SPIN: IMAGE IMMUNIZATION AGAINST DIFFUSION EDITING VIA SINGLE-STEP PROJECTION IN STOCHASTIC NEIGHBORHOODS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have greatly advanced instruction-guided image editing, while also raising concerns about unauthorized image manipulation. Image immunization addresses this risk by adding imperceptible perturbations to an input image to disrupt subsequent edits. Since editing requests are unknown at image release, protection should remain effective beyond the instruction used to construct the perturbation. Existing immunization methods either require costly full-trajectory backpropagation or use intermediate objectives whose effects may be weakened by subsequent denoising. Meanwhile, a single inference path provides limited feedback about alternative denoising continuations. To address these challenges, we propose SPIN, a framework for image immunization via one-step projection over local stochastic trajectory neighborhoods. Starting from an early denoising state, SPIN generates stochastic neighboring states under the same instruction and predicts their clean latents through one-step projection without full unrolling. We then optimize a bounded input perturbation to maximize the average deviation of these predictions from a clean-edit reference, encouraging the perturbation to disrupt multiple possible editing outcomes. Experiments on two image editors demonstrate substantial gains in protection performance, with SPIN outperforming compared methods across all six metrics under seen instructions and in the more challenging unseen instruction setting.

1 INTRODUCTION

Instruction-guided diffusion models have made it possible to modify a source image using natural-language instructions while preserving content unrelated to the requested changes (Brooks et al., 2023; Zhao et al., 2024). Publicly available image editing models have substantially lowered the barriers to personalized image manipulation. However, the same accessibility also makes publicly released images vulnerable to unauthorized editing, which may compromise personal privacy and reputation or facilitate misleading content. Deepfake detection (Rossler et al., 2019) and AI-generated image detection (Ojha et al., 2023) can identify manipulated content, but operate only after such content has been produced. This motivates proactive protection that is embedded imperceptibly in images before release.

Image immunization protects source images by adding imperceptible perturbations that disrupt subsequent unauthorized edits. A direct approach optimizes perturbations against the final editing output, but backpropagation through the denoising trajectory incurs substantial computational cost (Salman et al., 2023). More efficient alternatives construct surrogate objectives from denoising predictions or intermediate representations (Liang et al., 2023; Liang & Wu, 2023). Such objectives can target encoder representations (Xue et al., 2024), instruction-related cross-attention (Lo et al., 2024), or early denoising states (Choi et al., 2025), providing protection signals through shorter gradient paths.

Nevertheless, deviations induced by intermediate representations may be attenuated by subsequent denoising, weakening their effect on the final edit. Protection under a fixed editing instruction may also depend on the specific sampling trajectory. This dependence becomes particularly relevant when future editing requests are unknown, as a single trajectory offers limited feedback about other

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.

possible continuations. The challenge is therefore to construct an effective optimization signal that reflects the perturbation's impact on editing outcomes across multiple possible continuations, enabling protection beyond the original editing instruction.

We address this challenge by proposing SPIN (Single-step Projection in Stochastic Neighborhoods), an image-immunization framework that uses predicted editing outcomes to guide protective perturbations. Single-step projection predicts a clean latent from an early denoising state without completing the sampling trajectory. The discrepancy between this prediction and a fixed clean-edit reference provides an effective surrogate for optimizing perturbations that disrupt the final editing output. To extend this signal beyond a single inference path, SPIN generates neighboring candidate states through stochastic transitions from a shared early state. We then optimize a bounded input perturbation to maximize the average discrepancy between their projected outcomes and the reference, reducing dependence on a single construction path and promoting generalization to unseen editing instructions.

Experiments on SD3 and InstructPix2Pix show that SPIN outperforms all baselines across six metrics under both seen and unseen editing instructions. Immunization success rates reach 90% and 94% under seen instructions (85% and 91% under unseen instructions), consistently exceeding those of competing methods. Ablations demonstrate that single-step projection provides an effective protection signal, while stochastic neighborhood sampling further improves generalization over single-candidate projection.

Our main contributions are summarized as follows:

- We propose SPIN, an image-immunization framework that uses single-step projection to guide perturbation optimization with predicted editing outcomes, providing an effective signal for disrupting the final edit.
- We introduce stochastic neighborhood optimization to construct a single perturbation that targets multiple possible continuations of the editing process. By driving their projected outcomes away from the clean-edit reference, this strategy promotes disruption across different inference paths and improves generalization to unseen editing instructions.
- Experiments under standard editing pipelines show that SPIN consistently outperforms the evaluated baselines in immunization success rate under both seen and unseen instructions, across flow-matching and noise-prediction diffusion editors.

2 Related Work

2.1 Instruction-Guided Diffusion Image Editing

Diffusion models support image editing by combining a generative prior with a source image and user-specified control. SDEdit (Meng et al., 2022) perturbs and denoises an input through a stochastic differential equation to balance faithfulness and realism. Later methods introduced more explicit semantic control. Prompt-to-Prompt (Hertz et al., 2023) manipulates cross-attention maps to preserve spatial layout while changing textual content, whereas Imagic (Kawar et al., 2023) optimizes a text embedding and fine-tunes the model for semantic edits on a single image. InstructPix2Pix (Brooks et al., 2023) establishes a supervised learning paradigm for directly following natural-language editing instructions without per-image inversion or fine-tuning. UltraEdit (Zhao et al., 2024) advances this paradigm through large-scale editing supervision with diverse instructions, real-image anchors, and region annotations, supporting the adaptation of pretrained generative models, including Stable Diffusion 3 (Esser et al., 2024), to instruction-guided image editing. These advances improve the capability and accessibility of image editing, while also exposing publicly released images to increasingly diverse unauthorized transformations.

2.2 Adversarial Examples

Norm-bounded adversarial perturbations can alter model predictions (Goodfellow et al., 2015), and projected-gradient methods (Madry et al., 2018) formulate their construction as constrained optimization. To improve cross-model transferability, MI-FGSM (Dong et al., 2018) stabilizes updates through momentum, while VMI-FGSM (Wang & He, 2021) further refines them using neighborhood

gradient information. EOT (Athalye et al., 2018) averages attack objectives over input transformations to improve transformation robustness, whereas universal adversarial perturbations generalize across images (Moosavi-Dezfooli et al., 2017). SPIN targets the generation trajectories of diffusion editors, optimizing a bounded input perturbation over local stochastic continuations to disrupt generation and drive final editing outputs away from their clean counterparts.

2.3 Image Immunization Against Diffusion Editing

Image immunization methods mainly differ in the optimization signal used to construct protective perturbations. PhotoGuard exposes the central trade-off through an efficient encoder-targeting objective (PhotoGuard-E) and a more output-related but expensive full-diffusion objective (PhotoGuard-D) (Salman et al., 2023). AdvDM optimizes the diffusion denoising loss, while MIST combines semantic and textural objectives to disrupt the generation process at multiple representation levels (Liang et al., 2023; Liang & Wu, 2023). Subsequent methods shorten the gradient path by targeting specific internal components: SDS (Xue et al., 2024) applies score distillation at the vulnerable VAE bottleneck, EditShield (Chen et al., 2024) shifts instruction-conditioned latent representations, and SA (Lo et al., 2024) disrupts prompt-relevant cross-attention features. DiffusionGuard (Choi et al., 2025) places its objective at early denoising stages. Other approaches improve complementary aspects of protection: DCT-Shield (Bala et al., 2025) optimizes frequency-domain perturbations for visual fidelity and JPEG resistance, whereas SIFM (Dong et al., 2025) combines intermediate-feature manipulation with semantic and perceptual criteria for evaluating edit failure. Despite their different targets, these methods either require costly differentiation through the diffusion trajectory or rely on internal surrogates whose deviations may be attenuated during subsequent denoising.

3 Method

We present an image-immunization framework, termed Single-step Projection in Stochastic Neighborhoods (SPIN), for protecting images against unauthorized editing. To efficiently obtain a protection signal that reflects the perturbation's effect on the predicted edit, we use single-step projection to estimate the clean latent from an intermediate denoising state. We further optimize the perturbation over a local stochastic neighborhood to reduce dependence on a single inference path and promote generalization across editing instructions.

3.1 Preliminaries and Problem Formulation

Given a source image $x \in [0,1]^{H\times W\times 3}$ and an editing instruction c, a frozen editor $\mathcal{G}_\theta$ produces an edited image $y = \mathcal{G}_\theta(x, c; \omega)$, where ω denotes the sampling randomness. The editor iteratively denoises a latent state z_t conditioned on x and c, then decodes the final latent into y.

Image immunization seeks a bounded input perturbation that disrupts subsequent edits while preserving the source image's appearance. The protected image is $x_\delta = \Pi_{[0,1]}(x+\delta)$, where $\Pi_{[0,1]}$ clips pixel values to the valid range. For a given instruction c and sampling randomness ω, we formulate the protection objective as:

$$\delta^\star = \underset{\|\delta\|_\infty \le \varepsilon}{\arg\max}\, \mathcal{L}\left(\mathcal{G}_\theta(x_\delta, c; \omega), \mathcal{G}_\theta(x, c; \omega)\right), \tag{1}$$

where ε limits the perturbation magnitude to preserve input fidelity, and $\mathcal{L}$ measures the discrepancy between protected and clean editing outputs. Only δ is optimized, with all model parameters frozen. The perturbation remains fixed during evaluation under instructions and sampling randomness not used in optimization.

3.2 Overall Framework

Our goal is to construct a bounded input perturbation that disrupts the final editing output and remains effective under unseen editing instructions. To this end, SPIN combines two modules, as shown in Figure 1: (a) local trajectory sampling generates neighboring candidate states from a shared early state under a fixed construction instruction, providing feedback across multiple local continuations to reduce path dependence and support transfer to unseen instructions; (b) one-step

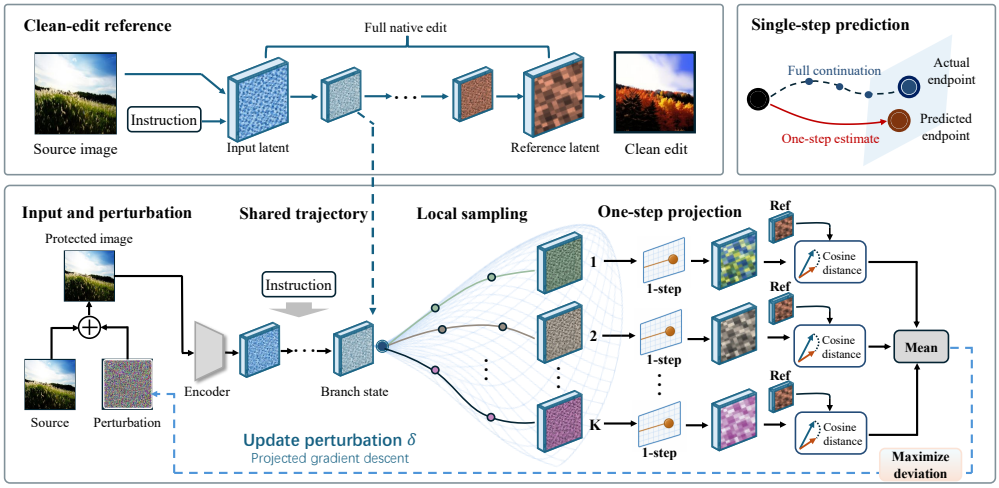


Figure 1: Overview of SPIN. From a shared native prefix, local trajectory sampling generates neighboring candidates and one-step projection estimates their clean endpoints. These estimates guide perturbation optimization to drive editing outputs away from the fixed clean-edit reference.

projection estimates each candidate's clean latent, enabling output-oriented evaluation without full sampling rollouts. We aggregate the projected candidates' deviations from a fixed clean-edit reference to optimize the same input perturbation. The following subsections detail the projection, neighborhood construction, and perturbation optimization.

3.3 Outcome-Oriented Single-Step Projection

To evaluate local continuations without completing each sampling rollout, we use single-step projection to predict the clean latent from an intermediate state. For a noise-prediction model, consider $z_t = a_t z_0 + b_t \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. For a flow-matching model with a linear probability path, $z_t = (1-t)z_0 + t\epsilon$. The corresponding projections are

$$\begin{aligned} \mathcal{P}_t^{\epsilon}(z_t; x_\delta, c) &= \frac{z_t - b_t \widehat{\epsilon}_\theta(z_t, t \mid E(x_\delta), c)}{a_t}, \\ \mathcal{P}_t^{v}(z_t; x_\delta, c) &= z_t - t\widehat{v}_\theta(z_t, t \mid E(x_\delta), c). \end{aligned} \tag{2}$$

Here, $\widehat{\epsilon}_\theta$ and $\widehat{v}_\theta$ denote the editor's predictions, including guidance operations. Each projection uses the prediction at the candidate state and bypasses the remaining sampling steps.

For population-optimal conditional noise or velocity predictors trained with squared error, the unguided projection equals the conditional mean of the clean latent. In practice, we apply Equation 2 to guided predictions. To evaluate intermediate states, SVDD-PM (Li et al., 2025b) approximates their soft values using rewards computed on predicted clean samples. Following this look-ahead evaluation principle, we use the projected latent's deviation from a clean-edit reference as a differentiable protection surrogate.

Specifically, we obtain the reference latent z_{ref} by running the complete native editor once on the clean input under instruction c. The reference is held fixed throughout perturbation optimization, and the discrepancy is computed in the shared clean-latent space.

Projection placement determines the prefix cost and the endpoint information available for evaluation. SuperEdit (Li et al., 2025a) observes that early sampling stages primarily affect global layout in InstructPix2Pix, while later stages refine local attributes and image details. This motivates using early projections to guide perturbation optimization toward disrupting the overall editing structure while keeping the sampling prefix short. The next subsection describes how we construct local candidates and aggregate their projected outcomes.

3.4 Optimization over Stochastic Trajectory Neighborhoods

Different editing instructions induce different denoising trajectories. Optimizing a protective perturbation along a single construction trajectory may cause it to overfit to that particular path. We therefore optimize over a stochastic neighborhood of local continuations, aiming to reduce trajectory-specific overfitting and improve transfer to unseen instructions.

Let j index the branch position in sampling order, and let $\mathcal{F}_{0:j}$ denote the native sampling prefix. For the current protected input, we compute a shared intermediate state, generate K candidates through a model-specific stochastic transition $\mathcal{T}_j$, and project each candidate:

$$\begin{gathered} z_j(\delta) = \mathcal{F}_{0:j}(x_\delta, c; \omega_{<j}), \qquad z_{j+1}^{(k)} = \mathcal{T}_j\big(z_j(\delta), \xi_k; x_\delta, c\big), \\ \widetilde{z}_0^{(k)} = \mathcal{P}_{t_{j+1}}\big(z_{j+1}^{(k)}; x_\delta, c\big). \end{gathered} \tag{3}$$

Here, $\xi_k \sim \mathcal{N}(0, I)$ are independently sampled branch noises for $k = 1, \ldots, K$, and t_{j+1} is the sampling time of the candidate states. Intuitively, these branches represent alternative ways the same early editing state could evolve. Recomputing the model prediction at each candidate yields a projected clean latent, allowing us to consider how the perturbation affects several possible editing outcomes without completing their denoising trajectories.

We maximize the expected cosine distance between candidate projections and the fixed clean-edit reference z_{ref}, approximated by a finite candidate average:

$$\begin{aligned} J(\delta) &= \mathbb{E}_\xi \left[d_{\cos}\big(\widetilde{z}_0(\delta, \xi), z_{\mathrm{ref}}\big) \right], \\ \widehat{J}_K(\delta) &= \frac{1}{K} \sum_{k=1}^{K} d_{\cos}\big(\widetilde{z}_0^{(k)}, z_{\mathrm{ref}}\big). \end{aligned} \tag{4}$$

The cosine distance is defined as $d_{\cos}(a, b) = 1 - \langle a, b \rangle / (\|a\|_2 \|b\|_2)$. This objective combines feedback from multiple local continuations to update the same input perturbation, encouraging a larger average predicted deviation across the sampled neighborhoods.

The candidate count K controls the number of samples used to approximate the expectation, while the stochastic transition scale determines the neighborhood's extent. We sample the candidate noises once and reuse them throughout optimization, yielding a fixed sample-average objective. The optimized perturbation is then held fixed and evaluated under unseen instructions using the complete native editing pipeline.

3.5 Instantiating the Neighborhood in Different Editors

SPIN adapts to flow-matching and noise-prediction editors through the transition $\mathcal{T}_j$, while retaining the same projection objective and perturbation optimization. For flow-matching editors such as SD3, we adopt the stochastic transition used in Flow-grpo (Liu et al., 2026) and Flow-TTRL (Chen et al., 2026). Let $\Delta t_j = t_{j+1} - t_j < 0$ and $v_j = \widehat{v}_\theta(z_j, t_j \mid E(x_\delta), c)$. We construct local candidates by adding Gaussian noise with a corresponding drift correction:

$$z_{j+1}^{(k)} = z_j + \left[v_j + \frac{g_j^2}{2t_j}\big(z_j + (1 - t_j) v_j\big) \right] \Delta t_j + g_j \sqrt{-\Delta t_j}\, \xi_k, \qquad g_j = \lambda \sqrt{\frac{t_j}{1 - t_j}}. \tag{5}$$

Here, $\xi_k \sim \mathcal{N}(0, I)$ and $\lambda \geq 0$ controls the stochasticity. Setting $\lambda = 0$ recovers the native Flow-Match Euler update. The drift correction follows from the velocity–score relation, with its derivation and endpoint handling provided in Appendix C.

For noise-prediction editors such as InstructPix2Pix (Brooks et al., 2023), the native Euler-Ancestral sampler already provides a stochastic transition. With clean-latent estimate $\widehat{z}_{0,j} = z_j - \sigma_j \widehat{\epsilon}_j$, we sample candidates as

$$z_{j+1}^{(k)} = z_j + \frac{z_j - \widehat{z}_{0,j}}{\sigma_j}\big(\sigma_{\mathrm{down},j} - \sigma_j\big) + \tau \sigma_{\mathrm{up},j} \xi_k. \tag{6}$$

The coefficients $\sigma_{\mathrm{down},j}$ and $\sigma_{\mathrm{up},j}$ are determined by the native scheduler. The temperature $\tau \geq 0$ controls the neighborhood scale, with $\tau = 1$ recovering the native transition. In both instantiations,

candidates share the sampling prefix and differ only in their branch noise. We recompute the model prediction at each candidate state and apply the corresponding single-step projection in Equation 2. These transitions construct the optimization neighborhood.

3.6 Perturbation Optimization

We reparameterize each candidate transition by adding a Gaussian noise term to its conditional mean. Holding the sampled noise fixed during backpropagation allows us to differentiate the projection loss with respect to the input perturbation through the candidate prediction, local transition, shared prefix, and image encoder. Throughout optimization, model parameters are frozen, and the clean reference z_{ref} is treated as a constant.

We initialize the perturbation at zero and maximize Equation 4 using projected gradient ascent (Madry et al., 2018). Define the feasible set as $\mathcal{C}(x) = \{\delta : \|\delta\|_\infty \le \varepsilon,\ x+\delta \in [0,1]^{H\times W\times 3}\}$. Each update is computed by:

$$\delta^{(n+1)} = \Pi_{\mathcal{C}(x)}\left[\delta^{(n)} + \eta\ \mathrm{sign}\left(\nabla_\delta \widehat{J}_K(\delta)\Big|_{\delta=\delta^{(n)}}\right)\right], \tag{7}$$

where η is the step size. At each iteration, we recompute the shared prefix for the current protected input, generate the candidates, aggregate their projection distances, and update the perturbation within the feasible set. After M iterations, we return $x_{\mathrm{adv}} = x + \delta^{(M)}$. The perturbation budget, step size, and iteration count are specified in the experimental setup.

4 Experiments

4.1 Setup and Evaluation Protocol

Benchmark Datasets. We construct a benchmark of 500 samples from Pico-Banana, covering landscapes, artworks, multi-object scenes, real-world photographs, news imagery, public figures, and indoor scenes. For text-conditioned protection methods, perturbations are optimized using the original prompt associated with each image. Each sample is evaluated with its original prompt and five semantically disjoint editing instructions, resulting in 3,000 image-instruction pairs. Additional dataset and instruction details are provided in Appendix A.1.

Editing Models and Evaluation Metrics. We evaluate InstructPix2Pix and Stable Diffusion 3 (SD3) as denoising-diffusion and flow-matching editors, respectively. We compare edits of clean and protected inputs using PSNR (Wang & Bovik, 2009), SSIM (Wang et al., 2004), FSIM (Zhang et al., 2011), VIFP (Sheikh & Bovik, 2006), and LPIPS (Zhang et al., 2018). Lower values of the first four and higher LPIPS indicate stronger protection; the directions reverse for perturbation imperceptibility. Following SIFM (Dong et al., 2025), we report Immunization Success Rate (ISR): the fraction of protected edits with instruction failure or substantial perceptual degradation, judged by consensus of Kimi K2.5 (Team et al., 2026) and Qwen3-VL (Bai et al., 2025). Higher ISR indicates stronger protection. See Appendices A.2 and D for evaluation details.

Implementation Details. For each image, we optimize an additive perturbation δ under $\|\delta\|_\infty \le 8/255$ while keeping the model parameters frozen. Starting from $\delta = 0$, we perform 20 PGD steps with a step size of $1/255$, projecting the perturbation and clipping the protected image after each step. We use branch index $j = 1$ for both editors and set $\tau = 2$ for InstructPix2Pix. Sampling noise is fixed during optimization. Evaluation uses unseen seeds, with clean and protected inputs sharing the same editing instruction and sampling noise. Further implementation details are provided in Appendix A.3.

4.2 Main Results

Comparison with State-of-the-Art Methods. We first compare SPIN with representative image-immunization methods, PhotoGuard-E/PhotoGuard-D (Salman et al., 2023), SDS (Xue et al., 2024), ED (Chen et al., 2024), AdvDM (Liang et al., 2023), MIST (Liang & Wu, 2023), SA (Lo et al., 2024), DCT (Bala et al., 2025), and SIFM (Dong et al., 2025) on SD3 and InstructPix2Pix under the instructions used to construct the protected inputs. Table 1 shows that SPIN achieves the greatest output discrepancy and highest immunization success rate on both editors. Compared with objectives defined on intermediate features or noisy states, our projection objective targets deviations in

predicted editing outcomes, aligning perturbation optimization more closely with final-edit disruption and thereby improving immunization performance.

Table 1: Comparison with image-immunization baselines under original instructions. Lower PSNR/SSIM/VIFp/FSIM and higher LPIPS/ISR indicate better immunization performance.

Method	SD3						InstructPix2Pix					
	PSNR ↓	SSIM ↓	VIFp ↓	FSIM ↓	LPIPS ↑	ISR ↑	PSNR ↓	SSIM ↓	VIFp ↓	FSIM ↓	LPIPS ↑	ISR ↑
PhotoGuard-E	18.96	0.6689	0.2134	0.7890	0.3250	58%	13.93	0.4680	0.1010	0.6919	0.5585	83%
PhotoGuard-D	17.16	0.5668	0.1873	0.7682	0.3946	66%	13.16	0.4042	0.1087	0.7053	0.5094	83%
SDS	17.92	0.5868	0.1878	0.7591	0.3798	64%	14.82	0.4704	0.1368	0.7330	0.4703	86%
ED	19.88	0.6530	0.2419	0.8084	0.3176	60%	16.06	0.5098	0.1538	0.7652	0.4199	77%
AdvDM	13.54	0.4519	0.1059	0.6600	0.5391	84%	15.41	0.4839	0.1473	0.7634	0.4388	85%
MIST	14.09	0.4664	0.1189	0.6808	0.5251	56%	15.44	0.4560	0.1219	0.7418	0.5060	86%
SA	17.56	0.5875	0.1850	0.7577	0.4535	68%	16.69	0.5009	0.1411	0.7718	0.4970	86%
DCT	17.25	0.5506	0.1671	0.7591	0.4206	70%	18.11	0.5765	0.1852	0.8107	0.4244	77%
SIFM	15.92	0.4945	0.1468	0.7027	0.4988	82%	15.02	0.4818	0.1346	0.7568	0.5129	91%
SPIN	**11.20**	**0.3450**	**0.0677**	**0.5845**	**0.6730**	**90%**	**10.93**	**0.3072**	**0.0649**	**0.6188**	**0.6974**	**94%**

Table 2: Generalization to five unseen instructions. Perturbations are optimized with the original instruction; Lower PSNR/SSIM/VIFp/FSIM and higher LPIPS/ISR indicate better immunization performance.

Method	SD3						InstructPix2Pix					
	PSNR ↓	SSIM ↓	VIFp ↓	FSIM ↓	LPIPS ↑	ISR ↑	PSNR ↓	SSIM ↓	VIFp ↓	FSIM ↓	LPIPS ↑	ISR ↑
PhotoGuard-E	19.80	0.6928	0.2397	0.8109	0.3007	67%	14.37	0.4884	0.1101	0.7076	0.5486	82%
PhotoGuard-D	20.15	0.7039	0.2471	0.8146	0.2988	64%	16.76	0.4974	0.1565	0.7855	0.4493	81%
SDS	18.79	0.6202	0.2147	0.7843	0.3549	69%	15.54	0.5068	0.1545	0.7536	0.4463	84%
ED	20.64	0.6789	0.2673	0.8305	0.2954	50%	17.10	0.5482	0.1770	0.7951	0.3833	79%
AdvDM	14.36	0.4775	0.1182	0.6834	0.5216	68%	16.65	0.5176	0.1665	0.7833	0.4179	82%
MIST	15.02	0.4942	0.1355	0.7063	0.5070	66%	16.00	0.4745	0.1324	0.7565	0.4940	83%
SA	18.73	0.5616	0.1762	0.7581	0.4662	58%	16.48	0.5231	0.1628	0.7692	0.4805	82%
DCT	18.59	0.5886	0.1981	0.7679	0.4259	67%	18.40	0.6198	0.2275	0.8245	0.3772	80%
SIFM	17.81	0.5408	0.1634	0.7403	0.4854	69%	15.62	0.5038	0.1557	0.7532	0.4889	85%
SPIN	**13.95**	**0.4396**	**0.1024**	**0.6486**	**0.5704**	**85%**	**14.13**	**0.4208**	**0.1044**	**0.6990**	**0.5543**	**91%**

Generalization to Unseen Editing Instructions. We evaluate protection on semantically disjoint instructions excluded from perturbation optimization. Table 2 shows that SPIN achieves ISR of 85% on SD3 and 91% on InstructPix2Pix, with a larger advantage over competing methods than under original instructions. This advantage is consistent with its optimization across multiple stochastic trajectories, which reduces dependence on a single editing path and helps protection remain effective when the instruction changes.

Qualitative Comparison. Figures 2 and 3 compare source images, protected inputs and editing results. Many baseline methods mainly introduce texture-like patterns or slight color shifts, while the requested edits remain largely intact. In contrast, SPIN causes substantial changes to the main subject or scene structure beyond the requested edits under both original and unseen instructions. More examples appear in Appendix E.

Input Imperceptibility and Computational Efficiency. Protection quality also depends on input fidelity and perturbation-generation cost. We measure fidelity between clean and protected images separately from output discrepancy. All efficiency measurements are conducted on an NVIDIA GeForce RTX 5090 GPU, using the same input resolution and optimization budgets across methods. As shown in Table 3, SPIN achieves better perturbation imperceptibility on most reported metrics and the highest immunization success rate, with a reasonable generation time per image.

Table 3: Input imperceptibility, immunization success, and time cost on SD3.

Method	PSNR ↑	SSIM ↑	VIFp ↑	FSIM ↑	LPIPS ↓	ISR ↑	Time/image (s) ↓
SDS	32.85	0.8439	0.5406	0.9700	0.1584	64%	10.7
PhotoGuard-E	31.71	0.8635	0.5405	0.9801	**0.0931**	58%	8.4
PhotoGuard-D	33.24	0.8473	0.5412	0.9770	0.1835	66%	311.0
AdvDM	33.69	0.8511	0.5540	0.9758	0.1564	84%	14.9
MIST	33.23	0.8433	0.5496	0.9753	0.1611	56%	25.6
SIFM	34.75	0.8836	0.5542	0.9806	0.1279	82%	19.3
SPIN	**35.44**	**0.8950**	**0.6129**	**0.9812**	0.1187	**90%**	16.0

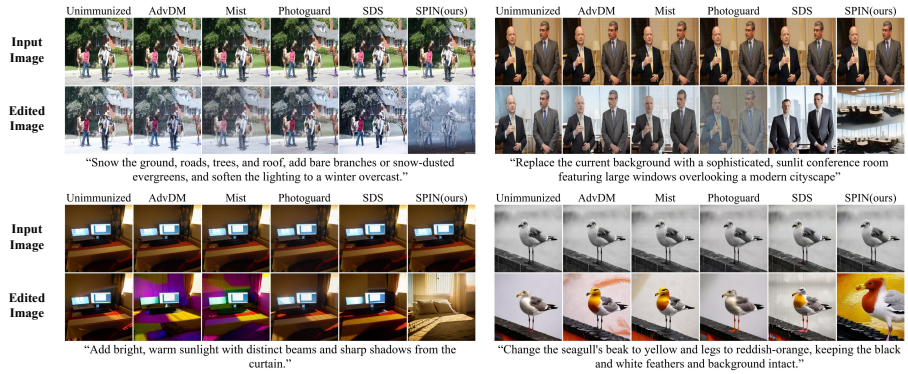


Figure 2: Qualitative comparisons under seen editing instructions. The first and second rows show examples from InstructPix2Pix and SD3, respectively.



Figure 3: Qualitative comparison under unseen editing instructions. The left and right examples use SD3 and IP2P, respectively, with editing instructions not used during protection optimization.

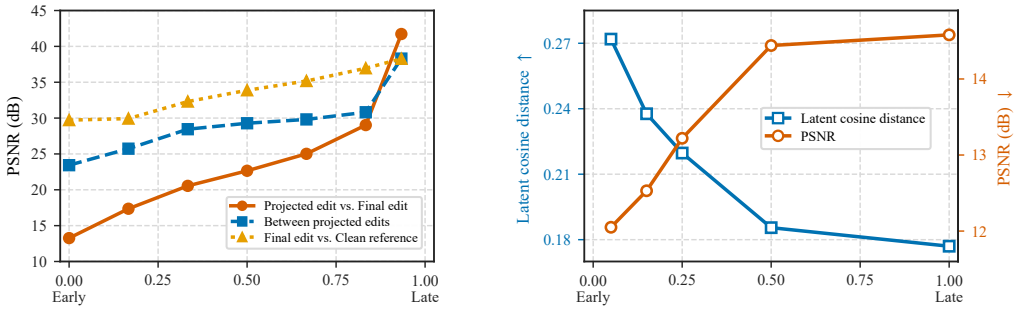
4.3 Ablation Studies

Effect of the Projection and Neighborhood Objectives. We measure latent discrepancy using the cosine distance between flattened latent representations. Table 4 reports this distance between the final latents of clean and protected edits after full denoising, alongside output PSNR and LPIPS. Under the same perturbation budget, one-step projection yields substantially larger latent cosine distances, lower PSNR, and higher LPIPS than random perturbations, demonstrating its effectiveness as a surrogate for disrupting final editing outputs. Compared with single-candidate projection, SPIN ($K = 4$) further improves all three metrics on both editors under the same PGD budget, demonstrating the benefit of aggregating feedback across stochastic local continuations.

Table 4: Effect of one-step projection and stochastic neighborhood aggregation on protection efficacy. The deterministic SD3 variant uses the native deterministic transition, while stochastic variants use one or four candidates.

Variant	Latent cosine dist. ↑	PSNR ↓	LPIPS ↑
SD3			
Random perturbation	0.0623	20.78	0.280
Deterministic one-step projection	0.2782	11.70	0.629
Stochastic one-step projection ($K = 1$)	0.2814	11.95	0.653
SPIN ($K = 4$)	**0.3141**	**11.20**	**0.673**
InstructPix2Pix			
Random perturbation	0.1012	19.84	0.195
Stochastic one-step projection ($K = 1$)	0.7327	11.24	0.685
SPIN ($K = 4$)	**0.7637**	**10.93**	**0.697**

Effect of Branch Position and Guidance Timing. We vary the branch position to examine candidate diversity and final-output deviation, and separately vary the timing of projection-guided latent updates that steer the trajectory away from the clean edit. Figure 4(a) shows that later branching improves projection accuracy but reduces variation among candidate projections and their final deviations from the clean edit. Figure 4(b) further shows that earlier latent updates produce larger final latent cosine distances and lower output PSNR, indicating stronger trajectory disruption. These observations motivate early branching and projection in SPIN.



(a) Candidate diversity and deviation (b) Output deviation under guidance

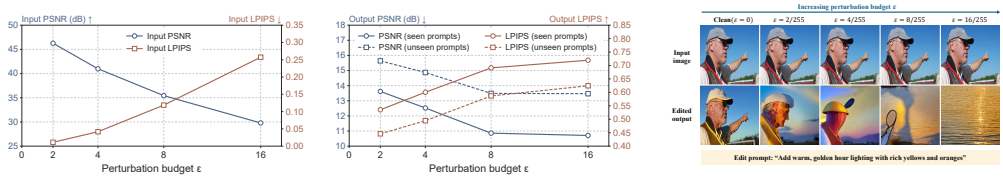
Figure 4: Effect of normalized sampling progress. (a) We report the PSNR between a candidate's one-step projected edit and its fully denoised final edit, between projected edits from different candidates, and between the final edit and a fixed clean reference. Lower PSNR indicates a larger difference. (b) Latent cosine distance and PSNR are both computed with respect to the clean edit; higher latent cosine distance and lower PSNR indicate a larger deviation.

Effect of the Number of Candidate Trajectories. We evaluate different numbers of candidate trajectories, $K \in \{1, 2, 4, 8\}$. As shown in Table 5, increasing K from 1 to 4 improves protection under both original and unseen instructions across the two editors. Increasing K to 8 does not yield consistent performance improvement while substantially increasing generation time. We therefore use $K = 4$ to balance protection effectiveness and computational cost.

Table 5: Effect of the number of candidate trajectories at a fixed editor-specific neighborhood scale.

Editor	K	Seen PSNR ↓	Seen LPIPS ↑	Unseen PSNR ↓	Unseen LPIPS ↑	Time/image (s) ↓
SD3	1	11.95	0.6532	14.57	0.5702	**10.51**
SD3	2	10.99	0.6424	14.01	0.5718	12.65
SD3	4	11.20	**0.6730**	**13.95**	0.5704	16.01
SD3	8	**10.94**	0.6479	13.96	**0.5828**	27.11
InstructPix2Pix	1	11.24	0.6848	15.51	0.5147	**5.61**
InstructPix2Pix	2	10.88	0.6916	14.36	0.5316	7.40
InstructPix2Pix	4	10.93	0.6974	**14.13**	**0.5543**	10.58
InstructPix2Pix	8	**10.41**	**0.7052**	14.53	0.5531	20.97

Effect of the Perturbation Budget. We vary the perturbation budget ε while fixing the optimization budget and editor-specific settings. Figure 5 shows that larger perturbation budgets strengthen protection but reduce input fidelity. Increasing ε beyond $8/255$ yields limited additional output discrepancy while introducing more visible input distortion, motivating our choice of $\varepsilon = 8/255$.



(a) PSNR trade-off **(b)** Output discrepancy **(c)** Qualitative results

Figure 5: Effect of the perturbation budget ε: (a) input fidelity; (b) output discrepancy under seen and unseen instructions; (c) qualitative examples. Both (a) and (b) report PSNR and LPIPS, with budgets in units of $1/255$.

5 Conclusion

We introduced SPIN to protect images against unauthorized editing by jointly considering the predicted outcomes of multiple local editing trajectories. This formulation makes output-oriented protection tractable without differentiating through the full denoising process. It also allows a protective perturbation to account for alternative continuations of an edit, rather than relying on a single construction path. Together, these ideas offer a practical direction for image immunization when future editing requests are unknown at the time of image release.

AI USE STATEMENT

In this work, we used generative AI tools to assist with portions of the code implementation and to improve the clarity and readability of the manuscript. These tools helped draft selected code segments and refine wording, sentence structure, and the expression of technical content. The core research idea was developed by the authors without the use of generative AI. We also used Kimi K2.5 and Qwen3-VL as evaluators to compute the Immunization Success Rate (ISR), following the protocol described in Appendix D. All AI-generated code was reviewed and checked by the authors. We also reviewed all AI-assisted text for accuracy and consistency with the intended meaning. We confirm the validity of the experimental results reported in this work. We take full responsibility for the final manuscript and accompanying code, including all claims, results, and content produced with AI assistance.

ETHICS STATEMENT

This work studies image immunization against unauthorized instruction-guided image editing. Although our method employs adversarial attacks, its objective is defensive: to help image owners protect their content by applying bounded perturbations before publication, reducing the risk of unwanted manipulation. The method operates on input images while keeping the editing models frozen. The same perturbation mechanism could also be used to disrupt legitimate editing workflows. We therefore encourage its use on images that users own or are authorized to protect. Our experiments use images from Pico-Banana, and the edited examples serve to evaluate protection effectiveness. They should not be interpreted as authentic depictions of people or events.

REPRODUCIBILITY STATEMENT

Section 3 describes the proposed method, and Section 4.1 summarizes the benchmark, editing models, metrics, and optimization settings. Appendices A and B provide additional evaluation details and the optimization algorithm. Evaluation uses seeds unseen during optimization, with each clean–protected pair sharing the same instruction and sampling noise to isolate the perturbation's effect. The anonymized implementation is available at https://anonymous.4open.science/r/SPIN-778F/.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pp. 284–293. PMLR, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Aniruddha Bala, Rohit Chowdhury, Rohan Jaiswal, and Siddharth Roheda. Dct-shield: a robust frequency domain defense against malicious image editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18876–18884. IEEE, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Jili Chen, Changqin Huang, Qionghao Huang, Yaxin Tu, Zhonglong Zheng, and Xiaodi Huang. Test-time reinforcement learning for flow matching. In *Forty-third International Conference on Machine Learning*, 2026.

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*, pp. 126–142. Springer, 2024.

William June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. Diffusionguard: A robust defense against malicious diffusion-based image editing. In *International Conference on Learning Representations*, volume 2025, pp. 27134–27180, 2025.

Shuai Dong, Jie Zhang, Guoying Zhao, Shiguang Shan, and Xilin Chen. Semantic mismatch and perceptual degradation: A new perspective on image editing immunity. *arXiv preprint arXiv:2512.14320*, 2025.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193. IEEE, 2018.

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 *Forty-first International Conference on Machine Learning*, 2024.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *The Third International Conference on Learning Representations*, 2015.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6007–6017. IEEE, 2023.

Ming Li, Xin Gu, Fan Chen, Xiaoying Xing, Longyin Wen, Chen Chen, and Sijie Zhu. Superedit: Rectifying and facilitating supervision for instruction-based image editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19206–19215. IEEE, 2025a.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *Advances in Neural Information Processing Systems*, pp. 95507–95545, 2025b.

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

Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 20763–20786. PMLR, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in Neural Information Processing Systems*, 38:40783–40818, 2026.

Ling Lo, Cheng Yu Yeo, Hong-Han Shuai, and Wen-Huang Cheng. Distraction is all you need: Memory-efficient image immunization against diffusion-based image editing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24462–24471. IEEE, 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*, 2018.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 86–94. Ieee, 2017.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24480–24489. IEEE, 2023.

Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious AI-powered image editing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29894–29918. PMLR, 2023.

Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1924–1933. IEEE, 2021.

Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

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

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*, volume 2024, pp. 25734–25758, 2024.

Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595. IEEE, 2018.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.