PERSISTEDIT: STEERING PERSISTENT NOISE FOR INVERSION-FREE IMAGE EDITING

Anonymous authors
Paper under double-blind review

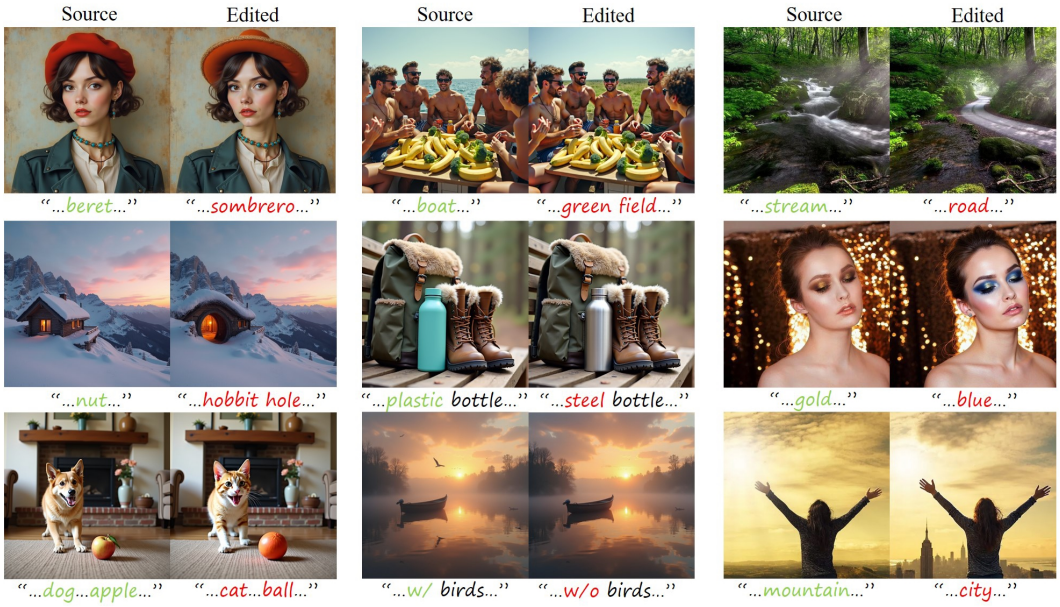

Figure 1: **Editing results of PersistEdit.** Our training-free and inversion-free method can achieve accurate target-semantic edits while maintaining source-image fidelity.

ABSTRACT

Recent inversion-free flow-based methods enable text-guided image editing without explicitly inverting source images into noise. They typically exploit averaging across timesteps through independent Gaussian noise sampling to approximate the noise-marginalized editing direction. However, finite sampling leaves realization-specific deviations that can favor different ways of realizing target semantics, shaping path-specific editing tendencies. Particularly, independent resampling repeatedly shifts these tendencies, disrupting the coherent accumulation of their effects over successive editing steps and leading to incomplete edits with residual source content or visible artifacts. To address this issue, we introduce PersistEdit, a training-free and inversion-free method to organize and selectively exploit realization-specific effects through noise control, which involves two key components, i.e., Noise Persistence and Semantic Steering. Specifically, Noise Persistence models noise evolution as a Gaussian-Markov process that combines inherited memory with newly sampled innovation, promoting coherent editing tendencies across updates. Semantic Steering selects target-favorable innovations using semantic cues within the estimated editing region while retaining inherited memory, allowing selected semantic preferences to persist across updates. Together, these designs produce a temporally coherent and semantically directed editing trajectory, improving target-semantic alignment while maintaining source-image fidelity. Extensive experiments on PIE-Bench show that PersistEdit outperforms existing methods in overall performance. Compared with FlowEdit, it improves background PSNR by 4.37 dB and edited-region CLIP similarity by 2.32%.

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.

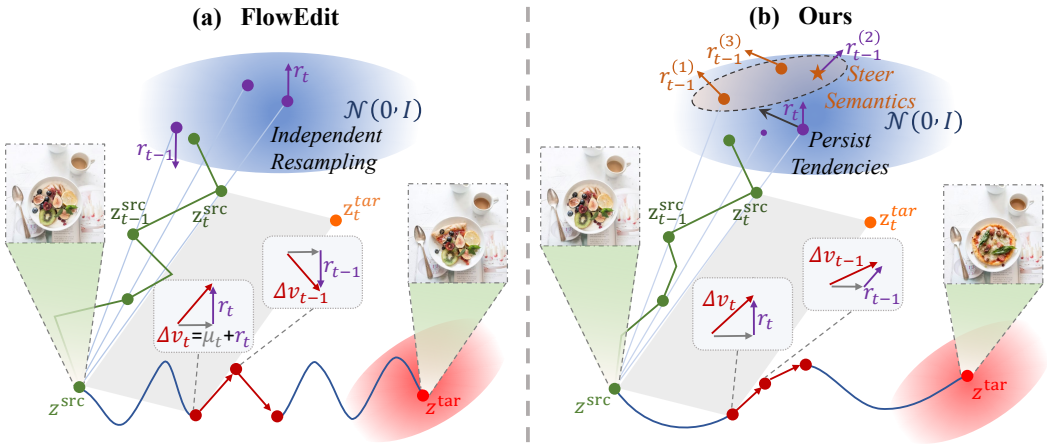

Figure 2: **Editing Process: FlowEdit vs. PersistEdit.** (a) FlowEdit independently resamples noise across timesteps, repeatedly changing the noise-induced realization-specific deviations r_t and thereby inducing fluctuations in the editing direction Δv_t, which can leave target changes partially realized with residual source content. (b) PersistEdit retains noise memory across timesteps and selects target-favorable innovations, allowing semantically aligned effects of r_t to accumulate coherently, supporting more consistent editing directions Δv_t and improved target-semantic alignment.

1 Introduction

Text-to-image generation has undergone a major transition from U-Net-based architectures (Ronneberger et al., 2015; Rombach et al., 2022) to Diffusion Transformers (DiTs) (Peebles & Xie, 2023), and from diffusion models (Ho et al., 2020; Song et al., 2021b; Rombach et al., 2022) to flow-based models. Recent systems such as Stable Diffusion 3 (Esser et al., 2024) and FLUX (Black Forest Labs, 2024) adopt rectified-flow formulations, enabling high-quality generation with efficient sampling. These advances have also made pretrained generative models increasingly effective for training-free image editing. By directly leveraging their generation and text-conditioning capabilities, training-free methods can perform image editing without task-specific fine-tuning, offering broad applicability and low deployment cost.

Training-free image editing methods can be broadly divided into inversion-based and inversion-free paradigms. Inversion-based methods first map a real image to a noisy latent and then regenerate it under a target prompt (Hertz et al., 2023; Mokady et al., 2023; Rout et al., 2025). Inversion-free methods instead inject noise into the source image to construct virtual intermediate states, and approximate the editing direction using the difference between target- and source-conditioned velocities, thereby bypassing explicit inversion (Xu et al., 2024; Kulikov et al., 2025). Recent flow-based approaches further improve source consistency and editing stability through trajectory control and structure-aware guidance (Kim et al., 2026; Jiang et al., 2026). Despite these advances, challenging edits can still suffer from incomplete target-semantic expression: the intended object or attribute may be only partially realized, residual source content may remain, and visible artifacts can persist in the edited region.

The key limitation lies in the mismatch between stepwise realization variation and cross-step pathwise accumulation. In finite-sample inference, the noise-marginalized editing direction can only be approximated, leaving realization-specific deviations under different stochastic couplings. Even when successive editing directions align with the target semantics, these deviations may favor different spatial configurations, extents, or appearances, yielding distinct editing tendencies. Independent resampling continually refreshes these deviations, switching tendencies before their effects can accumulate along a consistent path. This can lead to fluctuating editing directions and incomplete target-semantic expression, with residual source content or visible artifacts (see Fig. 2a). Although trajectory regularization can smooth the editing trajectory (Kim et al., 2026), it does not preserve the cross-step continuity of these effects. Therefore, effective inversion-free editing calls for a more coherent cross-step trajectory that supports consistent accumulation of useful pathwise effects.

To address this limitation, we propose **PersistEdit** (see Fig. 2b), a training-free and inversion-free image editing framework designed to promote coherent cross-step editing and more complete target-

semantic expression. Rather than repeatedly replacing realization-specific effects across successive updates, PersistEdit preserves their continuity throughout the editing process while steering the evolving trajectory toward the target semantics. This design mitigates cross-step inconsistency induced by repeated realization switching, allowing useful pathwise effects to accumulate more consistently and supporting more complete edits. Specifically, PersistEdit comprises two complementary components: (1) Noise Persistence models noise as a Gaussian-Markov process, making cross-step dependence an explicit control factor in inversion-free editing. By modeling persistence in the noise rather than directly constraining velocities, it reduces the repeated replacement of deviations while preserving the timestep-dependent adaptation of predicted velocities to changing noise levels and editing states. This persistence allows their pathwise effects to accumulate across successive updates, supporting more complete edits. (2) Semantic Steering makes newly introduced variation semantically controllable within the persistent process, steering it toward the desired semantics while retaining the inherited state. This steering shapes the spatial organization of the target content within the source layout, supporting a more complete and spatially coherent edit.

Our contributions are summarized as follows:

1. We analyze cross-step noise dependence in inversion-free flow editing, revealing its role in the coherence of path-specific editing tendencies. We propose PersistEdit, a training-free and inversion-free framework that organizes and selectively exploits realization-specific deviations through noise control to improve target-semantic alignment.
2. To this end, our PersistEdit presents two key components: Noise Persistence and Semantic Steering. Noise Persistence establishes controllable cross-step dependence through a Gaussian-Markov process while allowing timestep-dependent velocity adaptation. Semantic Steering selects target-favorable innovations based on target-token responses within the editing region while retaining inherited stochastic memory.
3. Extensive experiments demonstrate that PersistEdit improves target-semantic alignment and achieves superior overall performance, while validating the effectiveness of its components and cross-backbone transferability.

2 Related Work

Inversion-Free Image Editing. Inversion-free image editing leverages pretrained diffusion and flow models for text-guided manipulation without explicitly recovering a source inversion trajectory (Xu et al., 2024; Kulikov et al., 2025). Recent methods improve source consistency through feature guidance and structure-aware noise construction (Yang et al., 2025; Jiang et al., 2026), while trajectory guidance and regularization help control the editing path (Patel et al., 2025; Kim et al., 2026; Beaudouin et al., 2025). Independent resampling supports averaging across timesteps, helping reduce the influence of individual noise realizations on accumulated editing updates (Kulikov et al., 2025). Under finite sampling, however, realization-specific deviations remain and shape path-specific editing tendencies. Repeated resampling changes the noise realizations across steps (Kim et al., 2026; Jiang et al., 2026), shifting the associated tendencies across updates. Such shifts can disrupt the progressive formation of target content, leading to incomplete edits. This motivates selectively exploiting favorable realization-specific deviations and sustaining their influence across updates.

Noise Modeling in Image Editing. The noise used by pretrained generative models can influence semantic alignment and controllability, motivating methods that optimize or select initial noise for more faithful generation (Guo et al., 2024; Miao et al., 2025; Wang et al., 2025). For image editing built on these models, noise likewise provides a means of shaping the intended semantic changes. TiNO-Edit (Chen et al., 2024) jointly optimizes noise patterns and diffusion timesteps, while ELECT (Kim et al., 2025) selects reliable seeds by estimating background inconsistency at early denoising steps. Edit-Friendly DDPM (Huberman-Spiegelglas et al., 2024) extracts temporally dependent noise maps through inversion to support faithful reconstruction and semantic manipulation. DNAEdit (Xie et al., 2025) further refines noise using discrepancies between predicted and expected velocities to improve trajectory consistency, whereas SNR-Edit (Jiang et al., 2026) incorporates structure-aware priors into noise construction to preserve source structure. Despite these advances, the joint role of controllable cross-step noise dependence and target-semantic preference in sustaining coherent editing tendencies remains underexplored in inversion-free flow editing.

3 PERSISTEDIT

3.1 NOISE-INDUCED TENDENCIES IN INVERSION-FREE EDITING

Let $\boldsymbol{z}^{\mathrm{src}}$ denote the source-image latent, and c_{src} and c_{tar} the source and target text conditions. At timestep t, inversion-free flow editing uses a Gaussian noise realization $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to construct a source-target stochastic coupling:

$$\boldsymbol{z}_t^{\mathrm{src}}(\boldsymbol{\epsilon}_t) = (1-\sigma_t)\boldsymbol{z}^{\mathrm{src}} + \sigma_t \boldsymbol{\epsilon}_t, \quad \boldsymbol{z}_t^{\mathrm{tar}}(\boldsymbol{\epsilon}_t) = \boldsymbol{z}_t^{\mathrm{FE}} + \sigma_t(\boldsymbol{\epsilon}_t - \boldsymbol{z}^{\mathrm{src}}). \tag{1}$$

The corresponding source-to-target editing velocity is

$$\Delta \boldsymbol{v}_t(\boldsymbol{\epsilon}; \boldsymbol{z}_t^{\mathrm{FE}}) = \boldsymbol{v}_\theta(\boldsymbol{z}_t^{\mathrm{tar}}(\boldsymbol{\epsilon}), \sigma_t, c_{\mathrm{tar}}) - \boldsymbol{v}_\theta(\boldsymbol{z}_t^{\mathrm{src}}(\boldsymbol{\epsilon}), \sigma_t, c_{\mathrm{src}}). \tag{2}$$

At a fixed editing state, FlowEdit estimates the noise-marginalized velocity:

$$\boldsymbol{\mu}_t(\boldsymbol{z}_t^{\mathrm{FE}}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\Delta \boldsymbol{v}_t(\boldsymbol{\epsilon}; \boldsymbol{z}_t^{\mathrm{FE}})\right]. \tag{3}$$

Under finite sampling, a realization-conditioned velocity therefore contains both this marginalized component and a deviation around it:

$$\Delta \boldsymbol{v}_t(\boldsymbol{\epsilon}; \boldsymbol{z}_t^{\mathrm{FE}}) = \boldsymbol{\mu}_t(\boldsymbol{z}_t^{\mathrm{FE}}) + \boldsymbol{r}_t(\boldsymbol{\epsilon}; \boldsymbol{z}_t^{\mathrm{FE}}), \qquad \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\boldsymbol{r}_t(\boldsymbol{\epsilon}; \boldsymbol{z}_t^{\mathrm{FE}})\right] = \mathbf{0}. \tag{4}$$

Here, $\boldsymbol{\mu}_t$ denotes the noise-marginalized editing direction, while $\boldsymbol{r}_t$ captures the realization-specific deviation induced by a particular stochastic coupling. These deviations can reflect different spatial configurations, extents, or appearances under the same target prompt. Their evolving influence across successive updates shapes a path-specific editing tendency.

While noise averaging provides a principled estimate of the marginalized direction, finite sampling leaves realization-specific residuals whose favorable effects can be selectively exploited to guide the edit. Independent resampling replaces the noise realization at each step, repeatedly changing the residual effects and shifting the associated editing tendency. This cross-step inconsistency can disrupt the progressive formation of target content, leading to incomplete edits with residual source content or visible artifacts.

3.2 NOISE PERSISTENCE FOR COHERENT EDITING

To mitigate the cross-step inconsistency caused by repeated realization switching, we aim to promote the coherent accumulation of realization-specific effects across updates, allowing the intended semantic changes to develop more fully. We introduce persistence in the noise domain to regulate the cross-step evolution of source-target stochastic couplings without directly constraining editing velocities, allowing them to adapt to the evolving editing state and noise level. We further examine this design choice in Appendix D.2.

Specifically, we focus this control on the region associated with the intended semantic change. Within this region, we seek a noise process that preserves the standard Gaussian marginal at each timestep while introducing controllable dependence between successive noise realizations:

$$\tilde{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \qquad \mathrm{Cov}\left(\tilde{\boldsymbol{\epsilon}}_t, \tilde{\boldsymbol{\epsilon}}_{t-1}\right) = \rho \boldsymbol{I}, \tag{5}$$

where ρ controls the correlation between adjacent stochastic states, thereby regulating the cross-step evolution of realization-specific residuals. A natural and concise way to realize this dependency is through a first-order Gaussian-Markov process (Brockwell & Davis, 2002; Ho et al., 2020). Specifically, we model the noise sequence within the editing region as

$$\tilde{\boldsymbol{\epsilon}}_t = \underbrace{\rho \tilde{\boldsymbol{\epsilon}}_{t+1}}_{\text{memory}} + \underbrace{\sqrt{1-\rho^2}\boldsymbol{\eta}_t}_{\text{innovation}}, \qquad \boldsymbol{\eta}_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{6}$$

where $\tilde{\boldsymbol{\epsilon}}_{T+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\rho \in [0, 1)$. The memory term retains information from the preceding stochastic state, supporting coherent editing tendencies, while the innovation term introduces new stochastic variation that can be selectively steered toward target semantics. The process satisfies

$$\tilde{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \qquad \mathrm{Cov}\left(\tilde{\boldsymbol{\epsilon}}_t, \tilde{\boldsymbol{\epsilon}}_{t-k}\right) = \rho^k \boldsymbol{I}. \tag{7}$$

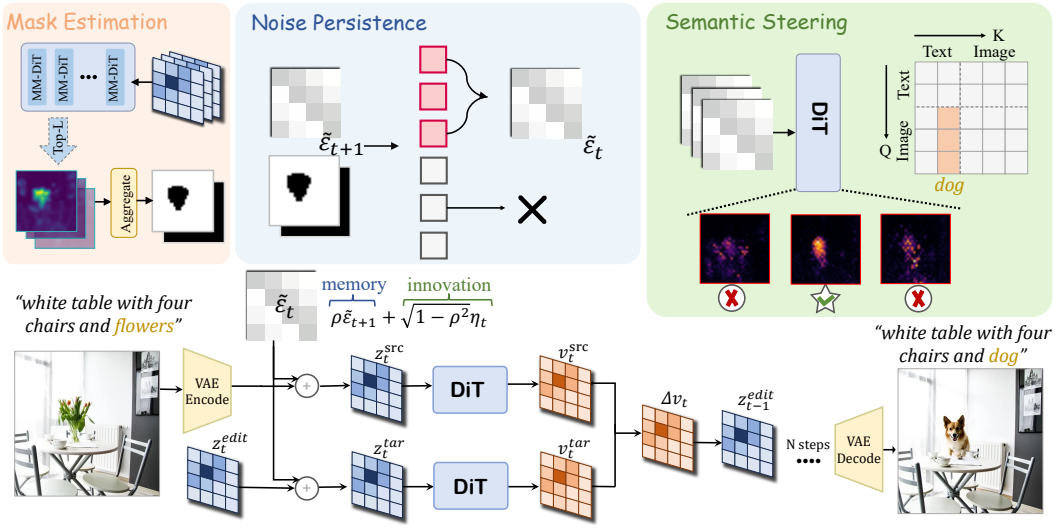

Figure 3: **Overview of PersistEdit.** Mask Estimation identifies the editing region by aggregating text-image attention across timesteps and high-confidence MM-DiT layers. Within this region, Noise Persistence introduces controllable cross-step noise dependence to retain noise memory, while Semantic Steering selects target-favorable innovations using target-token responses. Together, these complementary components promote coherent accumulation of target-favorable realization-specific effects, improving target-semantic alignment.

Therefore, ρ controls only the cross-step dependence of the noise sequence while leaving the marginal distribution at each timestep unchanged. When $\rho = 0$, the process naturally reduces to the independent resampling used in FlowEdit (Kulikov et al., 2025); as ρ increases, the correlation between adjacent stochastic states becomes stronger. Furthermore, for a noise dimension d,

$$\mathbb{E}\left[\|\tilde{\epsilon}_{t-1} - \tilde{\epsilon}_t\|_2^2\right] = 2d(1-\rho). \tag{8}$$

A larger ρ therefore reduces cross-step noise variation. By reducing repeated realization switching, successive editing steps operate under more persistent stochastic couplings. This noise control enables path-specific editing tendencies to evolve more consistently within the region associated with the intended semantic change, with limited impact on background consistency. To further reinforce this consistency, we use the clean source latent as a consistency reference along the clean-state trajectory, without relying on inversion-derived intermediate states. The constraint is applied only within a preservation window $\mathcal{T}_{\mathrm{P}}$ and released otherwise to accommodate background changes induced by the target edit, such as illumination, shadows, and reflections.

3.3 Region-Aware Semantic Steering

Noise Persistence promotes coherent editing tendencies across updates. To selectively exploit deviations better aligned with the target semantics, we introduce Semantic Steering. It steers new stochastic variation by selecting innovations based on semantic cues within the editing region, while preserving inherited memory.

We first estimate the editing region from the text-image attention responses of the pre-trained MM-DiT, without introducing an additional segmentation model. Attention responses are influenced by both noise level and Transformer layer: different noise levels provide localization cues at different spatial granularities, while different layers exhibit different localization confidence. Therefore, Temporal Attention Masking (TAM) aggregates attention responses over multiple timesteps and selected high-confidence layers. Let $\mathcal{J}_{\mathrm{src}}$ denote the token indices corresponding to the concept to be edited, and let $\alpha_{t,\ell}(j,p)$ denote the attention response between text token j and image token p at timestep t and layer ℓ. We compute

$$A(p) = \sum_{t\in\mathcal{T}_{\mathrm{M}}} \sum_{\ell\in\mathcal{L}_{\mathrm{M}}} \sum_{j\in\mathcal{J}_{\mathrm{src}}} \alpha_{t,\ell}(j,p), \quad \boldsymbol{M} = \mathrm{PostProcess}(\boldsymbol{A}), \tag{9}$$

where $\mathcal{T}_{\mathrm{M}}$ and $\mathcal{L}_{\mathrm{M}}$ specify the timesteps and selected layers used for aggregation, respectively, and $\boldsymbol{M}$ is the resulting binary editing mask. Details of layer selection and post-processing are provided in Appendix A.2.

Given the editing mask $\boldsymbol{M}$, we construct alternative stochastic continuations of the current editing path within an early selection window $\mathcal{T}_R$. At each timestep $t \in \mathcal{T}_R$, we sample K independent Gaussian innovations and combine them with the same inherited memory through Eq. 6. Within the editing region, the candidate noise states are given by

$$\tilde{\boldsymbol{\epsilon}}_t^{(k)} = \rho\tilde{\boldsymbol{\epsilon}}_{t+1} + \sqrt{1-\rho^2}\boldsymbol{\eta}_t^{(k)}, \quad \boldsymbol{\eta}_t^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad k = 1, \ldots, K. \tag{10}$$

The shared memory term carries information from preceding updates into each candidate, while the innovations provide alternative noise realizations. At the same editing state and noise level, these candidates induce different realization-specific deviations that can favor different expressions of the target semantics. We therefore score and select the candidates according to the target-token attention responses they induce within the editing region:

$$S_t^{(k)} = \sum_{p \in \Omega_{\boldsymbol{M}}} \sum_{j \in \mathcal{J}_{\mathrm{tar}}} \bar{\alpha}_t^{(k)}(j,p), \quad k_t^\star = \arg\max_{k \in \{1,\ldots,K\}} S_t^{(k)}, \tag{11}$$

where $\mathcal{J}_{\mathrm{tar}}$ denotes the token indices of the target concept, $\Omega_{\boldsymbol{M}} = \{p \mid M(p) = 1\}$ denotes the image-token positions inside the editing mask, and $\bar{\alpha}_t^{(k)}(j,p)$ is the target-branch attention response induced by candidate k, averaged over the scoring layers. This score measures target concept response strength within the editing region, serving as a local proxy for target-semantic alignment.

The selected candidate $\tilde{\boldsymbol{\epsilon}}_t^{(k_t^\star)}$ is used for the current editing update and retained as memory in the persistent noise process, allowing the semantic preference introduced by selection to influence subsequent updates. The distributional identities in Sec. 3.2 characterize the underlying process without semantic selection. We further observe that candidates induce distinguishable target concept responses in shallow Transformer blocks, enabling candidate scoring without a full forward pass for each candidate and substantially reducing the additional cost of Semantic Steering. Details are provided in Appendix A.3.

4 Experiments

4.1 Experiment Setting

Dataset and Baselines. We evaluate our method on PIE-Bench (Ju et al., 2024), which contains 620 image-prompt pairs from nine editing categories. Following (Li et al., 2024; Xu et al., 2024; Zhu et al., 2025), we exclude the style-transfer category and focus on semantic editing tasks. We compare with LDM-based methods (P2P (Hertz et al., 2023), P2P with PnP-Inv (Hertz et al., 2023; Ju et al., 2024), PnP (Tumanyan et al., 2023), MasaCtrl (Cao et al., 2023)) and DiT-based methods (RF-Inv (Rout et al., 2025), KV-Edit (Zhu et al., 2025), DNAEdit (Xie et al., 2025), FlowEdit (Kulikov et al., 2025), FlowAlign (Kim et al., 2026)).

Metrics. To comprehensively assess editing performance and background preservation, we use seven metrics for the main comparisons. HPSv2 (Wu et al., 2023) and aesthetic score (AS) (Schuhmann, 2022) evaluate edited image quality, while PSNR, LPIPS (Zhang et al., 2018), and MSE measure content preservation in the unedited region. For text-image alignment, we compute CLIP similarity (Radford et al., 2021) within the editing region and ImageReward (Xu et al., 2023) for the full edited image. We additionally use Structure Distance (Tumanyan et al., 2022) to assess source structural preservation in our analysis of ρ (Sec. 4.4). In the main comparisons, dataset-provided masks are used to separate editing and non-editing regions, but only during evaluation.

Implementation Details. We conducted experiments on FLUX.1-dev (Black Forest Labs, 2024), following the same framework as other rectified-flow-based editing methods (Rout et al., 2025; Kulikov et al., 2025). We set persistence strength $\rho = 0.6$, $\mathcal{T}_{\mathrm{P}}$ to the first 50% of the editing process, $\mathcal{T}_{\mathrm{R}}$ to the first 40%, and the candidate pool size to $K = 4$. We also follow FlowEdit's hyperparameters, using 28 timesteps in total, skipping the first 4 timesteps ($N = 24$), and setting the guidance values to 1.5 and 5.5 for the source and target branches, respectively. All experiments were conducted on a single NVIDIA RTX A6000 GPU (48GB).



Figure 4: **Qualitative comparison.** Green and red denote source and target prompt terms, respectively. Our method achieves strong semantic editing while maintaining background consistency.

4.2 Comparisons with Other Editing Methods

Table 1: **Quantitative comparison on PIE-Bench.** The best and second-best results are **bold** and underlined, respectively. The final column reports the average rank across all methods, with lower values indicating better overall performance. KV-Edit uses our mask construction method.

Model	Method	Image Quality		Background Preservation			Text Align		Rank
		HPS$_{\times 10^2}$↑	AS↑	PSNR↑	LPIPS$_{\times 10^3}$↓	MSE$_{\times 10^4}$↓	CLIP↑	IR$_{\times 10^2}$↑	Avg.↓
SD 1.5 50 steps	P2P (Hertz et al., 2023)	25.40	6.25	17.83	208.30	220.69	22.08	-4.47	8.71
	P2P(PnP-Inv) (Ju et al., 2024)	26.07	6.33	27.25	52.64	**34.13**	21.76	22.67	5.29
	PnP (Tumanyan et al., 2023)	26.59	6.38	22.41	109.89	81.19	22.33	40.89	6.29
	MasaCtrl (Cao et al., 2023)	24.93	6.13	22.83	81.14	79.18	21.14	5.44	8.00
FLUX.1 28 steps	RF-Inv (Rout et al., 2025)	27.50	6.46	19.34	211.26	167.74	21.93	40.86	7.14
	KV-Edit (Zhu et al., 2025)	27.22	6.42	**27.32**	56.49	43.45	22.12	58.76	3.71
	DNAEdit (Xie et al., 2025)	27.68	6.51	25.25	85.84	47.45	21.85	55.02	4.93
	FlowEdit (Kulikov et al., 2025)	28.32	**6.59**	22.57	97.81	83.36	22.01	56.48	4.71
	FlowAlign (Kim et al., 2026)	27.81	6.51	25.89	53.39	42.75	21.56	38.49	4.64
	PersistEdit (Ours)	**28.41**	6.54	26.94	**49.80**	36.86	**22.52**	**71.03**	**1.57**

Quantitative Comparison. As shown in Table 1, we conduct a comprehensive evaluation on PIE-Bench against both LDM-based and DiT-based editing methods. Our method demonstrates strong overall performance. Compared with FlowEdit, our method achieves higher text alignment scores together with better background preservation. Although KV-Edit achieves the highest PSNR, its strict background preservation limits adaptation to the target edit and can produce a collage-like appearance, consistent with its lower image quality scores. FlowAlign also achieves strong background preservation, but its lower text alignment scores suggest that global source-consistency regularization can constrain semantic editing. Overall, our method enables stronger semantic edits while effectively preserving background content.

Qualitative Comparison. Visual comparisons with other editing methods are presented in Fig. 4. FlowEdit independently resamples noise at each timestep without incorporating semantic guidance into the noise sampling process. Some of its results exhibit limited target-semantic changes alongside noticeable alterations to the source structure. FlowAlign maintains background structural consistency at the expense of semantic editing effectiveness. KV-Edit can exhibit stylistic inconsis-

tencies between the edited and background regions, resulting in visible collage-like artifacts. In contrast, our method produces more satisfactory results, achieving precise semantic edits while maintaining background consistency, in agreement with our quantitative results.

User Study. We conduct a user study with 25 participants, comparing our method against RF-Inv (Rout et al., 2025), KV-Edit (Zhu et al., 2025), DNAEdit (Xie et al., 2025), FlowEdit (Kulikov et al., 2025), and FlowAlign (Kim et al., 2026) on four criteria: background preservation, text alignment, visual naturalness, and overall satisfaction. As shown in Table 2, our method outperforms the baselines overall. These results suggest that more complete semantic transformations help reduce residual source content and visible artifacts, yielding more natural and satisfying results.

Table 2: **User study.** Our win rates (%) from pairwise evaluations by 25 participants across four criteria. A rate of 50% indicates equal preference, and ties count as half wins.

Ours vs.	Background↑	Text↑	Naturalness↑	Overall↑
RF-Inv	90.3%	65.5%	56.3%	84.3%
KV-Edit	48.8%	61.8%	78.0%	71.5%
DNAEdit	62.3%	74.0%	54.3%	68.8%
FlowEdit	78.5%	59.3%	58.5%	62.8%
FlowAlign	59.3%	73.5%	62.5%	67.8%

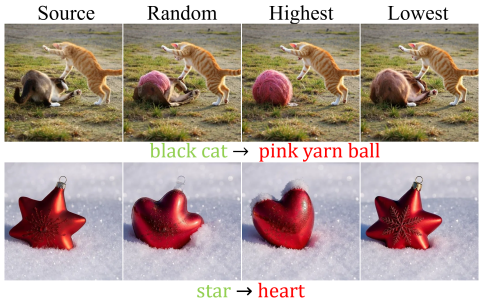

Figure 5: **Effect of Semantic Steering.**

4.3 Ablation Study

We evaluate the effects of *Noise Persistence* and *Semantic Steering*. Base denotes the FlowEdit-based pipeline with both components disabled. The editing region is defined using our TAM.

As shown in Table 3, Steering improves text alignment and image quality. Figure 5 shows that selecting candidates with the highest attention scores produces more complete edits than random or lowest-score selection, supporting the effectiveness of semantic selection. Persistence increases CLIP similarity from 22.01 to 22.35 and ImageReward from 56.48 to 67.89, alongside improved background preservation. These gains suggest that noise persistence strengthens semantic editing through coherent editing tendencies, while the background consistency constraint preserves source content. Combining both components further improves target alignment while retaining strong background preservation, supporting the complementary roles of semantic selection and cross-step persistence in enabling target-favorable effects to accumulate consistently along the editing trajectory.

Table 3: **Ablation study on Noise Persistence and Semantic Steering.**

Component	Image Quality		Background Preservation			Text Align	
	HPS$_{\times 10^2}$↑	AS↑	PSNR↑	LPIPS$_{\times 10^3}$↓	MSE$_{\times 10^4}$↓	CLIP↑	IR$_{\times 10^2}$↑
Base	28.32	6.59	22.57	97.81	83.36	22.01	56.48
+Steering	28.53	6.61	22.56	97.89	83.47	22.18	64.11
+Persistence	28.34	6.53	26.96	49.45	34.46	22.35	67.89
+Persistence+Steering	28.41	6.54	26.94	49.80	36.86	22.52	71.03

4.4 Effect of Persistence Strength

We isolate the effect of persistence strength ρ on editing dynamics using ground-truth masks, with both Semantic Steering and the background consistency constraint disabled.

Cross-Step Coherence of Editing Tendencies. To examine whether noise memory promotes more consistent editing tendencies, we compute the cosine similarity between the channel vectors of adjacent realization-specific residuals, $\boldsymbol{r}_t$ and $\boldsymbol{r}_{t-1}$, at each spatial location within the editing region, and average these similarities over timesteps. We estimate $\boldsymbol{\mu}_t$ at each fixed editing state by averaging editing velocities over a large number of independent Gaussian noise realizations. As shown in Fig. 6, increasing ρ yields higher similarity between adjacent residuals within the editing region, indicating stronger directional coherence of realization-specific effects across successive updates. These results support the role of noise persistence in promoting coherent editing tendencies across updates and reducing the repeated switching and averaging associated with independent resampling.

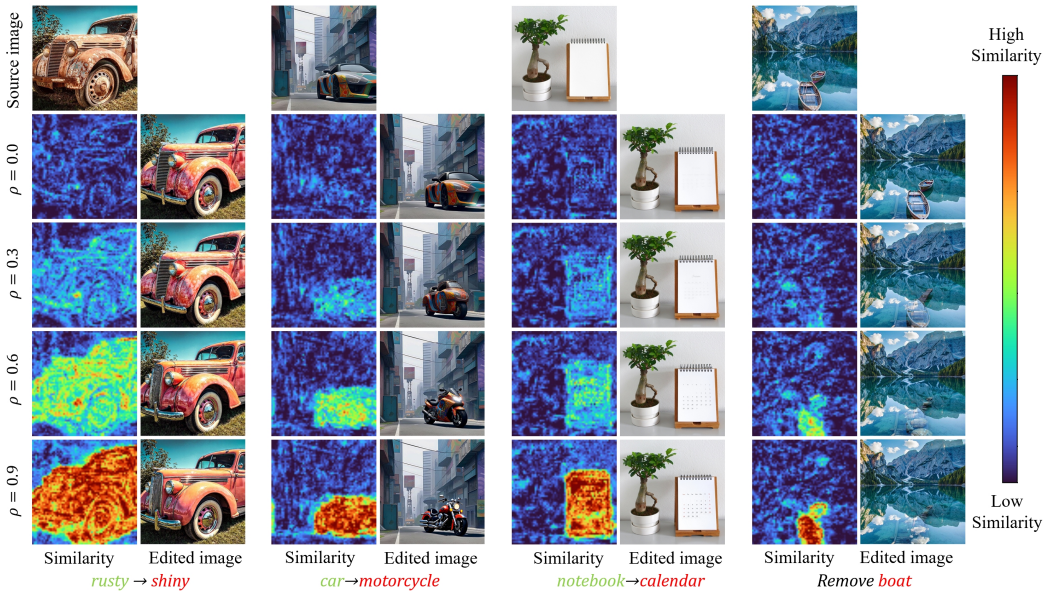

Figure 6: **Qualitative Results and Visualization of Editing-Tendency Coherence.** We compare cosine similarity maps of adjacent realization-specific residuals and the corresponding edited images across persistence strengths $\rho \in \{0.0, 0.3, 0.6, 0.9\}$. Increasing ρ yields higher residual similarity within the editing region, leading to more pronounced target-semantic changes. Similarity maps are computed as $\text{Sim} = \frac{1}{T-1}\sum_{t=2}^{T} \cos_{\text{channel}}(\boldsymbol{r}_t, \boldsymbol{r}_{t-1})$.

Sensitivity Analysis of ρ. As shown in Fig. 7, CLIP similarity increases with ρ, suggesting that more coherent trajectories improve editability and help alleviate under-editing associated with the source-image inductive bias. However, excessively high persistence strength may also allow unintended changes in object scale or appearance to accumulate across successive updates, leading to an increase in Structure Distance. These editing gains are accompanied by only a slight decrease in background PSNR, from 22.57 to 22.31 dB as ρ increases from 0 to 0.9. The qualitative comparisons below show that increasing persistence strength yields more complete edits, but excessively high values can lead to over-editing. Based on these results, we adopt $\rho = 0.6$ to balance semantic editing strength and source-structure preservation.

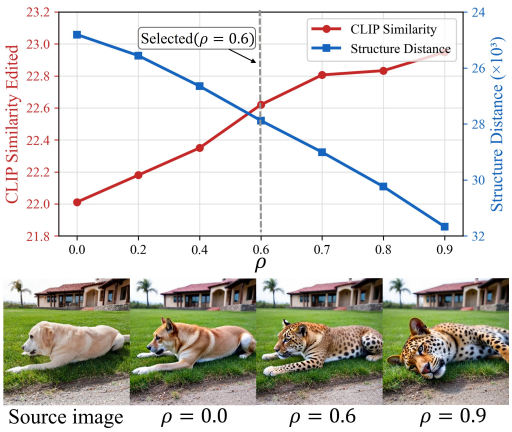

Figure 7: **Sensitivity Analysis of ρ.** We set $\rho = 0.6$ to balance editability and structural fidelity.

5 Conclusion

We propose PersistEdit, a training-free and inversion-free image editing framework that controls noise evolution to exploit realization-specific deviations in editing velocities. Noise Persistence promotes coherent editing tendencies across successive updates without directly constraining velocities, preserving their adaptation to the editing state and noise level. Semantic Steering selects innovations aligned with the target semantics while retaining inherited memory, directing the persistent process toward the intended edit. Empirical analysis shows greater directional coherence between adjacent residuals as persistence increases, and experiments demonstrate improved target-semantic alignment and superior overall performance. This work offers a perspective on inversion-free image editing in which realization-specific deviation is actively organized and exploited through noise control to support coherent, semantically directed editing.

AI USE STATEMENT

In this work, we used LLMs to assist with implementing experimental code, translating and refining manuscript text and user-study questions, identifying and summarizing relevant literature, and LaTeX formatting. We also used LLMs for feedback on experimental design and discussion of possible interpretations of experimental results. The authors tested AI-generated code, reviewed the core implementation, and manually checked the retrieved references and literature summaries. AI-assisted text was reviewed and revised by the authors. We take responsibility for the final manuscript, implementation, and reported results.

REFERENCES

Gaspard Beaudouin, Minghan Li, Jaeyeon Kim, Sung-Hoon Yoon, and Mengyu Wang. Delta rectified flow sampling for text-to-image editing. *arXiv preprint arXiv:2509.05342*, 2025.

Black Forest Labs. Flux.1 [dev]: A 12 billion parameter rectified flow transformer. *https://blackforestlabs.ai/*, 2024.

Peter J. Brockwell and Richard A. Davis. *ARMA Models*, pp. 83–110. Springer New York, New York, NY, 2002. ISBN 978-0-387-21657-7. doi: 10.1007/0-387-21657-X_3.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22560–22570, 2023.

Sherry X. Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. TiNO-Edit: Timestep and noise optimization for robust diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6337–6346, 2024.

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.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9380–9389, 2024.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12469–12478, 2024.

Lifan Jiang, Boxi Wu, Yuhang Pei, Tianrun Wu, Yongyuan Chen, Yan Zhao, Shiyu Yu, and Deng Cai. SNR-Edit: Structure-aware noise rectification for inversion-free flow-based editing. *arXiv preprint arXiv:2601.19180*, 2026.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Joowon Kim, Ziseok Lee, Donghyeon Cho, Sanghyun Jo, Yeonsung Jung, Kyungsu Kim, and Eunho Yang. Early timestep zero-shot candidate selection for instruction-guided image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18844–18854, 2025.

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

Yaowei Li, Yuxuan Bian, Xuan Ju, Zhaoyang Zhang, Ying Shan, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *arXiv preprint arXiv:2412.10316*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Boming Miao, Chunxiao Li, Xiaoxiao Wang, Andi Zhang, Rui Sun, Zizhe Wang, and Yao Zhu. Noise diffusion for enhancing semantic faithfulness in text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23575–23584, 2025.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047, 2023.

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. FlowChef: Steering of rectified flow models for controlled generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15308–15318, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 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*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

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

Christoph Schuhmann. LAION-Aesthetics, 2022. URL `https://laion.ai/blog/laion-aesthetics/`.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, et al. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10748–10757, 2022.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Ruoyu Wang, Huayang Huang, Ye Zhu, Olga Russakovsky, and Yu Wu. The silent assistant: Noisequery as implicit guidance for goal-driven image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17618–17628, 2025.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL https://arxiv.org/abs/2306.09341.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. DNAEdit: Direct noise alignment for text-guided rectified flow editing. *arXiv preprint arXiv:2506.01430*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kaixiang Yang, Boyang Shen, Xin Li, Yuchen Dai, Yuxuan Luo, Yueran Ma, Wei Fang, Qiang Li, and Zhiwei Wang. Fia-edit: Frequency-interactive attention for efficient and high-fidelity inversion-free text-guided image editing. *arXiv preprint arXiv:2511.12151*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

A Algorithm and Implementation Details

A.1 Overall Algorithm of PersistEdit

Here, we present the overall algorithm of PersistEdit. As shown in Algorithm 1, PersistEdit combines Noise Persistence and Semantic Steering to sustain semantically aligned editing tendencies across updates. The source and target concept token index sets $\mathcal{J}_{\mathrm{src}}$ and $\mathcal{J}_{\mathrm{tar}}$ are specified by the user according to the intended edit. The editing region is estimated by Temporal Attention Masking (TAM), with its settings specified offline.

Algorithm 1 Algorithm for PersistEdit

Input: source image X^{src}, source/target prompts $c_{\mathrm{src}}/c_{\mathrm{tar}}$, persistence strength ρ, candidate count K, preservation interval $\mathcal{T}_{\mathrm{P}}$, semantic steering interval $\mathcal{T}_{\mathrm{R}}$, editing steps T, source/target concept token index sets $\mathcal{J}_{\mathrm{src}}$ and $\mathcal{J}_{\mathrm{tar}}$.
Output: edited target image X_{tar}.
Estimate the editing mask $\boldsymbol{M}$ using Eq. 9.
$\boldsymbol{z}_T^{\mathrm{PE}} = \boldsymbol{z}^{\mathrm{src}} = X^{\mathrm{src}}, \quad \boldsymbol{\epsilon}_{T+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \dots, 1$ **do**
  if $t \in \mathcal{T}_{\mathrm{R}}$ **then**
    $\boldsymbol{\eta}_t^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad k = 1, \dots, K$
    $\boldsymbol{\epsilon}_t^{(k)} \leftarrow \boldsymbol{M} \odot \left(\rho \boldsymbol{\epsilon}_{t+1} + \sqrt{1-\rho^2}\boldsymbol{\eta}_t^{(k)}\right) + (1 - \boldsymbol{M}) \odot \boldsymbol{\eta}_t^{(k)}, \quad k = 1, \dots, K$
    Compute candidate scores $\{S_t^{(k)}\}_{k=1}^{K}$ using Eq. 11.
    $k_t^\star = \arg\max_{k \in \{1,\dots,K\}} S_t^{(k)}, \quad \boldsymbol{\epsilon}_t \leftarrow \boldsymbol{\epsilon}_t^{(k_t^\star)}$
  else
    $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $\boldsymbol{\epsilon}_t \leftarrow \boldsymbol{M} \odot \left(\rho \boldsymbol{\epsilon}_{t+1} + \sqrt{1-\rho^2}\boldsymbol{\eta}_t\right) + (1 - \boldsymbol{M}) \odot \boldsymbol{\eta}_t$
  end if
  $\boldsymbol{z}_t^{\mathrm{src}} = (1 - \sigma_t)\boldsymbol{z}^{\mathrm{src}} + \sigma_t \boldsymbol{\epsilon}_t$
  $\boldsymbol{z}_t^{\mathrm{tar}} = \boldsymbol{z}_t^{\mathrm{PE}} + \boldsymbol{z}_t^{\mathrm{src}} - \boldsymbol{z}^{\mathrm{src}}$
  $\Delta \boldsymbol{v}_t = \boldsymbol{v}_\theta(\boldsymbol{z}_t^{\mathrm{tar}}, \sigma_t, c_{\mathrm{tar}}) - \boldsymbol{v}_\theta(\boldsymbol{z}_t^{\mathrm{src}}, \sigma_t, c_{\mathrm{src}})$
  $\boldsymbol{z}_{t-1}^{\mathrm{PE}} \leftarrow \boldsymbol{z}_t^{\mathrm{PE}} + (\sigma_{t-1} - \sigma_t)\Delta \boldsymbol{v}_t$
  if $t \in \mathcal{T}_{\mathrm{P}}$ **then**
    $\boldsymbol{z}_{t-1}^{\mathrm{PE}} \leftarrow \boldsymbol{M} \odot \boldsymbol{z}_{t-1}^{\mathrm{PE}} + (1 - \boldsymbol{M}) \odot \boldsymbol{z}^{\mathrm{src}}$
  end if
end for
Return: $X_{\mathrm{tar}} = \boldsymbol{z}_0^{\mathrm{PE}}$

A.2 Methods for Estimating Mask

This section describes the implementation details of Temporal Attention Masking (TAM) for estimating the editing region. As shown in Fig. 8, attention localization varies across timesteps under different noise levels. We therefore aggregate attention responses over multiple timesteps to obtain a more stable editing mask. In the main experiments with 28 inference steps, we use steps 5–28 as the temporal aggregation window. The accumulated attention map is then post-processed to obtain the binary editing mask. Specifically, $\mathrm{PostProcess}(\cdot)$ denotes normalization, Gaussian smoothing, and Otsu thresholding.

We further evaluate the localization quality of different MM-DiT Transformer layers. Specifically, we sample 100 examples from COCO val2017 (Lin et al., 2014), each containing an image, a human-written caption, a target object, and its corresponding ground-truth mask. For each layer, we evaluate the estimated masks within the selected timestep window using BCE, soft mIoU, and MSE. The results are shown in the right part of Fig. 8, with FLUX.1-dev on the top row and SD3-Medium on the bottom row. Different layers exhibit clear differences in localization quality, while temporal aggregation produces more reliable masks than single-step masking. Based on these results, we select the 10 best-performing layers for FLUX.1-dev in our main experiments.

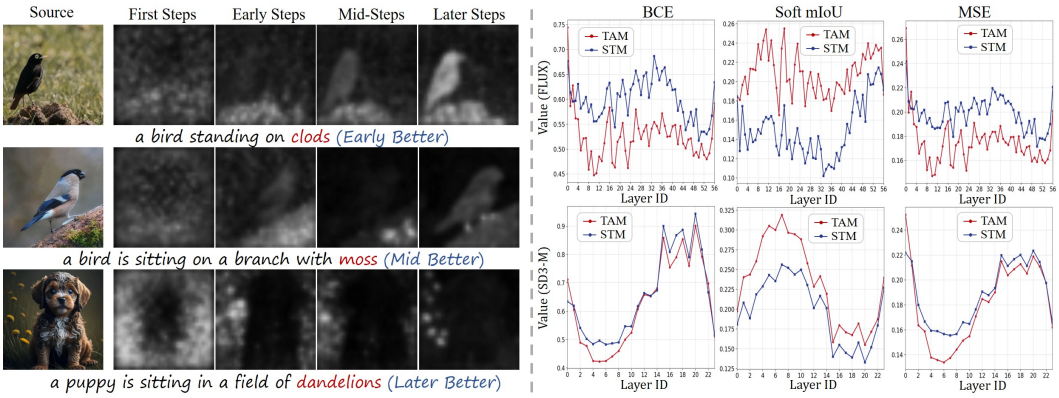

Figure 8: **Temporal Attention Masking (TAM).** The left part illustrates the variation of attention localization across timesteps, while the right part compares the localization quality of different MM-DiT layers on FLUX.1-dev (top) and SD3-Medium (bottom). Temporal aggregation provides more reliable localization than single-step masking (STM).

A.3 Accelerating Strategy for Semantic Steering

Semantic Steering evaluates multiple candidate continuations at selected editing steps, introducing additional inference cost. A naive implementation requires a full forward pass for every candidate before comparing their target-semantic responses. For FLUX.1-dev, whose Transformer backbone contains 57 blocks, this cost becomes substantial as the number of candidate innovations increases.

We find that full-depth evaluation is unnecessary for candidate selection. Differences in target-token responses already emerge in shallow MM-DiT blocks, providing sufficient evidence to distinguish favorable continuations. We therefore evaluate all candidates through the initial Transformer blocks, then complete the remaining forward pass only for the selected candidate.

Let $\mathcal{L}_{\mathrm{R}}^{\mathrm{early}}$ denote the shallow layers used for scoring. For candidate k, we compute

$$S_{t,\mathrm{early}}^{(k)} = \sum_{p\in\Omega_M} \sum_{j\in\mathcal{J}_{\mathrm{tar}}} \bar{\alpha}_{t,\mathrm{early}}^{(k)}(j,p), \qquad k_t^\star = \arg\max_{k\in\{1,\dots,K\}} S_{t,\mathrm{early}}^{(k)}, \tag{12}$$

where $\bar{\alpha}_{t,\mathrm{early}}^{(k)}(j,p)$ denotes the target-token response averaged over $\mathcal{L}_{\mathrm{R}}^{\mathrm{early}}$. The selected innovation determines the persistent noise state used for the current editing update, while the remaining candidates are discarded. For FLUX.1-dev, we use the first 19 Transformer blocks for early scoring.

As shown in Fig. 9, early-layer scoring achieves the same Whole CLIP similarity as full-layer scoring (25.82), while reducing the inference time from 28.11 s to 19.87 s. This indicates that semantic preference can be reliably identified from shallow responses, substantially reducing the additional cost of Semantic Steering.

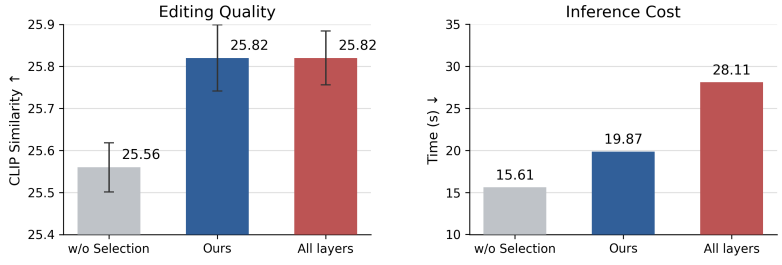

Figure 9: **Efficiency of Semantic Steering.** Early-layer candidate scoring using the first 19 Transformer blocks of FLUX.1-dev achieves the same CLIP similarity as full-layer scoring (25.82), while reducing inference time from 28.11 s to 19.87 s.

B User Study Details

To more comprehensively evaluate the quality of edited results, we conduct a pairwise user study. We compare our method with five baselines: RF-Inv, KV-Edit, DNAEdit, FlowEdit, and FlowAlign. The evaluation covers four criteria: Background Preservation, Text Alignment, Visual Naturalness, and Overall Preference. The images are randomly sampled from PIE-Bench after removing the style-transfer category.

We allow participants to choose a tie or no-preference option when the two results are difficult to distinguish, as shown in Figure 10. Candidate results are anonymized and randomly ordered for each sample to reduce bias from method names or display order. In total, we recruit 25 participants and collect 200 pairwise comparisons for each evaluated baseline. When computing the final comparison results, a tie is counted as half a win for both sides.

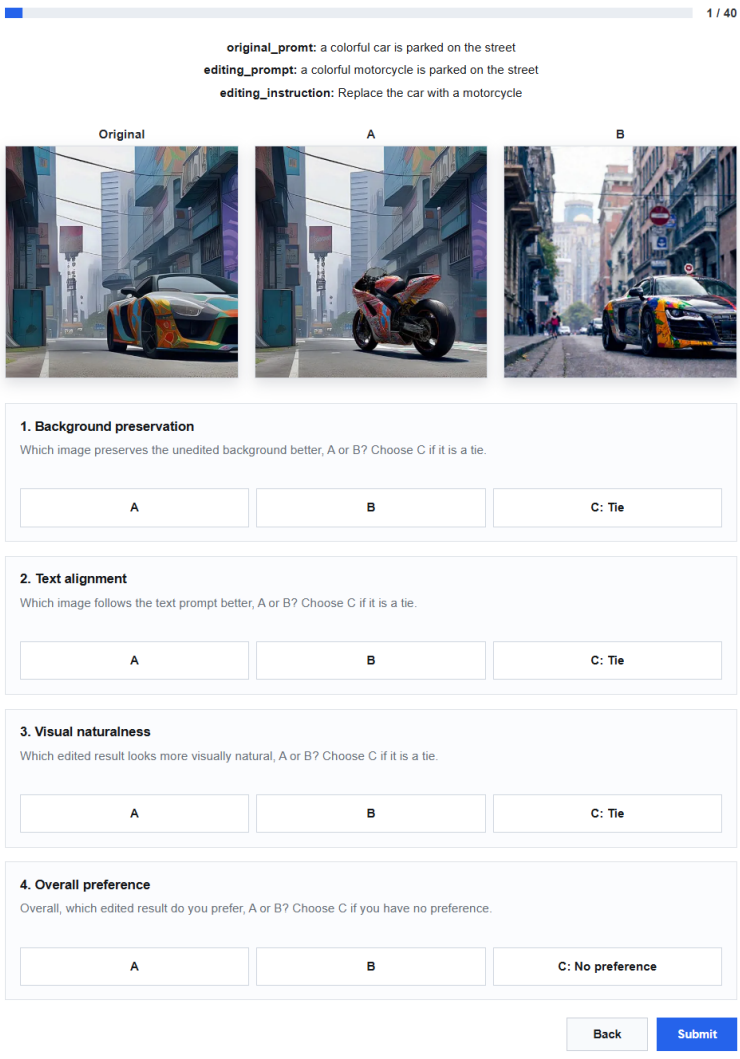

Figure 10: **User study interface.** Participants compare two anonymized editing results under four criteria and may choose a tie or no-preference option when the two results are difficult to distinguish.

C Comparison Methods and Details

In this section, we provide the experimental setup and hyperparameter configurations of the comparison methods reported in Table 1.

For P2P, PnP, and MasaCtrl, we adopt DDIM inversion (Song et al., 2021a) as the inversion backbone and use 50 sampling steps. All editing experiments are conducted with their default parameter settings. In particular, following the official recommendation for MasaCtrl, we use the intermediate latent trajectory obtained from DDIM inversion as the source trajectory during editing, which improves real-image reconstruction and editing stability.

For P2P (PnP-Inv), we combine PnP Inversion (Ju et al., 2024) with Prompt-to-Prompt using the official PnP-Inversion implementation and its default settings.

For RF-Inv, we use 28 inference steps. To obtain better structural consistency in the edited results, we follow the official recommendation and set the stopping time to 0.35. Other settings remain unchanged.

For DNAEdit, FlowEdit and FlowAlign, we use 28 inference steps. Apart from this, all experiments use the official implementations and keep all default settings unchanged.

For KV-Edit, we modify the official codebase by using the mask constructed by our method as its required editing region. Other settings remain unchanged.

D Additional Analyses

D.1 Noise Averaging and Noise Persistence

In FlowEdit, n_{avg} denotes the number of noise samples used at each timestep to estimate the editing velocity. Different noise realizations induce different velocity estimates, which are averaged to reduce stochastic variance and approximate the noise-marginalized editing direction. To further analyze the effects of within-step noise averaging and cross-step noise persistence, we vary $n_{\text{avg}} \in \{1, 2, 4\}$ for FlowEdit and the persistence strength ρ for our method. To isolate the effect of persistence strength, we disable Semantic Steering and the background consistency constraint in this analysis. All experiments are conducted with 28 inference steps. Quantitative results are reported in Table 4, and qualitative examples are shown in Fig. 11.

As shown in Table 4, increasing n_{avg} gradually decreases CLIP similarity in our evaluation setting. Averaging more noise realizations provides a more accurate estimate of the noise-marginalized editing direction by reducing deviation. Meanwhile, it also reduces the contribution of realization-specific deviations induced by individual stochastic couplings. Since these deviations can shape different pathwise realizations of the target change, stronger averaging may attenuate their contribution to consistent accumulation along the editing trajectory. In contrast, increasing ρ preserves stronger cross-step consistency among realization-specific effects, allowing their influence to persist across successive updates.

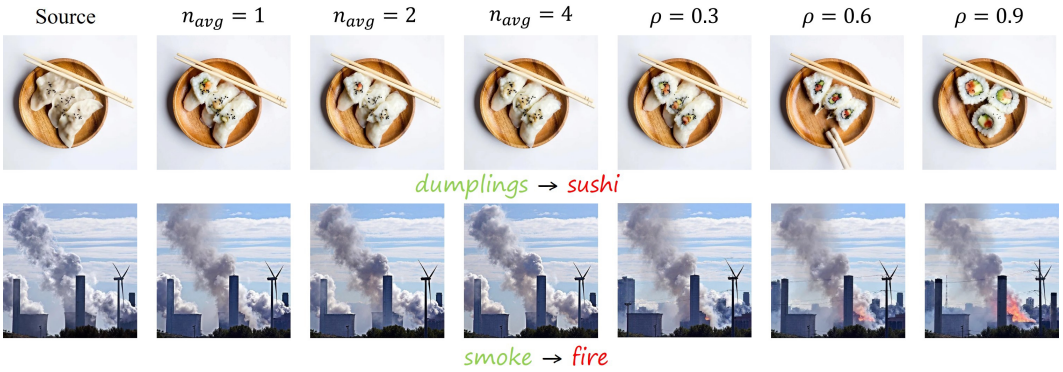

Figure 11: **Qualitative comparison of noise averaging and noise persistence.** Increasing n_{avg} attenuates realization-specific deviations and may weaken their contribution to target changes, whereas stronger noise persistence preserves more consistent editing tendencies and enables more complete edits.

Table 4: **Quantitative comparison of noise averaging and noise persistence.** We compare n_{avg} in FlowEdit with persistence strength ρ in our method using CLIP similarity.

Metric	n_{avg} (FlowEdit)			ρ (Noise Persistence)		
	1	2	4	0.3	0.6	0.9
CLIP Similarity	22.01	21.87	21.74	22.28	22.62	22.94

D.2 Effect of Velocity Smoothing

We examine direct smoothing of editing velocities as a way to encourage coherent editing tendencies across steps. Specifically, we apply an exponential moving average (EMA):

$$\overline{\Delta \boldsymbol{v}}_t = \beta \overline{\Delta \boldsymbol{v}}_{t+1} + (1-\beta)\Delta \boldsymbol{v}_t, \tag{13}$$

where $\overline{\Delta \boldsymbol{v}}_{t+1}$ denotes the running average from the preceding editing step and β controls the smoothing strength. Smoothing is restricted to the editing mask $\boldsymbol{M}$, with norm matching to preserve the Frobenius norm of the current unsmoothed velocity within this region:

$$\Delta \boldsymbol{v}_t^{\text{norm}} = (1-\boldsymbol{M}) \odot \Delta \boldsymbol{v}_t + \frac{\|\boldsymbol{M} \odot \Delta \boldsymbol{v}_t\|_F}{\|\boldsymbol{M} \odot \overline{\Delta \boldsymbol{v}}_t\|_F} \boldsymbol{M} \odot \overline{\Delta \boldsymbol{v}}_t. \tag{14}$$

The norm-matched velocity $\Delta \boldsymbol{v}_t^{\text{norm}}$ is used for the editing update, while the unscaled $\overline{\Delta \boldsymbol{v}}_t$ is retained for subsequent averaging. We vary $\beta \in \{0, 0.1, \ldots, 0.9\}$ with Semantic Steering and the background consistency constraint disabled. All experiments use a 28-step schedule, skipping the first four steps.

As shown in Fig. 12, moderate smoothing improves edited-region CLIP similarity from 22.01 to 22.34 around $\beta = 0.5$, consistent with the benefit of promoting coherent editing tendencies across updates. Under the same settings, noise persistence achieves 22.62 at the default $\rho = 0.6$ used in our main experiments, with further gains at larger ρ within the tested range. For velocity smoothing, increasing β beyond 0.5 reduces CLIP similarity and degrades image quality, with large β potentially introducing blurred boundaries and ghosting. Editing velocities also evolve with the editing state and noise level: early updates primarily involve coarse structural changes, while later updates increasingly involve finer details (Kulikov et al., 2025). Strong temporal smoothing of these velocities may restrict this evolution, suppressing adjustments needed at each stage and impairing edit quality. These observations support promoting coherent editing tendencies through noise-domain persistence while allowing velocities to adapt.

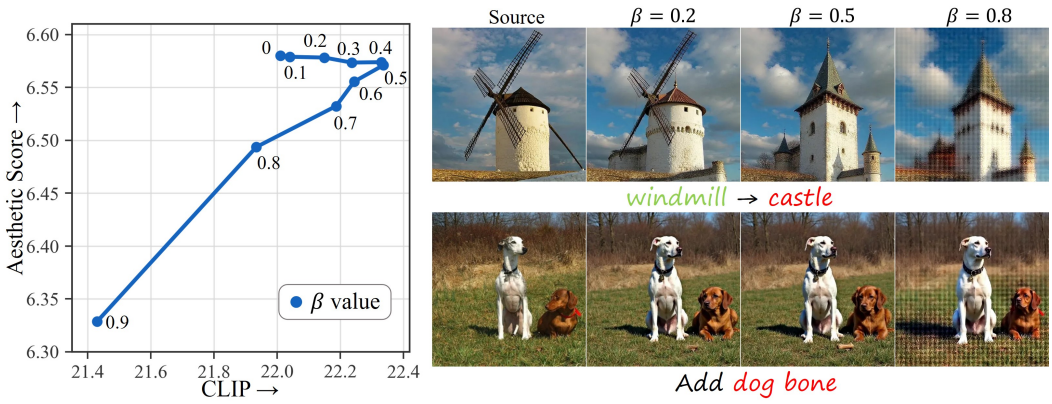

Figure 12: **Effect of velocity smoothing.** Moderate norm-matched smoothing improves edited-region CLIP similarity, whereas excessive smoothing reduces both CLIP similarity and aesthetic score (left), accompanied by blurred boundaries and ghosting in the edited images (right).

E More Ablation Studies

E.1 Semantic Steering Analysis

We first examine the temporal placement of Semantic Steering by applying it over three different intervals, $\mathcal{T}_{\mathrm{R}} \in \{5\text{–}12, 13\text{–}20, 21\text{–}28\}$. We use ground-truth masks to exclude the influence of mask estimation, while keeping all other settings consistent with the main experiments. As shown in Table 5 and Fig. 13, applying Semantic Steering during the early stage produces stronger target-semantic changes than applying it at middle or later stages. Qualitatively, early steering also better establishes the spatial configuration of the edited object, including its placement and spatial extent, suggesting that semantic preference is particularly influential during the early evolution of the editing process.

We further study the duration of Semantic Steering by progressively extending the steering interval from the early stage toward later timesteps (Table 6). The semantic improvement is mainly obtained during the early portion of the editing process, while further extending the steering interval provides only limited additional gains. We therefore apply Semantic Steering only during the first 40% of the editing process in the main experiments, reducing the additional time cost introduced by evaluating candidate innovations while retaining most of the semantic benefit.

Table 5: **Effect of the temporal placement of Semantic Steering.** Early-stage steering yields stronger target-semantic alignment than middle- or late-stage steering.

Metric	$\mathcal{T}_{\mathrm{R}}$		
	5–12	13–20	21–28
CLIP Sim.	22.54	22.39	22.34

Table 6: **Effect of Semantic Steering duration.** Most semantic gains are obtained within the early steering interval, with limited improvement from extending it to later steps.

Metric	$\mathcal{T}_{\mathrm{R}}$			
	5–8	5–12	5–16	5–28
CLIP Sim.	22.39	22.54	22.55	22.56



Figure 13: **Qualitative comparison of Semantic Steering at different stages.** Early-stage steering produces stronger target-semantic changes and better establishes the spatial configuration of the edited object.

E.2 Effect of Preservation Window

We further analyze the preservation window $\mathcal{T}_{\mathrm{P}}$, during which the non-editing region is constrained toward the source reference. Under 28 inference steps, we use ground-truth masks and vary the inclusive end step of $\mathcal{T}_P$, which starts at step 5, while keeping all other settings unchanged. As shown in Table 7, extending the preservation window consistently improves background-preservation metrics, while image-quality and text-alignment scores gradually decrease. This reflects a trade-off between source consistency and the flexibility required for edit-induced changes. We therefore set $\mathcal{T}_{\mathrm{P}}$ to the first 50% of the editing process in our main experiments.

Table 7: **Effect of preservation window $\mathcal{T}_{\mathrm{P}}$.** A longer preservation window improves source and background consistency, while slightly reducing semantic alignment.

$\mathcal{T}_{\mathrm{P}}$	Image Quality		Background Preservation			Text Align	
	HPS$_{\times 10^2}\uparrow$	AS$\uparrow$	PSNR$\uparrow$	LPIPS$_{\times 10^3}\downarrow$	MSE$_{\times 10^4}\downarrow$	CLIP$\uparrow$	IR$_{\times 10^2}\uparrow$
5–8	28.73	6.62	26.37	54.38	33.39	22.65	78.12
5–14	28.44	6.58	29.23	33.95	17.42	22.54	72.63
5–20	28.07	6.51	31.38	21.78	11.14	22.39	64.85

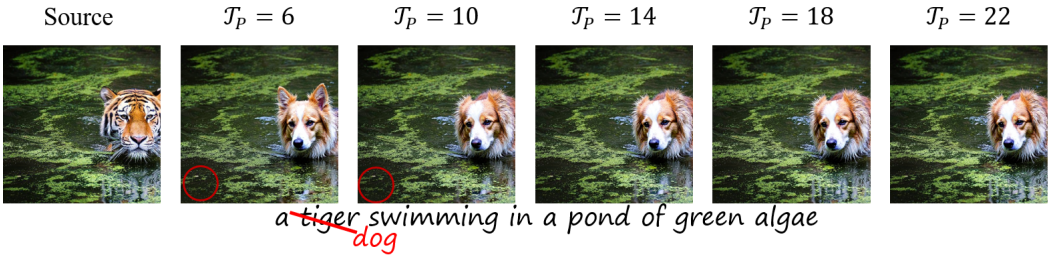

Figure 14: **Qualitative effect of the preservation window.** A short $\mathcal{T}_{\mathrm{P}}$ may cause unwanted background changes, whereas an overly long $\mathcal{T}_{\mathrm{P}}$ can hinder adaptive changes required by the edit.

The qualitative results in Fig. 14 further illustrate this trade-off. A short preservation window may lead to unwanted changes in non-editing regions, whereas an overly long window can suppress adaptive changes induced by the semantic edit.

F Additional Visualization Results on SD3

Our main experiments are conducted using the pre-trained flow model FLUX.1-dev. To further evaluate the applicability of PersistEdit across different flow-based backbones, we transfer our method to Stable Diffusion 3 Medium. Similar to FLUX.1-dev, SD3-Medium is based on the MM-DiT architecture and the flow matching sampling process. In implementation, we keep the overall pipeline unchanged and select the top-5 high-confidence layers for mask estimation. Figure 15 shows editing results of PersistEdit on SD3-Medium. The results indicate that our method still maintains strong background consistency while accurately performing the intended semantic modifications. This suggests that PersistEdit is broadly applicable to flow-based image editing models.

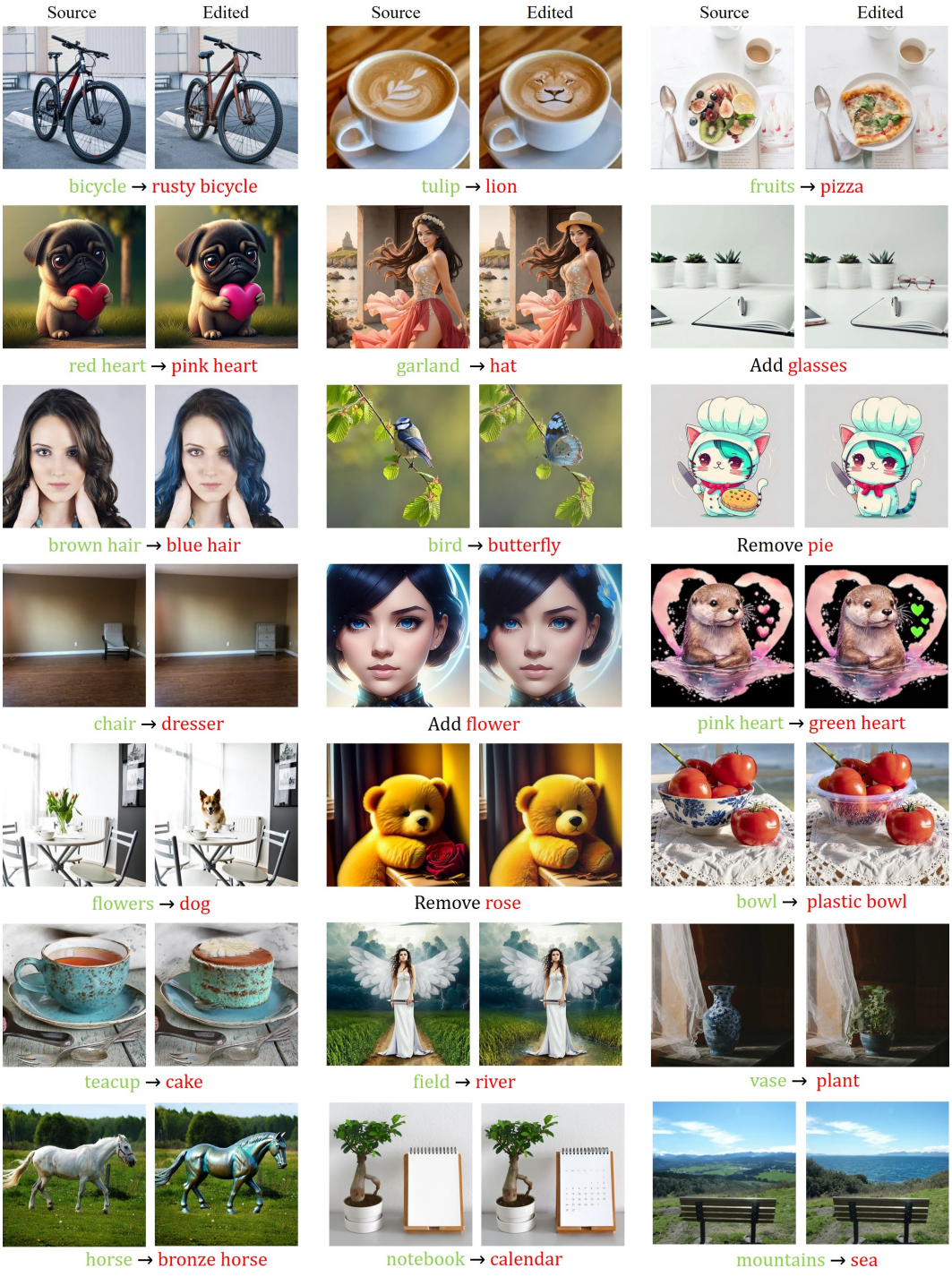

Figure 15: **Additional visualization results on SD3-Medium.**