FREEDIT: ONE-STEP SOURCE MEMORY FOR MASK-GUIDED IMAGE EDITING

Anonymous authors
Paper under double-blind review

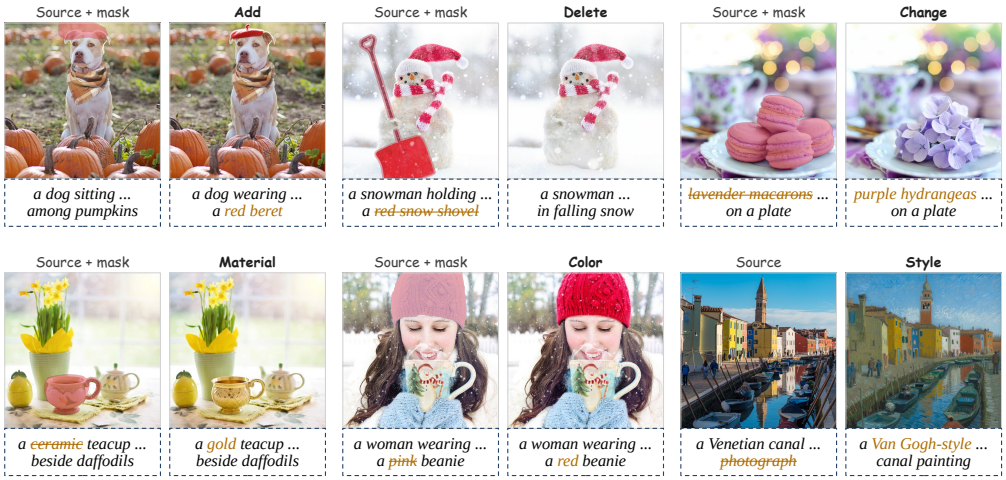


Figure 1: FreEdit accelerates mask-guided editing with one source evaluation and foreground-only denoising while preserving unedited regions across six edit types.

ABSTRACT

Text-to-image (T2I) models can also edit real images according to text prompts. Training-free methods often rely on inversion to preserve source content, requiring repeated evaluations at multiple noise levels and full-image denoising during generation. This is particularly costly for local editing, where the edit region is known in advance and the remaining content should stay unchanged. We present FreEdit (Free-inversion Editing), a training-free and highly efficient method that eliminates multi-step inversion and global denoising for local image editing. FreEdit adds noise to the encoded source at a single intermediate timestep and extracts its attention projections with only one network evaluation. During target denoising, only masked image tokens are updated, attending to stored source keys and values outside the mask for contextual guidance. The generated foreground is combined with the clean source latent, preserving untouched content by construction. For color and style edits, optional QK-CFG further improves source-structure preservation while keeping foreground appearance target-driven. Dedicated appearance evaluations demonstrate strong color fidelity, structural preservation, style alignment, and aesthetic quality. On FLUX.1-dev, FreEdit achieves a $\mathbf{3.58}\times$ speedup on the same hardware while improving all six reported mean metrics. Results across Klein, Qwen, and SD3.5 further demonstrate its consistent effectiveness across diverse generative backbones.

1 INTRODUCTION

Text-to-image (T2I) models based on diffusion or rectified flow generate images by transforming noise according to a text prompt (Rombach et al., 2022; Esser et al., 2024). The same pretrained models can also edit real images (Yang et al., 2026; Zhu et al., 2026). Given a source image and a new prompt, the goal is to produce the requested change while preserving details that the prompt does not mention, such as the viewpoint, lighting, background, and identity of unchanged objects. Unlike ordinary text-to-image generation, image editing is therefore constrained by two inputs: the

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.

prompt specifies what should change, while the source image specifies everything that should remain. Following the prompt without retaining the source produces a new image rather than an edit; retaining the source without following the prompt leaves the requested change incomplete. We consider the setting in which the user also marks the region that needs to change. The editor should generate new content in that region and leave the rest of the image unchanged. Figure 1 illustrates six representative edit types under this setting.

Training-free editors perform this task without learning new model weights. To edit an observed image with a pretrained generator, many methods first use *inversion*. A text-to-image model normally generates an image by starting from a noisy latent and progressively removing noise (Ho et al., 2020; Song et al., 2020). Inversion runs in the opposite direction: it starts from the encoded source image and recovers a sequence of progressively noisier source latents. During editing, the model denoises a target latent under the new prompt and reuses the corresponding source latent or its features at each step to retain source content (Mokady et al., 2023; Ju et al., 2024b; Rout et al., 2024; Wang et al., 2024). The target branch commonly denoises a complete image, even when only a small part of that image is meant to change. In this conventional pipeline, inversion provides source-specific information and target denoising produces the edited image. Both stages repeatedly run the model over all image tokens (Wu et al., 2026).

This redundancy becomes clear when the user has specified where the edit should occur. The marked region contains content that must be generated, whereas the rest of the desired output is already present in the source image. Nevertheless, multi-step inversion repeatedly records information from the complete source, including regions that will remain unchanged. Much of this record is redundant when the source background only needs to provide context for the new foreground. At the same time, global target denoising repeatedly updates background tokens whose output will eventually be replaced by the known source. A local editor therefore has two opportunities to reduce computation: obtain the source information needed for the edit without a full inversion, and generate only the region requested by the user.

These two costs are distinct. Restricting target denoising to the edit region reduces target-side computation, but does not by itself eliminate the repeated source evaluations used to collect features throughout inversion. This raises a further question: how much source computation is actually needed to guide a local edit?

For local editing, the source branch does not need to reconstruct the source image again. Its main role is to tell the generated region how it fits into the observed scene: where the boundary lies, what surrounds it, and how the new content should be organized relative to the background. The unchanged pixels themselves can be recovered directly from the clean source after generation. This suggests that time-matched source features from every inversion step may not be necessary. If one source evaluation can retain the contextual information needed by the foreground, the same fixed features can guide all target denoising steps.

We present FreEdit, a training-free editor that addresses both sources of computation. Instead of constructing a multi-step inversion, FreEdit adds noise to the encoded source at one intermediate timestep and evaluates the pretrained transformer once under the source prompt. As shown in Figure 2, the attention projections produced by this evaluation are stored as a fixed source memory and reused throughout target denoising. The source is not evaluated again during generation.

During target denoising, FreEdit updates only image tokens inside the mask. At each attention layer, these foreground tokens attend to stored source keys and values outside the mask and to current target keys and values inside it. The stored background features provide the context needed to align the new content with the source scene, without denoising target background tokens. After denoising, the generated foreground is inserted into the clean source latent, and the completed latent is decoded once.

Some edits should also retain the organization of the source foreground. A color or style change, for example, should alter appearance without changing the object's shape or layout. For these cases, we introduce an optional rule, QK-CFG, that moves target foreground queries and keys toward their stored source counterparts while leaving target values unchanged. We use this rule only for appearance-editing tasks.

Our contributions are:

- We show through controlled ablations that a single evaluation of the noised source at an intermediate timestep can replace multi-step source inversion for mask-guided editing while maintaining comparable overall editing quality.
- Based on this observation, we introduce FreEdit, which stores the attention projections from this source evaluation as fixed memory and reuses them throughout masked target denoising. Clean-latent composition restores the known region, while optional QK-CFG supports color and style editing; dedicated evaluations verify its effectiveness on both tasks.
- On the same FLUX.1-dev backbone and hardware, FreEdit reduces average editing time from 13.93 to 3.89 seconds, a $\mathbf{3.58}\times$ speedup, while improving all six reported mean metrics. Controlled ablations and quantitative and qualitative results on Klein, Qwen, and SD3.5 further validate the design and its transfer across backbones.

2 Related Work

Image editing and inversion. Imagic fine-tunes a generator for each image, whereas InstructPix2Pix learns instruction-based editing from paired examples (Kawar et al., 2023; Brooks et al., 2023). Without model training, SDEdit edits by adding noise and denoising (Meng et al., 2021); P2P and Null-text Inversion use attention control and inversion (Hertz et al., 2022; Mokady et al., 2023). EDICT and Edit-Friendly DDPM improve source reconstruction, while Negative-Prompt Inversion avoids per-image optimization (Wallace et al., 2023; Huberman-Spiegelglas et al., 2024; Miyake et al., 2025). PnP Inversion jointly processes source and target trajectories and introduces PIE-Bench (Ju et al., 2024b). For rectified flow, methods recover source states, reuse features, or control sampling trajectories (Rout et al., 2024; Wang et al., 2024; Beaudouin et al., 2026; Lupaşcu & Stupariu, 2026). KV-Edit reuses source-background key-value features to preserve unedited regions (Zhu et al., 2025). InfEdit and FlowEdit reduce inversion cost through virtual inversion or direct transport (Xu et al., 2024; Kulikov et al., 2025).

Feature control and local editing. ControlNet and T2I-Adapter learn spatial conditioning modules (Zhang et al., 2023; Mou et al., 2024). Training-free alternatives include Plug-and-Play Diffusion's feature injection, MasaCtrl's mutual attention, and FreeControl's structural and appearance guidance (Tumanyan et al., 2023; Cao et al., 2023; Mo et al., 2024; Liu et al., 2026). Attention guidance, routing, and fusion preserve source structure (Parmar et al., 2023; Yin et al., 2026; Xu et al., 2026; Lai et al., 2026); related methods extend such control to complex edits (Chen et al., 2026; Jiang et al., 2026). For local editing, Blended Diffusion and Blended Latent Diffusion combine edited and source regions during denoising (Avrahami et al., 2022; 2023), while DiffEdit infers an edit mask from text-conditioned predictions (Couairon et al., 2022). Trained systems such as BrushNet and FLUX.1 Fill [dev] learn to synthesize a masked region from its visible context (Ju et al., 2024a). FreEdit obtains the source context required for local editing from one evaluation and reuses its fixed attention features throughout foreground-only denoising.

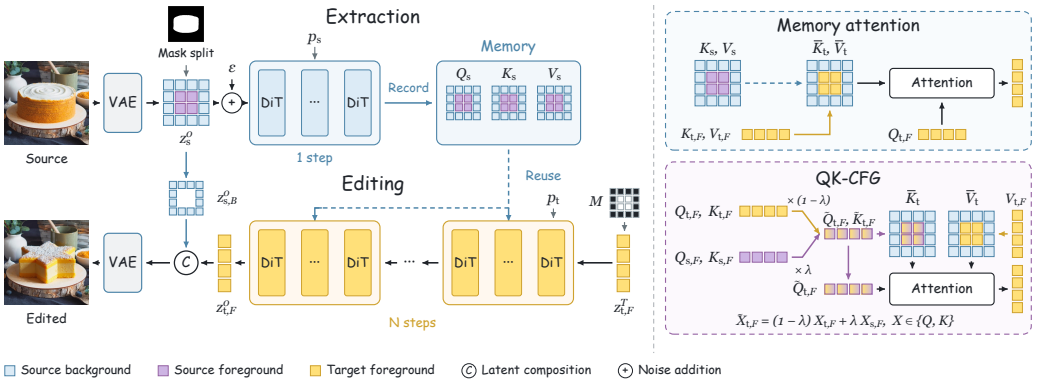


Figure 2: Overview of FreEdit. A single source evaluation records reusable attention memory for masked target denoising. Background K/V supplies source context through memory attention, while optional QK-CFG guides appearance edits.

3 METHOD

3.1 PROBLEM SETTING AND METHOD OVERVIEW

The input to FreEdit consists of a source image x_{s}, an optional source prompt p_{s}, a target prompt p_{t}, and an edit mask M. Appendix C.2 shows that replacing p_{s} with an empty string causes negligible change. Throughout, the roman subscripts s and t identify source and target quantities, respectively; superscripts identify a source extraction time, a target denoising step, or a transformer layer. The method follows three stages, as illustrated in Figure 2.

1. Read the source once. We add noise to the encoded source, evaluate the pretrained network once under p_{s}, and store the resulting Q/K/V projections as a fixed description of the source scene.

2. Generate only the masked region. We initialize the target branch with the same noise and denoise it under p_{t}. Only image tokens inside M are updated. The target keeps no evolving background tokens; its foreground queries instead attend to a mixed memory containing stored source K/V outside the mask and current target K/V inside it.

3. Compose the final image. We insert the generated foreground latent into the clean source latent and decode the completed latent once. The background comes directly from the source rather than from target denoising.

We next define the three stages formally. Let $M \in \{0,1\}^{H\times W}$, with one indicating a location that may change. We resample M to two resolutions: the latent grid used to compose the final image, and the image-token grid used by the transformer. On the token grid, let $\mathcal{F} = \{i : M_i = 1\}$ denote the editable foreground. Only positions in $\mathcal{F}$ carry target states that change over time. Their queries, however, can read source K/V from every unmasked position. This gives the foreground enough context to match boundaries and illumination, or to infer content hidden by a removed object, without generating a target background.

3.2 A ONE-STEP READ OF THE SOURCE

We describe the method using a pretrained rectified-flow transformer. The underlying flow formulation learns a vector field between noise and data (Liu et al., 2022; Lipman et al., 2022), while transformer backbones process image latents as sequences of patches with self-attention (Vaswani et al., 2017; Peebles & Xie, 2023; Esser et al., 2024). Our construction only requires access to the model's attention projections. Let z_{s}^0 be the latent encoding of the source image and let $\epsilon \sim \mathcal{N}(0, I)$ be sampled noise. We choose an extraction time $\tau \in [0,1]$ and form a noisy source latent:

$$z_{\mathrm{s}}^{\tau} = (1-\tau) z_{\mathrm{s}}^0 + \tau\epsilon. \tag{1}$$

Here, $\tau = 0$ corresponds to the clean source and $\tau = 1$ to pure noise. We evaluate the network once using z_{s}^{τ}, τ, and the source prompt p_{s}. The same noise ϵ later initializes the target branch, keeping the random perturbation spatially aligned between source and target token positions. We use this shared-noise initialization as a fixed alignment rule. The target then follows the standard denoising schedule under p_{t}. At each attention layer ℓ, the single source evaluation produces

$$\mathcal{S}^{\ell,\tau} = \left(Q_{\mathrm{s}}^{\ell,\tau}, K_{\mathrm{s}}^{\ell,\tau}, V_{\mathrm{s}}^{\ell,\tau}\right), \tag{2}$$

We store these projections as source memory and do not evaluate the source again. Source K/V outside the mask supplies background context (Section 3.3), while source Q/K inside the mask can optionally guide foreground organization through QK-CFG (Section 3.5).

The extraction time controls the information encoded in this fixed memory: it determines the background context available to every edit and, with QK-CFG, the source foreground relations used for appearance preservation. The same memory is reused at every target step instead of constructing a time-matched source trajectory. We use fixed $\tau = 0.6$ throughout the main evaluation. Section 4.4 shows that this default lies within a broad stable regime.

The stored projections form persistent memory rather than a time-indexed inversion trajectory. At each target step, new foreground projections are paired with fixed source background K/V; when QK-CFG is enabled, stored foreground Q/K can also guide source organization. One source evaluation therefore supports the entire edit.

3.3 Foreground Queries over Mixed Full-Scene Memory

At each target step, the network computes new Q/K/V only for editable tokens. To let them see the surrounding scene, we build a complete attention memory: positions outside the mask use stored source features, while positions inside the mask use features from the current target. Text tokens continue through the pretrained block normally under p_{t}; only the image-token memory is modified, so text K/V are omitted from the notation.

At target denoising step n, let $(Q^{\ell,n}_{\mathrm{t},\mathcal{F}}, K^{\ell,n}_{\mathrm{t},\mathcal{F}}, V^{\ell,n}_{\mathrm{t},\mathcal{F}})$ denote the projections of editable tokens under p_{t}. The operator $\mathsf{Scat}_{\mathcal{F}}(\cdot)$ places a foreground sequence on the full image-token grid and fills all other positions with zeros. The complete key and value memories are

$$\bar{K}^{\ell,n}_{\mathrm{t}} = (1 - M) \odot K^{\ell,\tau}_{\mathrm{s}} + \mathsf{Scat}_{\mathcal{F}}(K^{\ell,n}_{\mathrm{t},\mathcal{F}}), \tag{3}$$

$$\bar{V}^{\ell,n}_{\mathrm{t}} = (1 - M) \odot V^{\ell,\tau}_{\mathrm{s}} + \mathsf{Scat}_{\mathcal{F}}(V^{\ell,n}_{\mathrm{t},\mathcal{F}}). \tag{4}$$

Here, M is the mask on the image-token grid, with broadcasting over channels left implicit. The first term keeps source K/V outside the mask, and the second term inserts current target K/V inside the mask. The foreground attention output is

$$A^{\ell,n}_{\mathrm{t}} = \mathrm{softmax}\left(\frac{Q^{\ell,n}_{\mathrm{t},\mathcal{F}}(\bar{K}^{\ell,n}_{\mathrm{t}})^{\top}}{\sqrt{d}}\right)\bar{V}^{\ell,n}_{\mathrm{t}}, \tag{5}$$

The output $A^{\ell,n}_{\mathrm{t}}$ then passes through the original residual and MLP operations and contributes to the normal denoising update. Text-token processing and the rest of the transformer block remain unchanged. Repeating this operation lets each foreground token retrieve information from any unmasked source position while interacting with current target features inside the edit region, even though only the foreground has an evolving target state.

Source and target tokens share the same positional grid, so the mask provides their correspondence directly. If target sampling uses N network evaluations, an inversion-based method may also evaluate the source roughly N times. FreEdit evaluates it once and forms target queries only for the $|\mathcal{F}|$ editable tokens, while still giving those queries full-scene context.

3.4 Composing the Final Image

After denoising, we combine the generated foreground with the original background. Let z^0_{t} be the target latent at the end of denoising; final composition retains its masked values. Let M_z be the mask resampled to the latent grid. Using the clean source encoding z^0_{s}, we form

$$z_{\mathrm{edit}} = M_z \odot z^0_{\mathrm{t}} + (1 - M_z) \odot z^0_{\mathrm{s}}. \tag{6}$$

This operation takes target latents inside the mask and clean source latents outside it. We then decode the completed latent once. The target branch therefore never needs to generate the background. Because the VAE decodes the complete latent jointly, small pixel-level differences may still appear outside the mask in the final RGB image.

The attention memory and the final composition have different roles. During generation, source background K/V tells the model how the new foreground should fit the scene, while target K/V inside the mask allows the old foreground to change. After generation, latent composition restores the known source background directly.

3.5 QK-CFG for Appearance-Preserving Edits

Semantic edits should be free to create new foreground structure, whereas appearance edits require a different balance. For example, recoloring a flower should preserve its petal arrangement without preserving its old color. We therefore guide target attention with source relations without copying source values, which could directly reintroduce old foreground features.

For these edits, we also use the foreground portion of the one-step source memory. We move target queries and keys toward the corresponding source queries and keys, while leaving target values

unchanged:

$$\widetilde{Q}^{\ell,n}_{\mathrm{t},\mathcal{F}} = Q^{\ell,n}_{\mathrm{t},\mathcal{F}} + \lambda_q\big(Q^{\ell,\tau}_{\mathrm{s},\mathcal{F}} - Q^{\ell,n}_{\mathrm{t},\mathcal{F}}\big), \tag{7}$$

$$\widetilde{K}^{\ell,n}_{\mathrm{t},\mathcal{F}} = K^{\ell,n}_{\mathrm{t},\mathcal{F}} + \lambda_k\big(K^{\ell,\tau}_{\mathrm{s},\mathcal{F}} - K^{\ell,n}_{\mathrm{t},\mathcal{F}}\big), \tag{8}$$

$$\widetilde{V}^{\ell,n}_{\mathrm{t},\mathcal{F}} = V^{\ell,n}_{\mathrm{t},\mathcal{F}}. \tag{9}$$

When QK-CFG is enabled, $\widetilde{Q}^{\ell,n}_{\mathrm{t},\mathcal{F}}$ replaces $Q^{\ell,n}_{\mathrm{t},\mathcal{F}}$ in Eq. (5), while $\widetilde{K}^{\ell,n}_{\mathrm{t},\mathcal{F}}$ replaces the target foreground key in Eq. (3). The mixed value memory in Eq. (4) continues to use the target-derived $V^{\ell,n}_{\mathrm{t},\mathcal{F}}$. The independently adjustable coefficients λ_q and λ_k set the guidance strength and its step/layer schedule. QK-CFG is a CFG-inspired linear rule in Q/K projection space (Ho & Salimans, 2022): rather than combining model predictions, it biases target attention relations toward the source. Keeping V target-derived avoids directly reinserting stored source foreground values.

We apply QK-CFG to appearance edits that should preserve the source foreground structure, such as color and style changes, and disable it when the requested edit changes object structure or content. Source background K/V remains active in both cases.

4 Experiments

4.1 Evaluation Protocol

We use the 620-image PIE-Bench main subset adopted by KV-Edit (Ju et al., 2024b; Zhu et al., 2025), including its prompts, masks, and semantic edits. We compare with training-free P2P (Hertz et al., 2022), MasaCtrl (Cao et al., 2023), RF-Inversion (Rout et al., 2024), RF-Edit (Wang et al., 2024), FlowEdit (Kulikov et al., 2025), and KV-Edit (Zhu et al., 2025), and with trained BrushEdit (Li et al., 2026) and FLUX Fill. Results through KV-Edit come from its published comparison; FlowEdit uses its official implementation on the same inputs. VAE reconstruction is included as an unranked reference. Following KV-Edit, we report HPS v2 for perceptual quality (Wu et al., 2023); PSNR, LPIPS, and MSE outside the mask for preservation (Zhang et al., 2018); and Edit CLIP similarity and ImageReward for prompt alignment (Radford et al., 2021; Xu et al., 2023). We use FLUX.1-dev with one source evaluation at $\tau = 0.6$ and foreground-only target denoising. QK-CFG is disabled for semantic evaluation and uses fixed task-specific configurations for color and style evaluation. Appendix A gives all configurations and protocols.

4.2 Quantitative Comparison and Efficiency

Table 1 shows a consistent quality pattern. FreEdit gives the strongest outside-mask preservation under PSNR, LPIPS, and MSE, and the best prompt alignment under both Edit CLIP similarity and ImageReward, while ranking second on HPS. Compared with KV-Edit, the improvements extend across all six metrics. FlowEdit reaches a higher HPS, but its substantially lower preservation scores indicate a different trade-off. Overall, these results show that the computational simplifications in FreEdit do not come at the expense of editing quality, background preservation, or prompt alignment. FreEdit also remains stable across four sampling seeds (Appendix B).

Table 1: Quantitative comparison. VAE* is unranked; best and second-best editing results are bold and underlined.

Method	Training-free	Image Quality	Background Preservation			Text Alignment	
		HPS $\times 10^2$ ↑	BG PSNR↑	BG LPIPS $\times 10^3$ ↓	BG MSE $\times 10^4$ ↓	Edit CLIP Sim↑	IR $\times 10$ ↑
VAE*	–	24.93	37.65	7.93	3.86	19.69	-3.65
P2P	✓	25.40	17.86	208.43	219.22	22.24	0.02
MasaCtrl	✓	23.46	22.20	105.74	86.15	20.83	-1.66
RF-Inversion	✓	27.99	20.20	179.73	139.85	21.71	4.34
RF-Edit	✓	27.60	24.44	113.20	56.26	22.08	5.18
BrushEdit	✗	25.81	32.16	17.22	8.46	22.44	3.33
FLUX Fill	✗	25.76	32.53	25.59	8.55	22.40	5.71
FlowEdit	✓	**29.25**	21.95	107.66	93.13	22.59	8.61
KV-Edit	✓	28.05	<u>33.30</u>	<u>14.80</u>	<u>7.45</u>	<u>23.62</u>	<u>9.15</u>
FreEdit (ours)	✓	<u>28.57</u>	**33.82**	**12.48**	**6.78**	**23.72**	**9.54**

Efficiency. Under identical data, timing, and hardware, FreEdit reduces KV-Edit's runtime from 13.93 to 3.89 seconds ($\mathbf{3.58}\times$), the fastest result among FLUX.1-dev editors. Comparable peak memory shows that the speedup does not come from lower memory use. BrushEdit and FLUX Fill use different backbones and serve as system-level references.

Table 2: Per-image efficiency. Bold marks the fastest FLUX.1-dev runtime.

Method	Backbone	Time (s)↓	Peak memory (GiB)↓
P2P	SD v1.4	21.91	12.09
MasaCtrl	SD v1.4	13.59	10.78
BrushEdit	SD1.5+ BrushNetX	1.92	4.43
FLUX Fill	FLUX.1-Fill-dev	5.38	32.62
RF-Edit	FLUX.1-dev	20.68	33.31
FlowEdit	FLUX.1-dev	9.13	34.77
KV-Edit	FLUX.1-dev	13.93	33.29
FreEdit	FLUX.1-dev	**3.89**	34.08

Color and style editing. We further evaluate QK-CFG on the PIE-Bench color and style subsets; the complete protocol is given in Appendix A.2.

Table 3: Quantitative comparison of color and style editing. Best and second-best results are bold and underlined.

Method	Color editing			Style editing		
	ΔE_{00} ↓	Full SSIM↑	BG PSNR↑	Style CLIP↑	Text CLIP↑	Aesthetic↑
P2P	37.78	<u>0.76</u>	24.77	15.67	27.03	5.92
MasaCtrl	41.72	0.72	19.81	14.72	25.69	5.86
RF-Inversion	42.79	0.49	15.99	16.60	27.45	5.86
RF-Edit	42.42	0.72	20.59	16.44	27.30	<u>6.32</u>
BrushEdit	31.28	0.62	27.60	<u>17.73</u>	**29.48**	6.19
FLUX Fill	<u>28.51</u>	0.70	26.98	17.69	27.17	5.60
FlowEdit	39.54	0.74	19.42	15.48	26.37	6.10
KV-Edit	29.13	0.73	<u>31.41</u>	17.60	28.65	6.17
FreEdit (ours)	**26.48**	**0.77**	**31.92**	**18.58**	<u>28.84</u>	**6.52**

Table 3 shows that FreEdit improves appearance control without trading away source fidelity. It achieves the best color accuracy together with the strongest full-image and background preservation. The same advantage extends to style editing, where it leads in style alignment and aesthetics while remaining competitive in prompt alignment. Across the two tasks, FreEdit ranks first on five of six metrics and second on the remaining one, giving a consistent result rather than optimizing one aspect of appearance at the expense of another.

Transfer across backbones. FreEdit also transfers to Klein, Qwen, and SD3.5, retaining localized editing and strong background preservation across addition, deletion, and replacement. On Klein, it improves all six reported metrics over the native editing pipeline. The shared behavior across distinct transformer backbones shows that the source-memory formulation is not tied to FLUX.1-dev. Appendix D reports the complete results and backbone-specific configurations.

4.3 Qualitative Comparison

Semantic editing. Figure 3 covers addition, deletion, shape, and material changes. The comparisons expose two common failure modes. P2P can preserve too much of the original foreground: the cake remains round, and the dog is not fully removed. FLUX Fill makes stronger changes, but sometimes modifies nearby source structure. FreEdit instead changes the masked content while retaining the visible context: the cake becomes star-shaped, and the dog is removed without redrawing the person or bench. Together, these examples show that FreEdit localizes substantial semantic changes without sacrificing the surrounding scene. Further semantic comparisons are provided in Appendix E.

Color and style editing. Figure 4 shows color and style edits. In the color rows, the requested hue changes while the face, flower shape, and surrounding context remain recognizable. In the style rows, the wolf silhouette and landscape layout are retained while their rendering changes. Across both edit types, FreEdit retains the principal source geometry while giving the target prompt control over appearance. These examples are consistent with the appearance-specific results in Table 3.

Additional color results are provided in Appendix F, and Appendix G presents a complementary latent-space color refinement.

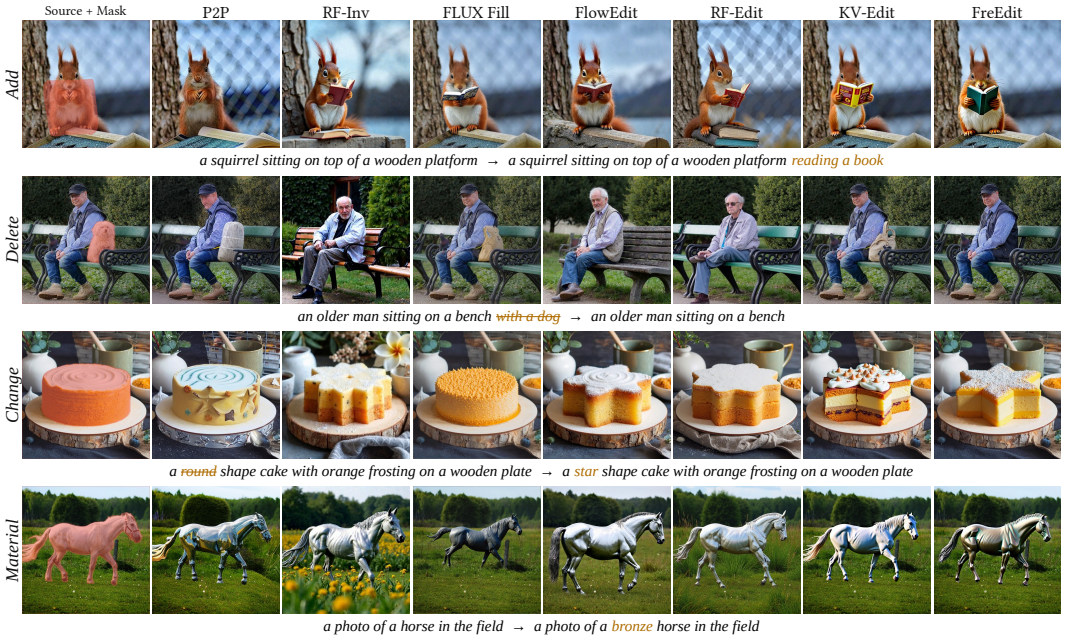


Figure 3: Semantic editing comparisons. "Source +" overlays the edit mask on the input. All methods receive the same source image, mask, and target prompt.

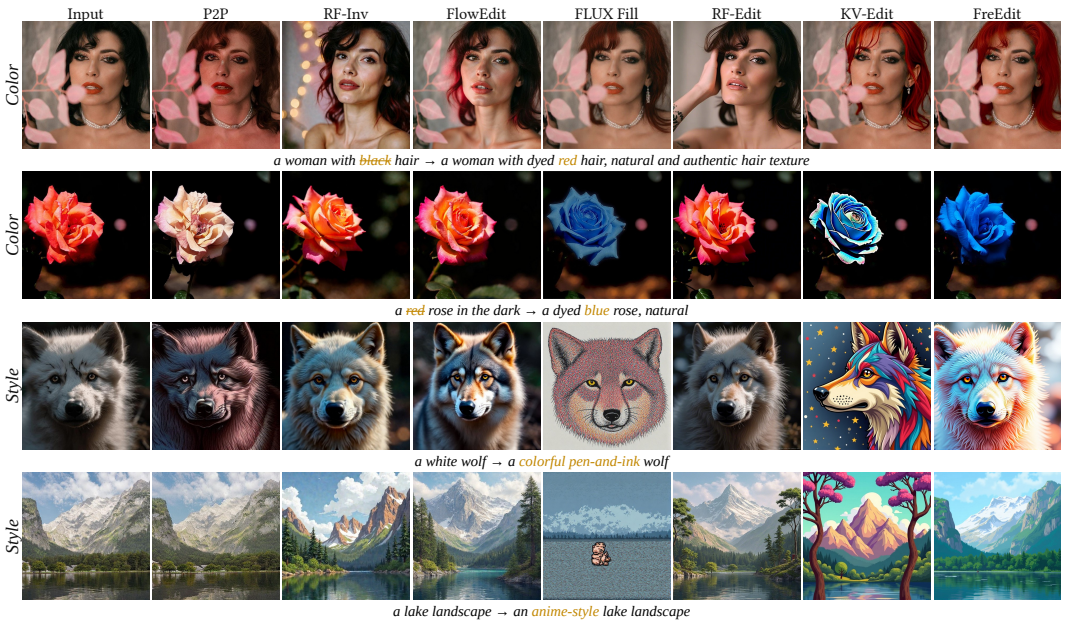


Figure 4: Color and style editing. For these appearance edits, QK-CFG guides selected foreground Q/K projections toward the source while leaving V target-derived.

4.4 Ablations and Sensitivity Analysis

This section examines when to read the source, whether source background memory is necessary, whether target denoising should be restricted to the foreground, and whether one source evaluation can replace full inversion.

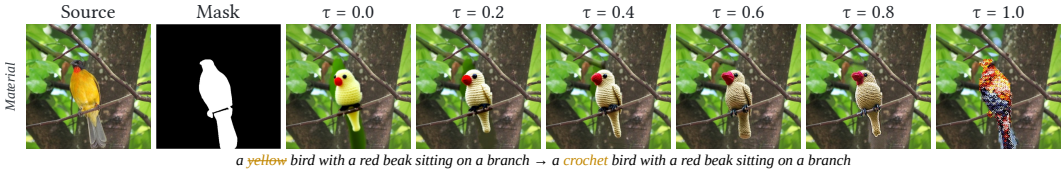


Figure 5: Qualitative sensitivity to the one-step extraction time for a material edit. Only τ changes; the input, mask, prompts, seed, and target denoising configuration remain fixed.

Extraction time. In Figure 5, a clean source read at $\tau = 0.0$ produces the requested crochet texture but compresses the bird into a small, toy-like shape. From $\tau = 0.2$ to 0.8, the generated bird shows more coherent proportions, a clearer elongated body and tail, and more natural contact with the branch while retaining visible crochet stitches. At $\tau = 1.0$, the generated foreground becomes unstable: the bird develops high-contrast patterned artifacts and loses the requested yellow body and red-beak appearance. The qualitative trajectory therefore suggests a broad stable regime rather than a sharp optimum: extraction times from $\tau = 0.4$ to 0.8 all produce coherent edits. The fixed default $\tau = 0.6$ lies near the center of this interval. Table 6 in Appendix C.1 further confirms that performance remains strong across intermediate extraction times, with different metrics attaining their best values at different points.

Table 4: Controlled component and source-inversion ablations.

(a) Spatial components

Variant	HPS $\times 10^2 \uparrow$	BG PSNR↑	Edit CLIP Sim↑	IR $\times 10 \uparrow$	Time (s)↓
FreEdit	**28.57**	**33.82**	**23.72**	**9.54**	3.89
w/o source background K/V	21.93	30.37	18.37	-6.23	**3.63**
All target queries	**28.57**	**33.82**	23.69	9.49	5.51

(b) One-step source memory versus full source inversion

Source memory	Source evals	HPS $\times 10^2 \uparrow$	BG PSNR↑	Edit CLIP Sim↑	IR $\times 10 \uparrow$	Time (s)↓	Peak GiB↓
One-step read	1	**28.57**	33.82	**23.72**	**9.54**	**3.89**	34.08
Full inversion	27	28.22	**34.00**	23.56	9.08	15.22	**33.59**

Source memory and foreground-only denoising. Table 4(a) establishes the complementary roles of the two spatial components. Removing source background K/V sharply degrades visual quality and prompt alignment: final latent composition can restore known pixels, but it cannot supply the scene context required while the new foreground is being generated. Conversely, updating all target queries preserves essentially the same quality but raises runtime from 3.89 to 5.51 seconds. Because final composition retains only the generated foreground, this additional computation is spent on output that is ultimately discarded. Complete metrics and the component runtime breakdown are reported together in Appendix C.3.

One source evaluation versus full inversion. As shown in Table 4(b), replacing 27 source evaluations with one while keeping the target editor fixed reduces runtime from 15.22 to 3.89 seconds ($\mathbf{3.91}\times$) with comparable overall editing quality. This controlled comparison isolates the benefit of the one-step source memory; the complete metric-wise results are reported in Appendix C.4.

5 Conclusion

We presented FreEdit, a training-free mask-guided editor that liberates local editing from multi-step inversion through one source pass and masked-region denoising. Fixed source memory supplies context, latent composition restores the clean background, and QK-CFG preserves foreground organization during appearance edits. Across appearance tasks, FreEdit leads in color accuracy, preservation, style alignment, and aesthetics. Relative to KV-Edit on FLUX.1-dev, it improves all six mean metrics and achieves a $\mathbf{3.58}\times$ speedup. Ablations validate background memory and foreground-only denoising; the one-step read matches full-inversion quality with a $\mathbf{3.91}\times$ speedup. Transfer to Klein, Qwen, and SD3.5 further demonstrates robustness across backbones.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and presentation-level editing of the manuscript, including improvements to clarity, formatting, and the presentation of figures and tables. All research ideas, method design, code implementation, experimental design and execution, data processing, evaluation, and scientific conclusions were developed and completed by the authors. AI tools were not used to generate experimental data or make research decisions. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of this submission.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproduction of our empirical results. Section 3 specifies the source-memory construction, masked target denoising, and QK-CFG operations, including the corresponding attention equations. Section 4 describes the evaluation benchmark, baselines, metrics, hardware, and timing protocol. The appendices provide the complete inference configurations, controlled component and extraction-time ablations, random-seed stability results, cross-backbone evaluations, and additional qualitative comparisons. Our experiments use publicly available model backbones and the PIE-Bench benchmark. Source code is available in our anonymous repository at https://anonymous.4open.science/r/FreEdit_-DBAF.

REFERENCES

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18187–18197. IEEE, 2022.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42(4):1–11, 2023.

Gaspard Beaudouin, Minghan Li, Jaeyeon Kim, Sung-Hoon Yoon, and Mengyu Wang. Delta rectified flow sampling for text-to-image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18662–18672, 2026.

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.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513. IEEE, 2023.

Zhuo Chen, Fanyue Wei, Runze Xu, Jingjing Li, Lixin Duan, Angela Yao, and Wen Li. The devil is in attention sharing: Improving complex non-rigid image editing faithfulness via attention synergy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8237–8246, 2026.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

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.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

Yilei Jiang, Zhen Wang, Yanghao Wang, Jun Yu, Yueting Zhuang, Jun Xiao, and Long Chen. FlowDC: flow-based decoupling-decay for complex image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25757–25766, 2026.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pp. 150–168. Springer, 2024a.

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*, volume 2024, pp. 23395–23422, 2024b.

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.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730. IEEE, 2025.

Yongwen Lai, Chaoqun Wang, and Shaobo Min. FusionEdit: Semantic fusion and attention modulation for training-free image editing. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11907–11911. IEEE, 2026.

Yaowei Li, Yuxuan Bian, Xuan Ju, Zhaoyang Zhang, Junhao Zhuang, Ying Shan, Yuexian Zou, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xiang Liu, Junjun Jiang, Wei Han, Kui Jiang, and Xianming Liu. Cross-controlnet: Training-free fusion of multiple conditions for text-to-image generation. In *International Conference on Learning Representations*, volume 2026, pp. 130155–130182, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Marian Lupaşcu and Mihai Sorin Stupariu. Optimal transport for rectified flow image editing: Unifying inversion-based and direct methods. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6764–6774. IEEE, 2026.

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. *arXiv preprint arXiv:2108.01073*, 2021.

Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2063–2072. IEEE, 2025.

Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7465–7475. IEEE, 2024.

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

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Mateusz Pach, Jessica Bader, Quentin Bouniot, Serge Belongie, and Zeynep Akata. The latent color subspace: Emergent order in high-dimensional chaos. *arXiv preprint arXiv:2603.12261*, 2026.

Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 conference proceedings*, pp. 1–11, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 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 *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22532–22541. IEEE, 2023.

Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024.

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.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Tao Wu, Senmao Li, Yaxing Wang, Shiqi Yang, Kai Wang, and Joost van de Weijer. Training-free image inversion for one-step diffusion models. *Pattern Recognition*, pp. 114063, 2026.

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.

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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9454–9461. IEEE, 2024.

Yu Xu, Fan Tang, Juan Cao, Xiaoyu Kong, Yuxin Zhang, Jintao Li, Oliver Deussen, and Tong-Yee Lee. HeadRouter: A training-free image editing framework for MM-DiTs by adaptively routing attention heads. *ACM Transactions on Graphics*, 45(3):1–14, 2026.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11613–11621, 2026.

Zixin Yin, Xili Dai, Ling-Hao Chen, Deyu Zhou, Jianan Wang, Duomin Wang, Gang Yu, Lionel Ni, Lei Zhang, and Heung-Yeung Shum. Training-free text-guided color editing with multi-modal diffusion transformer. In *International Conference on Learning Representations*, volume 2026, pp. 54707–54737, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

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.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16607–16617. IEEE, 2025.

Zifeng Zhu, Jiaming Han, Jiaxiang Zhao, Minnan Luo, and Xiangyu Yue. GIDE: Unlocking diffusion LLMs for precise training-free image editing. In *European Conference on Computer Vision*, pp. 464–482. Springer, 2026.

A Implementation and Evaluation Details

A.1 Inference Configurations

The main FLUX.1-dev configuration uses 28 target sampling intervals and skips the first, highest-noise interval. The source is read at $\tau = 0.6$. Both denoising guidance and source-extraction guidance are set to 4.0. Background K/V memory is enabled in both double-stream and single-stream blocks. QK-CFG is disabled for the quantitative semantic evaluation. The color and style configurations retain the same source-read time and guidance scales but use different fixed QK-CFG schedules. For color editing, $\lambda_q = \lambda_k = 0.8$ in every single-stream block from target step 4 through step 28. For style editing, $\lambda_q = 0.8$ and $\lambda_k = 0$, applied from single-stream layer 30 onward and from target step 1 through step 20. Each configuration is fixed across its evaluation set; no model weights or per-image parameters are optimized.

A.2 Appearance-Specific Evaluation

The appearance evaluation uses the corresponding PIE-Bench source images, edit masks, and target prompts for every method. The color protocol expands all 40 images in the color category to seven target hues—green, pink, yellow, blue, purple, orange, and cyan—for 280 edits. The style protocol uses all 80 images and their original style-editing prompts. KV-Edit uses its no-skip, re-initialized (NS+RI) variant.

For color, ΔE_{00} measures the CIEDE2000 distance between the target color and the mean edited-region color after conversion from RGB to Lab. Full SSIM compares grayscale source and edited images (Wang et al., 2004), while BG PSNR measures RGB preservation outside the edit mask and is omitted when no unmasked region exists. For style, Style CLIP and Text CLIP are CLIP ViT-L/14 cosine similarities, multiplied by 100, against the target style phrase and the complete target prompt, respectively. Aesthetic quality is measured with the `sac+logos+ava1-l14-linearMSE` predictor.

A.3 Timing Protocol

Except for the separate LCS study (see Appendix G), every runtime is averaged over all 620 source–mask–prompt tuples in the evaluation subset, processed in the same order. Each official implementation runs one image at a time on a dedicated modded RTX 4090 with 48 GB of VRAM. Model initialization is completed before timing starts, and every method is timed over the same interval from input to saved output. Peak memory is measured under the same hardware protocol.

B Random-Seed Robustness

To evaluate sensitivity to sampling randomness, we repeat the complete main evaluation with four random seeds under the same configuration. Table 5 reports the mean and sample standard deviation across the four run-level results. All six principal metrics vary only slightly; the largest coefficient of variation is 1.4%, and every value in the main comparison lies within one standard deviation of its four-seed mean. The reported quantitative behavior is therefore stable across sampling seeds rather than being tied to a particular random draw.

Table 5: Random-seed stability under the main evaluation protocol. Values are mean ± sample standard deviation across four run-level results.

Runs	HPS $\times 10^2 \uparrow$	BG PSNR$\uparrow$	BG LPIPS $\times 10^3 \downarrow$	BG MSE $\times 10^4 \downarrow$	Edit CLIP Sim$\uparrow$	IR $\times 10 \uparrow$
Four seeds	28.57 ± 0.08	33.79 ± 0.06	12.35 ± 0.17	6.78 ± 0.02	23.78 ± 0.07	9.53 ± 0.07

C Additional Ablations and Sensitivity Analyses

C.1 Extraction-Time Sensitivity

The main evaluation uses a fixed default of $\tau = 0.6$. To assess sensitivity to this setting, this ablation varies only the source extraction time while keeping the remaining configuration unchanged. Table 6 extends the single qualitative example in Figure 5 to the full evaluation set. Performance is weaker near the two ends of the extraction range but forms a broad plateau over intermediate times. Settings from $\tau = 0.4$ to 0.8 maintain strong overall performance, and different metrics peak at different points rather than identifying a single universally optimal extraction time. The default $\tau = 0.6$ lies near the center of this stable interval. Taken together, the sweep shows that intermediate source reads provide a reliable balance across perceptual quality, background preservation, and prompt alignment without requiring a narrowly tuned extraction time.

Table 6: Quantitative extraction-time ablation. Only τ differs across settings.

τ	HPS $\times 10^2 \uparrow$	BG PSNR$\uparrow$	BG LPIPS $\times 10^3 \downarrow$	BG MSE $\times 10^4 \downarrow$	Edit CLIP Sim$\uparrow$	IR $\times 10 \uparrow$
0.1	27.50	32.77	14.15	7.67	23.84	8.30
0.2	28.01	33.29	13.02	7.10	**24.01**	8.97
0.3	28.27	33.59	12.52	6.83	23.92	9.28
0.4	28.42	33.75	**12.33**	**6.72**	23.88	9.54
0.5	28.45	33.81	12.34	6.73	23.78	9.44
0.6 (default)	**28.57**	**33.82**	12.48	6.78	23.72	9.54
0.7	28.56	33.66	12.89	6.97	23.68	**9.56**
0.8	28.37	33.33	13.70	7.29	23.65	9.33
0.9	27.94	32.77	14.97	7.93	23.24	8.44

C.2 Source-Prompt Sensitivity

During the source read, the standard configuration encodes the source prompt with T5 and CLIP. Table 7 replaces that prompt with an empty string while leaving the target prompt and every other setting unchanged. The empty string is encoded normally rather than replaced by a zero embedding. All quality and preservation metrics remain essentially unchanged, while prompt-alignment scores improve slightly. The source description can therefore be omitted without materially changing editing quality, indicating that the visual source memory carries most of the source-specific conditioning in this setting.

Table 7: Source-prompt ablation. The only change is whether the source read uses the provided source prompt or an empty string.

Source text	HPS $\times 10^2 \uparrow$	BG PSNR$\uparrow$	BG LPIPS $\times 10^3 \downarrow$	BG MSE $\times 10^4 \downarrow$	Edit CLIP Sim$\uparrow$	IR $\times 10 \uparrow$
Source prompt	**28.57**	**33.82**	**12.48**	**6.78**	23.72	9.54
Empty string	28.56	33.79	12.58	6.82	**23.91**	**9.59**

C.3 Source Memory and Foreground-Only Denoising

Table 8 examines the two components that enable local generation. Removing source background K/V deprives the generated foreground of surrounding scene context and causes a broad collapse in visual quality and prompt alignment. We then keep the source memory but expand target denoising from masked queries to all image queries. This variant nearly matches the quality of FreEdit, but increases runtime from 3.89 to 5.51 seconds ($1.42\times$ slower). Equation 6 retains the masked foreground and discards the generated background. Restricting target denoising to the foreground is therefore computationally justified, with essentially unchanged measured quality in this controlled comparison.

Runtime breakdown. Table 9 breaks down the runtimes from Table 8. Each value is averaged after model loading; the underlying per-sample generation logs have a resolution of 0.1 seconds. Source extraction remains effectively constant across all variants, while updating all target queries

Table 8: Component ablations. BG PSNR, LPIPS, and MSE are computed outside the edit mask. Time is measured after model loading within the same controlled ablation run. "All target queries" denoises the complete target token grid, but final latent composition still retains only its masked foreground output.

Variant	Image quality	Background preservation			Text alignment		Time (s)↓
	HPS $\times 10^2$ ↑	BG PSNR↑	BG LPIPS $\times 10^3$ ↓	BG MSE $\times 10^4$ ↓	Edit CLIP Sim↑	IR $\times 10$ ↑	
FreEdit	**28.57**	**33.82**	12.48	**6.78**	**23.72**	**9.54**	3.89
w/o source background K/V	21.93	30.37	25.02	12.31	18.37	-6.23	**3.63**
All target queries	**28.57**	**33.82**	**12.47**	**6.78**	23.69	9.49	5.51

increases target-editing time from 3.582 to 5.200 seconds, an increase of roughly 45%. The breakdown therefore confirms that the additional runtime comes from denoising the target background that final composition later discards.

Table 9: Runtime decomposition for the component ablations.

Variant	Extraction (s)	Target editing (s)	Total (s)
FreEdit	0.306	3.582	3.888
w/o source background K/V	0.318	3.315	3.634
All target queries	0.312	5.200	5.512

C.4 One Source Evaluation versus Full Inversion

Table 10 directly compares one source evaluation with complete multi-step source inversion. The target editor, initial noise, prompts, masks, and final latent composition are identical. The only difference is whether target denoising reuses one fixed source memory or receives source K/V from a matching noise level at every step. The complete trajectory slightly improves pixel-level background preservation, whereas the one-step read gives higher perceptual quality and prompt reward with similar Edit CLIP alignment. Replacing 27 source evaluations with one therefore reduces runtime from 15.22 to 3.89 seconds ($\mathbf{3.91}\times$) while maintaining comparable overall editing quality.

Table 10: One source evaluation versus full multi-step inversion. BG metrics are computed outside the edit mask. Runtime uses batch size one on the same modded RTX 4090 with 48 GB VRAM after model loading.

Source memory	Source evals	HPS $\times 10^2$ ↑	BG PSNR↑	BG LPIPS $\times 10^3$ ↓	BG MSE $\times 10^4$ ↓	Edit CLIP Sim↑	IR $\times 10$ ↑	Time (s)↓	Peak GiB↓
One-step read	1	**28.57**	33.82	12.48	6.78	**23.72**	**9.54**	**3.89**	34.08
Full inversion	27	28.22	**34.00**	**11.74**	**6.45**	23.56	9.08	15.22	**33.59**

D Transfer to Other Backbones

The source-memory construction requires access to attention projections and an explicit edit mask, and is independent of a FLUX-specific training procedure. We transfer FreEdit to FLUX.2 [klein] 9B, Qwen-Image (Wu et al., 2025), and Stable Diffusion 3.5 Large Turbo (Esser et al., 2024), and evaluate each variant under the main evaluation protocol. FreEdit-Klein uses only Klein's text-to-image generation capability; we additionally evaluate Klein's native four-step editing pipeline. Table 11 also includes the main FLUX.1-dev configuration as the original FreEdit reference. FreEdit-FLUX.1-dev gives the strongest HPS, FreEdit-Klein gives the strongest LPIPS, MSE, Edit CLIP, and ImageReward results, and FreEdit-Qwen gives the strongest PSNR result. FreEdit-Klein improves all six reported metrics over native Klein, with especially large gains in background preservation. Together, these results show that the source-memory construction remains effective beyond FLUX.1-dev rather than depending on a single backbone.

The three transfer configurations follow the same source-read formulation as the main setup. To evaluate transfer without backbone-specific tuning, we keep the source-read time fixed at the main default, $\tau = 0.60$, for Klein, Qwen, and SD3.5. All three variants use four target denoising steps.

Figure 6 complements the quantitative evaluation with local addition, deletion, and replacement examples. In the butterfly-to-parrot edit, each backbone places the new bird on the original leaf;

Table 11: Cross-backbone quantitative results.

Variant	HPS $\times 10^2$ ↑	BG PSNR↑	BG LPIPS $\times 10^3$ ↓	BG MSE $\times 10^4$ ↓	Edit CLIP Sim↑	IR $\times 10$ ↑
Klein (native)	28.11	26.91	49.68	77.42	23.25	8.88
FreEdit-FLUX.1-dev	**28.57**	33.82	12.48	6.78	23.72	9.54
FreEdit-Klein	28.55	34.24	**10.32**	**5.81**	**24.26**	**9.77**
FreEdit-Qwen	28.31	**34.85**	20.85	6.47	24.00	9.73
FreEdit-SD3.5	27.06	33.61	12.87	7.21	23.86	8.71

in the pear-to-banana edit, the vase, sunflowers, table, and wall remain fixed. The collar and hat are added without changing the animals' faces, while the laptop and backpack are removed without changing the subjects or surrounding scenes. The main visual differences across backbones lie in the newly generated foreground details, such as plumage, collar shape, and hat style.

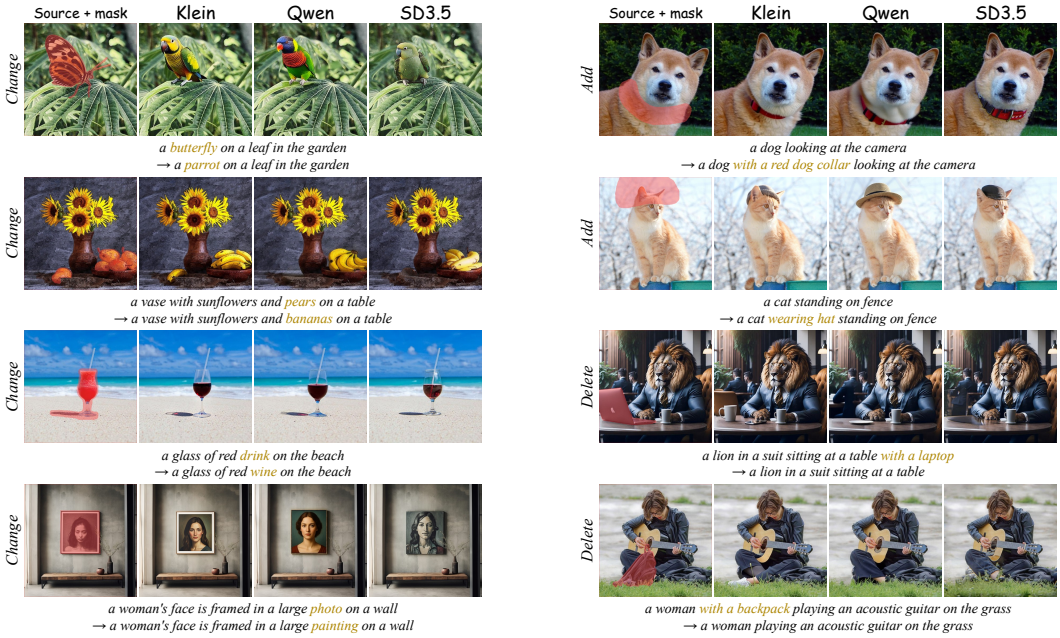


Figure 6: Qualitative transfer to Klein, Qwen, and SD3.5. The first column in each group overlays the local edit mask on the source image.

E Additional Editing Results

This section extends the main qualitative comparison with twelve additional PIE-Bench inputs that are disjoint from those shown in the main paper. The examples cover local replacement, addition, deletion, expression and pose changes, clothing edits, and background modification. Every method column comes from the same official baseline runs used elsewhere in the paper. The source column shows the edit mask overlaid on the input image.

Semantic and attribute edits. In Figure 7, FreEdit changes the beach drink into a wine glass and the illustrated rat into a pig while retaining their original settings. The earrings and laughing-expression edits remain localized to the face, and removing the laptop leaves the lion, its paws, the table, and the cafe scene intact. The straight-hair edit similarly preserves the woman's face, clothing, glasses, and city background.

Pose, material, and scene edits. Figure 8 shows that the running edit changes the woman's leg and skirt motion while retaining the sunset path. The sweater edit preserves the man's identity and pose, and the perch edit replaces the branch with a rock without redrawing the bird. For larger scene changes, FreEdit inserts water beneath the angel, mountains behind the man, and a floral wall behind the bed while leaving the unmasked subjects and foreground objects in place.

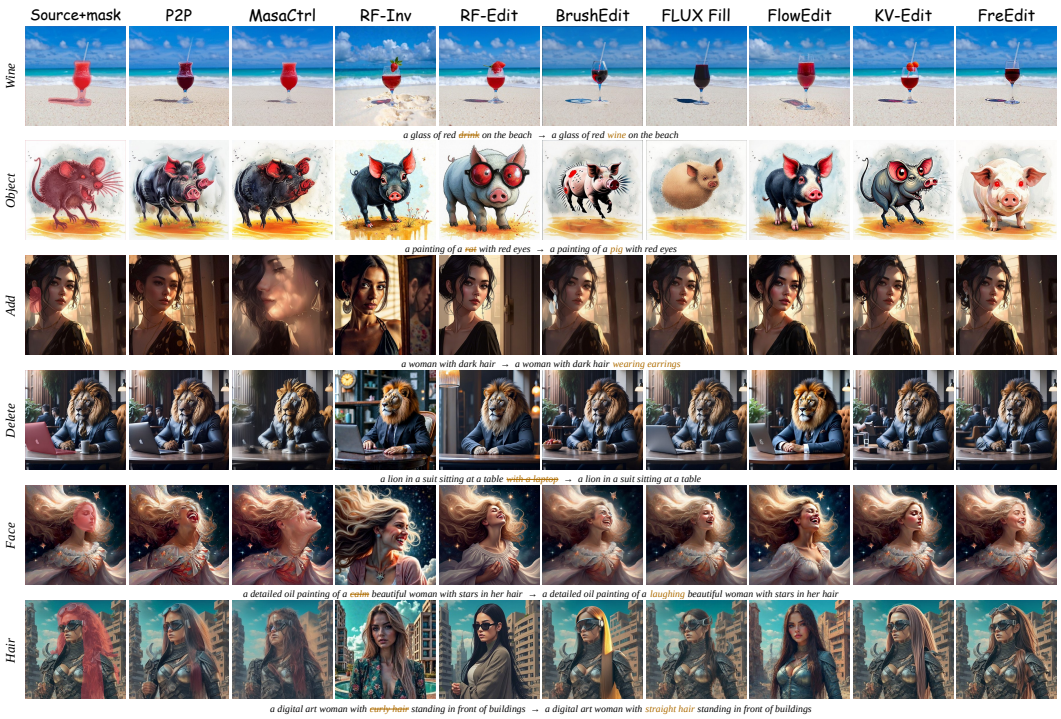


Figure 7: Additional editing comparisons. The six rows show object replacement, local addition and deletion, and facial-attribute edits.

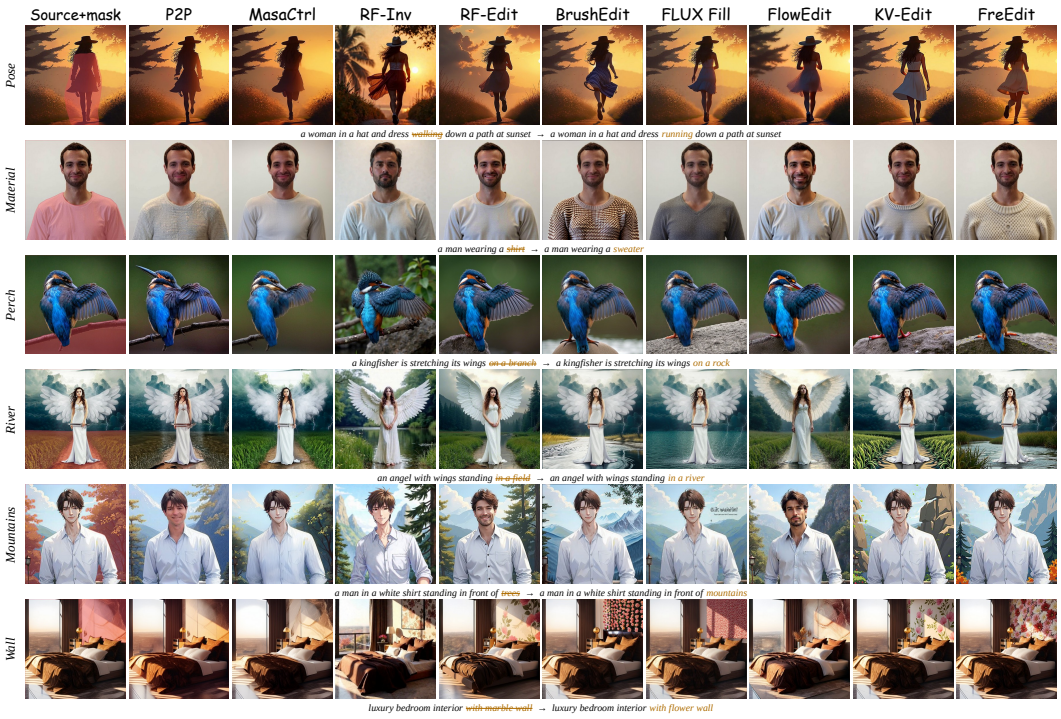


Figure 8: Further editing comparisons on pose, clothing, and background changes.

F Additional Color Editing Results

Figure 9 evaluates color editing across natural objects, clothing, furniture, and illustrations. The rose, umbrella, bird, mushroom, shirt, and berries reach distinct target hues while retaining their shapes and surrounding scenes. The more textured flower bouquet and teacup rows expose the remaining limitation: their layouts remain recognizable, but some target columns retain muted or residual source colors. This pattern exposes a trade-off in which strong structural preservation can coincide with an under-realized target hue. The next section studies this failure mode.

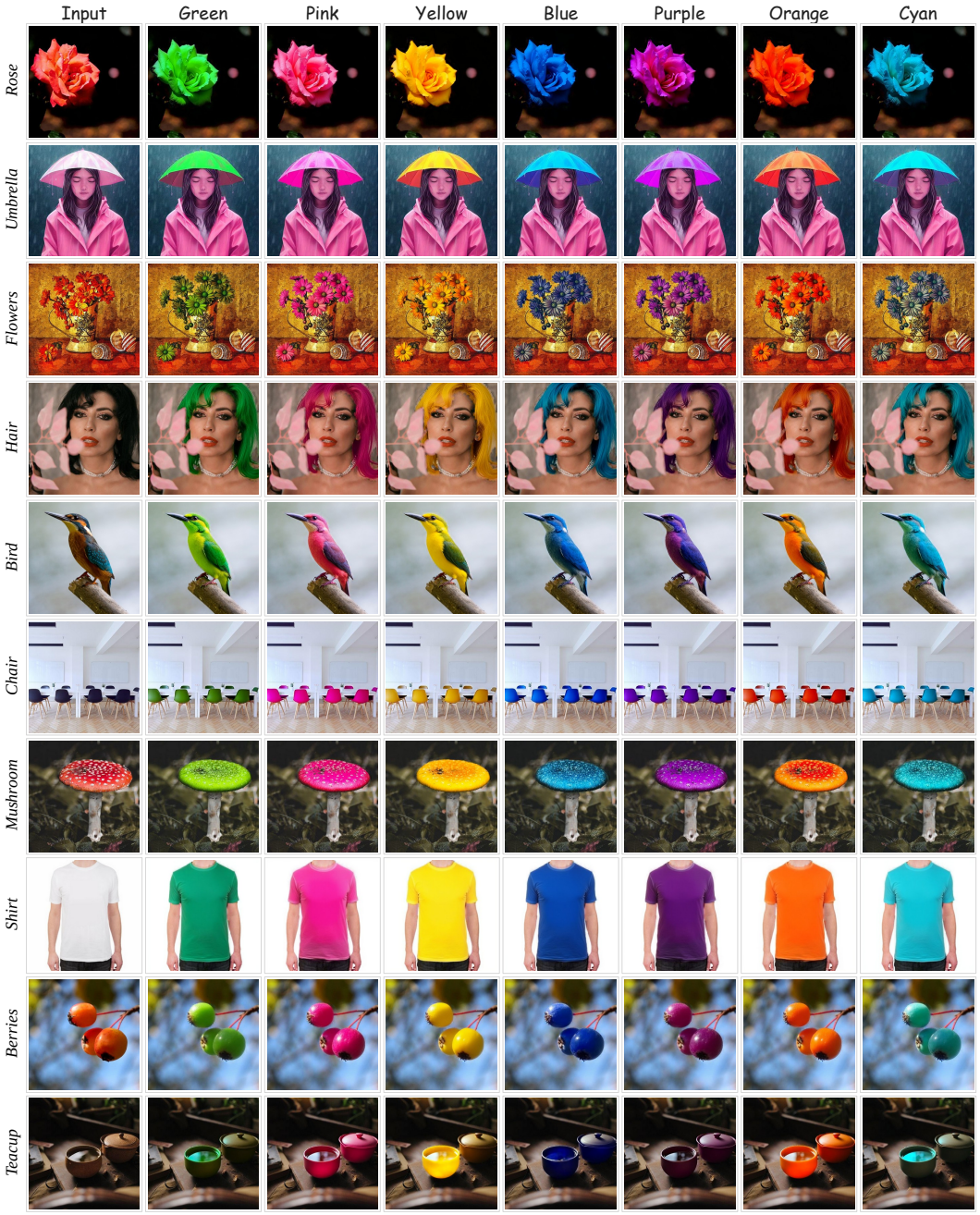


Figure 9: Multi-object color editing. Rows are source examples and columns are target colors; each result changes the local masked subject.

G Latent Color Steering

G.1 A Complementary Color Prior

QK-CFG supports appearance editing while retaining source organization, but the results in Figure 9 show that some requested hues remain under-realized. Increasing text guidance alone may strengthen the requested color while also changing object structure. This motivates an optional color control that acts in a representation designed specifically for color.

The Latent Color Subspace (LCS) identifies a three-dimensional color basis in the FLUX.1 VAE latent space (Pach et al., 2026). We use its direct translation operation as an optional refinement of FreEdit. FLUX encodes an image into a 16-channel VAE latent and groups each 2×2 spatial block into a 64-dimensional patch vector. The scale and shift applied by the FLUX autoencoder define what we call the processed VAE latent space. Let $p_i \in \mathbb{R}^{64}$ be the predicted clean vector for foreground patch i, where $i = 1, \ldots, N$ indexes the N foreground patches selected by the edit mask. Let $B \in \mathbb{R}^{64\times 3}$ be an orthonormal color basis in this space and let $\mu \in \mathbb{R}^{64}$ be its calibration mean. We decompose each patch into a color coordinate $c_i = B^\top(p_i - \mu)$ and a residual $r_i = (p_i - \mu) - Bc_i$. Given a target color coordinate $c^\star$, we modify the foreground as follows:

$$p_i' = \mu + B[c_i + \alpha\,(c^\star - \bar{c})] + r_i, \qquad \bar{c} = \frac{1}{N}\sum_{j=1}^{N} c_j, \tag{10}$$

Here, α controls the strength of the color change and $\bar{c}$ is the mean color coordinate of the current predicted foreground. Every foreground patch receives the same displacement in the color subspace, while its residual r_i remains unchanged. We obtain $c^\star$ by encoding a solid patch of the requested color with the same VAE. The basis is calibrated once from solid-color images, and its first three principal components explain 97.2% of the calibration variance.

We apply Equation 10 to the predicted clean latent instead of adding a color direction directly to the model velocity. At each selected target denoising step n, whose noise level is σ^n, the transformer predicts $v_{\mathrm{t}}^n = v_\theta(z_{\mathrm{t}}^n, \sigma^n)$. Under this parameterization, its prediction of the clean target latent is $\hat{z}_{\mathrm{t}}^{0|n} = z_{\mathrm{t}}^n - \sigma^n v_{\mathrm{t}}^n$. We apply LCS only to its foreground patches to obtain $\hat{z}_{\mathrm{t}}'^{0|n}$, and then replace the velocity used by the standard Euler update with

$$v_{\mathrm{t}}'^{n} = \frac{z_{\mathrm{t}}^n - \hat{z}_{\mathrm{t}}'^{0|n}}{\sigma^n}. \tag{11}$$

LCS therefore affects each selected sampling update through $v_{\mathrm{t}}'^{n}$, while the color operation itself remains defined in the clean VAE latent space. It does not change the cached source Q/K/V or the QK-CFG rule. Foreground Q/K continues to follow the source-biased QK-CFG rule in support of source organization, while LCS supplies an explicit color displacement. The same edit mask is resampled to the VAE patch grid, and FreEdit already stores only the selected foreground patches in its target latent sequence. The operation is therefore local and requires no additional segmentation model. We study LCS as a compatible color refinement on top of FreEdit; setting its strength to zero recovers the original editor exactly.

G.2 Paired Color Study

We evaluate LCS on the same 280 color edits and the same base FreEdit configuration used in Table 3. Within each pair, the source, mask, prompt, seed, and generation configuration are fixed; the only difference is whether LCS is enabled. When enabled, LCS uses $\alpha = 0.1$ from step 1 through step 28. Color and structure metrics use all pairs; background metrics are omitted when no unmasked region exists.

We evaluate color accuracy and structural preservation separately. Color errors are computed from the mean RGB value inside the original edit mask: CIEDE2000 distance (ΔE_{00}), circular HLS hue error in degrees (ΔH), and absolute saturation and lightness errors as percentages (ΔS and ΔL). For preservation, we convert the source and edited images to grayscale and report foreground SSIM inside the original mask, full-image SSIM, SqueezeNet LPIPS (Iandola et al., 2016; Zhang et al., 2018), and DINOv2 CLS-token cosine distance (Oquab et al., 2023). The LPIPS and DINOv2

values use their raw scales. Background PSNR is computed over pixels outside the original mask, while background SSIM uses only locations whose full 11×11 support lies outside the mask and image boundary.

Table 12: Paired color-editing results with $\lambda_q = \lambda_k = 0.8$. Panel (a) reports mean-foreground color accuracy; panel (b) reports grayscale structure, background preservation, and the total runtime including source extraction.

(a) Color accuracy

Method	$\Delta E_{00} \downarrow$	ΔH (°)↓	ΔS (%)↓	ΔL (%)↓
FreEdit	26.48	28.83	57.60	12.04
FreEdit + LCS	**15.00**	**8.39**	**36.71**	**7.66**

(b) Structure and background preservation

Method	FG SSIM↑	Full SSIM↑	LPIPS↓	DINOv2↓	BG PSNR↑	BG SSIM↑	Time (s)↓
FreEdit	**0.433**	**0.775**	**0.099**	**0.119**	**31.92**	**0.933**	**4.046**
FreEdit + LCS	0.431	0.769	0.108	0.132	30.79	0.925	4.078

As shown in Table 12, LCS markedly improves target-color accuracy. Mean ΔE_{00} decreases by 43%, from 26.48 to 15.00, while mean hue error falls by 71%, from 28.83° to 8.39°. Saturation and lightness errors each decrease by approximately 36%. The agreement across all four color measures shows that LCS corrects under-realized color changes rather than merely increasing saturation.

These color gains come with small structural and background trade-offs. Foreground grayscale SSIM remains essentially unchanged at 0.433 versus 0.431, while LPIPS, DINOv2 distance, background PSNR, and background SSIM degrade modestly. Total runtime increases only from 4.046 to 4.078 seconds. Overall, LCS substantially improves color fidelity with small structural and background trade-offs and negligible runtime overhead.

Figure 10 illustrates the same behavior qualitatively. The base editor leaves the requested green and yellow chairs close to gray or beige, the blue and purple teacups dark or weakly saturated, and the flower colors muted. LCS strengthens the requested hues while retaining the chair arrangement, bird pose, cup geometry, flower layout, and surrounding scenes.

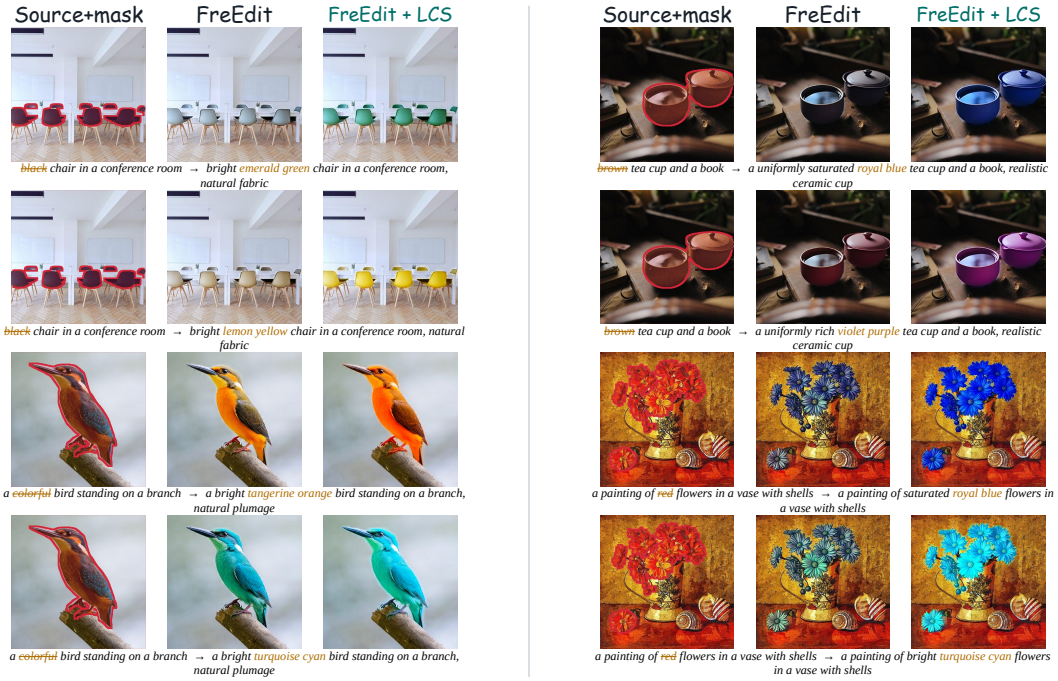


Figure 10: Selected under-realized color edits corrected by latent color steering. Each group compares the same source, mask, prompt, and seed.