GUARDFLOW: IMAGE IMMUNIZATION VIA TRAJECTORY DISRUPTION AGAINST FLOW-BASED EDITING

Anonymous authors
Paper under double-blind review

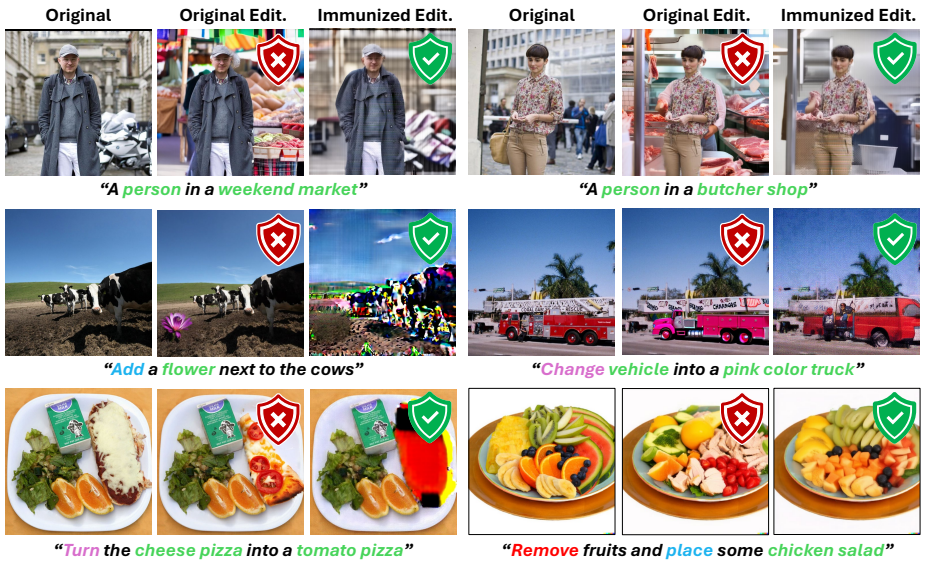


Figure 1: **Immunization results of GuardFlow**. Both inpainting (top row) and instruction-based editing (bottom two rows) demonstrate the applicability of GuardFlow across diverse editing scenarios. Edits on immunized images show noticeably weaker semantic alignment with the prompts than those on the original images.

ABSTRACT

Text-guided image editing enables powerful manipulation of visual content but also introduces serious security risks through unauthorized or malicious edits. Thus, image immunization aims to protect images with imperceptible perturbations that induce edit failure. However, existing methods are primarily designed for GAN or diffusion models and do not generalize to flow-based generative models, despite their growing adoption. Hence, we propose **GuardFlow**, the first immunization framework that *guards* images from *flow-based editing models*. The key insight is to exploit the trajectory-driven nature of flow models, where small perturbations accumulate and alter the generation path. We further derive an image-specific theoretical perturbation bound that enables effective disruption while preserving perceptual quality. Experiments on major flow-based editing frameworks show that GuardFlow effectively impedes text-guided editing while maintaining high visual fidelity, outperforming existing baselines in both effectiveness and robustness. *The code will be released upon publication.*

1 INTRODUCTION

Recent advances in generative models (Karras et al., 2019; Ho et al., 2020) have significantly improved the quality and controllability of text-guided image editing. While diffusion-based models such as Stable Diffusion (Rombach et al., 2022; Podell et al., 2024) and DALL-E (Ramesh et al., 2022) have been widely adopted, recent progress has increasingly shifted toward flow-based generative models (Liu et al., 2023; Lipman et al., 2023), which offer efficient generation through ordinary differential equations (ODEs) defined by learned vector fields. These modern frameworks enable realistic and semantically precise modifications based on natural language prompts (Esser et al., 2024;

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.

Labs, 2024), and they are rapidly being integrated into real-world applications such as graphic design (Adobe Inc., 2025) and advertisement (Wu et al., 2024). However, such recent progress also introduces serious security concerns (He et al., 2024), as such models can be abused to manipulate images in misleading or harmful ways, including the alteration of sensitive content (Mackenzie & Choi, 2024; Davies & McDermott, 2022; Cole, 2018) or the generation of deceptive visual information (Wojewidka, 2020; Delfino, 2022).

To address these risks, *image immunization* aims to protect images from unauthorized editing by introducing imperceptible perturbations designed to disrupt edit frameworks while preserving the visual fidelity of the original image (Salman et al., 2023). As immunization strategies typically tailor perturbations to ruin specific components of the editing pipeline, they do not naturally generalize across different generative editing frameworks. In particular, as diffusion-based models edit samples through an iterative denoising process, perturbations directly interfere with the predicted update at each generation step (Choi et al., 2025). In contrast, as flow-based models evolve samples along a continuous trajectory governed by a vector field, the model prediction corresponds to a local velocity rather than the next state itself. As a result, perturbations designed for diffusion-based editing may induce only limited local deviations in flow trajectories, making them less effective at protecting images from flow-based editing. Consequently, existing methods provide limited protection against flow-based editors, leaving images vulnerable to unauthorized manipulation.

Motivated by such limitations, we introduce **GuardFlow**, the first image-immunization framework to **guard** images from edits for **flow**-based generative models, to the best of our knowledge. The key insight is to train a global immunizer that exploits the trajectory-driven nature of flow models by generating noise that directly perturbs the underlying dynamics of the generation process. For this, we propose a trajectory disruption objective that induces deviation from the original flow, together with a terminal regression objective that discourages the intended edits at the output level. To further balance edit disruption and perceptual quality, we derive a theoretically grounded, image-specific perturbation bound. Specifically, it provides a lower bound on the perturbation magnitude required to alter the editing trajectory based on the local sensitivity of the trained vector field to input images. During training, this bound serves as a regularizer, enabling the immunizer to generate image-dependent perturbations that remain effective and visually imperceptible.

To summarize, GuardFlow is an integrated framework with the following **contributions**: **1)** We propose a trajectory disruption objective that interferes with text-conditioned flow-based image editing by inducing discrepancies in the vector field that push the trajectory away from the base trajectory. **2)** We introduce a terminal regression loss that encourages the edited output to resemble the original input image by leveraging terminal estimation. **3)** We derive a theoretical perturbation lower bound required for trajectory disruption, and leverage it to design an effective optimization objective. We present the first image immunization framework for flow-based generative editing models, providing a foundation for protecting images against unauthorized editing in emerging flow-matching systems. Extensive experiments based on major flow-based editing frameworks on variations of editing tasks demonstrate the efficacy of our method as GuardFlow outperforms all baseline methods.

2 Related Work

Generative models for image editing aim to modify a source image according to desired attributes while preserving its underlying structure and content. Early approaches focused on task-specific transformations such as style transfer (Gatys et al., 2016; 2015) and domain translation (Huang et al., 2018; Isola et al., 2017; Zhu et al., 2017), followed by latent-space manipulation (Chai et al., 2021; Abdal et al., 2019; Richardson et al., 2021). Subsequent methods (Kwon & Ye, 2022; Patashnik et al., 2021; Zheng et al., 2022) incorporated vision-language models (e.g., CLIP (Radford et al., 2021)) to enable text-guided semantic editing. With the remarkable success of diffusion models (Dhariwal & Nichol, 2021; Zhu et al., 2023) in high-fidelity image synthesis, recent approaches have increasingly adopted them as the foundation for image editing. In particular, diffusion-based methods (Meng et al., 2022; Brooks et al., 2023) edit images by first corrupting the input image then using a diffusion model to reconstruct it through iterative denoising conditioned on text prompts, enabling precise and controllable modifications. Recently, flow-based generation (Esser et al., 2024; Labs, 2024) has emerged as an efficient and deterministic alternative, attracting growing interest for image editing applications. Existing flow-based editing methods (Wang et al., 2025; Kim et al., 2025; Rout et al., 2025) adopt a corruption–restoration paradigm to balance editability and source consistency, achieving effective and high-quality text-guided image editing.

Image immunization introduces imperceptible perturbations to protect images from unauthorized or malicious edits by disrupting generative editing models. Early works (Yeh et al., 2020; Ruiz et al., 2020; Aneja et al., 2022) focused on GAN-based image editing, where perturbations interfere with latent manipulation or image translation processes. With the success of diffusion models, recent approaches have focused on disrupting the denoising process for text-guided editing. Existing methods manipulate latent representations (Chen et al., 2024; Salman et al., 2023), feature spaces (Wang et al., 2024), or noise prediction (Choi et al., 2025) dynamics to degrade editing results. Some (Chen et al., 2024; Choi et al., 2025; Salman et al., 2023; Xue et al., 2024) compute image-specific perturbations through direct optimization, while others, such as DiffVax (Ozden et al., 2026), learn a parametric model to generate perturbations for improved scalability. However, these methods are primarily designed for diffusion models and rely on perturbing the denoising process. Immunization for flow-based generative models remains unexplored, and our work fills this gap.

3 Preliminary

3.1 Flow-based models and Rectified flow

Flow-based generative models (Lipman et al., 2023) learn a continuous transformation that maps a simple source distribution p (e.g., a standard Gaussian) to a target data distribution q. Given $x_0 \sim p$, a continuous trajectory x_t, where $t \in [0, 1]$ denotes a continuous time variable, is defined such that x_1 follows the target distribution q. A time-dependent mapping $\psi_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ defines this trajectory, and the dynamics of x_t are governed by a time-dependent vector field v, which specifies the instantaneous rate of change along the trajectory:

$$x_t = \psi_t(x_0), \quad \frac{dx_t}{dt} = v(x_t, t). \tag{1}$$

Rectified flow (Liu et al., 2023) is a special case of flow models where, given samples $x_0 \sim p$ and $x_1 \sim q$, the trajectory follows a linear interpolation:

$$\psi_t(x_0) = (1 - t)x_0 + tx_1, \; v(x_t, t) = x_1 - x_0 \tag{2}$$

This formulation enables the use of a simple and efficient trajectory that has been widely adopted for high-quality image generation (Esser et al., 2024; Liu et al., 2023; Labs, 2024). While the proposed method is applicable to general flow-based generative models, we use rectified flow as a representative model throughout this work.

In training, samples are constructed along the trajectory Eq. (2) by sampling $t \sim \mathcal{U}(0, 1)$. A learned vector field v_θ is trained to regress target $x_1 - x_0$ using the flow matching loss (Lipman et al., 2023):

$$\mathcal{L}_{FM} = \mathbb{E}_{t,x_0,x_1} \| v_\theta(x_t, t) - (x_1 - x_0) \|_2^2. \tag{3}$$

After training, v_θ is used to solve the ODE to generate new samples from the target distribution.

For text-to-image generation, the vector field is conditioned on a text prompt c, yielding $v_\theta(x_t, t|c)$. In practice, classifier-free guidance (CFG) (Ho & Salimans, 2021) is employed to train a single v_θ by randomly dropping the condition during training, enabling both conditional and unconditional predictions. At inference, CFG constructs a guided vector field by combining conditional and unconditional predictions to enhance alignment with c:

$$v_\theta^{\text{cfg}}(x_t, t \mid c) = \underbrace{v_\theta(x_t, t \mid \emptyset)}_{\text{base}} + w \underbrace{\big(v_\theta(x_t, t \mid c) - v_\theta(x_t, t \mid \emptyset)\big)}_{\text{prompt correction}}, \tag{4}$$

where $w \geq 1$ is the guidance scale. Here, the first term represents the base dynamics shared across all conditions, while the second term introduces a prompt-specific correction. The generation process is performed by solving the ODE defined by the classifier-free guided vector field:

$$\frac{dx_{t|c}}{dt} = v_\theta^{\text{cfg}}(x_t, t \mid c), \quad x_{t|c} = \psi_{t|c}(x_0), \tag{5}$$

where $\psi_{t|c}$ denotes the flow induced by v_θ^{cfg}.

3.2 Text-prompted Image Editing using Pre-trained Flow Models

Given an input image x^{src} and a target prompt c, most flow-based text-prompted image editing consists of the following stages: *i)* sampling of x_0 from source distribution p, *ii)* partial inversion along trajectory x_t^{src} defined by x_0 and x^{src}, and *iii)* solving the ODE using text-conditioned v_θ^{cfg}.

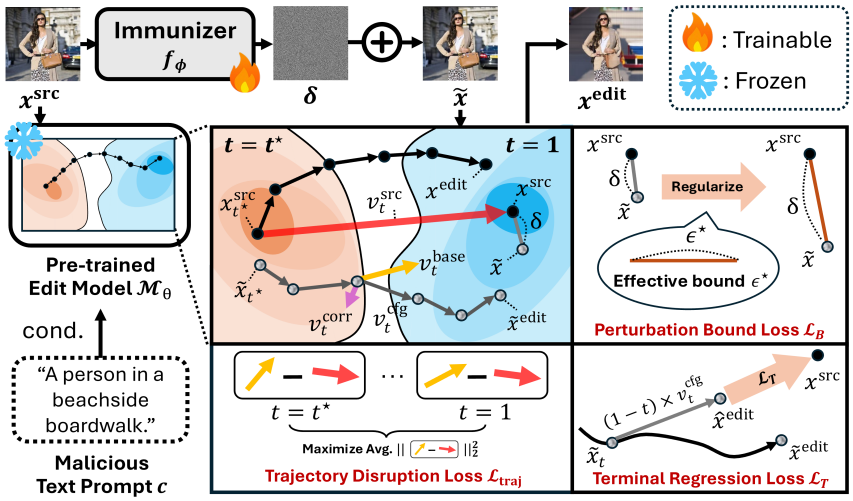


Figure 2: **Overall framework of GuardFlow**. Given an input image x^{src}, the immunizer f_ϕ generates an imperceptible perturbation δ, producing an immunized image $\tilde{x} = x^{\text{src}} + \delta$. The immunizer is trained using the proposed objective $\mathcal{L}_{\text{imm}}$, which disrupts the editing trajectory of a pre-trained flow-based editing model $\mathcal{M}_\theta$.

During partial inversion, x^{src} is mapped to an intermediate point along the trajectory x_t^{src} induced by the rectified flow towards p. Here, we assume p to be a standard Gaussian, following conventions in flow-based image generative models (Liu et al., 2023; Lipman et al., 2023). This process can be formulated as a conditional path $p_t(x \mid x^{\text{src}})$ that bridges the source distribution p and the given source image x^{src}. Specifically, a sample along this path is obtained as:

$$x_t^{\text{src}} = (1-t)x_0 + tx^{\text{src}}, \quad x_0 \sim p = \mathcal{N}(0, I), \tag{6}$$

where I denotes the identity matrix. By selecting an intermediate timestep $t^* \in (0, 1)$, we obtain a partially-inverted state $x_{t^*}^{\text{src}}$, where t^* determines the inversion strength which controls the degree of source preservation and editability. Subsequently, an edited image x^{edit} for c is generated by evolving $x_{t^*}^{\text{src}}$ to terminal state $x_{t=1|c}^{\text{src}}$ by solving the ODE using v_θ^{cfg}.

4 Method

Problem Definition. Given a pre-trained flow-based model $\mathcal{M}_\theta$, our goal is to learn an immunizer f_ϕ that produces an imperceptible perturbation $\delta = f_\phi(x^{\text{src}})$ for an arbitrary input image x^{src}, which yields a perturbed image $\tilde{x} = x^{\text{src}} + \delta$ that causes $\mathcal{M}_\theta$ to fail at text-guided image editing for arbitrary target prompt c. We define editing failure as reduced semantic alignment with c and deviation from the clean-image edit, measured by text–image and image similarity, respectively. Here, θ denotes the parameters of the vector field v_θ in $\mathcal{M}_\theta$, and ϕ denotes the parameters of f_ϕ. The perturbation δ is constrained to preserve the visual fidelity of the original input. Specifically, we formulate this as the following constrained optimization problem:

$$\phi^* = \arg\min_\phi \mathcal{L}_{\text{imm}}\big(x^{\text{src}} + f_\phi(x^{\text{src}}); \mathcal{M}_\theta\big) \quad \text{s.t.} \quad \|f_\phi(x^{\text{src}})\|_\infty \le \beta, \tag{7}$$

where $\mathcal{L}_{\text{imm}}$ is an immunization objective designed to induce editing failure, and β is a given perceptibility constraint that controls the perturbation magnitude while ensuring imperceptibility of δ. We optimize f_ϕ with respect to a fixed $\mathcal{M}_\theta$, reflecting a practical immunization scenario in which the editing model remains frozen throughout training.

4.1 Trajectory Disruption with Jacobian-Free Optimization

Unlike diffusion-based models, flow-based generative models evolve images by solving an ODE that defines a continuous trajectory governed by a time-dependent vector field. Consequently, perturbations to the predicted velocity can propagate along the trajectory and accumulate over the generation process. Motivated by this property, we propose a trajectory disruption objective that directly perturbs the base dynamics of the flow, causing the editing trajectory to deviate from its original path.

Let x_t^{src} denote the rectified flow trajectory with endpoint at $t = 1$ given by x^{src}, as defined in Eq. (6). We define the local trajectory derivative at time t as:

$$v_t^{\text{src}} := \frac{dx_t^{\text{src}}}{dt} = x^{\text{src}} - x_0, \tag{8}$$

which corresponds to the exact local derivative of the trajectory under the rectified flow. Note that as rectified flow assumes all trajectories to be linear, v_t^{src} remains constant along this path. Given an immunized input $\tilde{x}$, the corresponding trajectory is defined as:

$$\tilde{x}_t = (1-t)x_0 + t(x^{\text{src}} + \delta) = x_t^{\text{src}} + t\delta. \tag{9}$$

This induces a simple shift of the trajectory while preserving its linear structure. However, a perturbation that merely satisfies $\|\delta\|_\infty \leq \beta$ does not necessarily induce a sufficient change in the predicted vector field. In particular, $v_\theta(\tilde{x}_t, t)$ may remain closely aligned with the original trajectory derivative v_t^{src}, resulting in only limited output deviation. We therefore optimize δ to explicitly drive the predicted vector field away from the original trajectory. To induce meaningful disruption, we define a loss that drives the model-predicted vector field away from the original trajectory derivative:

$$\mathcal{L}_{\text{traj}} = -\mathbb{E}_{t,x_0}\left[\|v_\theta(\tilde{x}_t, t|\emptyset) - v_t^{\text{src}}\|_2^2\right], \quad t \sim \mathcal{U}(t^*, 1). \tag{10}$$

Notably, we employ the unconditional vector field prediction $v_\theta(\cdot \mid \emptyset)$ to perturb conditional generation as a whole. As the conditional editing flow is obtained by reinforcing this base vector field with a prompt-dependent term as in Eq. (4), disrupting the base term leads to trajectory deviations that persist regardless of the target prompt c.

The exact gradient of $\mathcal{L}_{\text{traj}}$ requires computing the Jacobian of the frozen flow network, which becomes computationally expensive for large-scale models (e.g., transformers (Dosovitskiy et al., 2021)). Therefore, inspired by the Score Distillation Sampling (SDS) and its rectified-flow extension (RFDS) (Poole et al., 2023; Yang et al., 2025), we employ a Jacobian-free approximation of $\nabla_\delta \mathcal{L}_{\text{traj}}$:

$$\nabla_\delta \mathcal{L}_{\text{traj}} = -2\,\mathbb{E}_{t,x_0}\left[t \cdot J_{v_\theta}(\tilde{x}_t, t)^\top \left(v_\theta(\tilde{x}_t, t|\emptyset) - v_t^{\text{src}}\right)\right] \approx -2\,\mathbb{E}_{t,x_0}\left[t \cdot \left(v_\theta(\tilde{x}_t, t|\emptyset) - v_t^{\text{src}}\right)\right] \tag{11}$$

where the Jacobian of the vector field $J_{v_\theta}(\tilde{x}_t, t) := \frac{\partial v_\theta(\tilde{x}_t, t|\emptyset)}{\partial \tilde{x}_t}$ is replaced by the identity I. Further analysis shows that the Jacobian-free gradient is positively aligned with the exact gradient (Sec. B).

4.2 Terminal Regression via x^{edit} Estimation

While trajectory disruption perturbs the underlying flow dynamics, preventing successful text-guided editing ultimately depends on the resulting output. Directly defining an objective on the edited image x^{edit}, however, requires backpropagation through the ODE-defined flow trajectory, which is prohibitively expensive for flow-based models. To avoid this cost, we define a terminal objective that approximates the final edited output from an intermediate state.

Specifically, given an intermediate sample x_t along the reverse flow, we approximate the terminal output x^{edit} using Tweedie's denoising estimate (Kim & Ye, 2021):

$$\hat{x}^{\text{edit}}(\tilde{x}_t) = \tilde{x}_t + (1-t)v_\theta^{\text{cfg}}(\tilde{x}_t, t \mid c). \tag{12}$$

$\hat{x}^{\text{edit}}(\tilde{x}_t)$ extrapolates $\tilde{x}_t$ along the learned velocity field to approximate the final edited output, as empirically validated in Sec. F.2. We therefore define the terminal objective to constrain the predicted edit outcome toward the source image:

$$\mathcal{L}_T = \mathbb{E}_t\left[\|x^{\text{src}} - \hat{x}^{\text{edit}}(\tilde{x}_t)\|_2^2\right], \quad t \sim \mathcal{U}(t^*, 1). \tag{13}$$

By constraining the predicted edit toward the source image, $\mathcal{L}_T$ also implicitly regularizes the perturbation magnitude, suppressing excessive perturbations as analyzed in Sec. C.

4.3 Minimum Perturbation Bound for Trajectory Disruption

In image immunization, the perturbation bound directly determines the range of input modifications and the degree of possible edit disruption. Hence, successful edit disruption requires a perturbation of adequate magnitude while satisfying imperceptibility constraints. Existing immunization methods set this bound as a manually selected global hyperparameter (Chen et al., 2024; Choi et al., 2025; Xue et al., 2024; Salman et al., 2023). However, as the effective perturbation scale varies across images, a single globally fixed budget cannot fully capture image-specific requirements. To address this limitation, we derive a theoretical perturbation bound w.r.t. the local vector-field sensitivity.

Given the perturbed state $\tilde{x}_t$, we quantify the effect of the perturbation as directional misalignment between the original and perturbed vector fields. We measure this deviation using cosine similarity:

$$\mathcal{D}(t) = \cos\left(v_\theta(\tilde{x}_t, t),\, v_\theta(x_t, t)\right), \tag{14}$$

where lower values of $\mathcal{D}(t)$ indicate stronger deviation. Here v_θ denotes the unconditional prediction $v_\theta(\cdot \mid \emptyset)$ so that the resulting bound depends only on x^{src}. Based on this measure of vector-field deviation, we derive the following perturbation bound:

Proposition 1. *Let $v_\theta(x_t, t)$ be a vector field differentiable with respect to x, with Jacobian $J_{v_\theta}(x_t, t) := \frac{\partial v_\theta(x_t,t)}{\partial x}$. Given $\tilde{x}_t = x_t + t\delta$ as defined in Eq. (9), under a local first-order Taylor approximation of the vector field around x_t, we have $v_\theta(\tilde{x}_t, t) \approx v_\theta(x_t, t) + J_{v_\theta}(x_t, t) \cdot t\delta$. For $\tau \in (-1, 1)$, if*

$$\|\delta\| < \frac{1}{t} \cdot \sqrt{1 - \max(\tau, 0)^2} \cdot \frac{\|v_\theta(x_t, t)\|}{\|J_{v_\theta}(x_t, t)\|},$$

then $\mathcal{D}(t) > \tau$ for every direction of δ. Equivalently, satisfying $\mathcal{D}(t) \leq \tau$ requires $\|\delta\|$ to be at least this value.

For $\tau < 0$, the bound saturates at its value for $\tau = 0$, as any negative threshold requires the same minimum velocity-change magnitude, while differing only in its directional requirement (Sec. A). Since the bound derived by Proposition 1 is defined for each (t, x_0), we construct an image-level estimate $\epsilon^*(x)$ by averaging the corresponding pointwise bounds over t and x_0:

$$\epsilon^*(x) = \sqrt{1 - \max(\tau, 0)^2} \cdot \mathbb{E}_{t,x_0}\left[\frac{1}{t} \cdot \frac{\|v_\theta(x_t, t)\|}{\|J_{v_\theta}(x_t, t)\|}\right], \quad t \sim \mathcal{U}(t^*, 1). \tag{15}$$

The norm $\|J_{v_\theta}\|$ is obtained by applying power iteration with Jacobian–vector products (Miyato et al., 2018), which requires no explicit construction of the Jacobian. Based on this estimate, we introduce a smooth lower-bound regularization term:

$$\mathcal{L}_B = \text{Softplus}\left(1 - \frac{\|\delta\|}{\epsilon^*}\right). \tag{16}$$

Since ϵ^* depends only on x^{src}, it is computed once per image prior to training and cached. Once trained, the immunizer generates image-dependent, bound-aware perturbations in a single forward pass, *without requiring empirical budget search or prompt-specific optimization.*

Training objective. We combine the proposed objectives to form the immunization loss $\mathcal{L}_{\text{imm}}$ as

$$\mathcal{L}_{\text{imm}} = \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_T\mathcal{L}_T + \lambda_B\mathcal{L}_B, \tag{17}$$

where λ_{traj}, λ_T, and λ_B control the contribution of each term. During optimization, we approximate the gradient of $\mathcal{L}_{\text{traj}}$ using Eq. (11), while $\mathcal{L}_T$ and $\mathcal{L}_B$ are directly differentiated.

5 Experiment

5.1 Experimental Setup

Datasets and Edit Frameworks. We evaluate GuardFlow under two text-guided editing settings: *i) Text-guided inpainting (w/ mask).* We use the dataset constructed by DiffVax (Ozden et al., 2026) based on the CCP dataset (Yang et al., 2014), where the inpainting masks and editing prompts were generated using SAM (Kirillov et al., 2023) and ChatGPT (OpenAI, 2024), respectively. The dataset consists of 800 training images and 200 validation images, each paired with two text prompts. *ii) Instruction-based editing (w/o mask).* We use PIE-Bench++ (Huang et al., 2024), which contains 700 image–instruction pairs across ten editing categories (e.g., object-level, attribute-level, background, and style edits). We perform a category-wise stratified 80/20 split, resulting in 560 training and 140 validation pairs. We report the main results on Stable Diffusion 3 (SD3.0) (Esser et al., 2024) under both settings, while additional results on FLUX (Labs, 2024) are provided in Sec. E.1.

Baselines. We compare our approach against representative image immunization methods, including per-image optimization methods and a global immunizer model. As a naive baseline, we include Random Noise, where bounded perturbations are applied without optimization. For per-image optimization, we consider EditShield (Chen et al., 2024), PhotoGuard (Salman et al., 2023), DiffusionGuard (Choi et al., 2025), and SDS (Xue et al., 2024), which generate image-specific perturbations by directly optimizing an objective for each input. For the global immunizer model, we include DiffVax (Ozden et al., 2026), which learns a parametric generator that produces perturbations conditioned on the input image. For instruction-based editing, we exclude DiffusionGuard and DiffVax, as these methods apply only to mask-based editing settings. Per-image optimization methods are evaluated directly on the corresponding validation split, whereas DiffVax and GuardFlow are trained on the corresponding training split and then evaluated on the validation split.

Table 1: **Quantitative results.** Results are reported as mean ± standard deviation over each validation set. "–" indicates that the corresponding value is unavailable. Without immunization, clean-edit CLIP-T scores are 28.46 (Inpainting) and 36.26 (Instruction-based). (**Bold**: best result, Underlined: second-best).

Category	Methods	Editing Failure				Imperceptibility		GPU Req.(MB) ↓	Runtime(s) ↓
		SSIM ↓	PSNR ↓	FSIM ↓	CLIP-T ↓	SSIM(Noise) ↑	PSNR(Noise) ↑		
Inpainting (w/ Mask)	Random Noise	$0.667^{\pm.091}$	$15.32^{\pm2.30}$	$0.567^{\pm.059}$	$28.02^{\pm3.99}$	$0.883^{\pm.025}$	$35.18^{\pm0.11}$	-	-
	EditShield	$0.648^{\pm.061}$	$16.22^{\pm1.68}$	$0.548^{\pm.045}$	$26.39^{\pm4.29}$	$0.857^{\pm.030}$	$34.13^{\pm0.13}$	38815	6.42
	DiffusionGuard	$0.640^{\pm.079}$	$16.29^{\pm1.78}$	$0.540^{\pm.048}$	$26.15^{\pm4.40}$	$0.893^{\pm.041}$	$\mathbf{35.50}^{\pm1.85}$	38252	26.46
	SDS	$0.656^{\pm.053}$	$18.33^{\pm1.66}$	$0.563^{\pm.042}$	$25.49^{\pm4.59}$	$0.889^{\pm.020}$	$34.82^{\pm0.35}$	43238	11.39
	PhotoGuard	$0.635^{\pm.060}$	$16.27^{\pm1.49}$	$0.531^{\pm.039}$	$26.38^{\pm4.40}$	$0.881^{\pm.023}$	$34.61^{\pm0.23}$	97052	51.42
	DiffVax	$0.634^{\pm.062}$	$15.14^{\pm1.41}$	$0.540^{\pm.041}$	$25.46^{\pm4.61}$	$0.880^{\pm.047}$	$34.01^{\pm2.01}$	93934	0.01
	Ours	$\mathbf{0.489}^{\pm.070}$	$\mathbf{9.93}^{\pm1.14}$	$\mathbf{0.455}^{\pm.040}$	$\mathbf{22.89}^{\pm4.67}$	$\mathbf{0.900}^{\pm.014}$	$34.95^{\pm0.69}$	**30840**	**0.01**
Instruction-based editing (w/o Mask)	Random Noise	$0.713^{\pm.157}$	$18.42^{\pm4.78}$	$0.583^{\pm.138}$	$35.64^{\pm4.04}$	$0.866^{\pm.045}$	$\mathbf{35.09}^{\pm0.28}$	-	-
	EditShield	$0.608^{\pm.104}$	$17.39^{\pm2.44}$	$0.490^{\pm.081}$	$34.90^{\pm4.87}$	$0.854^{\pm.049}$	$34.62^{\pm0.29}$	38815	6.00
	SDS	$0.531^{\pm.106}$	$16.26^{\pm2.02}$	$0.455^{\pm.061}$	$35.40^{\pm4.39}$	$0.864^{\pm.038}$	$34.11^{\pm0.72}$	42987	16.34
	PhotoGuard	$0.514^{\pm.087}$	$16.56^{\pm1.74}$	$0.434^{\pm.051}$	$34.54^{\pm4.83}$	$0.871^{\pm.044}$	$34.52^{\pm0.67}$	88655	111.8
	Ours	$\mathbf{0.497}^{\pm.111}$	$\mathbf{15.15}^{\pm1.84}$	$\mathbf{0.403}^{\pm.033}$	$\mathbf{30.50}^{\pm5.49}$	$\mathbf{0.894}^{\pm.031}$	$34.71^{\pm0.53}$	**38712**	**0.01**

Table 2: **Robustness against counterattacks.** Results are averaged over the validation set for inpainting.

Method	CLIP-T ↓								
	None	Rotation ($\pm\pi/6$)	Blur (σ=0.1)	Crop (10%)	Brightness (0.5∼1.5)	JPEG (65% quality)	IMPRESS (Cao et al., 2023)	AdverseCleaner (Zhang, 2023)	DiffPure (Nie et al., 2022)
EditShield	26.39	27.88	26.27	27.59	25.72	27.66	27.23	27.15	26.66
DiffusionGuard	26.15	27.83	26.82	27.70	25.61	27.10	27.52	27.12	27.00
SDS	25.49	27.38	26.19	26.07	25.92	26.21	26.15	26.05	26.17
PhotoGuard	26.38	27.82	26.64	27.98	25.38	27.68	26.90	27.33	26.79
DiffVax	25.46	26.87	25.94	25.91	25.79	25.88	25.90	26.00	25.97
Ours	**22.89**	**23.41**	**23.28**	**23.34**	**22.79**	**23.62**	**24.58**	**23.97**	**24.53**

Evaluation Protocol. We evaluate immunization in four aspects: *i) Editing disruption* compares edit outputs from immunized inputs with those from original inputs under identical conditions, using SSIM (Wang et al., 2004), PSNR, and FSIM (Zhang et al., 2011) for structural and perceptual differences, and CLIP-T (Ruiz et al., 2023) for semantic alignment with the target prompt. *ii) Imperceptibility* measures SSIM and PSNR between original and immunized images. *iii) Scalability* measures peak GPU memory during optimization (per-image for optimization-based methods and training for learned immunizers), while runtime is measured per image during immunization. *iv) Robustness* evaluates editing disruption under geometric (rotation, crop) and photometric (blur, brightness) transformations, JPEG-compression (Wallace, 1991), and purification-based defenses (IMPRESS (Cao et al., 2023), AdverseCleaner (Zhang, 2023), DiffPure (Nie et al., 2022)).

Implementation Details. All experiments are conducted on a single NVIDIA RTX Pro 6000 Blackwell Max-Q GPU. The immunizer f_ϕ is based on a UNet++ (Zhou et al., 2018) architecture with a 3-channel input. The model is optimized using Adam (Kingma & Ba, 2015) with a learning rate of $1/255$ for 100 epochs. The loss weights are set to $\lambda_{\text{traj}} = 10$, $\lambda_{\text{T}} = 10$ and $\lambda_{\text{B}} = 0.1$. The terminal objective $\mathcal{L}_{\text{T}}$ conditions the velocity field using the text prompt paired with each training sample. For the perturbation bound, we set $\tau = 0$ and $t^* = 0.6$, matching the inversion strength of the editing pipeline, and constrain the perturbation to $\|\delta\|_\infty \leq \beta = 16/255$ by clipping the immunizer output. All baselines are calibrated to a comparable PSNR between the original and immunized images for fair comparison. For consistent evaluation, we use the same random seed of 5 across all methods. In flow-based editing, the mask does not condition the velocity field, but is applied through latent compositing between sampling steps. Since $\mathcal{L}_{\text{imm}}$ is evaluated directly on the velocity field at independently sampled states, the immunizer is trained without incorporating the mask (Sec. E.3).

5.2 Quantitative Results

Editing Disruption & Imperceptibility As shown in Tab. 1, GuardFlow surpasses existing immunization baselines by a clear margin across both editing settings. In particular, GuardFlow achieves 1.86× (inpainting) and 3.35× (instruction-based editing) greater CLIP-T reduction from the clean-edit score than the strongest baseline. At the same time, both settings achieve the highest SSIM(Noise), showing that the perturbations remain visually imperceptible despite their strong disruptive effect. These results show that GuardFlow improves editing disruption while preserving the perceptual quality of immunized images.

Scalability Unlike optimization-based methods that require iterative perturbation search per image, GuardFlow learns a global immunizer that generates perturbations in a single forward pass, enabling



Figure 3: **Qualitative comparisons.** Immunized edit results based on SD3.0 framework on CCP dataset. Each row displays an original image and edit prompt pair, along with immunized results based on GuardFlow and other baseline methods. The second column from the left shows the immunization result using GuardFlow.

efficient inference (Tab. 1). This advantage is particularly important for flow-based frameworks, where generation involves costly trajectory evaluations. By avoiding backpropagation through the full trajectory, GuardFlow achieves the lowest GPU memory requirement during training.

Robustness We further assess the robustness of GuardFlow against various counterattacks. As shown in Tab. 2, GuardFlow consistently achieves the lowest CLIP-T across all attacks. GuardFlow remains robust to standard image transformations, e.g. rotation, blur, crop, as well as purification-based defenses such as IMPRESS, AdverseCleaner, and DiffPure. These results show that GuardFlow retains its immunization effect across various counterattacks.

5.3 Qualitative Results

Comparison with baselines. Fig. 3 illustrates the qualitative comparison of GuardFlow with existing baselines under the inpainting setting. GuardFlow preserves the overall structure of the source image while suppressing object-, color-, and texture-level changes associated with the target prompt. The terminal regression objective keeps the editing outcome close to the source image, while trajectory disruption perturbs the underlying editing dynamics and reduces the influence of prompt-guided editing. Consequently, the edited outputs retain the structural characteristics of the source image, while target-specific features are weakened and appear blurred. In contrast, baseline methods preserve most of the target-prompt features and produce results similar to the original edits. Additional qualitative results across diverse images and editing scenarios are provided in Sec. G.

Generalization to Unseen Images. To examine the generalizability of GuardFlow across diverse images and editing scenarios, we conduct a qualitative cross-dataset evaluation under the instruction-based editing setting. The immunizer trained on PIE-Bench++ is directly evaluated on samples from ImagenHub (Ku et al., 2024) without additional training or adaptation. As shown in Fig. 4, GuardFlow disrupts the intended edits across diverse examples, e.g., object manipulation, attribute modification, and style transfer. These results highlight the transferability of our learned immunizer. Further generalization to unseen masks, models, and inverse-free editing is provided in Sec. E.

5.4 Model Analysis

Component Analysis. To assess the contribution of each objective and the per-image perturbation bound, we conduct an ablation study in the inpainting setting (Tab. 3). Removing $\mathcal{L}_{\text{traj}}$ substantially weakens editing disruption, confirming its primary role in disrupting the editing process. Replacing $\mathcal{L}_T$ with a simple magnitude penalty $\|\delta\|_2^2$ results

Table 3: **Ablation Study.** $\mathcal{L}_B^{\text{fixed}}$ sets $\epsilon^* = 4/255$.

Loss	CLIP-T↓	SSIM(Noise)↑	$\|\delta\|_2/\sqrt{dim.}$
No immunization	28.46	–	–
$\mathcal{L}_T + \mathcal{L}_B$	26.08	0.909	4.38/255
$\mathcal{L}_{\text{traj}} + \|\delta\|_2^2 + \mathcal{L}_B$	24.41	0.889	4.82/255
$\mathcal{L}_{\text{traj}} + \mathcal{L}_T + \mathcal{L}_B^{\text{fixed}}$	23.84	0.876	5.55/255
$\mathcal{L}_{\text{traj}} + \mathcal{L}_T + \mathcal{L}_B$	22.89	0.900	4.58/255

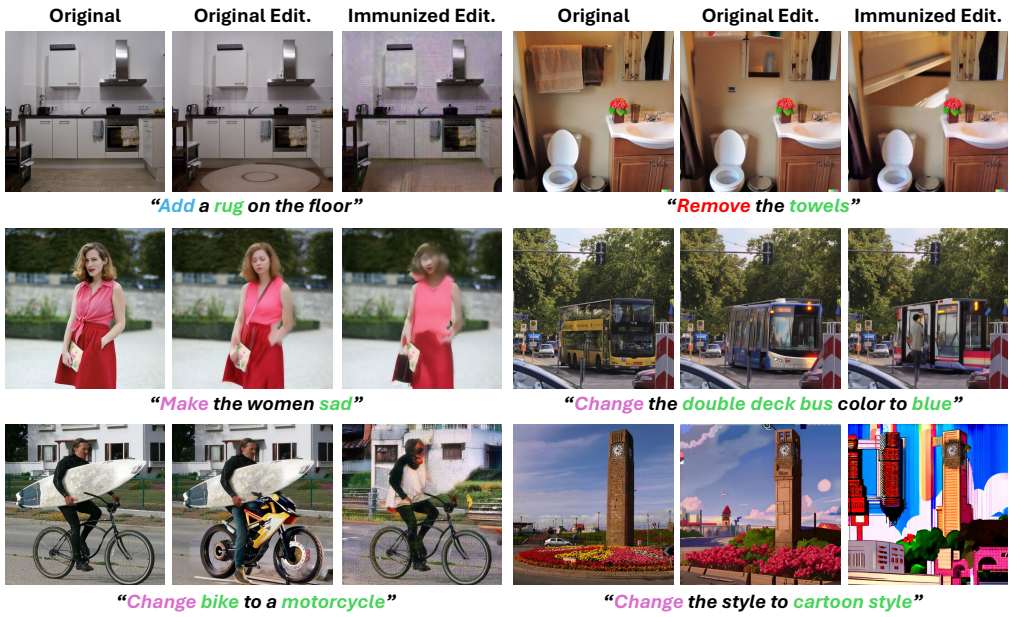


Figure 4: **Qualitative cross-dataset results on ImagenHub.** Each example shows the original image and the editing results before and after immunization with GuardFlow.

in weaker disruption despite a larger perturbation, showing that $\mathcal{L}_T$ contributes beyond its implicit regularization effect analyzed in Sec. C. Finally, replacing the per-image bound with a fixed perturbation budget yields weaker disruption with larger perturbations, demonstrating the benefit of adapting the perturbation magnitude to image-specific vector-field sensitivity.

Validity of the perturbation bound. To validate the image-specific perturbation bound, we examine whether ϵ^* reflects per-image vector-field sensitivity. Fig. 5a reports the per-dataset Spearman correlation (Spearman, 1961) between ϵ^* and five low-level statistics. ϵ^* correlates positively with structural statistics, including edge strength, entropy, and high-frequency energy, and negatively with brightness. The consistency across datasets suggests that ϵ^* captures a relation between image structure and vector-field sensitivity that generalizes across datasets. Additional attributes are provided in Sec. F.1. Fig. 5b shows that complex-textured images exhibit higher ϵ^*, whereas smoother images exhibit lower values, demonstrating image-dependent variation. These results confirm that ϵ^* provides an image-adaptive bound according to vector-field sensitivity.

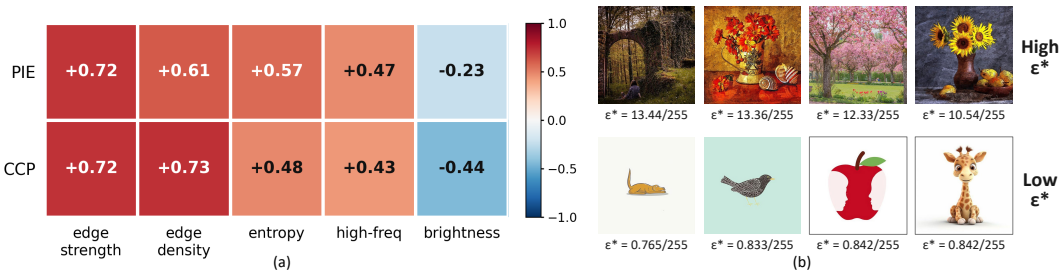


Figure 5: ϵ^* **Association with Image Attributes.** (a) Spearman correlation between low-level image statistics and ϵ^*, measured independently on each dataset. (b) Images with the highest and lowest ϵ^* on PIE-Bench++.

6 Conclusion

In this work, we introduced GuardFlow, the first image immunization framework for flow-based generative editing models. GuardFlow directly exploits flow trajectory dynamics by perturbing the learned velocity field and regressing the predicted edit toward the source image. We further derive an image-adaptive lower bound on the perturbation magnitude necessary for trajectory disruption based on per-image vector-field sensitivity. Extensive experiments show that GuardFlow effectively suppresses text-guided editing across different settings and consistently outperforms existing methods. These results demonstrate the effectiveness of our approach and provide a foundation for protecting images against emerging flow-based editing models.

AI USE STATEMENT

We used Large Language Models (LLMs) solely to aid in proofreading and improving the readability of this manuscript. No LLMs were used in the development of the proposed method, formulation of theoretical or mathematical claims, writing of proofs, design or execution of experiments, implementation, data analysis, or interpretation of results. The CCP-based inpainting benchmark of (Ozden et al., 2026), which we adopt without modification, includes editing prompts originally generated using ChatGPT and we use the released prompts as provided without any additional prompt generation or modification. All LLM-assisted edits to the manuscript were reviewed by the authors to ensure that they preserved the intended technical meaning and scientific content. We take full responsibility for the final content of this work, including text revised with the aid of LLMs.

REPRODUCIBILITY

To ensure reproducibility, we provide detailed implementation and hyperparameter settings for GuardFlow and all baseline methods in Sec. D. The theoretical derivations and analyses underlying the proposed perturbation bound and optimization strategy are also provided in the appendix. Experimental settings, including datasets, editing frameworks, evaluation protocols, and training configurations, are described in Sec. 5 and the appendix. We will release the source code and implementation upon publication.

ETHICS

This work aims to protect images from unauthorized or malicious manipulation by flow-based generative editing models. GuardFlow introduces imperceptible perturbations to source images to disrupt subsequent text-guided editing without modifying the underlying generative model. Our experiments use existing image-editing benchmarks and publicly available generative models, and no additional human-subject data are collected. While image immunization may interfere with legitimate editing when applied intentionally, the proposed method is developed for defensive use to give image owners greater control over unauthorized modification of their content.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the StyleGAN latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Adobe Inc. Adobe delivers new AI innovations, assistants and models across creative cloud to empower creative professionals. https://news.adobe.com/news/2025/10/adobe-max-2025-creative-cloud, 2025. Accessed: 2026-09-24.

Shivangi Aneja, Lev Markhasin, and Matthias Nießner. Tafim: Targeted adversarial attacks against facial image manipulations. In *European Conference on Computer Vision (ECCV)*. Springer, 2022.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in GANs. *International Conference on Learning Representations (ICLR)*, 2021.

Ruoxi Chen, Haibo Jin, Yixin Liu, Jinyin Chen, Haohan Wang, and Lichao Sun. Editshield: Protecting unauthorized image editing by instruction-guided diffusion models. In *European Conference on Computer Vision (ECCV)*. Springer, 2024.

June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. Diffusionguard: A robust defense against malicious diffusion-based image editing. *International Conference on Learning Representations (ICLR)*, 2025.

Samantha Cole. We are truly fucked: Everyone is making ai-generated fake porn now, 2018. URL https://web.archive.org/web/20240926135620/https://www.vice.com/en/article/reddit-fake-porn-app-daisy-ridley/. Accessed: 2024-11-14.

Jess Davies and Sarah McDermott. Deepfaked: 'they put my face on a porn video', 2022. URL https://www.bbc.com/news/uk-62821117. Accessed: 2024-11-14.

Rebecca A. Delfino. Deepfakes on trial: A call to expand the trial judge's gatekeeping role to protect legal proceedings from technological fakery. *SSRN Electronic Journal*, 2022. URL https://api.semanticscholar.org/CorpusID:246806628.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Juntao He, Haoran Dai, Runqi Sui, Xuejing Yuan, Dun Liu, Hao Feng, Xinyue Liu, Wenchuan Yang, Baojiang Cui, and Kedan Li. Evilpromptfuzzer: generating inappropriate content based on text-to-image models. *Cybersecurity*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Mingzhen Huang, Jialing Cai, Shan Jia, Vishnu S Lokhande, and Siwei Lyu. Paralleledits: Efficient multi-aspect text-driven image editing with attention grouping. *Advances in Neural Information Processing Systems(NeurIPS)*, 2024.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Guanlong Jiao, Biqing Huang, Kuan-Chieh Wang, and Renjie Liao. Uniedit-flow: Unleashing inversion and editing in the era of flow models. In *International Conference on Learning Representations(ICLR)*, 2026.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. In *International Conference on Learning Representations*, 2026.

Jimyeong Kim, Jungwon Park, Yeji Song, Nojun Kwak, and Wonjong Rhee. Reflex: Text-guided editing of real images in rectified flow via mid-step feature extraction and attention adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*. IEEE, 2023.

Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. *International Conference on Learning Representations (ICLR)*, 2024.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025.

Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations (ICLR)*, 2023.

Jean Mackenzie and Leehyun Choi. Inside the deepfake porn crisis engulfing korean schools. 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (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. *International Conference on Learning Representations (ICLR)*, 2022.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2018.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *International Conference on Machine Learning (ICML)*, 2022.

OpenAI. ChatGPT (oct 2 version) [large language model], 2024. URL https://chat.openai.com.

Tarik Can Ozden, Ozgur Kara, Oguzhan Akcin, Kerem Zaman, Shashank Srivastava, Sandeep P Chinchali, and James Matthew Rehg. Diffvax: Optimization-free image immunization against diffusion-based editing. In *International Conference on Learning Representations (ICLR)*, 2026.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *2021 IEEE/CVF international conference on computer vision (ICCV)*. IEEE, 2021.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations(ICLR)*, 2024.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *International Conference on Learning Representations (ICLR)*, 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 *International Conference on Machine Learning (ICML)*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2025.

Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *European conference on computer vision*. Springer, 2020.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *International Conference on Machine Learning (ICML)*, 2023.

Charles Spearman. The proof and measurement of association between two things. 1961.

Gregory K Wallace. The jpeg still picture compression standard. *ACM Communications*, 1991.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024.

Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *International Conference on Machine Learning (ICML)*, 2025.

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), 2004.

John Wojewidka. The deepfake threat to face biometrics. *Biometric Technology Today*, pp. 5–7, 2020. URL https://api.semanticscholar.org/CorpusID:212981964.

Pengfei Wu, Xiaomei Li, Ziming He, and Ling Fan. Advisorai: Transforming graphic advertising with generative ai-enhanced design strategies. In *International Conference on Human-Computer Interaction*, pp. 295–304. Springer, 2024.

Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *International Conference on Learning Representations(ICLR)*, 2024.

Wei Yang, Ping Luo, and Liang Lin. Clothing co-parsing by joint image segmentation and labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *International Conference on Learning Representations (ICLR)*, 2025.

Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, March 2020.

Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 2011.

Lvmin Zhang. Adversecleaner. https://github.com/lllyasviel/AdverseCleaner, 2023. GitHub repository.

Wanfeng Zheng, Qiang Li, Xiaoyan Guo, Pengfei Wan, and Zhongyuan Wang. Bridging clip and stylegan through latent alignment for image editing. *arXiv preprint arXiv:2210.04506*, 2022.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *International workshop on deep learning in medical image analysis*. Springer, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE international conference on computer vision (ICCV)*, 2017.

Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

APPENDIX

In this appendix, we provide additional materials supporting the main manuscript. Specifically, we include 1) theoretical analyses of the perturbation bound, Jacobian-free gradient approximation, and implicit regularization of the terminal objective, 2) implementation details and hyperparameter settings for GuardFlow and the baseline methods, 3) extended evaluations across flow-based editing models, editing settings, unseen masks, and cross-model transfer, 4) further analyses of the image-specific perturbation bound, terminal estimate, and hyperparameter sensitivity, and 5) qualitative results across diverse editing scenarios.

A PROOF OF PROPOSITION 1

Proposition 1. *Let $v_\theta(x_t, t)$ be a vector field differentiable with respect to x, with Jacobian $J_{v_\theta}(x_t, t) := \frac{\partial v_\theta(x_t,t)}{\partial x}$. Given $\tilde{x}_t = x_t + t\delta$ as defined in Eq. (9), under a local first-order Taylor approximation of the vector field around x_t, we have $v_\theta(\tilde{x}_t, t) \approx v_\theta(x_t, t) + J_{v_\theta}(x_t, t) \cdot t\delta$.*

For $\tau \in (-1, 1)$, if

$$\|\delta\| < \frac{1}{t} \cdot \sqrt{1 - \max(\tau, 0)^2} \cdot \frac{\|v_\theta(x_t, t)\|}{\|J_{v_\theta}(x_t, t)\|},$$

then $\mathcal{D}(t) > \tau$ for every direction of δ. Equivalently, satisfying $\mathcal{D}(t) \le \tau$ requires $\|\delta\|$ to be at least this value.

Proof. We begin with the local first-order approximation used throughout the derivation. From $\tilde{x}_t = x_t + t\delta$, a Taylor expansion of the vector field around x_t gives

$$v_\theta(\tilde{x}_t, t) = v_\theta(x_t, t) + J_{v_\theta}(x_t, t)\, t\delta + \mathcal{R}, \tag{18}$$

where $J_{v_\theta}(x_t, t) := \partial v_\theta(x_t, t)/\partial x_t$ and $\|\mathcal{R}\| = O(\|t\delta\|^2)$. Neglecting the higher-order term gives

$$v_\theta(\tilde{x}_t, t) \approx v + \Delta, \qquad v := v_\theta(x_t, t), \qquad \Delta := J_{v_\theta}(x_t, t)\, t\delta. \tag{19}$$

Throughout the proof, we assume $v \neq 0$; whenever $\mathcal{D}(t)$ is evaluated, $v + \Delta \neq 0$ is implicit in its definition. Under the first-order approximation in Eq. (19), we identify $v_\theta(\tilde{x}_t, t)$ with $v + \Delta$ in what follows.

We first consider $\tau \in [0, 1)$. Using Eq. (19), the directional alignment is

$$\mathcal{D}(t) = \cos(v + \Delta, v) = \frac{\langle v + \Delta, v\rangle}{\|v + \Delta\|\|v\|} = \frac{\|v\|^2 + \langle \Delta, v\rangle}{\|v + \Delta\|\|v\|}. \tag{20}$$

We show that $\mathcal{D}(t) \le \tau$ requires

$$\|\Delta\| \ge \sqrt{1 - \tau^2}\, \|v\|. \tag{21}$$

If $\langle v + \Delta, v\rangle < 0$, then

$$\|v\|^2 < -\langle \Delta, v\rangle \le \|\Delta\|\|v\|, \tag{22}$$

where the last inequality follows from Cauchy–Schwarz. Hence,

$$\|\Delta\| > \|v\| \ge \sqrt{1 - \tau^2}\, \|v\|, \tag{23}$$

and Eq. (21) directly follows. Otherwise, $\langle v + \Delta, v\rangle \ge 0$.

Since both $\langle v + \Delta, v\rangle \ge 0$ and $\tau \ge 0$, squaring preserves the inequality $\mathcal{D}(t) \le \tau$:

$$\big(\|v\|^2 + \langle \Delta, v\rangle\big)^2 \le \tau^2 \|v\|^2 \big(\|v\|^2 + 2\langle \Delta, v\rangle + \|\Delta\|^2\big), \tag{24}$$

where $\|v + \Delta\|^2 = \|v\|^2 + 2\langle \Delta, v\rangle + \|\Delta\|^2$. Letting $a := \langle \Delta, v\rangle$, for fixed $\|\Delta\|$ and $\|v\|$, the difference between the two sides of Eq. (24) becomes

$$f(a) = (\|v\|^2 + a)^2 - \tau^2 \|v\|^2 \big(\|v\|^2 + 2a + \|\Delta\|^2\big). \tag{25}$$

Since $f(a)$ is a quadratic with a positive leading coefficient, its global minimum over $a \in \mathbb{R}$ is attained at

$$a^* = -(1 - \tau^2)\|v\|^2. \tag{26}$$

Substituting a^* gives

$$f(a^*) = \tau^2\|v\|^2\Big[(1-\tau^2)\|v\|^2 - \|\Delta\|^2\Big]. \tag{27}$$

For Eq. (24) to hold at the actual $a = \langle \Delta, v\rangle$, the global minimum must satisfy $f(a^*) \le 0$. For $\tau \in (0,1)$, this gives

$$\|\Delta\|^2 \ge (1-\tau^2)\|v\|^2, \tag{28}$$

and therefore

$$\|\Delta\| \ge \sqrt{1-\tau^2}\,\|v\|. \tag{29}$$

For $\tau = 0$, $\mathcal{D}(t) \le 0$ requires $\langle v+\Delta, v\rangle \le 0$, and the same Cauchy–Schwarz argument with a non-strict inequality gives $\|\Delta\| \ge \|v\|$. Thus, Eq. (21) holds for all $\tau \in [0,1)$.

Next, from the definition of Δ,

$$\|\Delta\| = \|J_{v_\theta}(x_t,t)\,t\delta\| \le \|J_{v_\theta}(x_t,t)\|\,t\|\delta\|, \tag{30}$$

where $\|J_{v_\theta}(x_t,t)\|$ denotes the induced ℓ_2 operator norm. Combining Eqs. (21) and (30) yields

$$\sqrt{1-\tau^2}\,\|v_\theta(x_t,t)\| \le \|\Delta\| \le \|J_{v_\theta}(x_t,t)\|\,t\|\delta\|. \tag{31}$$

Therefore, for $\tau \in [0,1)$,

$$\|\delta\| \ge \frac{1}{t}\cdot\sqrt{1-\tau^2}\cdot\frac{\|v_\theta(x_t,t)\|}{\|J_{v_\theta}(x_t,t)\|}. \tag{32}$$

We next consider $\tau < 0$. If $\mathcal{D}(t) \le \tau$, then $\langle v+\Delta, v\rangle < 0$, which implies

$$\|v\|^2 < -\langle \Delta, v\rangle \le \|\Delta\|\|v\|. \tag{33}$$

Hence,

$$\|\Delta\| > \|v\|. \tag{34}$$

Combining Eqs. (30) and (34) gives

$$\|\delta\| > \frac{1}{t}\cdot\frac{\|v_\theta(x_t,t)\|}{\|J_{v_\theta}(x_t,t)\|}. \tag{35}$$

Since $\max(\tau,0) = 0$ for $\tau < 0$, this implies

$$\|\delta\| \ge \frac{1}{t}\cdot\sqrt{1-\max(\tau,0)^2}\cdot\frac{\|v_\theta(x_t,t)\|}{\|J_{v_\theta}(x_t,t)\|}. \tag{36}$$

Together, the two cases yield, for any $\tau \in (-1,1)$,

$$\mathcal{D}(t) \le \tau \quad\Longrightarrow\quad \|\delta\| \ge \frac{1}{t}\cdot\sqrt{1-\max(\tau,0)^2}\cdot\frac{\|v_\theta(x_t,t)\|}{\|J_{v_\theta}(x_t,t)\|}. \tag{37}$$

Taking the contrapositive gives the stated result. □

Remark (Tightness and saturation). For $\tau \in (0,1)$, the bound in Eq. (21) is attained when $\langle \Delta, v\rangle = -(1-\tau^2)\|v\|^2$ and $\|\Delta\| = \sqrt{1-\tau^2}\|v\|$, yielding $\mathcal{D}(t) = \tau$. Geometrically, $\sqrt{1-\tau^2}\|v\| = \|v\|\sin(\arccos\tau)$ is the distance from v to the set of nonzero vectors with cosine similarity at most τ.

At $\tau = 0$, this distance has infimum $\|v\|$, which remains unchanged for $\tau < 0$. Indeed, $\mathcal{D}(t) \le \tau < 0$ requires $\|\Delta\| > \|v\|$, while any $\|\Delta\| > \|v\|$ with $\Delta = -\|\Delta\|v/\|v\|$ gives $\mathcal{D}(t) = -1$. Thus, the required velocity-change magnitude saturates at its $\tau = 0$ value for negative thresholds.

The resulting bound on $\|\delta\|$ additionally requires the operator-norm inequality in Eq. (30) to be tight. These tightness statements hold under the first-order approximation in Eq. (19), whose neglected remainder is $O(\|t\delta\|^2)$. Thus, Proposition 1 provides a necessary magnitude condition for disruption.

B JACOBIAN-FREE GRADIENT APPROXIMATION VALIDATION

In Sec. 4.1, we replace $\partial v_\theta(\tilde{x}_t, t \mid \emptyset)/\partial \tilde{x}_t$ with the identity when computing $\nabla_\delta \mathcal{L}_{\text{traj}}$. Since this approximation directly determines the optimization direction for δ, a key question is whether the resulting update remains aligned with the exact gradient.

We therefore characterize the directional alignment between the Jacobian-free update and the exact gradient. For a tractable theoretical analysis, we consider the optimal rectified-flow vector field $v^\star$. We first study an isotropic Gaussian setting, where the Jacobian behavior can be derived in closed form, and then extend the analysis to a general target distribution. Finally, we empirically examine whether the resulting directional behavior also holds for the pre-trained vector fields v_θ used by GuardFlow.

B.1 CONDITION FOR POSITIVE GRADIENT ALIGNMENT

Let $g_t = v_\theta(\tilde{x}_t, t \mid \emptyset) - v_t^{\text{src}}$, and denote the Jacobian of the vector field with respect to its input by

$$J_v := \frac{\partial v_\theta(\tilde{x}_t, t \mid \emptyset)}{\partial \tilde{x}_t}. \tag{38}$$

Differentiation through v_θ transforms the residual g_t into $J_v^\top g_t$, whereas the Jacobian-free approximation uses g_t directly. Thus, the Jacobian-free direction is positively aligned with the exact Jacobian-transformed direction when

$$\left\langle J_v^\top g_t, g_t \right\rangle = g_t^\top J_v g_t > 0. \tag{39}$$

We next relate this condition to the gradient with respect to the perturbation δ. From Eq. (9),

$$\tilde{x}_t = x_t^{\text{src}} + t\delta, \qquad \frac{\partial \tilde{x}_t}{\partial \delta} = tI. \tag{40}$$

Therefore, up to a common positive scalar factor, the exact and Jacobian-free gradients with respect to δ are respectively given by

$$d_{\text{exact}} = tJ_v^\top g_t, \qquad d_{\text{JF}} = tg_t. \tag{41}$$

Their inner product satisfies

$$\langle d_{\text{exact}}, d_{\text{JF}} \rangle = t^2 g_t^\top J_v g_t. \tag{42}$$

Since $t > 0$ over the optimization interval, the two perturbation-level directions are positively aligned if and only if

$$g_t^\top J_v g_t > 0. \tag{43}$$

Hence, the directional effect of the Jacobian-free approximation can be characterized directly through the quadratic form $g_t^\top J_v g_t$.

B.2 GAUSSIAN TOY SETTING

We first consider a Gaussian toy setting in which both endpoints follow $x_0 \sim \mathcal{N}(0, I)$ and $x_1 \sim \mathcal{N}(0, I)$, and are connected through the rectified-flow interpolation $x_t = (1-t)x_0 + tx_1$. Because x_t is a linear combination of two independent Gaussian variables, its covariance is $\big((1-t)^2 + t^2\big)I$, while $\text{Cov}(x_1, x_t) = tI$. The conditional mean of x_1 given x_t is therefore

$$\mathbb{E}[x_1 \mid x_t] = \frac{t}{(1-t)^2 + t^2} x_t. \tag{44}$$

The optimal rectified-flow vector field is $v^\star(x_t, t) := \mathbb{E}[x_1 - x_0 \mid x_t]$. Since $x_0 = (x_t - tx_1)/(1-t)$, it can be expressed as

$$v^\star(x_t, t) = \frac{1}{1-t}\left(\mathbb{E}[x_1 \mid x_t] - x_t\right) = \frac{2t-1}{(1-t)^2 + t^2} x_t. \tag{45}$$

Its Jacobian is therefore

$$\boxed{J_{v^\star}(x_t, t) = \frac{2t-1}{(1-t)^2 + t^2} I.} \tag{46}$$

Define $\alpha_t := (2t-1)/\big((1-t)^2+t^2\big)$, such that $J_{v^\star}(x_t,t) = \alpha_t I$. Applying the alignment criterion in Eq. (39) gives

$$g_t^\top J_{v^\star} g_t = \alpha_t \|g_t\|_2^2. \tag{47}$$

Thus, the sign of the alignment is determined entirely by α_t: negative α_t for $t < \frac{1}{2}$ results in negative alignment, $\alpha_t = 0$ at $t = \frac{1}{2}$ gives zero alignment, and positive α_t for $t > \frac{1}{2}$ results in positive alignment. Moreover,

$$J_{v^\star}(x_t,t)^\top g_t = \alpha_t g_t. \tag{48}$$

Hence, for every $t > \frac{1}{2}$, the Jacobian-transformed and Jacobian-free directions are collinear with the same orientation. In this Gaussian setting, the Jacobian-free approximation therefore preserves the exact optimization direction up to a positive scalar factor for $t > \frac{1}{2}$.

The coefficient further satisfies $\lim_{t\to 1} \alpha_t = 1$, and consequently

$$\lim_{t\to 1} J_{v^\star}(x_t,t) = I. \tag{49}$$

Hence, positive directional alignment holds for $t > \frac{1}{2}$, while the Jacobian itself approaches the identity toward the target endpoint.

Table 4: **Jacobian coefficient α_t in the Gaussian toy setting.** Positive coefficients indicate directional preservation up to positive scaling, while α_t approaches 1 toward the target endpoint.

t	0.1	0.3	0.5	**0.6**	0.7	0.8	0.9
α_t	-0.98	-0.69	0.00	**0.38**	0.69	0.88	0.98

B.3 Structure of the Rectified-Flow Jacobian

We next extend the analysis to a general target distribution. For the rectified-flow interpolation $x_t = (1-t)x_0 + tx_1$ with $x_0 \sim \mathcal{N}(0,I)$, the conditional distribution satisfies $x_t \mid x_1 \sim \mathcal{N}(tx_1, (1-t)^2 I)$. Tweedie's formula therefore gives

$$\mathbb{E}[x_1 \mid x_t] = \frac{1}{t}\left(x_t + (1-t)^2 \nabla_{x_t} \log p_t(x_t)\right), \tag{50}$$

where p_t denotes the marginal density of x_t. Differentiating with respect to x_t yields

$$\frac{\partial}{\partial x_t}\mathbb{E}[x_1 \mid x_t] = \frac{1}{t}\left(I + (1-t)^2 \nabla_{x_t}^2 \log p_t(x_t)\right) = \frac{t}{(1-t)^2}\Sigma_t, \tag{51}$$

where $\Sigma_t := \mathrm{Cov}(x_1 \mid x_t)$.

Using $v^\star(x_t,t) = \big(\mathbb{E}[x_1 \mid x_t] - x_t\big)/(1-t)$, the Jacobian of the optimal vector field can be written as

$$\boxed{J_{v^\star}(x_t,t) = \frac{1}{t} I + \frac{1-t}{t}\nabla_{x_t}^2 \log p_t(x_t) = \frac{t}{(1-t)^3}\Sigma_t - \frac{1}{1-t} I.} \tag{52}$$

For any nonzero direction u, Eq. (52) gives

$$u^\top J_{v^\star} u = \frac{t}{(1-t)^3} u^\top \Sigma_t u - \frac{1}{1-t}\|u\|_2^2. \tag{53}$$

Therefore, $J_{v^\star}^\top u$ is positively aligned with u whenever

$$\boxed{\frac{u^\top \Sigma_t u}{\|u\|_2^2} > \frac{(1-t)^2}{t}.} \tag{54}$$

The left-hand side is the posterior variance of x_1 along the normalized direction $u/\|u\|_2$, while the right-hand side defines a timestep-dependent threshold. Thus, Eq. (54) provides a necessary and sufficient condition for positive directional alignment without requiring $\boldsymbol{u}$ to be an eigenvector of the Jacobian.

We further examine the behavior of the optimal Jacobian toward the target endpoint. Let $H_t := \nabla^2_{x_t} \log p_t(x_t)$. From Eq. (52),

$$J_{v^\star}(x_t, t) - I = \frac{1-t}{t}(I + H_t), \tag{55}$$

and consequently

$$\|J_{v^\star}(x_t, t) - I\|_2 \leq \frac{1-t}{t}\left(1 + \|H_t\|_2\right). \tag{56}$$

Therefore, under the sufficient regularity condition

$$(1-t)\|H_t\|_2 \to 0 \qquad \text{as } t \to 1, \tag{57}$$

we obtain

$$\boxed{\lim_{t\to 1} J_{v^\star}(x_t, t) = I.} \tag{58}$$

Together with Eq. (54), the result shows that directional alignment at finite timesteps and convergence of the Jacobian to the identity near the endpoint are distinct properties. In the next section, we empirically examine whether this directional alignment also holds for the learned vector fields used by GuardFlow.

B.4 Empirical Validation on Pre-trained Vector Fields

Since the preceding analysis considers the optimal vector field $v^\star$, we empirically validate the gradient alignment for the pre-trained vector fields v_θ used by GuardFlow. Using the exact and Jacobian-free directions defined in Eq. (41), we measure their cosine alignment as

$$A(t) = \frac{\langle d_{\text{exact}}, d_{\text{JF}} \rangle}{\|d_{\text{exact}}\|_2 \, \|d_{\text{JF}}\|_2}. \tag{59}$$

The common factor $t > 0$ cancels in $A(t)$, leaving the cosine alignment between $J_v^\top g_t$ and g_t.

We evaluate $A(t)$ on the training images from CCP and PIE-Bench++ used to train the immunizer with the Jacobian-free update. For each image, we use the perturbation δ generated by the immunizer trained for the corresponding backbone and sample five noise instances x_0. The Jacobian-transformed direction is computed using a vector–Jacobian product without explicitly forming the Jacobian. We report the mean and standard deviation of $A(t)$ at $t \in \{0.6, 0.7, 0.8, 0.9\}$, together with the fraction of (x^{src}, x_0, t) samples satisfying $A(t) > 0$ for $t \sim \mathcal{U}(t^*, 1)$.

Table 5: **Empirical directional alignment of the Jacobian-free update.** Cosine alignment $A(t)$ between the exact and Jacobian-free gradient directions at different timesteps. Positive (%) denotes the fraction of samples satisfying $A(t) > 0$ over $t \in [t^*, 1]$, where $t^* = 0.6$.

Model	$t = 0.6$	$t = 0.7$	$t = 0.8$	$t = 0.9$	**Positive (%)** $\uparrow$
SD3	0.43 ± 0.09	0.52 ± 0.08	0.55 ± 0.07	0.60 ± 0.05	**99.2**

As shown in Tab. 5, the exact and Jacobian-free gradient directions are positively aligned for over 99% of the samples. Moreover, the cosine alignment increases as t approaches 1. The result is consistent with the theoretical analysis above and empirically support the Jacobian-free update used by GuardFlow.

C $\mathcal{L}_T$ AS AN IMPLICIT REGULARIZER

In Sec. 4.2, $\mathcal{L}_T$ regresses the terminal estimate toward the source image. Beyond this regression objective, its dependence on the perturbation δ induces an implicit regularization effect on the perturbation magnitude. We make this structure explicit through a local expansion of the terminal objective.

From Eq. (9), the perturbed state is given by $\tilde{x}_t = x_t^{\mathrm{src}} + t\delta$. Applying a first-order expansion of the guided vector field around x_t^{src}, the terminal estimate becomes

$$\hat{x}^{\mathrm{edit}}(\tilde{x}_t) \approx \hat{x}^{\mathrm{edit}}(x_t^{\mathrm{src}}) + M_t\delta, \qquad M_t := t\left(I + (1-t)J_{v_\theta^{\mathrm{cfg}}}(x_t^{\mathrm{src}}, t)\right), \tag{60}$$

where

$$\hat{x}^{\mathrm{edit}}(x_t^{\mathrm{src}}) = x_t^{\mathrm{src}} + (1-t)v_\theta^{\mathrm{cfg}}(x_t^{\mathrm{src}}, t \mid c) \tag{61}$$

denotes the terminal estimate along the clean trajectory.

Let $r_t := x^{\mathrm{src}} - \hat{x}^{\mathrm{edit}}(x_t^{\mathrm{src}})$, which is independent of δ. Substituting Eq. (60) into the terminal objective gives

$$\begin{aligned}\mathcal{L}_T &\approx \mathbb{E}_t\left[\|r_t - M_t\delta\|_2^2\right] \\ &= \underbrace{\mathbb{E}_t[\|r_t\|_2^2]}_{\text{constant}} - \underbrace{2\mathbb{E}_t[\langle r_t, M_t\delta\rangle]}_{\text{linear}} + \underbrace{\delta^\top A\delta}_{\text{quadratic}}, \qquad A := \mathbb{E}_t[M_t^\top M_t] \succeq 0.\end{aligned} \tag{62}$$

The constant term does not affect optimization. The linear term preserves the edit-dependent regression signal toward the source, whereas the positive-semidefinite quadratic term penalizes the perturbation magnitude. Importantly, this regularization is anisotropic: A depends on the Jacobian of the guided vector field and therefore weights perturbation directions according to the local sensitivity of the terminal prediction.

The regularization effect becomes particularly clear near the terminal endpoint. As $t \to 1$, $x_t^{\mathrm{src}} \to x^{\mathrm{src}}$, $r_t \to 0$, and $M_t \to I$. Consequently,

$$\lim_{t\to 1} \mathcal{L}_T = \|\delta\|_2^2. \tag{63}$$

Thus, near the endpoint, $\mathcal{L}_T$ increasingly behaves as a direct quadratic penalty on the perturbation. At earlier timesteps, the nonzero residual r_t retains the regression signal toward the source. Sampling $t \sim \mathcal{U}(t^*, 1)$ therefore allows $\mathcal{L}_T$ to incorporate both terminal regression and perturbation regularization.

This implicit regularization complements the trajectory disruption objective in Sec. 4.1. While $\mathcal{L}_{\mathrm{traj}}$ promotes deviation from the source trajectory, the quadratic component of $\mathcal{L}_T$ penalizes excessive perturbation magnitude. Accordingly, $\mathcal{L}_T$ serves a dual purpose: regressing the predicted edit toward the source while implicitly regularizing the perturbation.

D Implementation Details

All immunization methods, including GuardFlow and the baseline methods, are evaluated using SD3.0-medium and FLUX.1-dev. Specifically, inpainting is evaluated using inpainting pipelines, whereas instruction-based editing is evaluated using img2img pipelines. Both models are tuned for optimal editing performance on each dataset over gs (guidance scale) $\in \{5, 7.5, 10, 12.5, 15\}$, NFE (number of function evaluations) $\in \{10, 20, 30, 40\}$. For SD3.0, we use gs of 7.5 and nfe of 20 for inpainting and gs of 10 nfe of 30 for instruction-based editing, yielding clean-edit CLIP-T scores of 28.46 and 36.26, respectively. For FLUX, we use gs of 10 and nfe of 20 and gs of 10 and nfe of 20, yielding scores of 25.64 and 33.55, respectively. For a fair comparison, all immunization methods use the same generation parameters.

D.1 GuardFlow Settings

The immunizer f_ϕ is a UNet++ (Zhou et al., 2018) operating directly in pixel space, mapping a three-channel input image to a three-channel perturbation at the same spatial resolution. Its encoder consists of 5 stages with filter widths $[32, 64, 128, 256, 512]$; the full network totals 9.2M parameters. All images are processed at a resolution of 512×512.

The spectral norm $\|J_{v_\theta}(x_t, t)\|$, required for ϵ^* derivation, is computed via power iteration using Jacobian–vector products (Miyato et al., 2018), thereby avoiding the explicit formation of the Jacobian matrix. Starting from a random unit vector u_0, we iterate for $k = 0, 1, \ldots$:

$$w_k = J_{v_\theta} u_k, \qquad u_{k+1} = \frac{J_{v_\theta}^\top w_k}{\|J_{v_\theta}^\top w_k\|}, \qquad \sigma^{(k)} = \|J_{v_\theta} u_k\|, \tag{64}$$

where $J_{v_\theta} u$ is a Jacobian–vector product and $J_{v_\theta}^\top w$ a vector–Jacobian product, both obtained through automatic differentiation at the cost of a single forward and backward pass respectively. We run at most 30 iterations and stop early once $|\sigma^{(k)} - \sigma^{(k-1)}|/\sigma^{(k-1)} < 10^{-3}$.

Based on this computation, $\mathbb{E}_{t,x_0}[\frac{1}{t} \cdot \frac{\|v_\theta(x_t,t)\|}{\|J_{v_\theta}(x_t,t)\|}]$ in Eq. (15) is approximated using N timesteps spaced uniformly over $(t^*, 1)$ and S noise samples x_0 per timestep, using the unconditional vector field prediction $v_\theta(\cdot \mid \emptyset)$. The model is kept in evaluation mode, and all computations are performed in `fp32` for a fixed x_0 per power iteration to ensure that the Jacobian measurement remains deterministic.

Since Adam is invariant to a global rescaling of the objective, only the ratios among the three loss weights are meaningful. We therefore fix λ_{traj} and search over the remaining two weights. We conduct a grid search on the training set over $\tau \in \{0, 0.25, 0.5, 0.75\}$, $t^* \in \{0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$, $N \in \{2, 3, 5, 6, 11, 21\}$, $S \in \{1, 2, 4, 8\}$, $\lambda_T \in \{1, 3, 10, 30, 100\}$, $\lambda_B \in \{0.01, 0.03, 0.1, 0.3, 1\}$, and the learning rate in $\{10^{-3}, 10^{-4}, 10^{-5}, 1/255\}$.

Latent-space implementation. SD3.0 and FLUX perform their flow dynamics in the latent space of their respective VAEs. Secs. 3 and 4 use x_t to describe the flow state for notational simplicity. Here we clarify its correspondence to the actual implementation.

Let $\mathcal{E}$ denote the frozen VAE encoder and $z^{\text{src}} = \mathcal{E}(x^{\text{src}})$. The immunizer produces a pixel-space perturbation δ, yielding $\tilde{x} = x^{\text{src}} + \delta$, which is subsequently encoded by the VAE. We define the induced latent displacement as

$$\Delta z = \mathcal{E}(x^{\text{src}} + \delta) - \mathcal{E}(x^{\text{src}}). \tag{65}$$

The perturbed flow trajectory is then

$$\tilde{z}_t = (1-t)z_0 + t\mathcal{E}(x^{\text{src}} + \delta) = z_t^{\text{src}} + t\Delta z, \tag{66}$$

where $z_t^{\text{src}} = (1-t)z_0 + tz^{\text{src}}$. Thus, Eq. (9) is implemented in the latent flow space, with the pixel-space perturbation δ inducing the latent displacement Δz through the VAE encoder.

All vector-field evaluations and the trajectory and terminal objectives are computed in this latent flow space, while gradients are propagated through the frozen VAE encoder to the pixel-space perturbation δ. The Jacobian-free approximation in Sec. 4.1 replaces only the input Jacobian of the

pre-trained vector field,

$$J_{v_\theta}(\tilde{z}_t, t) = \frac{\partial v_\theta(\tilde{z}_t, t \mid \emptyset)}{\partial \tilde{z}_t}, \tag{67}$$

while gradient propagation through the VAE encoder is retained. Accordingly, the Jacobian-free trajectory gradient with respect to the pixel-space perturbation is:

$$\begin{aligned} \nabla_\delta \mathcal{L}_{\text{traj}} &= -2\,\mathbb{E}_{t,z_0}\left[t\, J_{\mathcal{E}}(x^{\text{src}}+\delta)^\top J_{v_\theta}(\tilde{z}_t,t)^\top \left(v_\theta(\tilde{z}_t, t \mid \emptyset) - v_t^{\text{src}}\right)\right] \\ &\approx -2\,\mathbb{E}_{t,z_0}\left[t\, J_{\mathcal{E}}(x^{\text{src}}+\delta)^\top \left(v_\theta(\tilde{z}_t, t \mid \emptyset) - v_t^{\text{src}}\right)\right], \end{aligned} \tag{68}$$

where $J_{\mathcal{E}}(x) = \partial \mathcal{E}(x)/\partial x$, and $v_t^{\text{src}} = z^{\text{src}} - z_0$ is computed from the clean source latent and is independent of δ. The alignment $A(t)$ in Sec. B.4 is evaluated using the exact counterpart of Eq. (68), with $J_{v_\theta}(\tilde{z}_t, t)^\top$ retained.

The terminal regression objective is likewise computed in the latent flow space. Specifically, we use $\hat{z}^{\text{edit}}(\tilde{z}_t) = \tilde{z}_t + (1-t)v_\theta^{\text{cfg}}(\tilde{z}_t, t \mid c)$ as the latent terminal estimate and compute

$$\mathcal{L}_T = \mathbb{E}_t\left[\left\|z^{\text{src}} - \hat{z}^{\text{edit}}(\tilde{z}_t)\right\|_2^2\right], \qquad t \sim \mathcal{U}(t^*, 1). \tag{69}$$

Gradients are propagated through both the vector field and the frozen VAE encoder to the pixel-space perturbation δ. The implicit regularization derived in Sec. C extends accordingly to the latent-space implementation. Under the local approximation $\Delta z \approx J_{\mathcal{E}}(x^{\text{src}})\delta$, the quadratic component with respect to δ is characterized by

$$A_{\text{latent}} = \mathbb{E}_t\left[J_{\mathcal{E}}^\top M_t^\top M_t J_{\mathcal{E}}\right] \succeq 0, \tag{70}$$

where M_t is defined in Eq. (60). Near the terminal endpoint, this regularization approaches $\|J_{\mathcal{E}}\delta\|_2^2$.

We further account for the VAE encoder when computing the perturbation bound in Eq. (15). Under a local linearization of the frozen encoder, $\Delta z \approx J_{\mathcal{E}}(x^{\text{src}})\,\delta$. The corresponding change in the unconditional vector field is approximated as

$$v_\theta(\tilde{z}_t, t \mid \emptyset) - v_\theta(z_t^{\text{src}}, t \mid \emptyset) \approx J_{v_\theta}(z_t^{\text{src}}, t) J_{\mathcal{E}}(x^{\text{src}})\, t\delta. \tag{71}$$

where $J_{v_\theta}(z_t^{\text{src}}, t) = \partial v_\theta(z_t^{\text{src}}, t \mid \emptyset)/\partial z_t^{\text{src}}$. Applying the same argument as in Sec. A with the composite Jacobian $J_{v_\theta}(z_t^{\text{src}}, t)J_{\mathcal{E}}(x^{\text{src}})$ yields the corresponding bound on the pixel-space perturbation $\|\delta\|_2$. Accordingly, for the latent-space implementation, J_{v_θ} in Eqs. (15) and (64) is replaced by $J_{v_\theta}(z_t^{\text{src}}, t)J_{\mathcal{E}}(x^{\text{src}})$. The power iteration in Eq. (64) thus propagates Jacobian–vector products through both the frozen VAE encoder and the vector field. We use the posterior mean of the encoder to make this mapping deterministic. The resulting ϵ^* can therefore be directly compared with $\|\delta\|_2$ in Eq. (16). Note that the Jacobian-free approximation in Sec. 4.1 applies only to the gradient of $\mathcal{L}_{\text{traj}}$, while the computation of ϵ^* uses exact Jacobian–vector products.

D.2 Baseline Settings and Adaptation to Flow Model

Since GuardFlow is the first immunization method designed for flow-based image editing, all baseline methods were originally developed for diffusion-based editing models. We adapt each baseline to the flow-based setting by reformulating its diffusion-specific components under the corresponding flow-matching formulation, while preserving the original optimization objective and overall procedure. For each baseline, we perform a grid search over method-specific hyperparameters, selecting the configuration that achieves the strongest disruption while maintaining a comparable level of imperceptibility.

- **PhotoGuard.** PhotoGuard proposes two immunization strategies: PhotoGuard-E, an encoder attack that perturbs the latent representation of the input image, and PhotoGuard-D, a diffusion attack that directly disrupts the diffusion generation process. We adapt both strategies to the flow-based setting and select PhotoGuard-D based on its stronger editing disruption. The perturbation is optimized using signed-gradient Projected Gradient Descent (PGD) (Madry et al., 2018) within an ℓ_∞ ball, with a step size $a \in \{0.5/255, 1/255, 1.5/255\}$. We search over the number of PGD iterations $\{25, 50, 100\}$.

- **EditShield.** Following the expectation over transformations (EOT) implementation used in EditShield, we apply image transformations including resizing, center cropping, rotation, and Gaussian blurring. For each transformation, we tune the scaling constant $s \in \{0.25, 0.35, 0.5, 0.75, 0.9, 1.0\}$, crop constant $r \in \{1.0, 0.95, 0.90, 0.85, 0.80, 0.70\}$, rotation angle $\theta \in \{0°, 2°, 5°, 10°, 15°, 30°, 45°\}$, and blur parameter $\sigma \in \{0, 0.5, 1.0, 1.5, 2.0, 3.0\}$. We search over the step size $\alpha \in \{0.5, 1.0, 1.5, 3.0\}$ and the number of inner steps $S \in \{1, 2, 3, 5, 10, 15, 20\}$.
- **SDS.** The original paper considers several protection approaches based on either the diffusion-model objective or the score-distillation objective (SDS), and evaluates both gradient-ascent and gradient-descent formulations of these objectives. We adapt these configurations to the flow-based setting and select SDS(-), which applies gradient descent to the score-distillation objective, as it provides the strongest protection among the tested. We replace the score-distillation target with the flow-matching regression target $v^{\mathrm{gt}} = z^{\mathrm{src}} - z_0$ and inject the SDS gradient $v^{\mathrm{cfg}} - v^{\mathrm{gt}}$ directly into the latent. The perturbation is optimized using signed-gradient PGD within an ℓ_∞ ball, with a step size $a \in \{0.5/255, 1/255, 1.5/255\}$ and $n \in \{30, 50, 100, 150\}$ iterations.
- **DiffusionGuard.** We retain the published mask-contour sampling procedure and replace the early-timestep noise-norm objective of diffusion models $-\|\epsilon_\theta\|_2$ with its velocity counterpart of flow models $-\|v_\theta(z_t, t)\|_2$. Gradients are restricted to the masked region. We tune the mask-smoothing parameter $s \in \{0.05, 0.1, 0.5, 1.0\}$, mask perturbation range $\xi \in \{0.5, 1.0, 2.0, 4.0\}$, and the number of mask-augmentation iterations $N \in \{5, 10, 15, 25, 40\}$. We further tune the step size $a \in \{0.5, 1, 2\}/255$ and the number of attack iterations $n \in \{1, 2, 4, 8\}$. Since DiffusionGuard is mask-dependent, we evaluate it only in the inpainting setting.
- **DiffVax.** DiffVax learns a global immunizer rather than a per-image perturbation; therefore, we retrain it end-to-end on the flow backbone. We search over the edit-loss weight $\alpha \in \{1, 2, 4, 8, 16, 32\}$, noise-loss weight $\beta \in \{1, 2, 4, 8\}$, and learning rate $\in \{5 \times 10^{-6}, 10^{-5}, 3 \times 10^{-5}, 10^{-4}\}$. Since DiffVax is mask-dependent, we evaluate it only in the inpainting setting.

E Additional Experimental Results

E.1 Additional Evaluation on FLUX

To further examine model-dependent behavior, we evaluate GuardFlow using FLUX-based image editing. FLUX differs from SD3.0 in its model architecture and generation behavior, providing an additional setting to assess whether the proposed immunization strategy remains effective beyond the primary backbone used in our experiments. Unlike SD3.0, FLUX is guidance-distilled and takes the guidance scale as an input rather than combining separate conditional and unconditional predictions. We therefore approximate $v_\theta(\cdot \mid \emptyset)$ by the distilled prediction under an empty prompt with the same guidance scale used for editing, and use this prompt-free prediction in $\mathcal{L}_{\text{traj}}$ As shown in Tab. 6, GuardFlow consistently achieves strong editing disruption while preserving high perturbation imperceptibility in both inpainting and instruction-based editing settings. These results demonstrate that the effectiveness of GuardFlow is maintained on FLUX and is not restricted to a single model.

Table 6: **Quantitative results for FLUX.** Results are averaged over 400 image-prompt pairs for inpainting setting and 140 image-prompt pairs for instruction-based editing. "–" indicates that the corresponding value is unavailable. (**Bold**: best result, Underlined: second-best) CLIP-T scores for clean-edits are 25.64 for Inpainting and 33.55 for Instruction-based Editing.

Category	Methods	Editing Failure				Imperceptibility		GPU Req.(MB) ↓	Runtime(s) ↓
		SSIM ↓	PSNR ↓	FSIM ↓	CLIP-T ↓	SSIM(Noise) ↑	PSNR(Noise) ↑		
Inpainting (w/ Mask)	Random Noise	$0.694^{\pm.089}$	$18.27^{\pm2.59}$	$0.568^{\pm.072}$	$25.05^{\pm5.22}$	$0.885^{\pm.024}$	**35.26**$^{\pm0.12}$	-	-
	EditShield	$0.644^{\pm.088}$	$17.29^{\pm2.20}$	$0.524^{\pm.066}$	$23.27^{\pm5.34}$	$0.861^{\pm.028}$	$34.23^{\pm.132}$	88784	12.4
	DiffusionGuard	$\underline{0.615}^{\pm.097}$	$\underline{17.07}^{\pm2.35}$	$0.522^{\pm.062}$	$\underline{22.44}^{\pm5.33}$	$0.878^{\pm.021}$	$34.02^{\pm0.56}$	$\underline{81900}$	110.6
	SDS	$0.618^{\pm.078}$	$17.20^{\pm2.02}$	$\underline{0.518}^{\pm.057}$	$23.18^{\pm5.19}$	$\underline{0.887}^{\pm.022}$	$32.66^{\pm0.88}$	93310	74.2
	PhotoGuard	$0.698^{\pm.080}$	$19.67^{\pm2.26}$	$0.562^{\pm.060}$	$22.81^{\pm4.98}$	$0.881^{\pm.015}$	$34.43^{\pm0.71}$	87996	45.4
	DiffVax	$0.687^{\pm.101}$	$18.30^{\pm2.78}$	$0.571^{\pm.084}$	$23.43^{\pm5.53}$	$0.882^{\pm.052}$	$33.08^{\pm2.52}$	91138	$\underline{1.3}$
	Ours	**0.563**$^{\pm.059}$	**14.67**$^{\pm1.66}$	**0.497**$^{\pm.042}$	**21.50**$^{\pm5.24}$	**0.891**$^{\pm.022}$	$\underline{35.01}^{\pm0.036}$	**73174**	**0.009**
Instruction-based editing (w/o Mask)	Random Noise	$0.734^{\pm.136}$	$20.28^{\pm4.69}$	$0.556^{\pm.132}$	$33.45^{\pm4.81}$	$\underline{0.871}^{\pm.043}$	**35.31**$^{\pm0.27}$	-	-
	EditShield	$0.706^{\pm.125}$	$19.97^{\pm3.28}$	$0.540^{\pm.108}$	$32.34^{\pm4.90}$	$0.847^{\pm.051}$	$34.30^{\pm0.30}$	88784	$\underline{12.4}$
	SDS	$\underline{0.556}^{\pm.126}$	$\underline{16.96}^{\pm2.30}$	$\underline{0.452}^{\pm.081}$	$32.57^{\pm4.73}$	$0.845^{\pm.048}$	$33.88^{\pm0.40}$	93276	128.2
	PhotoGuard	$0.686^{\pm.126}$	$19.79^{\pm2.83}$	$0.515^{\pm.084}$	$\underline{32.16}^{\pm4.94}$	$0.861^{\pm.024}$	$33.84^{\pm1.16}$	$\underline{87760}$	45.3
	Ours	**0.513**$^{\pm.107}$	**15.66**$^{\pm1.80}$	**0.403**$^{\pm.046}$	**29.92**$^{\pm5.50}$	**0.885**$^{\pm.042}$	$\underline{34.88}^{\pm1.15}$	**72400**	**0.009**

E.2 Application of GuardFlow to Inverse-Free Image Editing

Most flow-based image editing methods rely on inversion to map the source image onto a generative trajectory before applying the target edit. Recent inverse-free approaches instead construct the editing trajectory directly from the source representation, providing a substantially different editing mechanism from the inversion-based setting considered in our main experiments. To examine whether GuardFlow remains effective under this alternative editing paradigm, we evaluate it with three representative inverse-free methods, FlowEdit (Kulikov et al., 2025), FlowAlign (Kim et al., 2026), and Uni-Inv (Jiao et al., 2026), using both SD3.0 and FLUX. These inverse-free methods require an explicit source prompt in addition to the editing instruction. Since only PIE-Bench++ provides source prompts, we evaluate inverse-free editing using the instruction-based editing setting of PIE-Bench++.

As shown in Table 7, GuardFlow remains effective across all three inverse-free editors for both SD3.0 and FLUX, with consistently low editing-failure metrics across the different editing trajectories. These results indicate that the effectiveness of GuardFlow is not restricted to the inversion-based trajectory used during its standard evaluation and extends to inverse-free flow-based editing.

E.3 Robustness to Diverse Editing Masks

A practical immunization method should remain effective across different editing regions at test time. Although spatial masks are used by the inpainting pipeline, GuardFlow does not rely on them for immunizer optimization. We therefore evaluate its protection under diverse test-time masks targeting the body, upper body, and face, reflecting practical editing scenarios in which users modify only specific regions of an image.

Table 7: **GuardFlow immunization under inverse-free image editing.** All results are evaluated on 140 image-prompt pairs from the PIE-Bench++ under the instruction-based editing setting. The Base rows use the standard inversion-based editor, while FlowEdit, FlowAlign, and Uni-Inv use their corresponding inverse-free editing procedures with symmetric editor parameters for clean and immunized inputs. (**Bold**: best result, Underlined: second-best)

Editing Model	Editing Method	Editing Failure			
		SSIM ↓	PSNR ↓	FSIM ↓	CLIP-T ↓
SD3.0	Base	$0.497^{\pm.111}$	$15.15^{\pm1.84}$	$0.403^{\pm.033}$	$30.50^{\pm5.49}$
	FlowEdit	$0.513^{\pm.135}$	$13.33^{\pm2.60}$	$0.495^{\pm.070}$	$30.29^{\pm4.87}$
	FlowAlign	$0.492^{\pm.129}$	$14.65^{\pm2.95}$	$0.436^{\pm.072}$	$30.87^{\pm4.96}$
	Uni-Inv	$0.490^{\pm.115}$	$15.09^{\pm2.17}$	$0.419^{\pm.059}$	$30.72^{\pm5.16}$
FLUX	Base	$0.513^{\pm.107}$	$15.66^{\pm1.80}$	$0.403^{\pm.046}$	$29.92^{\pm5.50}$
	FlowEdit	$0.485^{\pm.109}$	$12.15^{\pm2.06}$	$0.430^{\pm.059}$	$31.18^{\pm5.09}$
	FlowAlign	$0.533^{\pm.103}$	$13.85^{\pm2.51}$	$0.464^{\pm.062}$	$29.58^{\pm5.54}$
	Uni-Inv	$0.529^{\pm.135}$	$15.40^{\pm2.70}$	$0.443^{\pm.058}$	$30.48^{\pm5.71}$

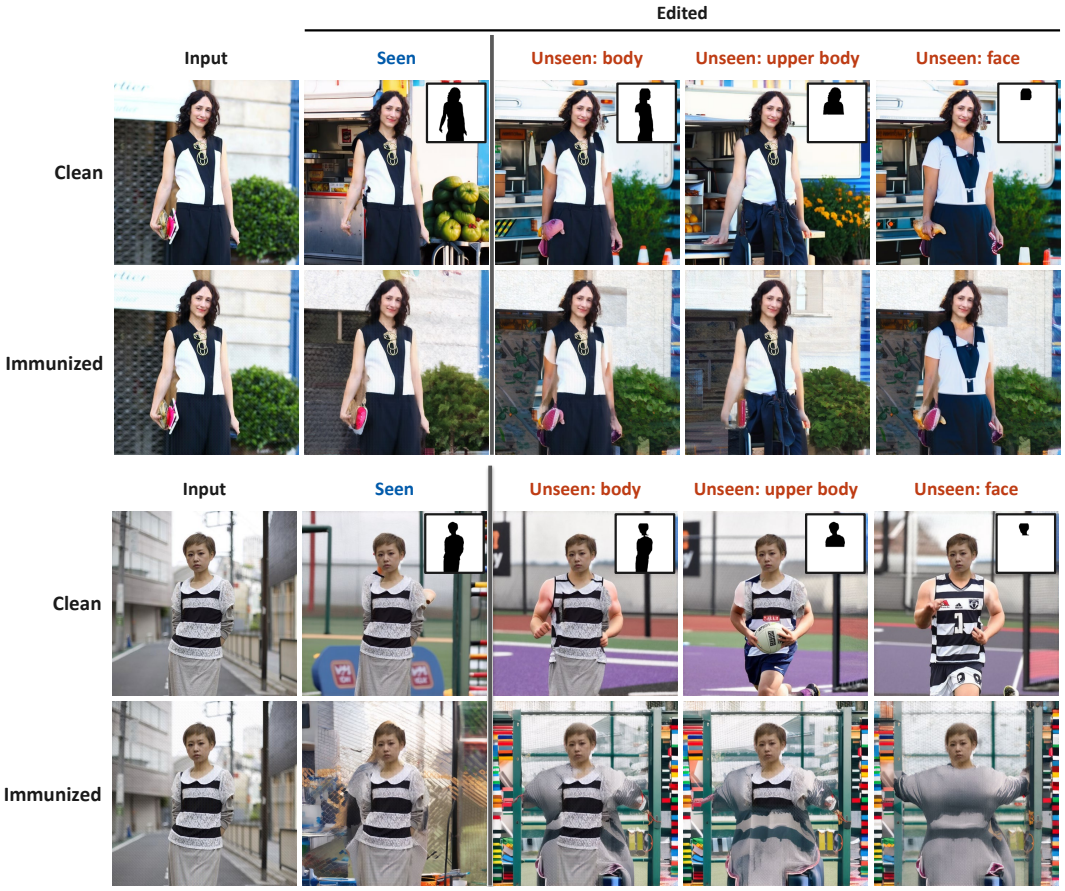


Figure 6: **Generalization to unseen editing masks.** Clean and immunized images are edited using the mask observed during optimization (*Seen*) and previously-unseen masks that target the body, upper body, and face (*Unseen*). The unseen masks represent practical editing scenarios in which only specific regions of the subject are modified. GuardFlow disrupts the intended edits across both seen and unseen editing regions without re-optimizing the immunized image.

Figure 6 shows that GuardFlow remains effective when the editing mask differs from the mask used during optimization. While clean inputs successfully follow the intended edits across different mask regions, the same edits are substantially disrupted for the immunized inputs. The results

demonstrate that GuardFlow is not restricted to the specific mask geometry used during optimization and generalizes to previously unseen editing regions.

E.4 Transferability to Unseen Editing Models

GuardFlow optimizes δ against a specific pre-trained editing model $\mathcal{M}_\theta$ and thus assumes white-box access to its vector field during immunization. This assumption is practical for publicly available editing models, but the model used for subsequent editing may differ from the one used to train the immunizer. We therefore evaluate whether an immunizer trained solely on SD3.0 transfers to unseen editing backbones. Specifically, the resulting immunized images are directly edited using SD3.5-Large and FLUX, without retraining the immunizer or re-optimizing the perturbations.

We consider SD3.5-Large and FLUX to evaluate transfer under different degrees of model shift. SD3.5-Large shares the VAE and latent representation with SD3.0 but uses a larger and modified vector-field architecture, representing a change in the learned flow field within a shared latent space. FLUX, in contrast, differs in both its transformer architecture and latent representation, providing a more challenging cross-representation transfer setting. We evaluate both models under inpainting and instruction-based editing.

Table 8: **Cross-model generalization of GuardFlow.** Immunized images generated by the SD3.0-trained GuardFlow immunizer are directly evaluated on unseen SD3.5 and FLUX editing backbones without retraining or perturbation re-optimization. Results are averaged over 400 image-prompt pairs for inpainting and 140 image-prompt pairs for instruction-based editing.

Cross Transfer	Category	Editing Failure			
		SSIM ↓	PSNR ↓	FSIM ↓	CLIP-T ↓
SD3.0 (source)	Inpainting	$0.489^{\pm.070}$	$9.93^{\pm1.14}$	$0.455^{\pm.040}$	$22.89^{\pm4.67}$
	Instruction-based	$0.497^{\pm.111}$	$15.15^{\pm1.84}$	$0.403^{\pm.033}$	$30.50^{\pm5.49}$
SD3.0 → SD3.5	Inpainting	$0.483^{\pm.076}$	$8.90^{\pm1.48}$	$0.453^{\pm.041}$	$23.96^{\pm5.12}$
	Instruction-based	$0.534^{\pm.113}$	$16.39^{\pm1.91}$	$0.421^{\pm.032}$	$30.39^{\pm5.57}$
SD3.0 → FLUX	Inpainting	$0.665^{\pm.083}$	$17.59^{\pm2.09}$	$0.546^{\pm.058}$	$22.63^{\pm5.14}$
	Instruction-based	$0.474^{\pm.125}$	$13.86^{\pm1.64}$	$0.377^{\pm.044}$	$31.39^{\pm5.22}$

As shown in Table 8, GuardFlow largely preserves its immunization effect when transferring from SD3.0 to SD3.5-Large across both editing settings. Transfer to FLUX is more setting-dependent as the perturbations remain effective under instruction-based editing, while their disruption effect is reduced under inpainting. These results suggest that GuardFlow transfers effectively across models sharing the latent representation used during immunization, while the current immunization paradigm mostly provides less consistent transfer across different latent representations.

E.5 Sensitivity to Editing Hyperparameters

Since GuardFlow is trained under a fixed editing configuration, we examine whether its protection remains effective under different inference-time configurations by varying three key hyperparameters of SD3.0: the guidance scale w, NFE, and the inversion strength t^*. We vary each hyperparameter independently while keeping the others at their default values (Sec. D), using the same immunizer without retraining.

Guidance scale and NFE. As shown in Fig. 7, GuardFlow remains stable across variations in both guidance scale and NFE. The mean CLIP-T varies by less than 1.5 across all configurations on both datasets, compared with reductions of 5.57 on CCP and 5.76 on PIE-Bench++ relative to clean editing under the default configuration. This indicates that the disruption achieved by GuardFlow is largely insensitive to these inference-time hyperparameters.

Inversion strength. The inversion strength t^* determines the portion of the trajectory traversed during editing and therefore interacts with the timestep range $t \sim \mathcal{U}(0.6, 1)$ used to optimize the immunizer. For $t^* < 0.6$, the editing trajectory includes the full optimized range, and GuardFlow largely retains its disruption capability. The degradation observed on PIE-Bench++ is associated with the additional trajectory segment $[t^*, 0.6)$ unseen during training and the reduced contribution

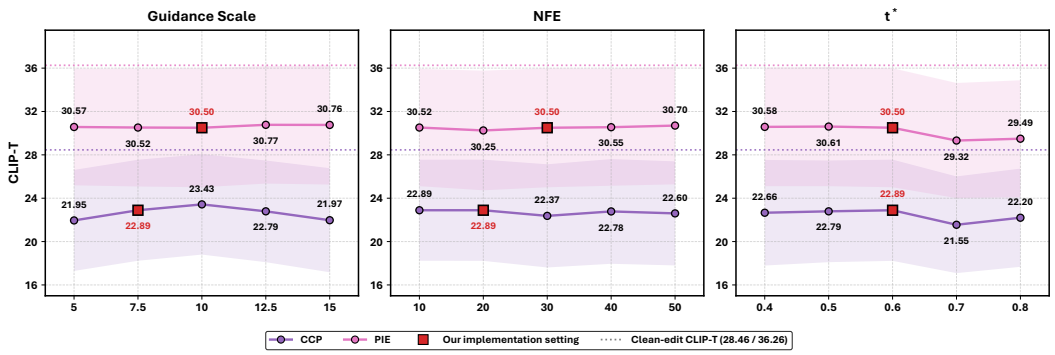


Figure 7: **Sensitivity to editing hyperparameters.** CLIP-T (↓) of immunized edits on SD3.0 under varying guidance scale, NFE, and inversion strength t^*. Each hyperparameter is varied independently while the others remain at their default values. Shaded regions denote ± 1 std., red squares indicate the default configuration, and dotted lines show the clean-edit CLIP-T under the default configuration.

of δ to the initial state ($t^* \Delta z$ in Eq. (65)). For $t^* > 0.6$, a smaller portion of the trajectory is traversed, resulting in weaker edits and correspondingly lower clean-edit CLIP-T. The lower CLIP-T of immunized edits in this regime should therefore be interpreted together with the reduced editability of the underlying pipeline. Overall, GuardFlow remains effective across a broad range of editing configurations without retraining.

F Model Behavior Analysis

F.1 Relationship between ϵ^* and Image Attributes

In Sec. 5.4 and Fig. 5, we showed that ϵ^* is strongly associated with several low-level structural image statistics, with structurally richer images tending to require a larger perturbation lower bound. Here, we extend this analysis to a broader set of sixteen image attributes and provide additional image-level results on CCP and PIE-Bench++.

Fig. 8 reports the full Spearman correlation profile between the image attributes and ϵ^*. The trend observed in the main text remains consistent across the extended attribute set. In both datasets, structural properties show the strongest positive correlations with ϵ^*. Sobel and gradient magnitude each reach correlations of $\rho = 0.72$ and $\rho = 0.71$ on both datasets respectively, while Canny edge density reaches $\rho = 0.61$ on PIE-Bench++ and $\rho = 0.73$ on CCP. Other structure-related quantities, including local contrast, entropy, Laplacian variance, and high-frequency components, also exhibit positive correlations. In contrast, attributes describing global photometric properties show substantially weaker or negative relationships. For example, mean intensity correlates negatively with ϵ^* in both datasets, whereas the relationship between RMS contrast and ϵ^* is negligible on PIE-Bench++. This broader comparison therefore supports the observation in the main text that the variation in ϵ^* is strongly associated with spatial structure.

The image-level relationship with Canny edge density provides a direct view of this tendency. As shown in Fig. 8, images with higher edge density exhibit larger ϵ^* on both PIE-Bench++ and CCP, with Spearman correlations of 0.61 and 0.73, respectively. Together with the results in Fig. 5, these additional analyses consistently indicate that images containing stronger edges and richer local structure have a larger perturbation lower bound. Thus, a larger perturbation magnitude is necessary before the editing trajectory of structurally-rich images can be sufficiently disrupted.



Figure 8: **Additional analysis of the relationship between ϵ^* and image attributes.** *Left:* Image-level relationship between Canny edge density and ϵ^* on PIE-Bench++ and CCP. *Right:* Spearman correlations between ϵ^* and 16 image attributes.

F.2 Validation of the Terminal Estimate

As described in Sec. 4.2, $\mathcal{L}_T$ uses the terminal estimate $\hat{x}^{\text{edit}}(\tilde{x}_t)$ as an efficient surrogate for the final edited image x^{edit}. $\hat{x}^{\text{edit}}(\tilde{x}_t)$ extrapolates an intermediate state $\tilde{x}_t$ along the predicted velocity to the data endpoint at $t = 1$, thereby avoiding backpropagation through the full editing trajectory. Here, we empirically evaluate how closely this estimate corresponds to the actual output x^{edit}.

We run the standard editing pipeline from clean images over the full PIE-Bench++ benchmark and record the states visited by the sampler. For each state, we compute $\hat{x}^{\text{edit}}(\tilde{x}_t)$ and compare it with the final edited image x^{edit} produced by the same run. The terminal estimates progressively capture the semantic content of the intended edit and exhibit increasing structural agreement with the final edited output as the trajectory evolves. Overall, the estimates preserve semantic alignment with the target edit while increasingly recovering its visual and structural characteristics. These results show that $\hat{x}^{\text{edit}}(\tilde{x}_t)$ provides a meaningful approximation of the actual edited output, supporting its use as the terminal representation in $\mathcal{L}_T$.

Figure 9 provides representative examples. Each row shows a clean source image, the terminal estimates $\hat{x}^{\text{edit}}(\tilde{x}_t)$ computed at successive states along the corresponding editing trajectory, and the final edited image. Although estimates from earlier states remain less complete, the target semantic

content emerges progressively along the trajectory, after which the estimates increasingly resemble the final edit.

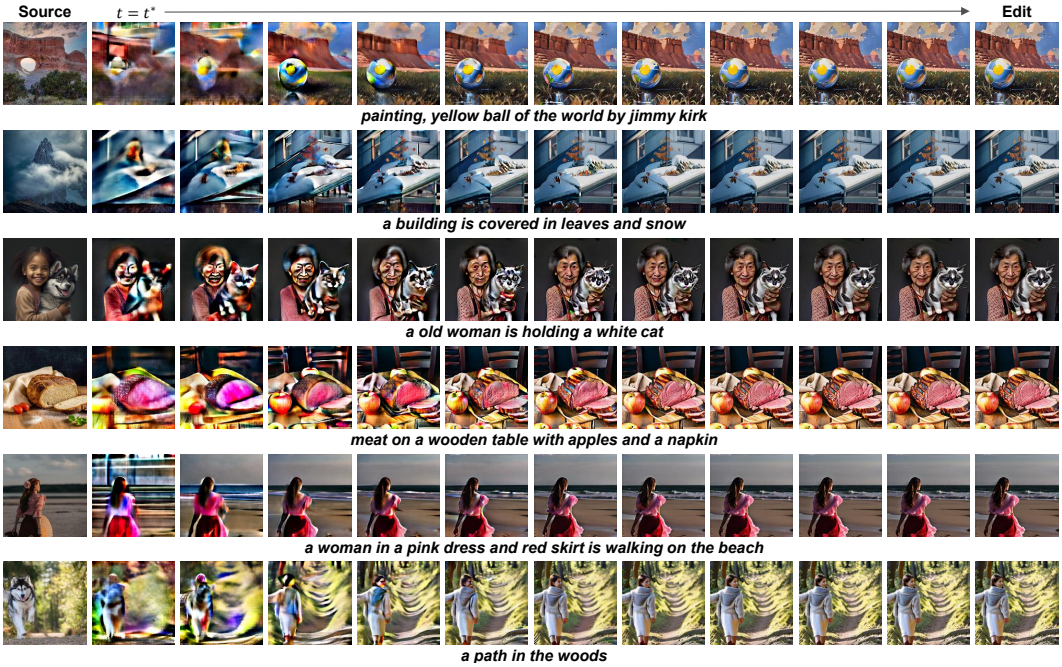


Figure 9: **One-step terminal estimates along clean-image editing trajectories.** Each row shows a clean PIE-Bench++ source image, the terminal estimates $\hat{x}^{\mathrm{edit}}(\tilde{x}_t)$ computed at successive states of the same editing trajectory, and the final edited image x^{edit}. The estimates progressively recover the semantic and visual content of the final edit, with later estimates closely matching the actual output. The prompt and sampling seed are fixed within each row.

G Additional Qualitative Results

We provide additional qualitative results across diverse images, prompts, masks, and editing settings. Specifically, Fig. 11 presents additional results in the inpainting setting on the CCP dataset, covering diverse source images, target prompts, and spatial masks. Fig. 12 provides additional results in the instruction-based editing setting on PIE-Bench++, demonstrating GuardFlow across a wide range of editing instructions and image content. Finally, Fig. 10 presents qualitative comparisons with baseline methods on PIE-Bench++ using the SD3.0 framework, further illustrating the effectiveness of GuardFlow in disrupting intended edits while preserving the visual quality of the source image. These examples complement the quantitative evaluation and further demonstrate the consistency of GuardFlow across diverse editing scenarios.

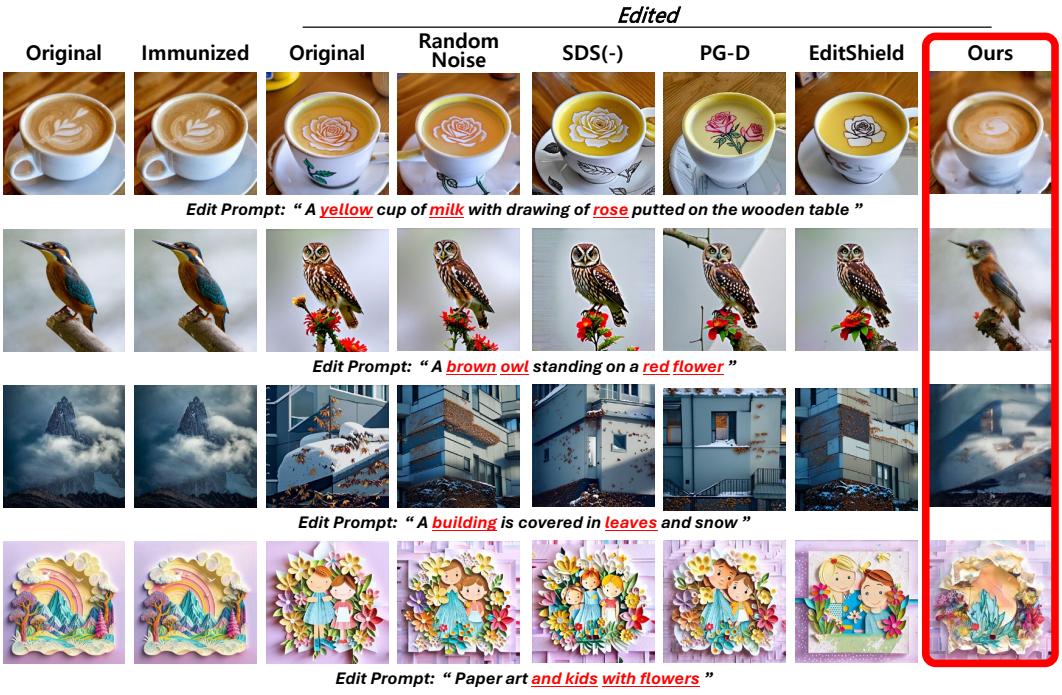


Figure 10: **Qualitative comparisons on PIE-Bench++.** Immunized editing results of GuardFlow and baseline methods using the SD3.0 framework.

H Limitations and Future Work

Despite its consistent performance across the evaluated settings, several aspects of GuardFlow remain to be further explored. Since the perturbation affects the editing trajectory through the noised initial state ($t^*\Delta z$), its influence decreases as t^* approaches zero. At the same time, smaller t^* reduces the dependence of the edit on the protected source image, increasingly departing from our intended immunization scenario. The immunizer is trained at a fixed resolution and on a finite set of editing scenarios, leaving broader generalization beyond the training distribution to be further explored. Our evaluations across diverse visual domains and editing scenarios in Sec. 5.3 and E, however, show that GuardFlow maintains its effectiveness under substantial variations in image content and editing conditions. While GuardFlow is robust to the existing purification and perturbation-removal strategies evaluated in Tab. 2, robustness to future adaptive defenses specifically designed against the immunizer remains an open direction.



Figure 11: **Additional qualitative results on the CCP dataset.** GuardFlow under diverse source images, target prompts, and masks in the inpainting setting.



Figure 12: **Additional qualitative results on PIE-Bench++.** GuardFlow under diverse source images and editing instructions in the instruction-based editing setting.