	135.771	0.356	-1.351	1.406
Anti-Pure	133.025	0.332	-3.121	1.742	113.313	0.383	7.773	3.762	118.241	0.409	8.852	3.019	135.382	0.345	3.822	1.524
PhotoGuard	127.933	0.318	5.027	1.865	119.382	0.365	5.066	4.222	122.728	0.339	6.684	2.472	149.653	0.349	5.715	1.633
ACE	126.739	0.310	2.402	1.882	123.009	0.376	7.851	4.635	128.956	0.340	4.313	2.402	151.146	0.340	3.731	1.529
BlurGuard	131.473	0.310	4.098	2.073	117.949	0.373	6.508	4.823	121.626	0.438	5.316	2.773	153.018	0.324	0.656	1.528
Ours	**141.788**	**0.276**	**-13.531**	**1.282**	**181.538**	**0.307**	**-4.981**	**3.614**	**169.372**	**0.321**	**-14.109**	**1.950**	**190.437**	**0.302**	**-20.078**	**1.005**

C Evaluation on More Customization Methods

Beyond **DreamBooth**, we evaluate protection efficacy and robustness across four mainstream customization approaches: **SDEdit**, **Textual Inversion**, **LoRA**, and **Custom Diffusion**.

C.1 Protection Effectiveness

Experiments are conducted on datasets CelebA-HQ and VGGFACE2. As shown in Table 8-Table 11, PuriShield consistently outperforms baseline methods across all metrics under different customization settings. These results, together with those on DreamBooth presented in the main text, indicate that structured protective perturbations can effectively disrupt various customized generation.

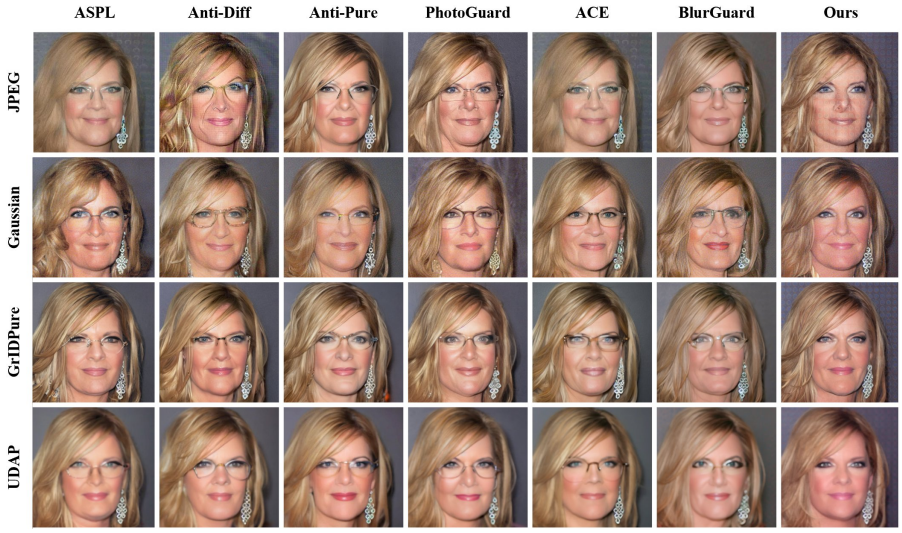


Figure 8: Output images generated by **SDEdit** after various purification methods. The instance is from CelebA-HQ and the prompt is "a close-up portrait photo of sks person wearing **eyeglasses**".

Table 13: Post-purification protection robustness against **Textual Inversion** on CelebA-HQ.

Methods	Gaussian				Noisy Upscaling				GrIDPure				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	150.306	0.394	-0.201	1.930	164.747	0.425	0.098	3.724	148.691	0.421	7.195	3.674	230.635	0.397	-1.495	2.182
Anti-Diff	176.940	0.356	-0.977	2.176	223.472	0.334	-5.164	2.284	183.718	0.367	-3.815	2.246	150.068	0.456	4.496	2.467
Anti-Pure	166.330	0.379	-0.867	2.400	183.694	0.438	7.070	2.497	130.810	0.404	-1.208	3.258	177.082	0.440	8.086	3.428
PhotoGuard	154.363	0.444	8.367	2.977	187.020	0.377	8.523	2.342	159.429	0.507	-2.797	2.565	183.420	0.435	0.824	2.212
ACE	150.572	0.387	1.233	1.950	166.272	0.365	11.883	3.351	175.210	0.385	6.918	3.438	255.607	0.412	4.016	2.953
BlurGuard	132.921	0.449	1.887	2.456	168.757	0.499	-1.037	3.148	172.516	0.372	3.449	3.557	251.977	0.357	1.709	2.406
Ours	**426.764**	**0.083**	**-16.266**	**1.867**	**242.530**	**0.294**	**-7.002**	**2.196**	**408.777**	**0.089**	**-21.063**	**1.853**	**311.785**	**0.301**	**-8.102**	**2.045**

Table 14: Post-purification protection robustness against **LoRA** on CelebA-HQ.

Methods	Gaussian				Noisy Upscaling				GrIDPure				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	116.175	0.669	-8.047	1.736	102.529	0.755	5.164	4.785	116.450	0.741	-0.836	2.722	170.019	0.722	-6.055	1.418
Anti-Diff	124.691	0.606	-3.346	1.518	108.900	0.769	4.383	4.310	118.946	0.739	-2.881	2.859	107.235	0.670	-12.922	1.633
Anti-Pure	115.996	0.707	5.797	2.656	101.235	0.745	8.633	4.996	102.176	0.687	16.484	3.449	131.694	0.745	-3.990	1.236
PhotoGuard	109.763	0.576	-10.359	1.302	109.645	0.729	9.750	4.964	109.304	0.762	5.090	3.868	146.368	0.702	0.598	1.510
ACE	171.017	0.671	-4.707	1.480	107.491	0.731	14.594	4.979	120.648	0.729	0.024	3.367	123.238	0.706	-4.168	1.312
BlurGuard	110.356	0.577	-1.782	2.272	108.879	0.714	6.836	4.936	125.768	0.709	-1.893	2.944	128.009	0.722	-9.805	1.266
Ours	**181.788**	**0.479**	**-13.531**	**1.182**	**185.538**	**0.658**	**-4.981**	**3.014**	**205.372**	**0.567**	**-14.109**	**1.650**	**190.437**	**0.605**	**-15.078**	**1.005**

C.2 Protection Robustness

We compare the protection robustness against four representative purification methods including Gaussian, Noisy Upscaling, GrIDPure, and UDAP. The experiments are conducted on CelebA-HQ dataset. As shown in Table 12-Table 15, PuriShield consistently yields the best robustness after diverse purification across all four metrics under different customization settings. These results, together with those on DreamBooth presented in the main text, validate that structured protective perturbations effectively enhance robustness against various forms of unauthorized customization.

As shown in Figs. 8-11, adversarial purification largely removes the protective perturbations of baseline methods, thereby restoring identity information across different personalization methods. In contrast, PuriShield produces purification-resistant structural perturbations and consistently preserves its defensive effectiveness across diverse personalization methods.

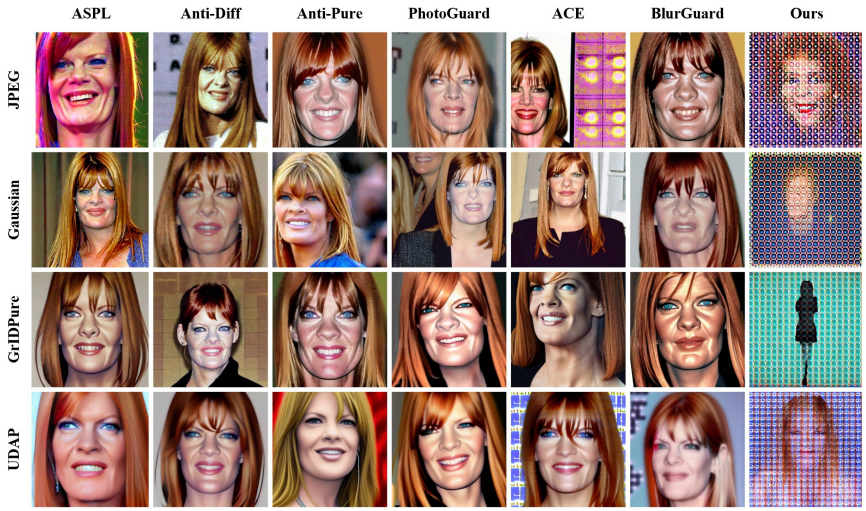


Figure 9: Output images generated by **Text Inversion** after purification (CelebA-HQ).

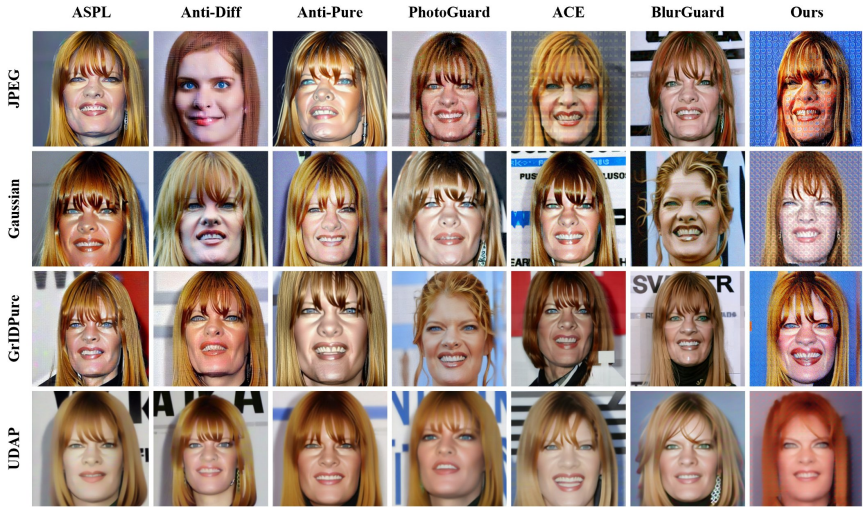


Figure 10: Output images generated by **LoRA** after purification (CelebA-HQ).

Table 15: Post-purification protection robustness against **Custom Diffusion** on CelebA-HQ.

Methods	Gaussian				Noisy Upscaling				GrIDPure				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	130.619	0.474	-7.590	1.865	190.993	0.314	2.562	4.589	146.800	0.421	-4.082	3.502	169.141	0.509	-2.975	2.739
Anti-Diff	122.705	0.470	-8.617	2.233	187.610	0.318	-1.198	3.833	165.731	0.398	-10.828	2.045	177.913	0.465	-13.773	1.865
Anti-Pure	114.790	0.509	-6.754	2.534	141.718	0.442	5.090	4.824	154.012	0.439	-8.656	3.528	165.516	0.505	-9.008	1.967
PhotoGuard	124.772	0.506	-2.945	1.880	155.060	0.341	4.227	4.781	174.402	0.480	-12.094	2.035	182.050	0.507	-6.277	1.829
ACE	129.537	0.474	-4.289	2.162	161.670	0.327	0.244	4.498	158.186	0.459	3.854	4.450	181.849	0.478	-0.351	2.721
BlurGuard	125.638	0.491	-5.922	2.072	179.743	0.304	3.357	4.695	172.710	0.421	6.754	4.885	187.930	0.476	-2.898	2.253
Ours	**197.113**	**0.342**	**-9.477**	**1.651**	**208.551**	**0.288**	**-10.461**	**2.962**	**253.714**	**0.326**	**-16.141**	**1.988**	**242.309**	**0.404**	**-15.156**	**1.706**

D More Results on Black-box Performance

We assess black-box transferability across unseen versions (**SD v1.4** and **SD v2.1base**), unseen architectures (**SDXL 1.0** and **SD v3**), and real-world models (**FLUX.1-dev1** and **Qwen3.5**). Experiments are conducted on CelebA-HQ, using SD v1.5 as the white-box model.

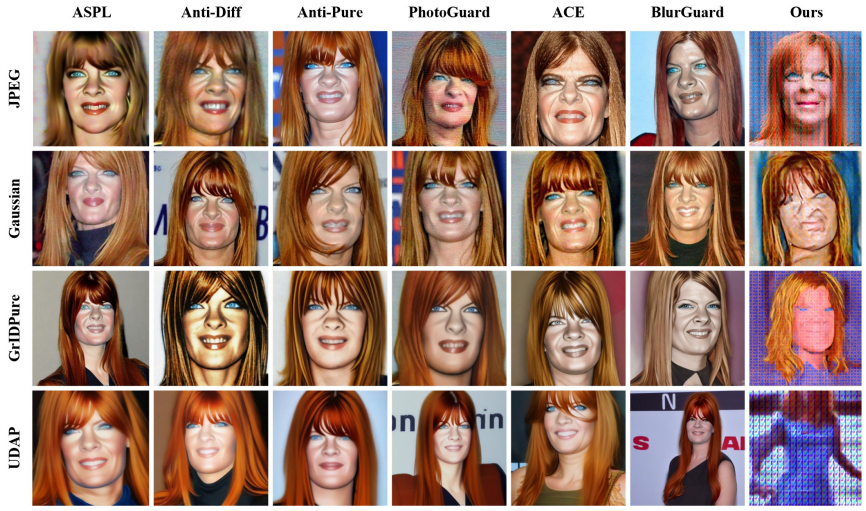


Figure 11: Output images generated by **Custom Diffusion** after purification (CelebA-HQ).

Table 16: Cross-version transferability before and after UDAP purification (CelebA-HQ).

Models	Methods	Before UDAP Purification				After UDAP Purification			
		FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
SD v1.4	ASPL	195.198	0.276	-15.492	1.677	212.163	0.376	-14.914	1.623
	Anti-Diff	187.732	0.500	-10.375	1.637	180.945	0.395	-18.609	1.883
	Anti-Pure	147.662	0.481	-3.791	2.248	158.860	0.458	-9.539	2.024
	PhotoGuard	183.188	0.386	-10.555	1.620	187.380	0.465	-13.344	1.546
	ACE	316.581	0.135	-13.984	1.716	225.499	0.363	-14.359	1.708
	BlurGuard	139.855	0.635	-7.508	2.550	157.310	0.465	-11.688	2.188
	Ours	**363.308**	**0.082**	**-28.656**	**1.445**	**243.648**	**0.306**	**-21.422**	**1.382**
SD v2.1-base	ASPL	269.895	0.390	-18.312	1.767	182.140	0.401	-15.266	1.322
	Anti-Diff	269.479	0.388	-17.578	2.057	193.089	0.433	-16.500	1.289
	Anti-Pure	157.604	0.536	-2.320	2.136	210.382	0.386	-11.578	1.310
	PhotoGuard	219.853	0.430	-15.531	1.937	239.478	0.301	-15.750	1.345
	ACE	288.163	0.363	-8.445	1.862	265.193	0.285	-13.250	1.313
	BlurGuard	198.335	0.420	-7.637	1.892	227.800	0.357	-16.547	1.304
	Ours	**433.174**	**0.041**	**-19.891**	**1.678**	**271.093**	**0.264**	**-18.984**	**1.184**

Table 17: Cross-architecture transferability before and after UDAP purification (CelebA-HQ).

Models	Methods	Before UDAP Purification				After UDAP Purification			
		FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
SDXL 1.0	ASPL	216.373	0.531	-15.547	1.204	161.294	0.475	-9.352	1.210
	Anti-Diff	235.223	0.532	-3.022	1.305	164.432	0.412	-2.391	1.302
	Anti-Pure	152.896	0.568	-2.006	1.174	204.498	0.415	-5.945	1.158
	PhotoGuard	139.871	0.544	-1.671	1.167	173.436	0.458	-4.777	1.084
	ACE	154.176	0.422	-13.781	1.521	196.268	0.341	-11.406	1.080
	BlurGuard	141.065	0.532	-3.800	1.112	180.753	0.434	-5.731	1.072
	Ours	**387.394**	**0.146**	**-29.109**	**1.008**	**218.236**	**0.305**	**-18.094**	**1.018**
SD v3	ASPL	164.631	0.528	-0.390	2.491	156.063	0.396	-8.234	1.341
	Anti-Diff	149.031	0.536	1.915	2.581	162.752	0.477	-6.047	1.315
	Anti-Pure	181.798	0.521	6.309	4.359	129.610	0.494	-1.301	1.249
	PhotoGuard	169.641	0.444	-2.982	2.432	193.543	0.419	-3.750	2.155
	ACE	176.388	0.504	-7.531	2.831	160.846	0.404	-9.484	1.751
	BlurGuard	154.904	0.382	-10.016	3.118	175.492	0.391	-11.031	1.403
	Ours	**203.125**	**0.141**	**-15.781**	**2.221**	**195.648**	**0.328**	**-17.328**	**1.017**

D.1 Unseen Model Versions and Architectures

Firstly, we evaluate the protect efficacy when transferring from SD v1.5 to SD v1.4 (CompVis, 2022) and SD v2.1base (Quang, 2023). As shown in Table 16, PuriShield consistently achieves superior performance before and after UDAP purification, yielding higher FID while maintaining lower ISM, IQS and LIQE compared to baselines. This demonstrates its strong robustness against purification and strong cross-version transferability ability in disrupting diffusion customization.

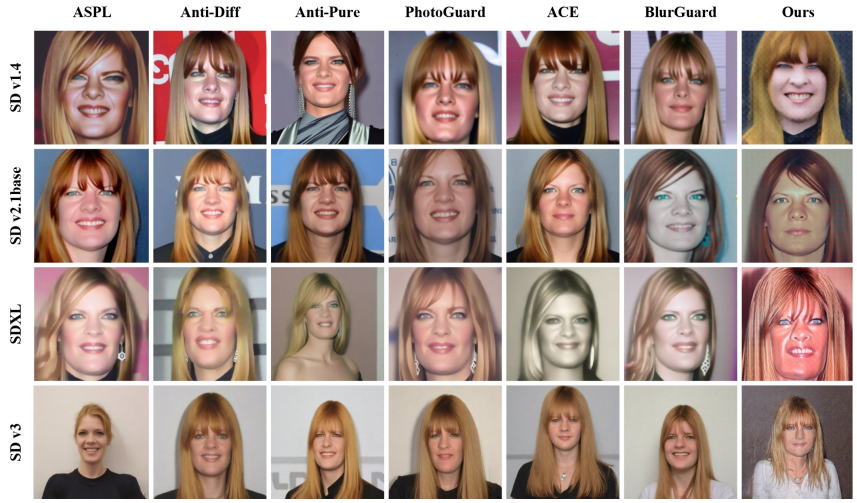


Figure 12: Output images from unseen diffusion model versions and unseen model architectures after UDAP purification. The instance is from CelebA-HQ.

Table 18: Black-box transferability against real-world image generation services.

Methods	Flux.1 dev				Qwen3.5			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	108.125	0.581	2.823	3.827	119.335	0.538	2.269	3.582
Anti-Diff	101.332	0.543	2.205	3.685	98.648	0.537	2.360	3.863
Anti-Pure	113.255	0.556	3.605	4.025	100.231	0.542	3.897	3.968
PhotoGuard	106.834	0.553	3.487	3.483	101.036	0.531	3.692	3.596
ACE	116.325	0.518	2.011	3.503	113.193	0.496	2.152	3.420
BlurGuard	99.837	0.564	3.741	4.026	104.850	0.584	3.983	4.089
Ours	**128.153**	**0.405**	**1.482**	**3.113**	**151.413**	**0.420**	**1.337**	**3.218**



Figure 13: Output images from real-world image generation service **Flux.1 dev** (CelebA-HQ).

Secondly, we evaluate the cross-architecture transferability of all anti-customization methods, by using the protective images crafted from SD v1.5 to defend against unauthorized customization based on SDXL 1.0 and SD v3. Speicifcally, SDXL 1.0 adopts a substantially larger UNet and uses two text encoders, namely CLIP ViT-L/14 and OpenCLIP ViT-bigG/14, to provide richer text conditioning. Unlike SD v1.5 that uses a UNet-based latent diffusion architecture with a single CLIP ViT-L/14 text encoder, SD3 adopts a Multimodal Diffusion Transformer (MMDiT) and three fixed pretrained text encoders, including CLIP ViT-L/14, OpenCLIP ViT-bigG/14, and T5-v1.1-XXL. These architectural changes improve text conditioning and image generation quality, making SD v3 substantially different from SD v1.5. Table 17 shows that PuriShield achieves the best protection performance before and after UDAP purification. Fig. 12 further shows that PuriShield exhibits optimal protection robustness against unauthorized personalization when transferred to unseen model versions and architectures. These results demonstrate that our perturbations exhibit excellent cross-version and cross-architecture transferability, while maintaining robustness against diverse purification.

Table 19: Quantitative comparison of new metrics before and after purification on CelebA-HQ.

Methods	w/o Purification		Gaussian		Noisy Upscaling		GrIDPure		UDAP	
	FDFR ↑	FDS ↓	FDFR ↑	FDS ↓	FDFR ↑	FDS ↓	FDFR ↑	FDS ↓	FDFR ↑	FDS ↓
ASPL	0.125	0.604	0.001	0.656	0.063	0.598	0.125	0.511	0.001	0.514
Anti-Diff	0.005	0.615	0.001	0.601	0.002	0.727	0.002	0.663	0.188	0.424
Anti-Pure	0.010	0.733	0.001	0.686	0.001	0.768	0.005	0.677	0.125	0.468
PhotoGuard	0.010	0.616	0.006	0.597	0.002	0.668	0.002	0.691	0.188	0.394
ACE	0.375	0.620	0.006	0.736	0.001	0.715	0.010	0.486	0.001	0.449
BlurGuard	0.001	0.755	0.003	0.760	0.063	0.705	0.063	0.486	0.063	0.449
Ours	**0.916**	**0.367**	**0.063**	**0.382**	**0.125**	**0.504**	**0.265**	**0.362**	**0.225**	**0.340**

Table 20: Visual quality of protected images under varying perturbation budgets (CelebA-HQ).

Methods	$\epsilon = 4/255$			$\epsilon = 8/255$			$\epsilon = 12/255$		
	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
ASPL	0.936	38.059	0.288	0.823	32.455	0.414	0.699	28.217	0.497
Anti-Diff	0.876	33.656	0.332	0.783	30.654	0.343	0.637	28.541	0.514
Anti-Pure	0.982	**43.516**	0.104	0.829	**35.360**	0.381	0.643	29.635	0.489
PhotoGuard	0.916	37.424	0.329	0.768	31.938	0.454	0.639	28.971	0.591
ACE	0.928	37.559	0.447	0.810	32.244	0.540	0.697	29.391	0.662
BlurGuard	**0.990**	38.016	**0.097**	**0.961**	33.822	**0.133**	**0.912**	**31.494**	**0.160**
Ours	0.943	37.866	0.386	0.807	32.421	0.465	0.694	28.990	0.572

D.2 Real-World Scenarios

To evaluate the practical applicability of our method, we conduct transferability tests on two state-of-the-art commercial image generation services under real-world black-box settings: FLUX.1-dev[1] and Qwen3.5[2]. Despite the proprietary preprocessing and image compression adopted by these online platforms, which may inadvertently act as strong black-box purifications, our method still effectively disrupts their generation pipelines. As shown in Table 18 and Fig. 13, our approach consistently outperforms all baselines across all metrics, demonstrating its strong practical robustness and superior black-box transferability against commercial online services.

E Evaluation on More Metrics

This section evaluates the protection effectiveness and robustness of all methods under the newly introduced **FDFR** and **FDS** metrics, and measures the imperceptibility of our protected images via peak signal-to-noise ratio (**PSNR**), structural similarity index (**SSIM**), and **LPIPS**.

E.1 Effectiveness and Robustness

In the main text, we evaluate the protective efficacy by using four metrics: FID, ISM, IQS and LIQE. In this section, we adopt additional metrics for assessing the effectiveness and robustness of generated images: FDFR and FDS on CelebA-HQ. As shown in Table 19, our PuriShield consistently outperforms all baselines across FDFR and FDS metrics before and after purification. This table further compares protection robustness against four representative purification methods including Gaussian, Noisy Upscaling, GrIDPure, and UDAP. Our method yields the highest FDFR and lowest FDS across all tested scenarios, demonstrating stable protection performance against these purification methods, particularly exhibiting superior performance against DP.

E.2 Imperceptibility of Protected Images

We use PSNR (Hore & Ziou, 2010), SSIM (Wang et al., 2004), and LPIPS to measure the imperceptibility of the protected images. Table 20 quantitatively compares the visual perceptibility of images protected under different perturbation magnitudes (ϵ). As expected, increasing ϵ introduces

[1] https://replicate.com/

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



Figure 14: Protected images from PuriShield with varying perturbation budgets (CelebA-HQ).

Table 21: Quantitative comparison of protection efficacy before and after UDAP purification under different DreamBooth prompts and subject identifiers on CelebA-HQ. S_* is "*sks*" for the first prompt setting and "*t@t*" for the second prompt setting.

Prompts	Methods	Before Purification								After Purification							
		a photo of S_* person				a close-up photo of S_* person				a photo of S_* person				a close-up photo of S_* person			
		FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
a close-up photo of *sks* person	ASPL	240.992	0.347	-11.742	2.349	206.598	0.390	-15.078	2.316	211.789	0.393	-10.126	1.536	168.182	0.661	-10.297	1.292
	Anti-Diff	227.055	0.286	-12.586	2.380	170.829	0.609	-15.297	2.534	183.906	0.398	-6.781	1.824	179.885	0.516	-11.859	1.597
	Anti-Pure	222.220	0.348	-3.143	2.438	119.468	0.646	1.091	2.769	280.874	0.295	-8.641	1.629	168.690	0.581	-3.190	1.512
	PhotoGuard	198.333	0.310	-14.125	2.451	189.386	0.478	-15.547	2.410	187.317	0.397	-12.656	1.663	163.511	0.570	-8.805	1.369
	ACE	258.754	0.307	-9.195	2.379	333.750	0.197	-5.844	3.233	220.040	0.444	-5.305	2.474	175.956	0.612	-11.109	1.307
	BlurGuard	236.811	0.379	-6.578	2.367	186.161	0.700	-4.645	2.878	207.150	0.310	-8.391	2.045	168.996	0.626	-6.563	2.057
	Ours	**396.150**	**0.039**	**-25.000**	**1.869**	**416.474**	**0.068**	**-17.891**	**1.716**	**295.234**	**0.283**	**-18.219**	**1.203**	**225.437**	**0.478**	**-23.828**	**1.119**
sks → *t@t*	ASPL	222.329	0.351	-4.371	2.346	175.739	0.283	6.547	2.869	201.165	0.196	-3.297	1.995	162.536	0.695	5.267	2.211
	Anti-Diff	207.906	0.441	-3.982	2.379	163.919	0.415	1.893	3.073	223.171	0.182	-2.313	2.017	141.562	0.660	-8.523	1.636
	Anti-Pure	123.382	0.570	4.762	3.838	140.316	0.463	16.047	3.688	225.863	0.186	-4.469	2.198	130.348	0.681	-3.990	1.557
	PhotoGuard	186.187	0.367	-7.539	2.351	181.259	0.295	16.047	3.189	246.858	0.192	-8.820	1.947	161.020	0.698	-6.340	1.677
	ACE	264.473	0.292	-5.004	3.561	276.078	0.293	3.510	2.744	201.559	0.183	-6.641	1.978	178.003	0.657	-8.628	1.283
	BlurGuard	112.576	0.593	-2.426	3.541	148.610	0.366	19.938	4.879	216.039	0.182	-8.336	1.866	131.772	0.746	-3.254	1.450
	Ours	**289.796**	**0.231**	**-8.305**	**2.184**	**408.604**	**0.023**	**-12.336**	**2.682**	**323.587**	**0.153**	**-11.023**	**1.774**	**216.072**	**0.651**	**-9.969**	**1.185**

more noticeable distortions, reflected by the continuous drop in SSIM/PSNR and the rise in LPIPS. To balance visual naturalness and protection robustness of protected images, we adopt $\epsilon = 8/255$ as the default setting, which yields an optimal trade-off. As shown in Fig. 14, when $\epsilon = 8/255$, the protected images generated by our method look natural and achieve visually acceptable quality, demonstrating a favorable balance between protection effectiveness and visual imperceptibility.

F More Results on Effectiveness

Unseen prompts. We use the prompt "a photo of *sks* person" to generate protective perturbations, while using the following prompts for unauthorized personalized generation: ① a close-up photo of *sks* person; ② *sks* → *t@t*. As shown in Table 21, our PuriShield consistently achieves the best protection performance across all types of unseen prompts. These results demonstrate that our PuriShield generalizes effectively across diverse prompts and identifiers.

Unfrozen text encoder. In the main text, the text encoder is kept frozen during fine-tuning DreamBooth. Taking into account potential modifications to the text encoder, we also evaluate the protection efficacy against DreamBooth personalization when the text encoder is unfrozen during training.

Table 22: Protection efficacy under unfrozen text encoder before and after UDAP (CelebA-HQ).

Methods	Before Purification				After Purification			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	239.168	0.379	-15.391	2.051	126.628	0.438	-7.242	1.374
Anti-Diff	249.739	0.486	-10.289	2.115	103.721	0.518	-7.805	1.618
Anti-Pure	139.660	0.615	8.727	3.197	129.416	0.550	3.219	1.612
PhotoGuard	235.748	0.464	-11.859	2.300	98.323	0.475	-6.695	1.333
ACE	237.115	0.389	-13.002	2.626	115.683	0.570	-4.715	1.606
BlurGuard	121.951	0.556	-8.055	2.429	133.825	0.435	-1.655	1.793
Ours	**384.807**	**0.068**	**-18.578**	**1.051**	**255.322**	**0.284**	**-12.969**	**1.062**

Table 23: Protection efficacy across perturbation budgets before and after UDAP (CelebA-HQ).

ϵ	Methods	Before Purification				After Purification			
		FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
4/255	ASPL	167.794	0.516	-5.234	2.312	156.487	0.526	-3.379	2.365
	Anti-Diff	154.800	0.552	-4.125	2.278	151.395	0.563	-11.867	2.288
	Anti-Pure	132.098	0.555	2.654	3.825	127.930	0.569	-2.164	3.831
	PhotoGuard	168.150	0.568	-6.421	2.465	160.055	0.578	-6.578	2.674
	ACE	191.240	0.488	-6.234	2.900	161.622	0.503	-7.503	2.952
	BlurGuard	152.661	0.630	-0.923	3.240	147.304	0.633	-4.641	3.288
	Ours	**209.945**	**0.403**	**-13.820**	**2.067**	**194.323**	**0.428**	**-16.781**	**1.831**
12/255	ASPL	259.983	0.339	-12.875	1.931	178.493	0.352	0.420	2.121
	Anti-Diff	171.194	0.428	-13.867	2.191	156.287	0.530	-7.683	2.243
	Anti-Pure	158.988	0.509	0.237	2.944	158.284	0.551	0.412	3.051
	PhotoGuard	206.047	0.365	-12.094	1.992	202.465	0.386	-1.869	2.133
	ACE	362.873	0.158	-12.875	2.475	182.450	0.463	-2.181	2.553
	BlurGuard	178.253	0.529	-8.758	2.276	177.427	0.536	-6.628	2.526
	Ours	**408.512**	**0.022**	**-18.631**	**1.465**	**322.968**	**0.216**	**-22.484**	**1.207**



Figure 15: DreamBooth outputs from PuriShield across perturbation budgets (CelebA-HQ).

As shown in Table 22, our PuriShield consistently outperforms all baselines across all quantitative metrics, demonstrating its strong adaptability to different text encoder settings.

Different perturbation magnitudes. In the main text, the perturbation budget is constrained by L_∞ norm with $\epsilon = 8/255$. For comprehensiveness, we conduct experiments under varying perturbation budgets ($\epsilon = 4/255$ and $\epsilon = 12/255$). As shown in Table 23, our PuriShield achieves better protection performance as the perturbation budget increases. Furthermore, our method consistently outperforms all baselines across varying perturbation budgets. Fig. 15 shows that larger perturbation

Table 24: Protection robustness against transformation-based purification methods on VGGFace2.

Methods	JPEG				Gaussian				Noisy Upscaling				SR			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	249.389	0.379	-2.541	1.652	198.433	0.498	-3.806	2.198	211.607	0.445	6.796	3.027	276.480	0.374	-5.531	2.106
Anti-Diff	255.834	0.378	-7.289	1.576	220.614	0.445	-4.285	2.071	210.287	0.412	7.875	3.032	273.299	0.383	-8.718	2.160
Anti-Pure	231.595	0.382	5.015	2.359	213.615	0.483	0.046	1.982	234.549	0.437	9.203	3.261	207.444	0.496	5.004	2.476
PhotoGuard	237.652	0.402	-5.695	1.628	215.440	0.405	0.694	2.062	216.044	0.444	8.765	3.478	258.086	0.381	-9.625	2.174
ACE	270.387	0.375	-1.449	2.323	221.102	0.385	-0.122	2.082	191.406	0.415	7.367	3.605	289.649	0.396	-4.996	2.305
BlurGuard	212.174	0.550	5.324	2.392	205.598	0.449	1.358	2.456	211.439	0.411	8.976	3.549	248.772	0.414	-2.982	2.393
Ours	**303.448**	**0.327**	**-15.078**	**1.385**	**239.612**	**0.310**	**-13.234**	**1.701**	**255.711**	**0.353**	**-10.074**	**2.097**	**321.322**	**0.356**	**-17.781**	**1.851**

Table 25: Protection robustness against diffusion-based purification methods on VGGFace2.

Methods	DiffPure				GrIDPure				Impress				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	231.315	0.439	0.717	2.492	218.412	0.457	5.171	2.469	309.714	0.310	-6.929	1.935	225.233	0.426	-3.355	2.252
Anti-Diff	216.631	0.444	-0.518	2.077	231.380	0.372	6.011	2.074	313.319	0.360	-7.375	2.198	249.597	0.316	-3.982	1.870
Anti-Pure	231.755	0.457	4.105	2.656	188.777	0.506	7.710	3.015	197.410	0.548	6.570	2.433	252.245	0.319	-1.570	2.090
PhotoGuard	224.854	0.520	-0.953	1.876	224.503	0.401	2.502	2.597	261.810	0.290	-7.891	2.002	218.094	0.309	0.557	1.962
ACE	246.474	0.352	-1.037	2.032	203.295	0.484	5.211	2.981	306.539	0.313	-3.685	2.644	225.910	0.387	-3.359	1.840
BlurGuard	228.678	0.412	5.242	2.888	214.902	0.398	2.212	2.877	214.256	0.549	6.804	3.053	227.593	0.475	2.082	2.398
Ours	**285.935**	**0.326**	**-13.585**	**1.195**	**288.754**	**0.291**	**-16.016**	**1.750**	**331.313**	**0.262**	**-16.688**	**1.636**	**272.112**	**0.249**	**-12.406**	**1.432**



Figure 16: Output images using unseen prompts and subject identifiers after UDAP (CelebA-HQ).

budgets lead to growing visual distortions and identity degradation in generated images, verifying improved protection against unauthorized personalization. Even with moderate budgets, our method can evidently disrupt the DreamBooth model, achieving favorable defensive performance.

G More Results on Robustness

Results on VGGFace2 dataset. Due to limited space, the main text only evaluates the protection robustness on CelebA-HQ dataset. For completeness, we provide the quantitative results on VGGFace2 dataset. Tables 24 and 25 present the robustness evaluation of different anti-customization methods against transformation-based and diffusion-based purification methods on the VGGFace2 dataset, respectively. Consistent with the results on CelebA-HQ, our method significantly outperforms existing baselines across most evaluation metrics. Notably, it sustains superior FID and ISM scores even after aggressive purification processes. This indicates that the strong protection capability of structured protective perturbations generalizes effectively across diverse facial datasets, confirming that such perturbations yield enhanced protection robustness.

Table 26: Robustness vs additional baselines: before and after purification (CelebA-HQ).

Methods	Before Purification				Noise				Noisy Upscaling				GrIDPure				UDAP			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
AdvDM	211.630	0.199	-6.328	2.331	130.676	0.654	-5.371	2.478	137.463	0.679	6.148	3.797	192.419	0.386	-0.839	2.610	162.329	0.540	-2.541	2.056
Mist	314.463	0.186	-6.251	3.238	126.585	0.567	-2.486	2.517	141.722	0.669	8.586	3.914	146.548	0.482	1.511	3.488	178.369	0.496	-6.273	1.449
VaC	**385.654**	**0.029**	**-26.593**	**1.465**	130.052	0.581	-1.999	2.352	116.612	0.670	9.765	4.061	146.777	0.617	7.820	3.724	194.919	0.449	-3.822	1.534
TAFAP	193.491	0.469	-3.639	1.970	160.084	0.649	-10.547	1.932	134.111	0.561	6.593	3.491	177.564	0.455	-2.731	1.914	159.172	0.477	-8.461	1.601
Ours	347.495	0.138	-17.547	1.895	**171.830**	**0.303**	**-11.313**	**1.466**	**217.711**	**0.414**	**-3.846**	**2.566**	**214.095**	**0.375**	**-9.602**	**1.825**	**216.215**	**0.365**	**-12.172**	**1.284**

Table 27: Protection robustness against more purification methods (CelebA-HQ).

Methods	Resizing				FD				FreqPure				LightShed				CAT			
	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓	FID ↑	ISM ↓	IQS ↓	LIQE ↓
ASPL	170.922	0.483	-6.883	2.174	184.678	0.545	-11.461	1.682	175.487	0.476	-1.175	2.519	182.665	0.589	-5.188	2.754	234.115	0.415	-12.922	2.010
Anti-Diff	188.692	0.457	-5.762	2.362	155.358	0.510	-10.148	1.710	188.751	0.448	0.542	2.156	198.901	0.425	-8.261	2.308	256.058	0.409	-12.227	2.056
Anti-Pure	147.846	0.574	-3.426	2.615	146.851	0.622	2.381	2.737	181.269	0.363	2.061	2.967	113.308	0.593	7.109	2.811	147.747	0.537	-3.289	2.389
PhotoGuard	170.689	0.490	-9.875	2.157	178.894	0.592	-9.492	1.563	186.033	0.449	-6.918	2.219	157.246	0.551	-2.450	2.848	190.315	0.481	-15.180	1.979
ACE	135.956	0.448	-8.266	3.252	144.499	0.556	-8.023	2.834	155.402	0.492	-3.785	2.630	163.114	0.485	-1.931	3.013	305.746	0.398	-5.805	2.763
BlurGuard	128.108	0.588	-6.949	3.007	127.370	0.674	0.129	3.840	179.323	0.459	5.188	2.445	151.720	0.604	-4.973	2.826	147.736	0.723	-6.469	2.406
Ours	**248.984**	**0.418**	**-11.133**	**1.931**	**240.277**	**0.421**	**-12.406**	**1.446**	**215.275**	**0.334**	**-8.277**	**2.003**	**221.320**	**0.361**	**-13.555**	**1.863**	**335.710**	**0.323**	**-16.172**	**1.686**

Unseen prompts. We still adopt the prompt "a photo of *sks* person" to generate protective perturbations, while adopting different prompts or subject identifiers for unauthorized personalized training and inference. Table 21 also presents the quantitative evaluation results after applying UDAP purification. It is clearly observed that despite the interference introduced by the purification process, our method still maintains optimal protection effectiveness across various prompts. Fig. 16 shows that our method retains effective protection against unauthorized personalization under unseen prompts even after UDAP purification. This further verifies the efficacy and robustness of the proposed method against diverse text-driven generations in complex adversarial environments.

Unfrozen text encoder. Table 22 also presents the robustness of different methods against UDAP purification when the text encoder is jointly fine-tuned during personalization on CelebA-HQ. This table shows that our method consistently surpasses all baselines across all evaluation metrics, maintaining highly robust protection performance.

Different perturbation magnitudes. Table 23 also evaluates the robustness of our method against UDAP purification under varying perturbation budgets. The results are consistent with those in the main text, validating the strong robustness of our method. Besides, our method exhibits enhanced protection robustness as the perturbation budget increases.

More baselines. We further compare the protection robustness with 4 additional baselines: **AdvDM** (Liang et al., 2023), **Mist** (Liang & Wu, 2023), **VaC** (Yang et al., 2025a), and **TAFAP** (Lee et al., 2026) on CelebA-HQ using four representative purification methods including Gaussian, Noisy Upscaling, GrIDPure, and UDAP. As shown in Table 26, all baseline methods suffer drastic defensive degradation after purification, while only our method can effectively resist purification. Notably, VaC achieves the highest protection efficacy before purification, yet suffers severe degradation of protective performance, with its metrics declining substantially after purification. Although our method attains slightly lower performance than VaC before purification, it consistently maintains the best defensive performance post-purification, demonstrating the robustness of our PuriShield.

More purification methods. To comprehensively evaluate the robustness of PuriShield, we further assess PuriShield against five additional purification methods, including **Resizing** (Xie et al., 2018), **FD** (Liu et al., 2019), **FreqPure** (Pei et al., 2025), **LightShed** (Foerster et al., 2025), and **CAT** (Peng et al., 2025). In particular, **LightShed** performs data-side depoisoning by learning to reconstruct protection perturbations and subtracting the reconstructed perturbations from protected images. In contrast, **CAT** performs model-side adaptation by inserting LoRA adapters into the VAE encoder and decoder and realigning the distorted latent representations of protected images. As detailed in Table 27, PuriShield consistently achieves the best protective performance across all evaluated metrics under these diverse attacks. Robustness against **LightShed** and **CAT** further demonstrates the adaptive robustness of PuriShield against both perturbation removal and latent-space model adaptation. This confirms that our structured protective perturbations are highly resilient, successfully surviving various purification attempts and continuously disrupting unauthorized customization.

Adaptive attacks. Following PID (Li et al., 2024) and VaC (Yang et al., 2025a), we evaluate our method under adaptive attacks that explicitly manipulate the latent variance during DreamBooth

Table 28: Robustness to various adaptive attacks on CelebA-HQ.

Methods	FID ↑	ISM ↓	IQS ↓	LIQE ↓
Clean	97.917	0.678	0.000	4.335
VaC (w/ Zero σ^2)	136.585	0.523	-5.632	3.826
VaC (w/ Clipped σ^2)	124.889	0.507	-4.185	3.652
VaC (w/ Fixed σ^2)	115.828	0.539	-4.229	3.831
Ours (w/ Zero σ^2)	216.332	**0.347**	-14.864	**2.026**
Ours (w/ Clipped σ^2)	220.379	0.351	-12.983	2.269
Ours (w/ Fixed σ^2)	**253.628**	0.349	**-15.650**	2.084
Ours (w/ PDAP)	252.541	0.436	-16.063	2.284
Ours (w/ PDAP & Zero σ^2)	246.841	0.463	-15.938	2.364
Ours (w/ PDAP & Clipped σ^2)	252.115	0.407	-16.906	2.486
Ours (w/ PDAP & Fixed σ^2)	235.762	0.470	-14.031	2.801

Table 29: Off-manifold deviations before and after GrIDPure purification.

Method	Before GrIDPure ↓	After GrIDPure ↓
ASPL	147.03	125.13
Anti-Diff	117.07	87.50
Anti-Pure	116.40	103.69
PhotoGuard	157.10	118.84
ACE	99.74	91.72
BlurGuard	105.27	89.48
Ours	**88.62**	**87.06**

fine-tuning. The motivation is that PID, VaC and our method all operate by disturbing the latent distribution of protected images, and thus an adaptive exploiter may attempt to neutralize the defense by correcting the abnormal variance σ^2. Specifically, we consider three attack strategies: **Zero** σ^2, which sets the latent standard deviation to zero; **Clipped** σ^2, which clips σ^2 to a relatively normal range; and **Fixed** σ^2, which fixes σ^2 to a constant value during the whole training process. These adaptive variants provide a stronger evaluation protocol by directly targeting the defense mechanism itself. As shown in Table 28, our method remains consistently more robust than the state-of-the-art anti-customization baseline, VaC, across all three adaptive attack settings. This is mainly because VaC achieves protection by enlarging the latent variance of protected images, which introduces abnormal distribution statistics and can thus be weakened by variance-adjustment-based adaptive attacks. In contrast, our method makes protected images to be far from original images yet close to dynamically updated perturbations in the latent space, rather than enlarging the latent variance, thereby significantly reducing the effectiveness of such adaptive attacks.

Besides, we further design a white-box Pattern-Density-Aware Purification (PDAP) attack, which exploits knowledge of the pattern-density parameter Ω to suppress the corresponding periodic components and their harmonics in the Fourier domain. Table 28 shows that our PuriShield exhibits strong robustness against PDAP. This is because the pattern density Ω only provides a structural prior, while the learned amplitude and mixing modulation distributes the perturbation across broader frequency components, making it difficult to remove without damaging natural image content.

Finally, we construct three types of joint adaptive attacks that combine latent-variance and PDAP attacks. As shown in Table 28, PuriShield remains robust even under these joint attacks, indicating that structured protective perturbations are more resilient to various adaptive attacks. Even under such joint attacks, our method outperformances VaC tested on the standalone latent-variance attack.

Quantitative evidence of protection efficacy and robustness. We quantitatively analyze the protection efficacy and robustness to verify the inherent advantages of structured protective perturbations. We first demonstrate that *structured protective perturbations adhere more closely to the natural data manifold than unstructured counterparts*. Specifically, we use VAE-based latent round-trip metric (VAE reconstruction error) to quantify the off-manifold deviation of protected samples: $d_{off} = \|\hat{z} - \mathcal{E}(\mathcal{D}(\hat{z}))\|_2$, where $\mathcal{D}$ and $\mathcal{E}$ denote the VAE decoder and encoder, respectively, and $\hat{z}$ is the latent representation of the protected images. The decode–encode round trip provides an approximate projection onto the natural data manifold. Accordingly, a smaller d_{off} indicates that the latent representations of protected image lies closer to the natural data manifold. As shown in Table 29, PuriShield achieves the smallest d_{off} both before and after GrIDPure purification (88.62

Table 30: Riemannian distances from clean representations before and after UDAP purification.

Method	Before UDAP ↑	After UDAP ↑
ASPL	138.62	87.12
Anti-Diff	117.73	90.40
Anti-Pure	121.44	87.32
PhotoGuard	118.64	89.45
ACE	171.98	87.38
BlurGuard	68.65	92.09
Ours	**192.47**	**115.48**

Table 31: Protection robustness against MasaCtrl customization after UDAP on CelebA-HQ.

Metrics	ASPL	Anti-Diff	Anti-Pure	PhotoGuard	ACE	BlurGuard	Ours
IQS ↓	-4.128	-4.076	-1.855	-5.043	1.686	2.831	**-7.442**
LIQE ↓	2.817	2.580	2.734	2.501	2.335	2.859	**2.087**

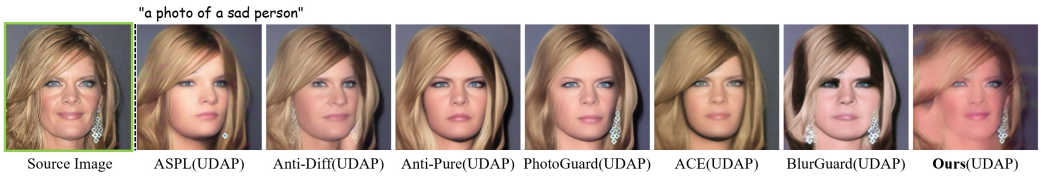


Figure 17: Visualized protection robustness against MasaCtrl (CelebA-HQ).

and 87.06, respectively), indicating that its protected representations remain closer to the natural data manifold under the approximate projection.

We further evaluate protection efficacy and robustness via the Riemannian feature distances measured between the clean and protected images, as well as between the clean and UDAP-purified images. As shown in Table 30, protected images generated by PuriShield yield the largest Riemannian feature distances from clean images, signifying the greatest separation from clean images along the data manifold. After UDAP purification, PuriShield still retains the largest Riemannian distance of 115.48, exceeding the strongest baseline by 23.39. Moreover, adversarial purification generally narrows the Riemannian distances between clean and protected images, pulling purified protected images back toward clean representations along the data manifold. BlurGuard is an exception, where the distance rises after UDAP, possibly due to purification-induced feature shifts. Overall, PuriShield maintains greater deviation from the clean feature manifold pre- and post-purification, implying superior protection efficacy and robustness. Combined with the VAE reconstruction error and Riemannian-distance results, we conclude that PuriShield achieves substantial manifold separation from clean images while keeping off-manifold deviation low. This dual benefit underpins the superior protection efficacy and strong robustness of structured protective perturbations against adversarial purification.

H More Results on Generalization

In this section, we evaluate the generalization ability of our method across more tasks and datasets.

H.1 Cross-Task Generalization

In the main text, we evaluate the protection robustness against unauthorized diffusion-based editing tasks based on the CelebA-HQ dataset and DiffEdit. In this section, we further evaluate the cross-task generalization across other editing task. Specifically, based on the CelebA-HQ dataset and **MasaCtrl framework** (Cao et al., 2023b), we compare robustness of different anti-customization methods against diffusion-based editing tasks after UDAP purification. As shown in Table 31, our method achieves the best performance on both metrics, yielding the lowest IQS and LIQE. From Fig. 17, we can see that our PuriShield remains effective after UDAP, where the editing instruction

Table 32: Protection efficacy of PuriShield against inpainting tasks on Helen before and after UDAP purification. ↑ and ↓ indicate improved protection efficacy compared to learning from clean images.

Dataset	Setting	LPIPS ↑	PSNR ↓	SSIM ↓	IQS ↓	LIQE ↓
Helen	w/o UDAP	0.601↑0.258	15.827↓6.448	0.394↓0.186	-5.336↓6.111	1.658↓0.985
	w/ UDAP	0.528↑0.185	18.425↓3.850	0.483↓0.097	0.496↓0.279	2.368↓0.275



Figure 18: Visualization of cross-domain inpainting generalization against **Stable Diffusion Inpainting** (**Helen**). The 2nd row displays the generation results from clean images, while the 3rd row shows the generation results from the protected images generated by our method.

Table 33: Protection efficacy of PuriShield against cross-domain style mimicry before and after purification. ↑ and ↓ indicate improved protection efficacy compared to learning from clean images.

Imitated Style	Setting	FID ↑	IQS ↓	LIQE ↓	LPIPS ↑	MS ↓
Oil Paintings	w/o UDAP	359.112↑126.431	-9.257↓4.632	2.277↓0.816	0.673	0.355
	w/ UDAP	277.770↑45.089	-6.020↓1.395	2.318↓0.775	0.582	0.414

"a photo of a sad person" fails to craft the intended edit. These results show that our method remains more effective than all baselines at disrupting editing-based customization even after purification.

H.2 Cross-Domain Generalization

Since all the compared baseline methods are based on iterative optimization and do not support cross-domain generation, we report only the results of our method.

Cross-domain inpainting task. We evaluate the protection robustness of PuriShield against unauthorized coss-domain inpainting task. Specifically, we craft protective perturbations for Helen dataset (Le et al., 2012) and Stable Diffusion Inpainting framework (Stable Diffusion v1.5 Community, 2024), by using the RPPG trained on VGGFace2 dataset. We evaluate protective efficacy before and after UDAP purification using the LPIPS, PSNR, SSIM, IQS, and LIQE metrics. As shown in Table 32, our method effectively disrupts the generative inpainting process. It significantly degrades both pixel-level and structural fidelity (indicated by decreased PSNR and SSIM), while introducing noticeable perceptual artifacts (indicated by increased LPIPS and decreased LIQE). This confirms the robust protection capability of PuriShield against localized image manipulation. Fig. 18 visualizes the cross-domain inpainting effects before purification. Notably, attempts to inpaint a localized masked region on a protected image cause PuriShield to produce prominent grid-like noise globally. This indicates that structured protective perturbations induce pervasive corruption, which destroys the visual integrity of the entire canvas and thereby fully thwarts localized editing attempts.

Cross-domain style mimicry. In the main text, we directly apply the RPPG trained on CelebA-HQ dataset to generate protective perturbations for WikiArt images. In this section, we craft protective

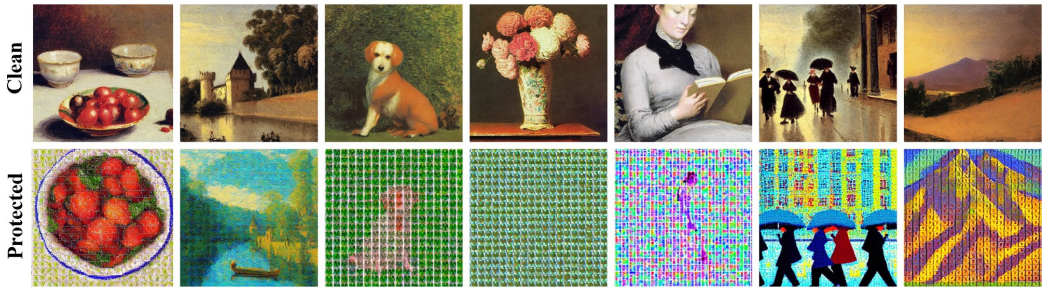


Figure 19: Visualization of cross-domain generalization against unauthorized style mimicry using **Textual Inversion** (**Paintings**). The top row displays the generation results from clean images, while the bottom row shows the generation results from the protected images generated by our method.

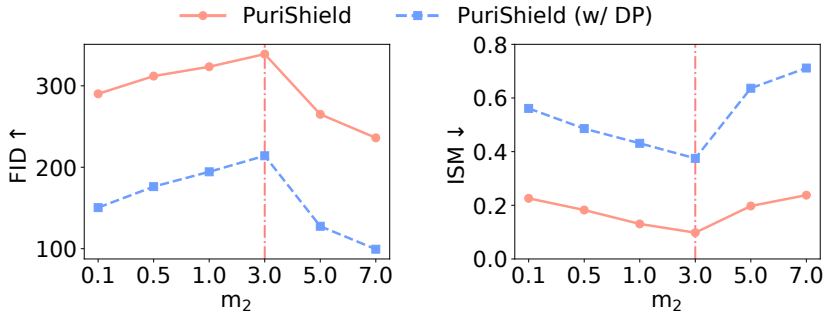


Figure 20: Ablation study on the different $m_1 : m_2$ ratios (With $m_1 = 1.0$ fixed, we set $m_2 \in \{0.1, 0.5, 1.0, 3.0, 5.0, 7.0\}$).

perturbations for Paintings dataset (Kaggle, 2017) by using the RPPG trained on VGGFace2 dataset. Specifically, we first add the perturbations generated from VGGFace2 dataset into oil paintings in the Paintings dataset. We then use Textual Inversion to learn the target painting style from the protected images and evaluate the protection performance before and after UDAP purification. Table 33 shows that RPPG retains effective protection capability even under this cross-domain setting. Fig. 19 visualizes the results of defending against cross-domain style mimicry before purification. Structured protective perturbations induce severe stylistic degradation in synthesized outputs, demonstrating that they can effectively prevent Textual Inversion from learning new style attributes.

I More Results on Ablation Studies

Following the settings in the main text, we conduct additional ablation studies on CelebA-HQ dataset using the FID and ISM metrics, with GrIDPure as the default DP method.

The impact of margins in loss $\mathcal{L}_{ldc}$. The latent distribution contrastive loss $\mathcal{L}_{ldc}$ defined in Eq. (2) of the main text involves two margin parameters m_1 and m_2, where m_1 constrains the latent mean distance between positive and negative pairs, and m_2 constrains the latent log-variance distance between positive and negative pairs. To explore the impact of these parameters, we design the following ablation scenarios: With $m_1 = 1.0$ fixed, we vary m_2 over $\{0.1, 0.5, 1.0, 3.0, 5.0, 7.0\}$ to investigate the influence of different $m_1 : m_2$ ratios. As shown in Fig. 20, a small m_2 (≤ 1.0) limits latent variance separation, reducing initial protection efficacy. Conversely, an excessively large m_2 (≥ 5.0) dominates the optimization, degrading structured patterns into unstructured noise and severely compromising robustness against DP. PuriShield achieves the optimal balance at $m_2 = 3.0$, (i.e., $m_1 : m_2 = 1 : 3$) indicating that a larger variance margin is required to effectively separate latent distributions without destroying structural priors.

The impact of hyperparameter λ in loss $\mathcal{L}_{total}$. According to Eq. (5) of the main text, λ weights the contribution of fine-tuning loss $\mathcal{L}_{ft}$ in total loss $\mathcal{L}_{total}$. Owing to the large difference in magni-

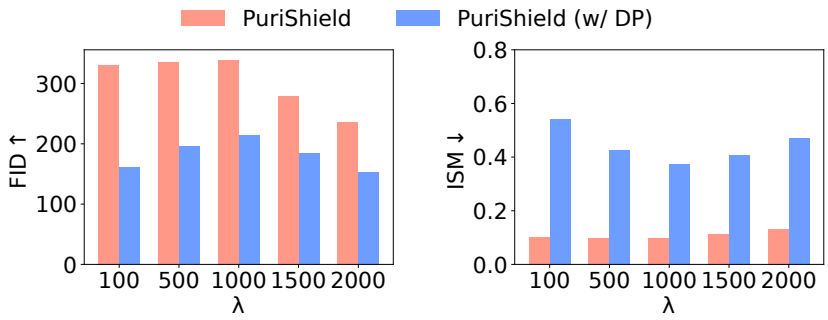


Figure 21: Ablation study on the weight λ of $\mathcal{L}_{ft}$.

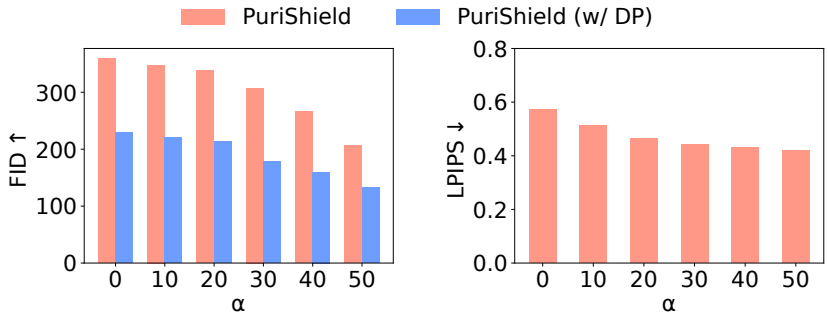


Figure 22: Ablation study on the weight α of $\mathcal{L}_p$.

tude between $\mathcal{L}_{ft}$ and $\mathcal{L}_{ldc}$, a larger weight is assigned to $\mathcal{L}_{ft}$ to balance their gradient contributions and prevent $\mathcal{L}_{ldc}$ from dominating the optimization. As shown in Fig. 21, a small λ fails to sufficiently adapt perturbations to the updated data manifold, weakening robustness against purification. Conversely, an excessively large λ overwhelms $\mathcal{L}_{ldc}$, reducing the protection efficacy. PuriShield achieves the optimal balance when $\lambda = 1000$.

The impact of hyperparameter α and feature layer l in loss $\mathcal{L}_{total}$. We use FID to assess protection efficacy, while using LPIPS to evaluate imperceptibility of protected images. The weight α controls the contribution of the feature perceptual loss $\mathcal{L}_p$. A small α provides insufficient perceptual constraint, while an excessively large α may weaken protection efficacy. The feature layer l determines the representation used in $\mathcal{L}_p$, with different layers capturing visual information at different levels. As shown in Fig. 22, when $\alpha = 0$, the perceptual loss $\mathcal{L}_p$ is removed, resulting in stronger protection but more perceptible perturbations. Increasing α improves imperceptibility but weakens protection efficacy. Beyond $\alpha = 20$, LPIPS decreases only marginally while FID drops substantially. Therefore, we set $\alpha = 20$ to balance imperceptibility and protection efficacy.

For the feature layer l, we focus on the high-resolution Up Blocks of the diffusion U-Net, which better capture local textures and edge details. We denote the four Up Blocks as U_1–U_4. As shown in Fig. 23, U_4 achieves the strongest protection with only a slight increase in LPIPS. We therefore select U_4 for feature extraction to balance imperceptibility and protection effectiveness.

The effect of parameter P in DBMO. To enhance the robustness of protective perturbation against DP, the DBMO module partitions the perturbation into an $P \times P$ grid, and randomly discards partial blocks with a predefined probability p. Following DIM (Xie et al., 2019), we set the drop probability p to 0.5, while varying the grid partition parameter $P \in \{2, 4, 6, 8, 16\}$ to evaluate its effect in the DBMO module. As shown in Fig. 24, an excessively small P (e.g., $P = 2$) creates massive mask blocks. Dropping these blocks deprives the generator of global contextual information, severely degrading the initial protection efficacy. Conversely, a large P (e.g., $P \geq 8$) results in tiny blocks that behave similarly to random pixel-level dropout. This fails to adequately simulate the contiguous spatial disruption caused by DP, leading to weaker robustness after purification. PuriShield achieves the optimal balance of protection and robustness when $P = 4$.

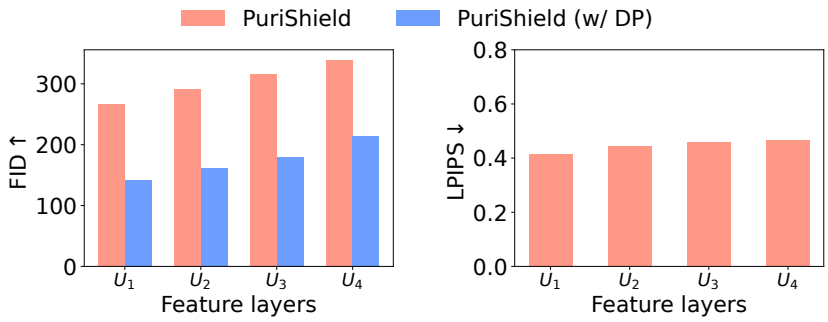


Figure 23: Ablation study on the different feature layers for $\mathcal{L}_p$.

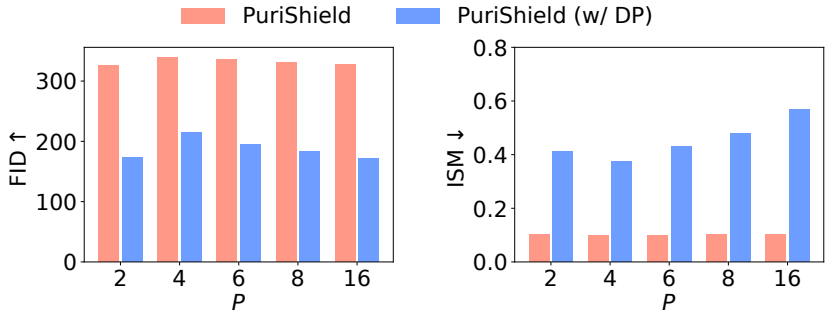


Figure 24: Ablation study on parameter P in DBMO.

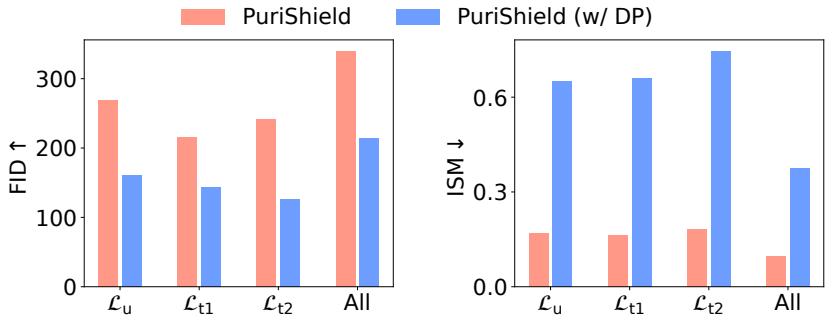


Figure 25: Ablation study on loss design.

The impact of loss design. To evaluate the impact of latent distribution contrastive loss $\mathcal{L}_{ldc}$, we design the following ablation scenarios: ① $\mathcal{L}_u$ replaces loss $\mathcal{L}_{ldc}$ with the untargeted loss of AdvDM (Liang et al., 2023): $\delta = \arg\max_{||\delta||_\infty \leq \epsilon} \mathcal{L}_{ldm}(\theta, c, \hat{z}_0)$, where $\hat{z}_0$ is the latent representation of the protected image $\hat{x} = x + \delta$. ② $\mathcal{L}_{t1}$ replaces loss $\mathcal{L}_{ldc}$ with the targeted loss of PhotoGuard (Salman et al., 2023): $\delta = \arg\min_{\|\delta\|_\infty \leq \epsilon} \|\hat{z} - z^{tgt}\|_2^2$, where $\hat{z}$ is latent representation of protected image $\hat{x}$. ③ $\mathcal{L}_{t2}$ replaces loss $\mathcal{L}_{ldc}$ with the targeted loss of Mist (Liang & Wu, 2023), which follows the same optimization objective as $\mathcal{L}_{t1}$ but adopts the Mist target image. As shown in Fig. 25, replacing $\mathcal{L}_{ldc}$ with the untargeted loss $\mathcal{L}_u$ or the targeted losses $\mathcal{L}_{t1}$ and $\mathcal{L}_{t2}$ leads to significant performance degradation, highlighting the necessity of employing contrastive loss. Here, **All** denotes that PuriShield is trained using the full total loss $\mathcal{L}_{total}$.

The impact of target design. We further investigate the impact of different target designs on protection performance. Specifically, for loss L_{ldc}, we still take the dynamically updated protected image as the anchor and the fixed original image as the negative sample, with three settings: ① A fixed gray image adopted by PhotoGuard (Salman et al., 2023) as the positive sample; ② The

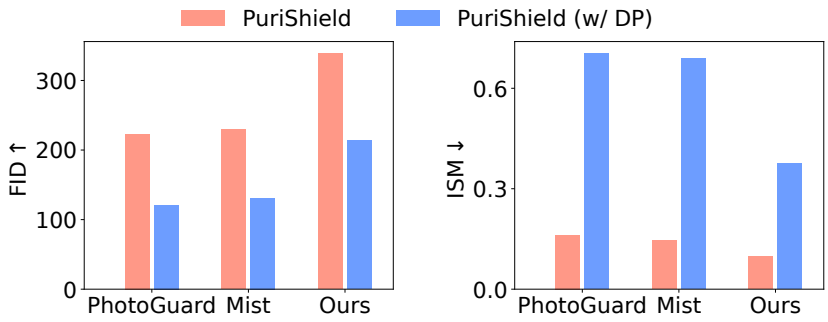


Figure 26: Ablation study on fixed and dynamic targets.

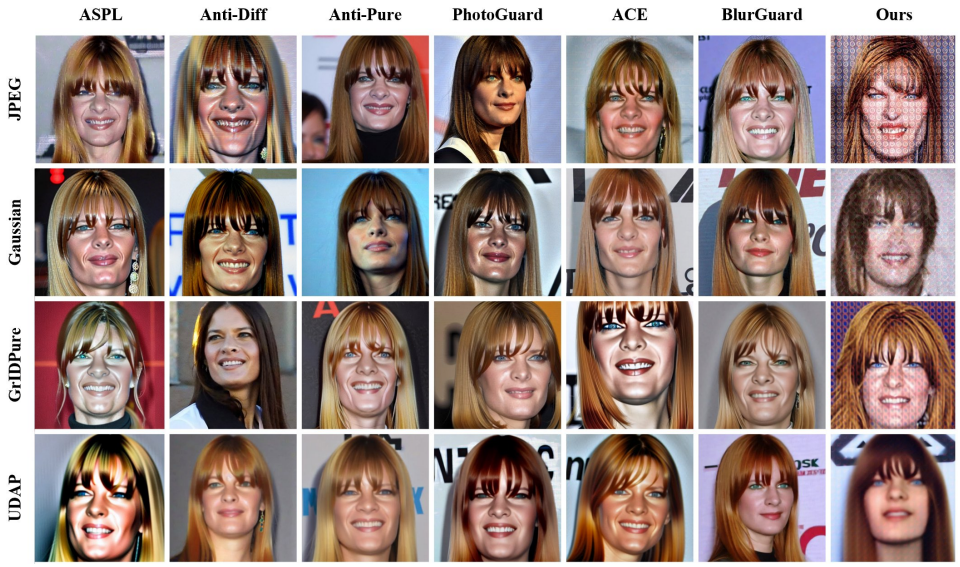


Figure 27: Output images generated by DreamBooth after four purification methods (CelebA-HQ).

fixed target image used in Mist (Liang & Wu, 2023) as the positive sample; ③ Our dynamically updated perturbation as the positive sample. As shown in Fig. 26, the dynamic target outperforms the fixed-target variants. This demonstrates that dynamically optimizing the target during training improves the effectiveness and robustness of protective perturbations.

J More Visualization Results

Protection robustness on CelebA-HQ. Fig. 27 visualizes the generated images from DreamBooth on CelebA-HQ using protected images that have undergone various purification methods. As shown in Fig. 27, PuriShield yields maximally perturbed subject identifiers relative to all baseline defenses, which verifies its strong protective performance.

Protection robustness on VGGFace2. Fig. 28 provides qualitative results for DreamBooth fine-tuning on the VGGFACE dataset, following various purification methods. Despite the aggressive noise removal by purification methods, our method demonstrates superior resilience. It persistently forces the fine-tuned model to generate chaotic and perceptually meaningless content, significantly outperforming other baselines across different facial datasets.

Unfrozen text encoder. Fig. 29 presents qualitative results of DreamBooth fine-tuning with an unfrozen text encoder on the CelebA-HQ dataset following various purification methods. Our method achieves substantial improvements over other baselines, verifying its excellent practicality.

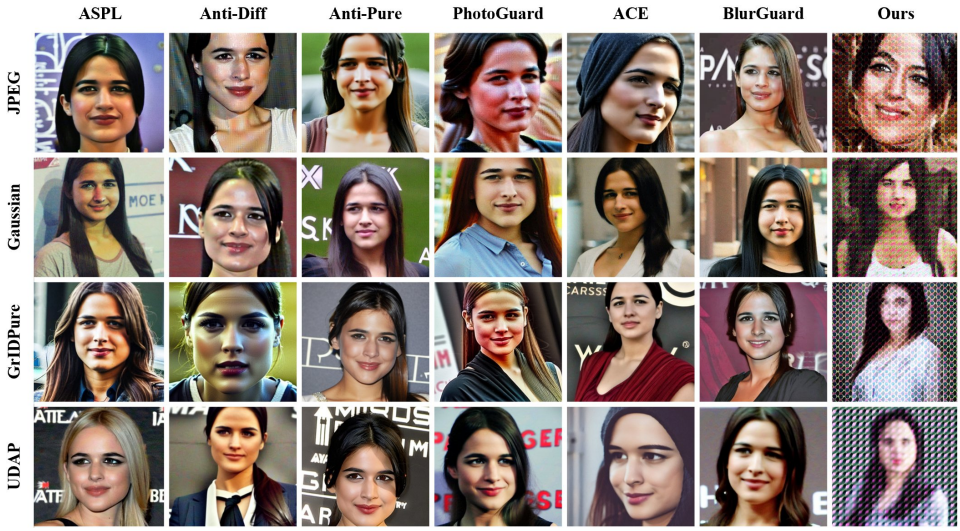


Figure 28: Output images generated by DreamBooth after four purification methods (VGGFace2).

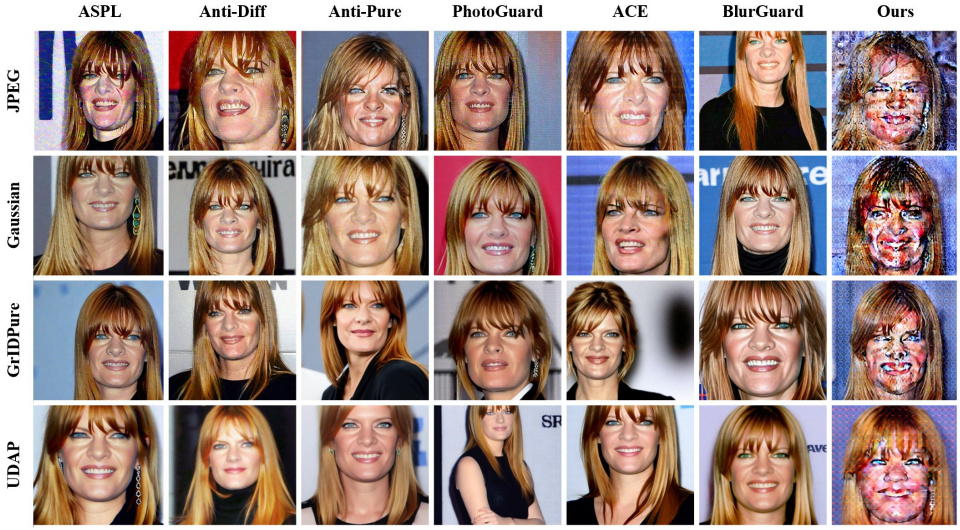


Figure 29: Output images generated by DreamBooth with an unfrozen text encoder after four purification methods (CelebA-HQ).

Custom Diffusion outputs on CelebA-HQ under varying perturbation budgets. Fig. 30 displays Custom Diffusion personalized outputs on CelebA-HQ dataset, produced from UDAP-purified protected images under varying perturbation budgets. As the budget increases, the images generated from our protected images exhibit stronger visual distortions and identity degradation, indicating the robustness of our method across different personalization methods and perturbation strengths.

Imperceptibility on VGGFace2 under varying perturbation budgets. Fig. 31 presents protected images by our method under varying perturbation budgets. As expected, larger ϵ leads to more severe distortions. At $\epsilon = 8/255$, our protected images remain visually natural with satisfactory perceptual quality, achieving a desirable trade-off between protection efficacy and visual imperceptibility.

Protection robustness under different perturbation budgets. Figs. 32 and 33 show qualitative results on CelebA-HQ with perturbation budgets of $\epsilon = 4/255$ and $\epsilon = 12/255$, respectively. After purification, most baselines produce recognizable facial identities. In contrast, PuriShield consistently causes strong identity distortion and structural artifacts after purification. This effect

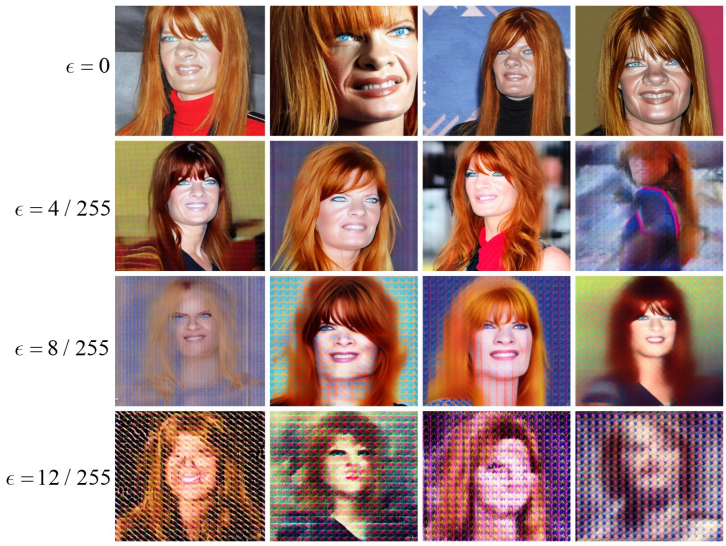


Figure 30: Custom Diffusion outputs after UDAP across perturbation budgets (CelebA-HQ).



Figure 31: Protected images generated by our method across perturbation budgets (VGGFace2).

is already clear at $\epsilon = 4/255$ and becomes stronger at $\epsilon = 12/255$. These results demonstrate the strong robustness of PuriShield across different perturbation budgets and purification methods.

Browser extension usage. Fig. 34 demonstrates the practical utility of the PuriShield browser extension. Following installation, users can protect images directly within the browser. All inference runs locally on the user's device, and original images are never uploaded to external servers. This enables a simple, privacy-preserving deployment of PuriShield for personal image protection.

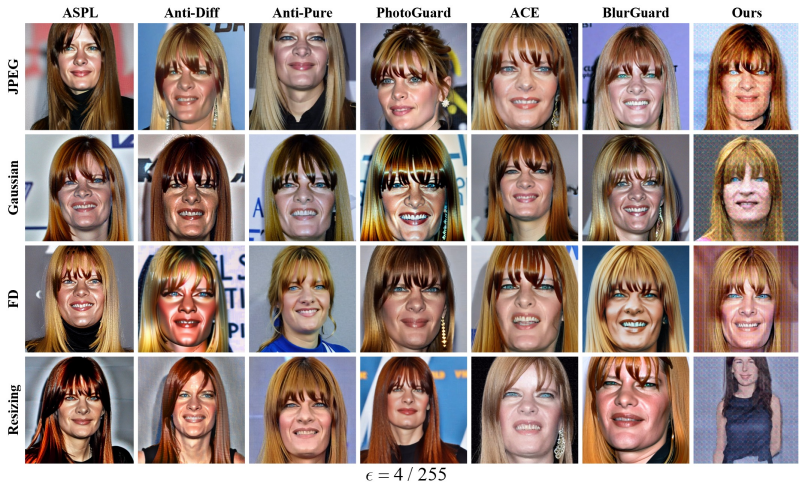


Figure 32: DreamBooth outputs after four purification methods. ($\epsilon = 4/255$ on CelebA-HQ).

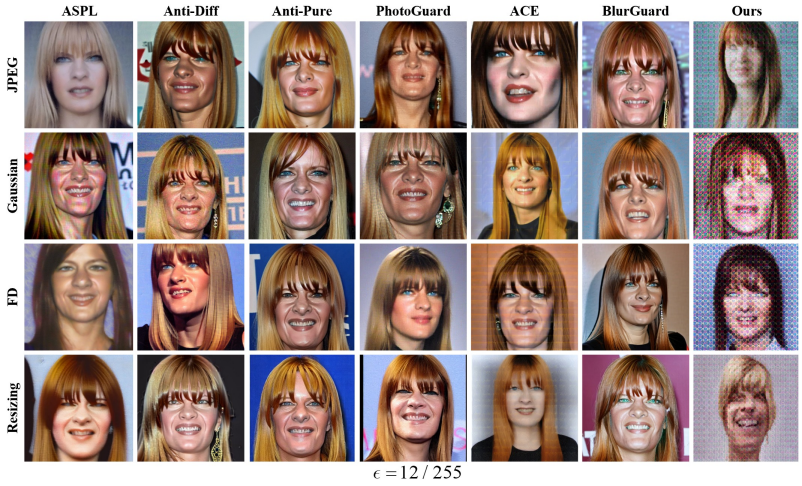


Figure 33: DreamBooth outputs after four purification methods. ($\epsilon = 12/255$ on CelebA-HQ).

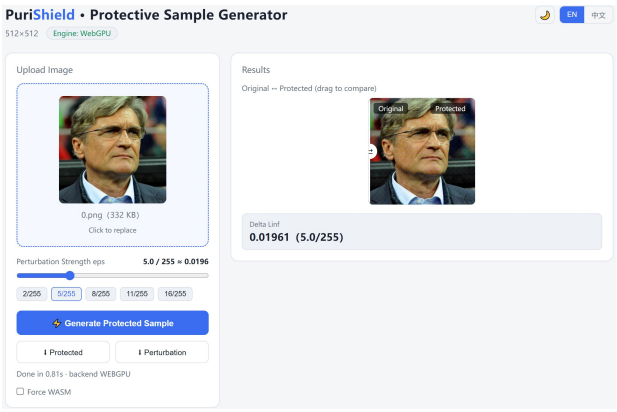


Figure 34: Usage of the PuriShield browser extension for local image protection.