Why Do DiT Editors Drift? Plug-and-Play Low Frequency Alignment in VAE Latent Space

Xiaoce Wang[†] [♠] **Sifan Zhou**[†] [♡] **Kaifei Wang**[†] [◇] **Leli Xu** [♠]
Xuerui Qiu [♣] **Tao He** [△] **Ming Li** [⋆‡]
[♠]Tsinghua University [♡]Carnegie Mellon University [◇]Peking University
[♣]CASIA [△]University of Electronic Science and Technology of China
[‡]School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen
[†]These authors contributed equally. [⋆]Corresponding author.
wangxc23@mails.tsinghua.edu.cn



Figure 1: **VAE-LFA improves both visual quality and style consistency during multi-turn editing.**

Abstract

Recent advances in diffusion transformers (DiTs) have enabled promising single-turn image editing capabilities. However, multi-turn editing often leads to progressive semantic drift and quality degradation. In this work, we study this problem from a latent-space frequency perspective by decomposing the editing process into two functional components: VAE and DiT. Through systematic analysis in the VAE latent space, we uncover that the DiT introduces dominant low-frequency drift that accumulates as semantic misalignment across editing rounds, while the VAE contributes comparatively stable reconstruction bias. Based on this insight, we propose **VAE-LFA** (Low Frequency Alignment), a training-free, plug-and-play method that performs alignment in VAE latent space. VAE-LFA decomposes latent discrepancies across editing rounds via low-pass filtering, and aligns low-frequency statistics to an exponential moving average of previous rounds, effectively suppressing accumulated semantic drift while preserving high-frequency details. Our method requires *no retraining, ground-truth priors, or access to diffusion parameters*, making it applicable to both white-box and black-box DiT editors. For white-box models, VAE-LFA is seamlessly integrated into the editing pipeline by eliminating redundant VAE round trips; for black-box models, it operates via an off-the-shelf VAE to perform inter-round latent alignment. Extensive experiments demonstrate that VAE-LFA improves semantic consistency and visual fidelity across diverse multi-turn editing scenarios, including both controlled and in-the-wild images.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Recent advances in instruction-based image editing models built upon diffusion transformers (DiTs) and flow matching have achieved remarkable improvements in prompt-following, realism, and visual quality for single-round editing [29, 19, 38, 2, 9, 13, 8, 44]. Despite these advances, their performance degrades significantly under multi-turn (*e.g.*, 5+ turns) iterative editing, often leading to severe semantic drift and quality deterioration [48, 21, 14]. This limitation poses a critical challenge for real-world deployment as users cannot be expected to articulate all editing intents with sufficient precision in a single prompt, and iterative refinement is inherent to any creative workflow. While prior works have proposed various remedies for multi-turn degradation, they either demand full-model retraining [27, 31, 1], require internal access to denoising trajectories [14], or rely on model-specific inference signals such as internal features or velocity fields [21]. Nevertheless, the community heavily relies on black-box editing models like GPT Image, Qwen Image 2.0 [38], Seedream 4.0 [30] and GLM-Image [43], yet these methods cannot be directly deployed on black-box or API-based editors without modification. This calls for a simple and effective remedy for editing degradation that **requires no internal access.**

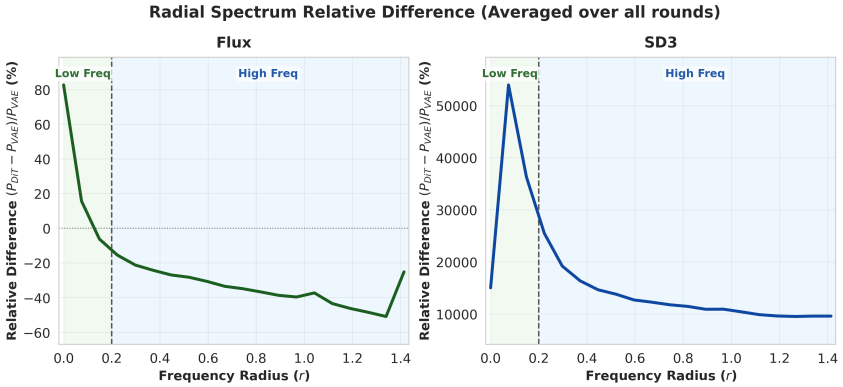

Figure 2: **Cross-architecture validation of frequency-wise drift.** Relative radial-spectrum difference between DiT-only and VAE-only loops over 10 rounds: $\Delta P(r) = \frac{P_{\text{DiT}}(r) - P_{\text{VAE}}(r)}{P_{\text{VAE}}(r)} \times 100\%$. On both FLUX.2 and SD3, DiT transitions dominate low-frequency drift ($r < 0.2$), while VAE bias is stronger at higher frequencies. The vertical line marks $r = 0.2$.

Moreover, fundamental questions remain unanswered: *Why do DiT editors drift? Which component is the primary cause?* To answer these questions, we draw upon recent evidence that frequency-based analysis in latent space provides a principled perspective [26, 25]. We focus on the **VAE latent space**, as the VAE encoder–decoder bottleneck is a fundamental component shared across virtually all modern DiT-based editors, making it a universal interface for component-level drift attribution.

Specifically, we decompose the editing pipeline into two components, the VAE for encoding and decoding, and the DiT for iterative refinement, and systematically analyze frequency-wise drift in the latent space using no-operation prompts to isolate intrinsic degradation. As shown in Fig. 2 and Fig. 6, our analysis reveals a striking asymmetry: semantic degradation is predominantly driven by accumulated low-frequency deviations introduced by the DiT over multiple iterations, whereas the VAE loop merely induces a stable, spectrally uniform reconstruction bias. This finding is counter-intuitive, challenging the prevailing hypothesis that repeated VAE cycles dominate multi-turn error accumulation [1, 42]. We further discover that VAE acts as an implicit regularizer that partially curbs DiT-induced low-frequency drift, rather than purely introducing degradation, as shown in Fig. 6.

Motivated by this asymmetry, we propose **VAE-LFA** (Low Frequency Alignment) (Fig. 3), a training-free and plug-and-play framework that operates directly in the VAE latent space. The key idea is to selectively constrain the low-frequency components that dominate semantic drift, while leaving high-frequency components unconstrained to preserve fine details and editing flexibility. By stabilizing the low-frequency latent statistics across editing rounds, our method effectively mitigates the accumulation of semantic misalignment without restricting the model's ability to perform meaningful edits. Crucially, VAE-LFA requires no retraining, no ground-truth priors, and no access to internal model parameters, making it applicable to both white-box and black-box DiT-based editors. This design enables seamless integration into existing editing pipelines, either by directly operating within

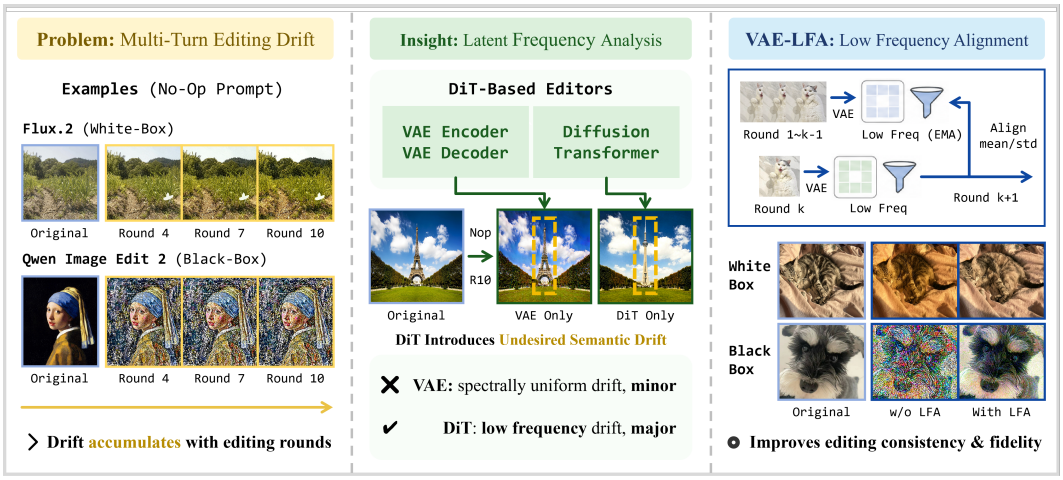

Figure 3: **Overview of VAE-LFA**: component-wise drift analysis and low-frequency latent alignment.

the model (for white-box settings) or by interfacing through an external latent space (for black-box settings), without modifying the underlying model. Extensive experiments demonstrate that VAE-LFA mitigates quality degradation across diverse editing scenarios. Across no-op, cycle, and long-chain protocols, our method improves semantic consistency and perceptual fidelity while preserving instruction-following capability (Fig. 1, Fig. 7, Tab. 1, and Tab. 2). Overall, our contribution can be summarized as:

- We present an interpretable analysis of multi-turn editing degradation from a VAE latent frequency perspective, showing that semantic drift is mainly driven by low-frequency drift.
- We provide a controlled decomposition of the editing pipeline, revealing an asymmetry between DiT and VAE, where the former dominates semantic drift while the latter introduces stable reconstruction bias with partial regularization effects.
- We propose VAE-LFA, a training-free and plug-and-play method that mitigates drift via low-frequency latent alignment, requiring no retraining or internal model access and supporting both white-box and black-box DiT-based editors with consistent performance improvements.

2 Related Works

Instruction-based and multi-turn image editing. Instruction-based editing modifies an input image according to natural-language instructions while preserving irrelevant content [2, 44, 8, 9, 13]. Recent DiT- and flow-matching-based editors further improve prompt following and realism [19, 38, 30, 43]. However, these systems are mostly optimized and evaluated in one-shot settings; under iterative use, small per-turn deviations accumulate into identity drift, color shift, and artifacts [48, 21, 14], especially when the editor is accessible only through an API.

Remedies for iterative degradation. Existing remedies differ in the access and supervision they require. Training-based methods learn iterative or in-context editing from richer supervision, *e.g.*, from videos [27] or multi-task data [31]. Others target degradation mechanisms directly: REED-VAE retrains the autoencoder to reduce accumulated artifacts [1], Layer-wise Memory stores internal latents and prompt embeddings from previous steps [14], and FreqEdit injects high-frequency features from reference velocity fields [21]. These methods are complementary to ours, but require training, internal representations, or inference-process access; we instead seek a minimal post-hoc intervention at the VAE latent interface, applicable to both white-box and black-box editors.

VAE bottlenecks and latent-space analysis. Most DiT-based generators employ an autoencoder bottleneck to reduce computation [29, 15]. The autoencoder is not an identity map: repeated pixel–latent transitions accumulate reconstruction bias in iterative editing [1, 42], and latent-space structure itself affects generative modeling [17, 10]. These works motivate analyzing the VAE latent space, but do not attribute multi-turn drift to pipeline stages; we use the latent space as a shared interface to separate the DiT transition from the VAE round trip.

Frequency perspectives in generative models. Latent power spectra and frequency-to-frequency correspondence have been linked to diffusion quality [26, 25], and low/high frequencies are associated with global semantics and local details, respectively [24, 46]. Frequency-aware editing exploits this split for controllability [39, 16], but does not address multi-turn drift attribution or black-box stabilization. We decompose drift into frequency-specific components in the VAE latent space, revealing which pipeline stage corrupts which semantic band, and derive a training-free remedy that requires no internal model access. A detailed discussion is deferred to Appendix B.

3 Observations

We formalize multi-turn editing bias (Sec. 3.1), build a frequency perspective on VAE latents (Sec. 3.2), and disentangle VAE- and DiT-induced drift through controlled analysis (Sec. 3.4).

3.1 Formulation: Multi-turn Editing Bias

A multi-turn editor induces a trajectory $\mathbf{x}^{(k)} = \mathcal{T}_{p^{(k)}}(\mathbf{x}^{(k-1)})$, where $p^{(k)}$ is the instruction at turn k. Let $\mathcal{P}_{\text{no-op}}$ denote no-operation prompts (*e.g.*, *"keep the image unchanged"*). Ideally, no-op editing is an identity map, $\mathcal{T}_{p_{\text{no-op}}}(\mathbf{x}) = \mathbf{x}$ for $p_{\text{no-op}} \in \mathcal{P}_{\text{no-op}}$; its violation in practice defines *multi-turn editing bias*, which accumulates into visible semantic drift over repeated turns.

In the VAE latent space, let E, D denote the VAE encoder/decoder and G_p the prompt-conditioned DiT transition, with $\Phi_p(\mathbf{z}) = E(D(G_p(\mathbf{z})))$. The no-op latent bias decomposes into

$$\Phi_{p_{\text{no-op}}}(\mathbf{z}) - \mathbf{z} = \underbrace{G_{p_{\text{no-op}}}(\mathbf{z}) - \mathbf{z}}_{\text{DiT bias}} + \underbrace{\Phi_{p_{\text{no-op}}}(\mathbf{z}) - G_{p_{\text{no-op}}}(\mathbf{z})}_{\text{VAE round-trip bias}}. \tag{1}$$

This motivates our component-wise analysis, since each per-turn bias is evaluated at an already contaminated latent and can accumulate over long trajectories. We stress that no-op editing serves as a diagnostic probe: under meaningful prompts, intended displacement and intrinsic drift are coupled rather than additively separable. A detailed formulation is provided in Appendix C.

3.2 Viewing Bias from a Frequency Perspective

Frequency decomposition in latent space. A VAE latent $\mathbf{z} \in \mathbb{R}^{C\times H\times W}$ can be viewed as C two-dimensional spatial feature maps. We define frequency by applying a 2D Fourier transform to each channel independently over the spatial dimensions (H, W), while the channel dimension is not transformed. Under this view, each latent channel contains a zero-frequency component and non-zero-frequency components, also known as DC and AC components, respectively.

For a latent tensor $\mathbf{u} \in \mathbb{R}^{C\times H\times W}$, we compute channel-wise spatial statistics

$$\mu_c(\mathbf{u}) = \frac{1}{HW}\sum_{i,j}\mathbf{u}_{c,i,j}, \quad \sigma_c(\mathbf{u}) = \left(\frac{1}{HW}\sum_{i,j}\big(\mathbf{u}_{c,i,j} - \mu_c(\mathbf{u})\big)^2\right)^{1/2}. \tag{2}$$

The mean $\mu_c(\mathbf{u})$ corresponds to the DC component of channel c up to the Fourier normalization factor. After removing this DC component, the centered signal $\mathbf{u}_c - \mu_c(\mathbf{u})$ contains only AC components. By Parseval's identity, $\sigma_c^2(\mathbf{u})$ is proportional to the total energy of the non-DC Fourier coefficients, and thus measures the average AC energy of channel c.

We then define low- and high-frequency latent components using a spatial low-pass filter. Let L denote a low-pass filter applied independently to each latent channel. A latent $\mathbf{z}$ is decomposed as

$$\mathbf{z}_{\text{low}} = L(\mathbf{z}), \quad \mathbf{z}_{\text{high}} = \mathbf{z} - L(\mathbf{z}). \tag{3}$$

Computed on $\mathbf{z}_{\text{low}}$, the statistics in Eq. (2) capture the channel-wise DC offset and the energy scale of low-band AC components; these are exactly the quantities aligned by VAE-LFA (Sec. 4.1).

3.3 Empirical Analysis of Low and High Frequency Components

To analyze the specific characteristics of low and high frequency components, we employ a filter in the VAE latent space of FLUX.2 [18], a DiT-based editor, to isolate the two, before decoding them back into pixel space. As demonstrated in Fig. 4, we identify that the low-frequency clearly represents

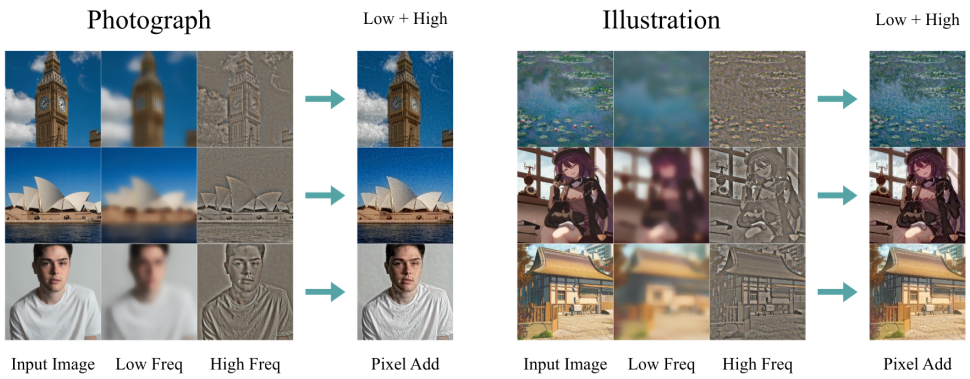

Figure 4: **Low/high-frequency components in the FLUX.2 VAE latent space.** Low frequencies capture global appearance (e.g., color); high frequencies retain local details (e.g., texture).

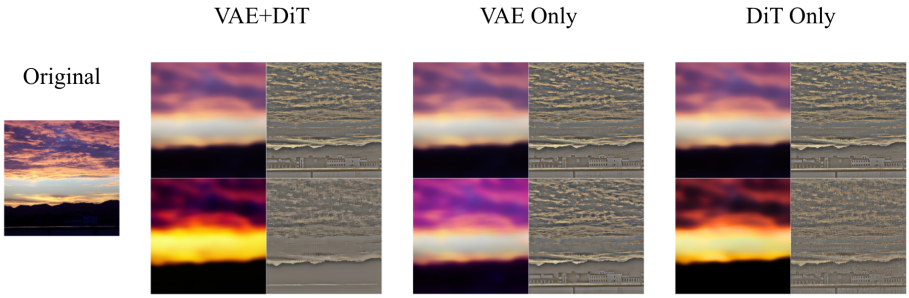

Figure 5: Low/high-frequency drift between round 1 (up) & 10 (down) under three different settings.

macro information, such as tone, style and overall structure, while high frequency components represent **micro details**, such as textures and outlines.

While VAE decoders are nonlinear maps from latent space to image space [15], our experiments in Fig. 4 suggest approximate additivity for the frequency decomposition considered here. Specifically, the visual effects observed by decoding low and high frequency components separately are consistent to a degree with their roles in the jointly decoded image. This makes separate low/high decoding a meaningful diagnostic of their image space representations. It also supports our choice to process the low and high frequency residuals separately, before recombining them by simple latent addition.

3.4 Disentangling VAE and DiT in VAE Latent Space

VAE-induced drift and DiT-induced drift. We isolate the two terms of Eq. (1) via two controlled no-op transitions: **DiT-only**, $\mathbf{z}^{(k)} = G_{p_{\text{no-op}}}(\mathbf{z}^{(k-1)})$, which feeds DiT outputs directly to the next turn without VAE round trips, and **VAE-only**, $\mathbf{z}^{(k)} = E(D(\mathbf{z}^{(k-1)}))$, which runs pure encode–decode cycles without the DiT. Using an average-pooling filter to separate frequency bands, we perform 10 rounds of spectral analysis on FLUX.2 and Stable Diffusion 3 (Fig. 2). It is evident that DiT by itself contributes more significantly towards the low-frequency component. We then conduct visualizations of the low/high frequency status of the two settings, alongside the default baseline, as demonstrated in Fig. 5.

Motivations of VAE-LFA. As shown in Fig. 6, removing intermediate VAE encode–decode operations substantially reduces high-frequency drift, but has a weaker effect on low-frequency drift. In fact, the inter-round VAE round trip sometimes preserves global attributes such as color more stably, suggesting that the VAE may partially regularize low-frequency drift. Motivated by the VAE objective, which encourages the latent posterior to match a Gaussian prior parameterized by mean and variance [15], we align the channel-wise mean and standard deviation of the low-frequency component. In the frequency domain, the mean corresponds to the DC component, i.e., the global channel-wise offset, while the variance controls the energy scale of centered non-DC components.

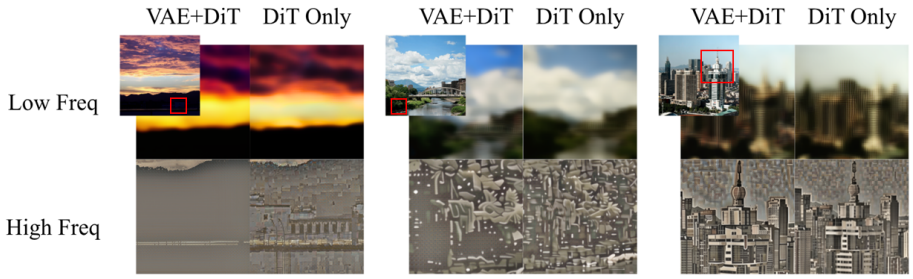

Figure 6: Removing VAE round trips improves high-frequency fidelity but leaves low-frequency drift.

Thus, mean/std alignment calibrates low-frequency statistics without directly constraining high-frequency residuals, leaving room for valid edits.

4 Method

4.1 VAE-LFA

Based on our observations, DiT is the primary source of semantic drift, while VAE introduces stable, spectrally uniform noise. This makes the VAE component easier to handle: when possible, we remove inter-round VAE encode-decode round trips. Therefore, VAE-LFA mainly alleviates the degradation induced by DiT by aligning the low-frequency components of the VAE latent space.

Mitigating drift induced by DiT. VAE-LFA inserts an alignment step in the VAE latent space after the DiT transition and before the next editing turn. Following the frequency decomposition in Sec. 3.2, we instantiate the low-pass operator L as channel-wise average pooling: $L(\mathbf{z}) = \mathrm{AvgPool}_\rho(\mathbf{z})$, where ρ is the pooling window size. For the k-th turn, we apply this filter to the DiT output latent and decompose it into a low-frequency component and a high-frequency residual as: $\boldsymbol{\ell}^{(k)} = L(\tilde{\mathbf{z}}^{(k)}), \mathbf{h}^{(k)} = \tilde{\mathbf{z}}^{(k)} - \boldsymbol{\ell}^{(k)}$.

To provide a reference for alignment, we maintain momentum statistics computed only from pre-alignment low-frequency components. The momentum anchors are initialized with $\boldsymbol{\ell}^{(0)}$ and update after each turn according to the following rules:

$$\mathbf{m}_\mu^{(k)} = \alpha_\mu \mathbf{m}_\mu^{(k-1)} + (1-\alpha_\mu)\mu(\boldsymbol{\ell}^{(k)}), \quad \mathbf{m}_{\log\sigma}^{(k)} = \alpha_\sigma \mathbf{m}_{\log\sigma}^{(k-1)} + (1-\alpha_\sigma)\log\sigma(\boldsymbol{\ell}^{(k)}), \tag{4}$$

where $\alpha_\mu, \alpha_\sigma \in (0,1)$ are momentum coefficients. The momentum anchor stores exponential moving averages (EMA) of the mean and log-standard-deviation. Given these target statistics, VAE-LFA aligns the current low-frequency component channel-wise:

$$\boldsymbol{\ell}_{\mathrm{align},c}^{(k)} = \mathbf{m}_{\mu,c}^{(k-1)} + \frac{\exp\left(\mathbf{m}_{\log\sigma,c}^{(k-1)}\right)}{\sigma_c\left(\boldsymbol{\ell}^{(k)}\right) + \epsilon}\left(\boldsymbol{\ell}_c^{(k)} - \mu_c\left(\boldsymbol{\ell}^{(k)}\right)\right), \quad c = 1, \ldots, C. \tag{5}$$

The final latent is obtained by combining the aligned low-frequency component with the original high-frequency residual: $\hat{\mathbf{z}}^{(k)} = \boldsymbol{\ell}_{\mathrm{align}}^{(k)} + \mathbf{h}^{(k)}$.

4.2 The Implementation of VAE-LFA

On white-box models. For open-source editors with full internal access, VAE-LFA is inserted **between consecutive DiT forward passes**, eliminating redundant VAE round trips. At turn k, the DiT output $\tilde{\mathbf{z}}^{(k)}$ is aligned into $\hat{\mathbf{z}}^{(k)}$, which is directly fed to the next DiT turn. The VAE decoder is used only to obtain the visible image $\mathbf{x}^{(k)} = D(\hat{\mathbf{z}}^{(k)})$, while the encoder is bypassed between turns. This reduces latency and removes VAE round-trip noise from the drift accumulation chain; the only persistent item is the EMA statistics.

On black-box models. For API-only editors, we access only image-level inputs and outputs, without internal features. We interleave an off-the-shelf VAE $E_{\mathrm{ext}}, D_{\mathrm{ext}}$ (e.g., SD-VAE-ft-ema [29]) between rounds: $\mathbf{x}^{(k-1)} \xrightarrow{\text{API}} \tilde{\mathbf{x}}^{(k)} \xrightarrow{E_{\mathrm{ext}}} \tilde{\mathbf{z}}^{(k)} \xrightarrow{\text{VAE-LFA}} \hat{\mathbf{z}}^{(k)} \xrightarrow{D_{\mathrm{ext}}} \mathbf{x}^{(k)}$. Although the external VAE may differ from the editor's native latent space, VAE-LFA remains effective under cross-VAE deployment (Sec. 5), likely because it acts on low-frequency statistics. Since low-frequency

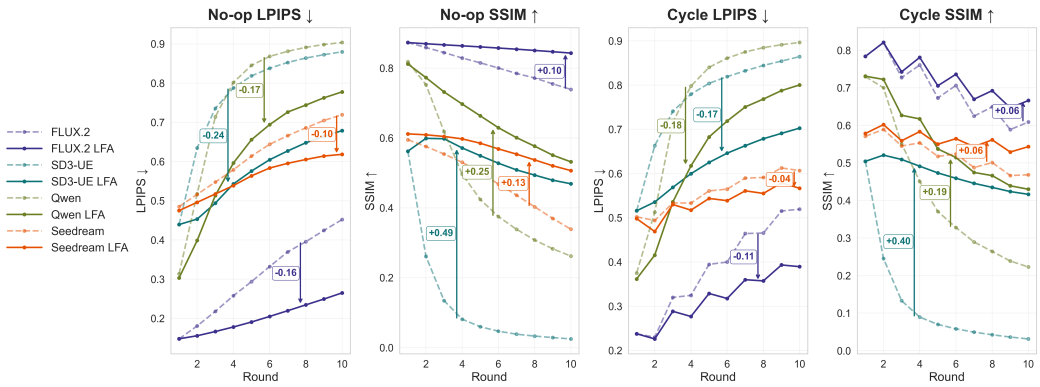

Figure 7: **Per-round metrics curve of no-op and cycle editing.** On average, VAE-LFA improves visual consistency both from a macro (LPIPS) and micro (SSIM) perspective.

Table 1: **No-op and cycle editing results for all models.** We abbreviate Qwen Image 2.0 as Qwen and Seedream 4.0 as Seedream. The better result between LFA/non-LFA variants is shown in **bold**.

Model	photograph									illustration								
	creature			architecture			scenery			creature			architecture			scenery		
	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑
No-Op, Round 10																		
White-Box																		
FLUX.2	0.36	0.09	0.81	0.42	0.15	0.70	0.43	0.13	0.77	0.46	0.15	0.70	0.50	0.16	0.72	0.56	0.15	0.71
FLUX.2+LFA	**0.17**	**0.04**	**0.93**	**0.27**	**0.09**	**0.79**	**0.25**	**0.07**	**0.83**	**0.26**	**0.07**	**0.88**	**0.29**	**0.09**	**0.80**	**0.33**	**0.08**	**0.83**
SD3-UE	0.87	0.42	0.03	0.90	0.48	0.01	0.88	0.41	0.01	0.82	0.46	0.08	0.89	0.52	0.02	0.92	0.44	0.01
SD3-UE+LFA	**0.66**	**0.26**	**0.50**	**0.68**	**0.32**	**0.42**	**0.69**	**0.30**	**0.47**	**0.61**	**0.27**	**0.55**	**0.69**	**0.33**	**0.45**	**0.74**	**0.32**	**0.43**
Black-Box																		
Qwen	0.82	0.22	0.44	0.77	0.22	0.43	0.78	0.22	0.46	0.72	0.21	0.52	0.69	0.22	0.44	0.80	0.21	0.50
Qwen+LFA	**0.66**	**0.12**	**0.68**	**0.64**	**0.15**	**0.62**	**0.60**	**0.11**	**0.71**	**0.57**	**0.14**	**0.66**	**0.54**	**0.16**	**0.60**	**0.68**	**0.15**	**0.67**
Seedream	0.66	0.15	0.54	**0.61**	0.19	0.42	0.61	0.18	0.48	0.64	0.20	0.45	0.59	0.21	**0.42**	**0.61**	0.17	0.56
Seedream+LFA	**0.65**	**0.13**	**0.57**	0.62	**0.17**	**0.45**	0.61	**0.16**	**0.51**	**0.62**	**0.19**	**0.50**	0.59	**0.20**	0.41	0.62	**0.16**	**0.60**
Cycle, Round 10																		
White-Box																		
FLUX.2	0.42	0.12	0.67	0.48	0.17	0.62	0.52	0.17	0.64	0.51	0.17	0.60	0.53	0.17	0.59	0.66	0.20	0.53
FLUX.2+LFA	**0.30**	**0.08**	**0.73**	**0.31**	**0.12**	**0.67**	**0.40**	**0.13**	**0.67**	**0.31**	**0.09**	**0.70**	**0.40**	**0.13**	**0.63**	**0.53**	**0.16**	**0.58**
SD3-UE	0.84	0.41	0.03	0.88	0.48	0.01	0.89	0.42	0.01	0.81	0.45	0.10	0.86	0.51	0.03	0.90	0.44	0.01
SD3-UE+LFA	**0.68**	**0.28**	**0.44**	**0.71**	**0.33**	**0.37**	**0.71**	**0.31**	**0.43**	**0.65**	**0.30**	**0.49**	**0.71**	**0.35**	**0.38**	**0.75**	**0.34**	**0.38**
Black-Box																		
Qwen	0.82	0.23	0.39	0.79	0.22	0.42	0.80	0.24	0.40	0.70	0.22	0.48	0.67	0.22	0.40	0.83	0.24	0.39
Qwen+LFA	**0.69**	**0.16**	**0.53**	**0.65**	**0.15**	**0.57**	**0.66**	**0.17**	**0.56**	**0.57**	**0.15**	**0.62**	**0.55**	**0.17**	**0.54**	**0.74**	**0.19**	**0.50**
Seedream	**0.61**	0.16	0.51	0.57	0.18	0.43	**0.61**	0.19	0.47	0.63	0.21	0.43	0.58	0.22	0.43	0.64	0.18	0.54
Seedream+LFA	0.68	**0.15**	**0.55**	**0.54**	**0.14**	**0.53**	0.64	**0.17**	**0.55**	**0.57**	**0.16**	**0.49**	**0.46**	**0.17**	**0.56**	**0.51**	**0.16**	**0.58**

components are comparatively robust to domain and architectural variations [41, 23, 22], external-VAE alignment can still suppress semantic drift. Implementation parameters are in Appendix F.7.

5 Experiments

5.1 Experiment Setup

Base models and data. We evaluate VAE-LFA on four DiT-based editors: two white-box models, FLUX.2 Klein 9B [18] and SD3-UE [4, 47], and two black-box API models, Qwen Image 2.0 [38] and Seedream 4.0 [30]. For black-box editors, VAE-LFA uses an off-the-shelf SD-VAE-ft-ema [29] between API calls, as described in Sec. 4.2. We evaluate on 120 images spanning illustration/photograph domains and creature/architecture/scenery categories; details are in Appendix F.1.

Evaluation protocol. Since VAE-LFA is a training-free post-hoc module, our primary comparison is paired: each editor is evaluated against its own +LFA variant under identical images, prompts, seeds, and model/API settings. This isolates the effect of inter-round low-frequency alignment without confounds from different training data, model capacity, or optimization recipes.

Tasks and metrics. We use three 10-turn protocols. **No-op editing** applies identity-preserving prompts to isolate intrinsic drift; we report LPIPS [45], SSIM [36], and normalized L1 drift to the initial image. **Cycle editing** alternates inverse instructions and uses the same metrics. **Long-**

Table 2: **Long-chain editing results for all models.**

Model	Round 5								Round 10							
	photograph				illustration				photograph				illustration			
	clear object		salient object		clear object		salient object		clear object		salient object		clear object		salient object	
	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑	DINO↑	VLM↑
White-Box																
FLUX.2	0.46	78.1	0.44	74.2	0.60	75.1	0.47	72.5	0.38	72.0	0.31	70.8	0.35	**68.1**	0.35	68.3
FLUX.2+LFA	0.46	**78.4**	**0.48**	**75.6**	**0.62**	**76.5**	**0.48**	**73.2**	**0.40**	**74.4**	**0.34**	**71.7**	**0.54**	67.0	**0.37**	**71.0**
SD3-UE	0.22	63.6	0.20	59.0	0.19	56.6	0.18	47.3	0.11	53.2	0.11	48.8	0.27	**50.0**	0.13	47.6
SD3-UE+LFA	**0.44**	**64.0**	**0.36**	**60.8**	**0.34**	**58.1**	**0.29**	**56.8**	**0.20**	**56.6**	**0.19**	**51.3**	**0.31**	48.2	**0.28**	**49.7**
Black-Box																
Qwen	0.43	63.2	0.51	61.1	0.46	64.9	0.52	62.7	0.29	55.1	0.33	48.1	0.31	54.6	0.32	50.1
Qwen+LFA	**0.49**	**66.7**	**0.55**	**62.1**	**0.51**	**68.3**	**0.54**	**65.6**	**0.37**	**60.9**	**0.44**	**53.8**	**0.39**	**58.7**	**0.41**	**55.2**
Seedream	0.50	72.1	0.52	**74.3**	0.48	**76.6**	0.53	74.5	0.37	70.0	0.35	67.7	0.32	68.3	0.41	69.0
Seedream+LFA	**0.61**	**73.0**	**0.56**	72.6	**0.60**	74.7	**0.59**	**75.9**	**0.44**	**70.6**	**0.36**	**68.0**	**0.42**	**69.4**	**0.47**	**70.1**

Table 3: **FLUX score breakdown on long-chain editing.** IP, SC, VQ, and CP denote instruction progress, state consistency, visual quality, and content preservation. Best paired results are in **bold**.

Round	Model	Illustration										Photograph									
		Clear Object					Salient Object					Clear Object					Salient Object				
		IP↑	SC↑	VQ↑	CP↑	Tot↑	IP↑	SC↑	VQ↑	CP↑	Tot↑	IP↑	SC↑	VQ↑	CP↑	Tot↑	IP↑	SC↑	VQ↑	CP↑	Tot↑
R5	FLUX.2	28.5	16.4	14.1	16.1	75.1	**28.0**	15.7	14.2	14.6	72.5	30.2	**17.0**	14.8	**16.1**	78.1	29.1	16.3	**14.3**	14.6	74.2
	+LFA	**29.2**	16.4	**14.7**	**16.2**	**76.5**	27.3	**16.2**	**14.7**	**15.0**	**73.2**	**31.0**	16.4	**15.2**	15.8	**78.4**	**29.6**	**16.8**	14.1	**15.1**	**75.6**
R10	FLUX.2	**23.7**	15.7	14.2	14.5	**68.1**	26.4	**16.0**	13.5	12.4	68.3	28.9	15.2	**13.6**	**14.4**	72.0	29.7	15.5	**13.2**	12.4	70.8
	+LFA	22.0	**15.9**	**14.3**	**14.8**	67.0	**29.2**	15.6	**13.8**	12.4	**71.0**	**31.1**	**15.9**	13.5	13.9	**74.4**	**30.5**	15.5	12.8	**12.9**	**71.7**

chain editing applies cumulative edits and reports DINOv3 [33] subject consistency and Qwen-3.6-based [28] VLM scores, following EdiVal-Agent [3]. Implementation details are in Appendix F.1–F.6.

5.2 Quantitative Analysis

Fig. 7 and Tab. 1 report 10-round no-op and cycle editing results. VAE-LFA improves consistency metrics across both white-box and black-box editors. Tab. 2 reports long-chain results, and Tab. 3 provides the FLUX.2 VLM score breakdown. For long-chain edits, we group samples by clear-object and salient-object images (Appendix F.5), since object presence is more relevant than the original category. VAE-LFA improves consistency, instruction following, and overall visual quality on average. Bootstrap confidence intervals for no-op and cycle improvements are reported in Appendix D.1.

5.3 Comparison Study

Comparison with frequency-based stabilization. We compare end-to-end VAE-LFA with FreqEdit [21] on FLUX.1 Kontext [19], the editor native to its official implementation, using identical settings. As shown in Tab. 6, VAE-LFA achieves the strongest no-op consistency and outperforms FreqEdit on all round-10 cycle metrics. Relative to the base editor it substantially improves L1 and LPIPS. On long-chain editing, VAE-LFA remains competitive with a slightly higher overall VLM score (75.5 *vs.* 74.6/74.4), while additionally supporting black-box editors.

Comparison with training-based stabilization. We fine-tune FLUX.2 Klein with LoRA on a no-op reconstruction objective (60 images, 400 steps) as a naive training-based baseline. Tab. 7 shows that explicit identity supervision reduces accumulated drift at later rounds, supporting the hypothesis that missing no-op supervision contributes to multi-turn bias. Nevertheless, VAE-LFA achieves stronger consistency without any training and remains applicable to API-only editors for which retraining is unavailable.

5.4 Qualitative Analysis

Fig. 8, Fig. 9, and Fig. 10 show qualitative results for the three settings. Baseline editors accumulate color shifts, artifacts, and semantic leakage over turns. VAE-LFA better preserves appearance and structure without visibly degrading instruction following. More examples are provided in Appendix E.

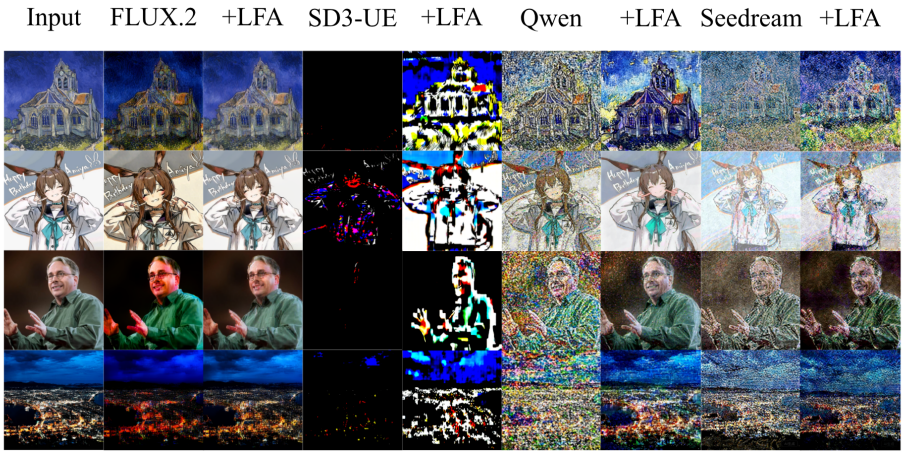

Figure 8: **Visualization of No-op editing trajectories at round 10.**

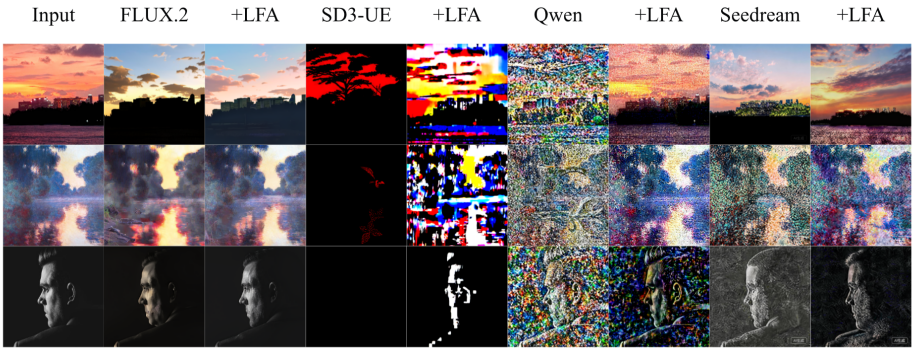

Figure 9: **Visualization of Cycle editing results at round 10.**

5.5 Ablation Studies

Frequency scope. Tab. 4 compares aligning low-frequency, high-frequency, and both components on FLUX.2. Low-frequency alignment achieves nearly the same no-op gains as full alignment, supporting our observation that drift is mainly low-frequency. High-frequency alignment helps little, while full alignment can over-constrain valid edits in the cycle setting. We therefore use low-frequency-only alignment by default.

Anchor update. Tab. 5 compares fixed, previous-round, and EMA anchors for low-frequency alignment. The fixed anchor preserves the initial appearance but is rigid for meaningful edits, while the previous-round anchor can drift with the trajectory. EMA offers a smoother historical reference that balances stability and edit flexibility, and is used by default.

External VAE. For black-box APIs, VAE-LFA introduces an external VAE encode–decode round trip, a potential confounder. We therefore include a +VAE baseline in Appendix D.2, using the same VAE round trip without low-frequency alignment. Further ablations on the external VAE choice,

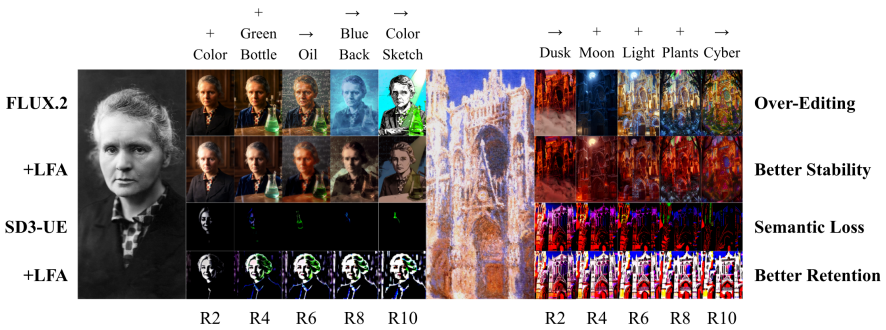

Figure 10: **Visualization of white-box long-chain edits at round 2, 4, 6, 8, 10.**

Table 4: **Ablation on frequency alignment scope for FLUX.2.** Best results are in **bold**. * uses the DiT-only setting; Low only is VAE-LFA.

Variant	photograph									illustration								
	creature			architecture			scenery			creature			architecture			scenery		
	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑
No-Op, Round 10																		
FLUX.2* (Base)	0.26	0.07	0.90	0.36	0.13	0.78	0.33	0.11	0.81	0.28	0.09	0.84	0.33	0.11	0.78	0.38	0.12	0.77
+ Both	**0.16**	0.05	**0.93**	**0.25**	**0.09**	**0.79**	**0.36**	**0.07**	**0.83**	**0.26**	**0.07**	**0.91**	**0.29**	**0.09**	0.79	**0.33**	**0.07**	**0.83**
+ High only	0.22	0.07	0.90	0.32	0.12	0.78	0.31	0.11	0.82	0.28	0.08	0.87	0.31	0.11	0.77	0.24	0.11	0.79
+ Low only (Ours)	0.17	**0.04**	**0.93**	0.27	**0.09**	**0.79**	0.25	**0.07**	**0.83**	**0.26**	**0.07**	0.88	**0.29**	**0.09**	**0.80**	**0.33**	0.08	**0.83**
Cycle, Round 10																		
FLUX.2* (Base)	0.33	0.10	0.70	0.43	0.16	0.65	0.44	0.15	0.63	**0.31**	0.11	0.69	0.46	0.15	0.59	0.67	0.20	0.50
+ Both	**0.30**	0.09	0.71	0.33	**0.12**	0.65	0.41	**0.13**	0.65	**0.31**	0.10	0.69	0.42	**0.13**	0.61	0.55	0.18	0.56
+ High only	0.32	0.11	0.71	0.40	0.15	0.66	0.45	0.16	0.64	0.33	0.12	0.68	0.45	0.14	0.60	0.65	0.19	0.52
+ Low only (Ours)	**0.30**	**0.08**	**0.73**	**0.31**	**0.12**	**0.67**	**0.40**	**0.13**	**0.67**	**0.31**	**0.09**	**0.70**	**0.40**	**0.13**	**0.63**	**0.53**	**0.16**	**0.58**

Table 5: **Ablation on anchor update strategy for FLUX.2.** EMA is the default VAE-LFA setting.

Variant	photograph									illustration								
	creature			architecture			scenery			creature			architecture			scenery		
	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑
No-Op, Round 10																		
FLUX.2* (Base)	0.26	0.07	0.90	0.36	0.13	0.78	0.33	0.11	0.81	0.28	0.09	0.84	0.33	0.11	0.78	0.38	0.12	0.77
+ Fixed	0.18	0.05	**0.93**	0.29	**0.09**	**0.79**	0.26	0.08	0.82	**0.26**	**0.07**	0.86	0.30	**0.09**	0.79	**0.33**	0.09	**0.83**
+ Prev	**0.17**	**0.04**	0.92	0.28	**0.09**	**0.79**	**0.25**	**0.07**	0.84	0.27	0.08	0.87	**0.29**	0.10	0.79	0.35	0.09	0.81
+ EMA (Ours)	**0.17**	**0.04**	**0.93**	**0.27**	**0.09**	**0.79**	**0.25**	**0.07**	**0.83**	**0.26**	**0.07**	**0.88**	**0.29**	**0.09**	**0.80**	**0.33**	**0.08**	**0.83**
Cycle, Round 10																		
FLUX.2* (Base)	0.33	0.10	0.70	0.43	0.16	0.65	0.44	0.15	0.63	**0.31**	0.11	0.69	0.46	0.15	0.59	0.67	0.20	0.50
+ Fixed	0.32	0.09	0.72	0.32	0.13	0.66	0.41	0.13	0.65	**0.31**	0.10	**0.70**	0.42	**0.13**	**0.62**	0.56	0.17	0.54
+ Prev	0.31	0.09	0.71	0.34	0.14	**0.67**	0.42	0.14	0.65	**0.31**	0.11	0.69	0.43	0.14	**0.62**	0.59	0.18	0.53
+ EMA (Ours)	**0.30**	**0.08**	**0.73**	**0.31**	**0.12**	**0.67**	**0.40**	**0.13**	**0.67**	**0.31**	**0.09**	**0.70**	**0.40**	**0.13**	**0.63**	**0.53**	**0.16**	**0.58**

Table 6: **Comparison with FreqEdit on FLUX.1 Kontext.** Best result per column in **bold**.

	No-Op R2			*No-Op R10*			*Cycle R2*			*Cycle R10*		
Method	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓
Kontext	0.045	0.974	0.082	0.120	0.887	0.390	0.088	0.943	**0.186**	0.216	**0.547**	0.626
Kontext+FreqEdit	0.075	0.845	0.172	0.169	0.634	0.450	0.150	0.721	0.354	0.252	0.317	0.720
Kontext+LFA	**0.036**	**0.980**	**0.073**	**0.088**	**0.904**	**0.364**	**0.073**	**0.950**	0.199	**0.198**	0.431	**0.576**

Table 7: **Comparison with a training-based baseline** on FLUX.2 Klein (no-op editing, held-out evaluation images).

	No-Op R1			*No-Op R5*			*No-Op R10*		
Method	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓
FLUX.2	**0.102**	**0.780**	**0.229**	0.142	0.740	0.341	0.195	0.659	0.546
FLUX.2+LoRA	0.102	0.757	0.240	0.122	0.727	0.306	0.153	0.670	0.461
FLUX.2+LFA	**0.102**	**0.780**	**0.229**	**0.109**	**0.767**	**0.275**	**0.119**	**0.748**	**0.359**

low-pass filter design, and EMA momentum, together with an evaluation under intentional global edits and an exploratory analysis of the origin of DiT drift, are provided in Appendix D.

6 Conclusion

We analyze multi-turn degradation in DiT-based image editors from a VAE-latent frequency perspective, showing that semantic drift is mainly driven by DiT-induced low-frequency deviations. Motivated by this, we propose VAE-LFA, a training-free low-frequency alignment method for both white-box and black-box editors. Experiments across no-op, cycle, and long-chain editing show improved consistency and fidelity.

Discussion and limitations. VAE-LFA can mildly suppress intended large global changes under strong prompts; prompt gating resolves this trade-off (Appendix D.6), and identity-preserving post-training alleviates the faster degradation observed under no-op editing (Tab. 7).

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 62502317), and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2026A1515011198). The authors declare no competing interests.

References

[1] Gal Almog, Ariel Shamir, and Ohad Fried. Reed-vae: Re-encode decode training for iterative image editing with diffusion models. In *Computer Graphics Forum*, volume 44, page e70020. Wiley Online Library, 2025.

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[3] Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, et al. Edival-agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing. *arXiv preprint arXiv:2509.13399*, 2025.

[4] 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.

[5] Han Fang, Kejiang Chen, Yupeng Qiu, Zehua Ma, Weiming Zhang, and Ee-Chien Chang. Dero: Diffusion-model-erasure robust watermarking. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2973–2981, 2024.

[6] Xiang Fang and Wanlong Fang. Slap: The semantic least action principle for variational video-language modeling. In *International Conference on Machine Learning*, 2026.

[7] Xiang Fang, Wanlong Fang, and Changshuo Wang. Cogniverse: Revolutionizing multi-modal retrieval-augmented generation with cognitive reflection and geometric reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[8] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023.

[9] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 12709–12720, 2024.

[10] Jing Gu, Morteza Mardani, Wonjun Lee, Dongmian Zou, and Gilad Lerman. Understanding latent diffusability via fisher geometry, 2026.

[11] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *European Conference on Computer Vision*, pages 56–73, 2022.

[12] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Toward a unified transformer-based framework for scene graph generation and human-object interaction detection. *IEEE Transactions on Image Processing*, 32:6274–6288, 2023.

[13] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.

[14] Daneul Kim, Jaeah Lee, and Jaesik Park. Improving editability in image generation with layer-wise memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7889–7898, 2025.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Gwanhyeong Koo, Sunjae Yoon, Ji Woo Hong, and Chang D. Yoo. Flexiedit: Frequency-aware latent refinement for enhanced non-rigid editing, 2024.

[17] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling, 2025.

[18] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

[19] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[20] Junzhe Li, Sifan Zhou, Liya Guo, Xuerui Qiu, Linrui Xu, Delin Qu, Tingting Long, Chun Fan, Ming Li, Hehe Fan, et al. Unif^{2}ace: A unified fine-grained face understanding and generation model. *arXiv preprint arXiv:2503.08120*, 2025.

[21] Yucheng Liao, Jiajun Liang, Kaiqian Cui, Baoquan Zhao, Haoran Xie, Wei Liu, Qing Li, and Xudong Mao. Freqedit: Preserving high-frequency features for robust multi-turn image editing. *arXiv preprint arXiv:2512.01755*, 2025.

[22] Yilin Liu, Yunkui Pang, Jiang Li, Yong Chen, and Pew-Thian Yap. Architecture-agnostic untrained network priors for image reconstruction with frequency regularization. In *European Conference on Computer Vision*, pages 341–358. Springer, 2024.

[23] Xianping Ma, Xiaokang Zhang, Xingchen Ding, Man-On Pun, and Siwei Ma. Decomposition-based unsupervised domain adaptation for remote sensing image semantic segmentation, 2024.

[24] Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. *arXiv preprint arXiv:2511.19365*, 2025.

[25] Tejaswini Medi, Hsien-Yi Wang, Arianna Rampini, and Margret Keuper. Missing fine details in images: Last seen in high frequencies, 2025.

[26] Mang Ning, Mingxiao Li, Le Zhang, Lanmiao Liu, Matthew B. Blaschko, Albert Ali Salah, and Itir Onal Ertugrul. Spectrum matching: a unified perspective for superior diffusability in latent diffusion, 2026.

[27] Leigang Qu, Feng Cheng, Ziyan Yang, Qi Zhao, Shanchuan Lin, Yichun Shi, Yicong Li, Wenjie Wang, Tat-Seng Chua, and Lu Jiang. Vincie: Unlocking in-context image editing from video. In *The Fourteenth International Conference on Learning Representations*, 2025.

[28] Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026.

[29] 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*, pages 10684–10695, 2022.

[30] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

[31] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024.

[32] Zitao Shuai, Chenwei Wu, Zhengxu Tang, Bowen Song, and Liyue Shen. Latent space disentanglement in diffusion transformers enables precise zero-shot semantic editing. *arXiv preprint arXiv:2411.08196*, 2024.

[33] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[34] Jinzhe Tu, Ruilei Guo, Zihan Guo, Junxiao Yang, Shiyao Cui, and Minlie Huang. Smsp: A plug-and-play strategy of multi-scale perception for mllms to perceive visual illusions, 2026.

[35] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[36] 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.

[37] Ziyi Wang, Songbai Tan, Gang Xu, Xuerui Qiu, Hongbin Xu, Xin Meng, Ming Li, and Fei Richard Yu. Safe-var: Safe visual autoregressive model for text-to-image generative watermarking. *arXiv preprint arXiv:2503.11324*, 2025.

[38] 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.

[39] Wei Wu, Qingnan Fan, Shuai Qin, Hong Gu, Ruoyu Zhao, and Antoni B. Chan. Freediff: Progressive frequency truncation for image editing with diffusion models, 2024.

[40] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, volume 2025, pages 20383–20407, 2025.

[41] Wei Yang, Rui Zhong, Yiqun Chen, Chi Lu, and Peng Jiang. Structured spectral reasoning for frequency-adaptive multimodal recommendation, 2026.

[42] Jinghan Yu, Junhao Xiao, Chenyu Zhu, Jiaming Li, Jia Li, HanMing Deng, Xirui Wang, Guoli Jia, Jianjun Li, Xiang Bai, Bowen Zhou, and Zhiyuan Ma. I2e: From image pixels to actionable interactive environments for text-guided image editing, 2026.

[43] Z.ai. GLM-Image: Auto-regressive for dense-knowledge and high-fidelity image generation, January 2026.

[44] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[45] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[46] Chen Zhao, Jiawei Chen, Hongyu Li, Zhuoliang Kang, Shilin Lu, Xiaoming Wei, Kai Zhang, Jian Yang, and Ying Tai. Luve: Latent-cascaded ultra-high-resolution video generation with dual frequency experts. *arXiv preprint arXiv:2602.11564*, 2026.

[47] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 3058–3093. Curran Associates, Inc., 2024.

[48] Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15792–15801, 2025.

A Impact Statement

VAE-LFA aims to improve the stability and consistency of multi-turn image editing, which can benefit creative workflows, visual design, content production, and iterative human-AI collaboration. By reducing unintended semantic drift and quality degradation, the method may make image editing systems more controllable and predictable for users.

At the same time, improved multi-turn editing consistency may also increase the realism and persistence of manipulated visual content, including misleading edits, impersonation, or other forms of synthetic media misuse. VAE-LFA does not introduce a new image generation model or bypass the safety mechanisms of existing editors, but it can be composed with such systems. Therefore, deployment should rely on the safeguards of the underlying image editing platforms, such as content provenance, watermarking, misuse detection, safety filtering, and policy-based restrictions for harmful editing requests.

B Detailed Related Works

Instruction-based and multi-turn image editing. As the development of generative AI [12, 20, 11, 6, 37, 7], instruction-based image editing aims to modify an input image according to natural-language instructions while preserving irrelevant content. Early and recent systems improve single-turn editing through paired edit data, manually annotated instruction-edit datasets, multimodal instruction enhancement, and stronger generative backbones [2, 44, 8, 9, 13]. More recently, diffusion-transformer and flow-matching based editors have further advanced prompt following, realism, and in-context editing capabilities [19, 38, 30, 43]. However, these systems are still commonly optimized and evaluated in one-shot settings. When applied iteratively, small deviations introduced at each turn can accumulate, resulting in identity drift, color shift, oversmoothing, artifact amplification, or unintended semantic changes. Recent works have begun to study this multi-turn setting explicitly [48, 21, 14], but robust iterative editing remains challenging, especially when the underlying editor is accessible only through an API.

Remedies for iterative degradation. Existing remedies for multi-turn degradation differ in the type of access and supervision they require. Training-based methods improve iterative or in-context editing by learning from richer supervision. For example, VINCIE learns in-context image editing from videos using interleaved multimodal sequences and a diffusion-transformer architecture [27], while Emu Edit improves instruction-based editing through multi-task training over recognition and generation tasks [31]. Other works target degradation mechanisms more directly. REED-VAE retrains the VAE to reduce artifacts and noise accumulated through repeated transitions between pixel and latent spaces [1]. Layer-wise Memory stores internal latent representations and prompt embeddings from previous editing steps to maintain consistency during sequential modification [14]. FreqEdit identifies progressive high-frequency information loss as a cause of multi-turn quality degradation and injects high-frequency features from reference velocity fields to stabilize editing [21]. These approaches are complementary to ours, but they typically require additional training, access to internal representations or velocity fields, or assumptions about the editor's inference process. In contrast, our goal is to design a minimal post-hoc intervention that can operate at the VAE latent interface and remain applicable to both white-box and black-box editors.

VAE bottlenecks and latent-space analysis. Most modern latent diffusion and DiT-based image generators employ an autoencoder bottleneck that maps images into a lower-dimensional latent space before generative modeling [29, 15]. This design substantially reduces the computational cost of diffusion modeling compared with operating directly in pixel space [29]. At the same time, the autoencoder is not an identity map: its encode–decode process can introduce reconstruction bias, and repeated transitions between pixel and latent spaces have been observed to accumulate artifacts and noise in iterative editing [1]. Recent studies further show that the structure of the latent space affects generative modeling, for example through equivariance properties of autoencoder latents and through latent diffusability under diffusion training [17, 10]. These works motivate analyzing the VAE latent space, but they do not directly attribute multi-turn editing drift to different stages of a VAE-DiT editing pipeline. In contrast, we use the VAE latent space as a shared interface to separate the DiT transition from the VAE round trip. Our controlled no-operation analysis shows that the VAE

mainly contributes a comparatively stable reconstruction bias, whereas progressive semantic drift is dominated by low-frequency deviations introduced by the DiT.

Frequency perspectives in generative models. Frequency-domain analysis provides a useful way to inspect how generative models distribute information across spatial scales. Spectrum Matching studies VAE latent diffusability from a spectral perspective, arguing that latent power spectra and frequency-to-frequency decoder correspondence are important for latent diffusion quality [26]. Other work on latent tokenizers shows that fine visual details are strongly tied to high-frequency fidelity, and that common reconstruction objectives may prioritize low-frequency reconstruction at the expense of high-frequency details [25]. Frequency separation has also been used in generation architectures, where low-frequency components are associated with global semantics and high-frequency components with local detail synthesis [24, 46].

Frequency-aware editing methods further show that manipulating different spectral bands can improve edit controllability. FreeDiff truncates excessive low-frequency guidance during diffusion sampling to reduce editing spillover [39], while FlexiEdit refines high-frequency DDIM latent components to better support non-rigid edits [16]. Beyond editing, DERO uses frequency-domain analysis of diffusion denoising to improve watermark robustness against diffusion-model erasure [5]. These studies support the utility of frequency analysis, but they do not attribute multi-turn drift to VAE versus DiT components or design a black-box-compatible stabilization module.

Works on multimodal LLMs [34] and DiT latent semantic editing [32] also indicate that latent representations can expose controllable semantic structure. In contrast, we analyze editing bias directly in the VAE latent space by decomposing each latent channel into low- and high-frequency components. Our experiments show that the DiT-induced drift is concentrated in low-frequency latent components, while high-frequency residuals are better left unconstrained to preserve local details and editing flexibility. This observation motivates VAE-LFA: after each DiT transition, we align only the channel-wise mean and standard deviation of the low-frequency latent component to an EMA anchor, and then recombine it with the original high-frequency residual.

C Detailed Formulation of Multi-turn Editing Bias

Here we formulate semantic drift in VAE-DiT models during multi-turn editing. For simplicity, we ignore the stochasticity of the generative process and present a deterministic formulation.

Image-space multi-turn drift. Let $\mathcal{X}$ denote the image space and let $\mathbf{x}^{(k)} \in \mathcal{X}$ be the image after the k-th editing turn. A multi-turn editor induces a sequence $\mathbf{x}^{(k)} = \mathcal{T}_{p^{(k)}}(\mathbf{x}^{(k-1)})$, where $p^{(k)}$ is the instruction at turn k and $\mathcal{T}_{p^{(k)}} : \mathcal{X} \to \mathcal{X}$ is the image-space editing transition. Let $\mathcal{P}_{\text{no-op}}$ denote a set of no-operation prompts, such as "keep the image unchanged", and let $p_{\text{no-op}} \in \mathcal{P}_{\text{no-op}}$ be a representative prompt. Ideally, no-operation editing should be an identity map, i.e., $\mathcal{T}_{p_{\text{no-op}}}(\mathbf{x}) = \mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}$. In practice, however, repeated no-operation editing violates this fixed point and produces visible degradation, including color noise, whitening, detail loss, and semantic ambiguity. Such degradation becomes increasingly pronounced as the number of editing turns increases. We refer to this no-operation fixed-point violation as *multi-turn editing bias*. This definition distinguishes degradation from valid edits, since meaningful editing is expected to change the image.

Latent space transition. We analyze this bias in the VAE latent space. Let $E : \mathcal{X} \to \mathcal{Z}$ and $D : \mathcal{Z} \to \mathcal{X}$ denote the VAE encoder and decoder. Given an input latent $\mathbf{z}$, the prompt-conditioned DiT editor produces a raw latent $G_p(\mathbf{z})$, and the standard VAE-DiT pipeline maps it to the next latent by $\Phi_p(\mathbf{z}) := E(D(G_p(\mathbf{z})))$. Analogous to the pixel space transition, for any no-operation prompt $p_{\text{no-op}}$, $\Phi_{p_{\text{no-op}}}$ should be an identity map ideally, such that $\Phi_{p_{\text{no-op}}}(\mathbf{z}) = \mathbf{z}$. Its violation defines the latent no-operation bias, which can be decomposed into two terms as:

$$\Phi_{p_{\text{no-op}}}(\mathbf{z}) - \mathbf{z} = \underbrace{G_{p_{\text{no-op}}}(\mathbf{z}) - \mathbf{z}}_{\text{DiT no-operation bias}} + \underbrace{\Phi_{p_{\text{no-op}}}(\mathbf{z}) - G_{p_{\text{no-op}}}(\mathbf{z})}_{\text{VAE round-trip bias}} \tag{6}$$

Prompted editing as bias-contaminated displacement. For a meaningful prompt p, the one-step latent displacement contains both prompt-induced change and the no-operation bias:

$$\Phi_p(\mathbf{z}) - \mathbf{z} = \underbrace{\Phi_p(\mathbf{z}) - \Phi_{p_{\text{no-op}}}(\mathbf{z})}_{\text{prompt-induced displacement}} + \underbrace{\Phi_{p_{\text{no-op}}}(\mathbf{z}) - \mathbf{z}}_{\text{no-operation bias}} \tag{7}$$

Table 8: Bootstrap 95% confidence intervals of paired improvements of +LFA over base editors at round 10. Positive values indicate improvement.

Protocol	LPIPS ↓	L1 ↑	SSIM ↑
No-op	0.1363 [0.1139, 0.1487]	0.0783 [0.0458, 0.0909]	0.1929 [0.1486, 0.2272]
Cycle	0.1179 [0.1036, 0.1292]	0.0692 [0.0563, 0.0817]	0.1633 [0.1302, 0.1969]

In prompted editing, the instruction-induced change and the model's intrinsic distribution drift are coupled through the same latent state and editing dynamics. The bracketed term is not guaranteed to be a clean semantic edit, and Eq. (7) should not be interpreted as an independent additive decomposition. We use the no-operation transition only as a reference to expose the drift that exists even without a semantic edit. Although two terms are not independent, later experiments empirically verify that reducing this no-operation drift also mitigates degradation under meaningful prompts.

Multi-turn accumulation. Total latent displacement over K turns is an accumulation of one-step residuals. Each residual is evaluated at the current latent state, which already contains the effects of previous rounds. Thus, even small per-turn no-operation biases can accumulate over long trajectories.

D Additional Quantitative Results

D.1 Experiment Statistical Significance

All no-op and cycle comparisons are paired over identical images and instruction sequences, so each difference is computed between a base editor and its corresponding +LFA variant on the same sample. To quantify uncertainty, we report non-parametric percentile bootstrap 95% confidence intervals (CIs) for the aggregated round-10 improvements in Tab. 8. For LPIPS and L1, where lower is better, the paired improvement is defined as $\Delta_i = s_i^{\text{base}} - s_i^{+\text{LFA}}$; for SSIM, where higher is better, it is defined as $\Delta_i = s_i^{+\text{LFA}} - s_i^{\text{base}}$. Positive values therefore indicate improvement.

For each model and protocol, we first average the round-10 metric across two repeated runs for every sample, and then compute paired differences. We pool paired differences from FLUX.2 ($n = 120$), SD3-UE ($n = 120$), Qwen Image 2.0 ($n = 120$), and Seedream 4.0 ($n = 120$), yielding $N = 480$ paired observations per protocol. We draw $B = 10{,}000$ bootstrap resamples of size N with replacement, compute the mean improvement for each resample, and use the 2.5-th and 97.5-th percentiles as the CI bounds. This percentile bootstrap procedure is non-parametric and does not assume normality of the paired differences.

All reported confidence intervals in Tab. 8 are strictly positive, indicating that +LFA yields statistically significant paired improvements over the corresponding base editors for both no-op and cycle protocols at the 95% confidence level. These results support the reliability of the consistency gains reported in the main tables.

Data preparation. For each model–protocol combination, per-sample metrics (LPIPS, L1, SSIM) are extracted from the round-10 records of all available samples. Each condition (model, protocol, VAE status) was run twice; when a sample appears in multiple runs, its round-10 values are averaged to obtain a single observation per sample.

D.2 Additional Ablation on Black-Box Models

We provide an additional ablation experiment on black-box models, where we compare VAE-LFA with an external-VAE-only baseline. As shown in Tab. 9, editing consistency is not consistently improved when applying an external VAE only, illustrating the effectiveness of VAE-LFA.

D.3 Ablation on External VAE Choice

For black-box editors, VAE-LFA operates through an external VAE, raising the question of how the choice of external autoencoder affects performance. We test three substantially different external

Table 9: Ablation studies on black-box models. The best result among variants of each base editor is shown in **bold**.

Model	photograph									illustration								
	creature			architecture			scenery			creature			architecture			scenery		
	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑
No-Op, Round 10																		
Qwen	0.82	0.22	0.44	0.77	0.22	0.43	0.78	0.22	0.46	0.72	0.21	0.52	0.69	0.22	0.44	0.80	0.21	0.50
Qwen+VAE	0.80	0.19	0.48	0.73	0.21	0.46	0.75	0.22	0.47	0.69	0.19	0.54	0.65	0.18	0.49	0.78	0.20	0.52
Qwen+LFA	**0.66**	**0.12**	**0.68**	**0.64**	**0.15**	**0.62**	**0.60**	**0.11**	**0.71**	**0.57**	**0.14**	**0.66**	**0.54**	**0.16**	**0.60**	**0.68**	**0.15**	**0.67**
Seedream	0.66	0.15	0.54	**0.61**	0.19	0.42	**0.61**	0.18	0.48	0.64	0.20	0.45	**0.59**	0.21	**0.42**	**0.61**	0.17	0.56
Seedream+VAE	0.68	0.16	0.54	0.64	0.20	0.39	0.63	0.21	0.46	0.64	0.21	0.48	0.62	0.23	0.39	0.64	0.19	0.54
Seedream+LFA	**0.65**	**0.13**	**0.57**	0.62	**0.17**	**0.45**	**0.61**	**0.16**	**0.51**	**0.62**	**0.19**	**0.50**	**0.59**	**0.20**	0.41	0.62	**0.16**	**0.60**

Table 10: **Ablation on external VAE choice** with Qwen Image 2.0 as the black-box editor. Each VAE+LFA variant is compared against its matched VAE-only control.

	No-Op R10			*Cycle R10*		
Setting	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓
Native editor (no VAE)	0.292	0.252	1.012	0.298	0.224	0.982
FLUX.2 VAE-only	0.295	0.245	0.995	0.299	0.217	0.939
FLUX.2 VAE+LFA	0.198	0.432	0.535	0.213	0.362	0.578
Qwen VAE-only	0.281	0.280	0.919	0.283	0.257	0.886
Qwen VAE+LFA	0.217	0.414	0.608	0.228	0.359	0.637
SANA VAE-only	0.172	0.575	0.397	0.205	0.448	0.472
SANA VAE+LFA	0.109	0.704	0.324	0.179	0.460	0.468

autoencoders—FLUX.2 (AutoencoderKLFlux2), Qwen-Image (AutoencoderKLQwenImage), and SANA (AutoencoderDC [40])—with the same black-box Qwen Image 2.0 editor. For each VAE, we compare a VAE-only control (external encode–decode without alignment) against the matched VAE+LFA pipeline under identical prompts, seeds, and API settings, which isolates the effect of LFA from the external VAE's own reconstruction behavior.

As shown in Tab. 10, LFA improves over its matched VAE-only control for every autoencoder family and both protocols, indicating that the gains are not explained by external-VAE reconstruction alone and do not depend on a specific latent parameterization or architecture. At the same time, the absolute VAE-only results vary considerably across families, confirming that the external VAE's reconstruction behavior does affect black-box performance; in practice, a VAE with strong reconstruction fidelity and compatible input resolution is preferable.

D.4 Ablation on Low-Pass Filter Design

We compare three low-pass operator families on FLUX.2 Klein: average pooling ($p \in \{5, 9, 13\}$), FFT-based Gaussian filtering ($\sigma \in \{0.15, 0.25, 0.35\}$), and second-order Butterworth filtering (cutoff $c \in \{0.15, 0.25, 0.35\}$, normalized spatial frequency). All variants share the default EMA anchor and identical paired 10-round settings.

As shown in Tab. 11, all tested operators substantially improve over the base editor, while their differences are small and no filter family consistently dominates across L1, SSIM, and LPIPS. This indicates that VAE-LFA relies on stable low-frequency statistics rather than the precise cutoff characteristics of a particular filter. We retain 9×9 average pooling as the default for its balanced performance and simplicity, without FFT/IFFT overhead or additional kernel design.

D.5 Ablation on EMA Momentum

We sweep $\alpha_\mu \in \{0.90, 0.95, 0.98\}$ with $\alpha_\sigma = 0.85$, and $\alpha_\sigma \in \{0.70, 0.85, 0.95\}$ with $\alpha_\mu = 0.95$, on FLUX.2 Klein. As shown in Tab. 12, performance remains highly stable throughout these ranges, and every setting substantially improves over the base editor. An EMA anchor has an effective time scale of approximately $1/(1-\alpha)$ rounds: the tested values span roughly 10–50 rounds for the mean and 3–20 rounds for the standard deviation. Earlier rounds therefore receive exponentially diminishing weight over a 50-round trajectory by design; the anchor is a rolling history updated from every pre-alignment latent, rather than a memory intended to preserve initial features indefinitely. This prevents the anchor from freezing meaningful edits while still smoothing inter-round drift. The

Table 11: **Ablation on low-pass filter design** on FLUX.2 Klein. The default setting is shown in **bold**.

	No-Op R10			*Cycle R10*		
Variant	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓
Base (no LFA)	0.197	0.659	0.561	0.215	0.468	0.610
Average $p = 5$	0.101	0.795	0.263	0.155	0.616	0.351
Average $p = 9$ (default)	**0.101**	**0.796**	**0.264**	**0.155**	**0.616**	**0.352**
Average $p = 13$	0.101	0.797	0.265	0.154	0.624	0.353
Gaussian $\sigma = 0.15$	0.101	0.795	0.263	0.155	0.615	0.351
Gaussian $\sigma = 0.25$	0.101	0.795	0.262	0.156	0.612	0.350
Gaussian $\sigma = 0.35$	0.101	0.795	0.262	0.155	0.612	0.349
Butterworth $c = 0.15, n = 2$	0.101	0.795	0.262	0.155	0.615	0.350
Butterworth $c = 0.25, n = 2$	0.101	0.795	0.262	0.155	0.614	0.349
Butterworth $c = 0.35, n = 2$	0.101	0.795	0.261	0.156	0.610	0.349

Table 12: **Ablation on EMA momentum coefficients** on FLUX.2 Klein. The default setting is shown in **bold**.

			No-Op R10			*Cycle R10*		
Variant	α_μ	α_σ	L1↓	SSIM↑	LPIPS↓	L1↓	SSIM↑	LPIPS↓
Base	—	—	0.197	0.659	0.561	0.215	0.468	0.610
LFA	0.90	0.85	0.101	0.796	0.265	0.154	0.618	0.352
LFA (default)	**0.95**	**0.85**	**0.101**	**0.796**	**0.264**	**0.155**	**0.616**	**0.352**
LFA	0.98	0.85	0.100	0.796	0.263	0.155	0.617	0.352
LFA	0.95	0.70	0.101	0.797	0.264	0.154	0.625	0.353
LFA	0.95	0.95	0.101	0.795	0.264	0.155	0.613	0.352

default $(\alpha_\mu, \alpha_\sigma) = (0.95, 0.85)$ provides a balanced time scale, and the sweep shows that the method is not sensitive to this exact choice.

D.6 Intentional Global Edits and Prompt Gating

VAE-LFA targets unwanted inter-round drift rather than every low-frequency change: under meaningful prompts, changes in color, lighting, style, or background can constitute valid low-frequency displacement. To probe this boundary, we construct a challenging 10-round protocol ($n = 30$) that alternates local edits with deliberately large global edits, and compare the native editor, default VAE-LFA, and a prompt-gated variant on FLUX.2 Klein. The gated variant uses an LLM judge (Qwen-3.6-Plus [28]) that receives the previous-round image and the current instruction, and decides whether the instruction calls for a large global change; on such rounds, alignment is disabled while the EMA still absorbs the new state.

As shown in Tab. 13, default fixed-strength alignment becomes more conservative under deliberately strong global changes, reflected in its lower instruction progress. The gated variant resolves this trade-off, achieving the best instruction progress, visual quality, and total score. This identifies explicit large global edits as the relevant boundary of fixed-strength alignment, and prompt-aware gating as a simple remedy.

Table 13: **Evaluation under intentional global edits** on FLUX.2 Klein (VLM scores).

Method	IP (0–40) ↑	SC (0–20) ↑	VQ (0–20) ↑	CP (0–20) ↑	Total (0–100) ↑
Native editor	31.1	17.1	13.2	15.2	76.6
Default VAE-LFA	30.2	17.4	12.2	14.8	74.6
Gated VAE-LFA	**31.9**	17.2	**14.1**	15.1	**78.4**

D.7 A Data-Structure Hypothesis for DiT Drift

Our work is primarily empirical, and formal causal attribution of DiT drift is beyond its scope; we nevertheless provide a testable data-structure hypothesis. Low-frequency latent components encode globally aligned attributes such as brightness, color tone, and coarse layout, which can exhibit a coherent dataset-level expectation, whereas high-frequency components have sample-dependent locations and phases with near-zero expectation. If no-op editing is not learned as an exact invariant mapping, a systematic residual may persist at each turn: under an approximately stationary residual

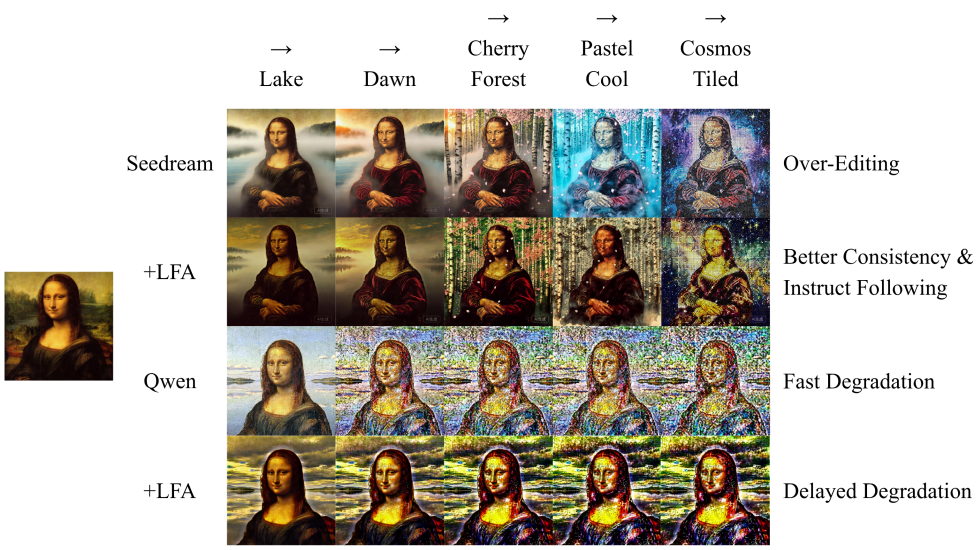

Figure 11: **Visualization of black-box long-chain edits.** On Seedream 4.0, VAE-LFA improves image consistency and instruction following capabilities considerably, while on Qwen Image 2.0, VAE-LFA delays image quality degradation significantly.

model, a coherent per-turn bias $\boldsymbol{\beta}$ grows as $K\boldsymbol{\beta}$ over K turns, while phase-incoherent fluctuations behave like a random walk of magnitude $\mathcal{O}(\sqrt{K})$, so low-frequency drift eventually dominates.

As a premise check, we analyze 1,024 randomly sampled UltraEdit [47] source images in the FLUX.2 VAE latent space. The global spatial-mean component shows a mean-to-RMS ratio of 0.345, about $11\times$ the finite-sample incoherent reference $1/\sqrt{1024}$, with 66% sign consistency across images. The coherence is concentrated in the lowest radial band (mean/RMS 0.103 for $r < 0.1$, about $3\times$ the reference) and approaches the incoherent reference above $r > 0.2$. These results support the premise that globally coherent structure concentrates in the lowest latent frequencies. We present them as an exploratory analysis of this premise, not as a complete causal explanation of DiT drift.

E Additional Qualitative Results

We provide visualizations of the long-chain editing results of black-box models, as shown in Fig. 11 and Fig. 12. VAE-LFA effectively improves consistency and visual quality even for black-box models.

F Implementation Details, Hardware and Parameters

F.1 Experimental Details

Dataset. We curate a 120-image evaluation set, partitioned into two coarse domains (*illustration* and *photograph*) and three fine-grained categories per domain (*creature*, *architecture*, and *scenery*), yielding 6 subsets of 20 images each. This taxonomy covers both subject-centric and in-the-wild scenes with varying structural complexity and frequency characteristics. The dataset is provided in the supplementary material.

Black-box setup. For Qwen Image 2.0 and Seedream 4.0, only image-level API inputs and outputs are used. We do not access model weights, native latents, denoising trajectories, velocity fields, attention maps, or internal features. VAE-LFA interleaves SD-VAE-ft-ema between editing rounds for latent alignment. We use identical prompts, seeds, and API settings for each base editor and its +LFA variant; details on exposed parameters are provided in Appendix F.7.

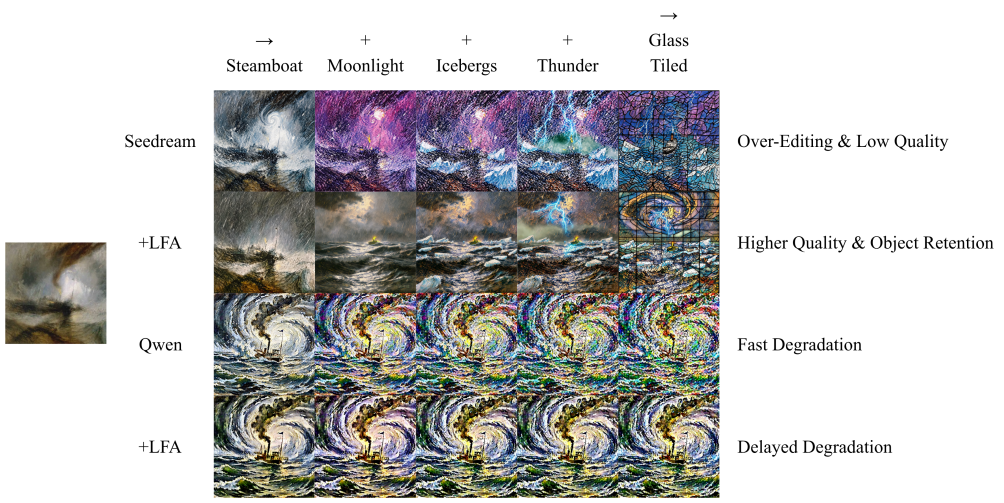

Figure 12: **Visualizations of black-box long-chain edits.** On Seedream 4.0, VAE-LFA improves image consistency and instruction following capabilities considerably, while on Qwen Image 2.0, VAE-LFA delays image quality degradation significantly.

Editing protocols. For no-op editing, we use multiple identity-preserving prompts, e.g., *"keep the image unchanged"*, for 10 consecutive turns; the full prompt list is in Appendix F.3. For cycle editing, we alternate semantically inverse instructions, e.g., *"add a red hat"* and *"remove the red hat"*, for 10 turns. For long-chain editing, we apply 10 cumulative instructions, such as changing attributes or altering backgrounds, to produce large semantic shifts.

Metrics. For no-op and cycle editing, we report LPIPS, global SSIM, and normalized L1 drift to the initial image at round 10. For long-chain editing, we use DINOv3 for subject consistency and a Qwen-3.6-based VLM judge for a 0–100 score at rounds 5 and 10. The VLM score decomposes into instruction progress, state consistency, visual quality, and content preservation; the detailed scoring protocol is in Appendix F.2.

F.2 The Design of VLM Metrics

The VLM metric measures iterative image editing quality at an arbitrary round r, rather than only the final round. The protocol is explicitly overwrite-aware: later instructions are allowed to replace earlier edits, so the model is judged against the intended cumulative state after applying instructions up to round r, instead of being penalized for not preserving superseded earlier states. The final VLM score is a 100-point composite score, formed by summing four judges:

- **Instruction Progress (0–40)**: Measures how well the edited image matches the intended cumulative target state after applying the instructions up to the evaluated round.
- **State Consistency (0–20)**: Measures whether the edited image forms a coherent overwrite-aware final state without contradictions between earlier and later edits.
- **Visual Quality (0–20)**: Measures the perceptual quality of the edited image, including artifact severity, boundary cleanliness, texture plausibility, and overall visual coherence.
- **Content Preservation (0–20)**: Measures how well non-target content, subject identity, and scene structure are preserved when they are not supposed to change.

Each judge returns subscores instead of a single holistic number. We aggregate those subscores locally to reduce score collapse and improve resolution across samples. The VLM metric is used to grade long-chain editing tasks because traditional methods are often object aware, making their utility limited when allowing edits that drastically modify the main object. For detailed prompts of each subagent, refer to Appendix F.6.

F.3 No-Op Prompts

We composed 10 no-op prompt choices for better experiment coverage, listed here.

No-Op Prompts

```
(NULL)
Keep the image unchanged.
Do not modify the image.
Output the original image.
Keep image the same.
Return the same image.
Make no edits.
Edit nothing.
Do not change anything.
Retain the identical image.
```

F.4 Data Synthesis

We synthesize cycle editing and long-chain editing samples using Qwen-3.6-Plus [28] and structured prompts to ensure both data collection efficiency and quality.

System Prompt: Cycle Editing

```
You are an expert image-editing instruction generator specialized in creating
reversible, paired editing prompts for academic research on multi-turn image
editing.
Your task:  Given an input image, analyze its main subject(s) and generate
exactly 10 editing prompts (5 strictly inverse pairs) for "cycle editing".
Strategy Selection (chosen by user):
There are two mutually exclusive strategies.  You MUST follow the one specified
by the user in the user message:
Strategy A: Accessory Add/Remove (50% cases)
- Add a specific, visually distinct accessory or prop to the main subject in
Turn 1, then remove it in Turn 2.  Repeat for 5 pairs (10 turns total).
- Examples:  "Add a red beret on the girl's head" / "Remove the red beret from
the girl's head"; "Put a silver wristwatch on the man's left wrist" / "Remove
the silver wristwatch from the man's left wrist".
- The accessory must be physically detachable and clearly visible.  Do NOT
change the subject's intrinsic attributes (hair color, species, etc.).
Strategy B: Subject Attribute Modify-Revert (50% cases)
- Modify a reversible visual attribute of the main subject in Turn 1, then
revert it exactly in Turn 2.  Repeat for 5 pairs.
- Examples:  "Change the dog's fur color to bright blue" / "Restore the dog's
fur to its original color"; "Convert the building to Gothic architecture style"
/ "Revert the building to its original architectural style"; "Change the car
paint to matte black" / "Restore the car paint to its original color".
- The attribute must be visually reversible.  Do NOT add/remove objects.
Universal Constraints:
1.  Each pair (Turn 2k-1, Turn 2k) must be strict semantic opposites.  After
Turn 10 the image should ideally return to the original state.
2.  Prompts must be concrete and executable:  include specific colors,
materials, positions, or styles.  Avoid vague words like "make it better" or
"adjust".
3.  Do NOT introduce irreversible changes (e.g., permanent background
destruction, subject replacement, cropping, or adding watermarks).
4.  Do NOT refer to previous turns with pronouns like "it" or "the previous
change".  Each prompt must be self-contained.
5.  Output format:  a single JSON array of exactly 10 strings.  No markdown, no
explanation, no code block fences.  Just the raw JSON array.
```

System Prompt: Long-Chain Editing

```
You are an expert image-editing instruction generator specialized in creating
progressive, cumulative editing prompts for academic research on multi-turn
image editing.
Your task: Given an input image, analyze its main subject(s) and generate
exactly 10 sequential editing prompts for "long-chain editing" where edits
accumulate over turns.
Strategy Selection (chosen by user):
There are two strategies. You MUST follow the one specified by the user:
Strategy A: Subject-Preserving Cumulative Transformation (80% cases)
- The main subject(s) must remain recognizable and structurally intact across
all 10 turns.
- Apply progressive changes to background, environment, lighting, weather,
season, artistic style, color tone, atmosphere, or camera perspective.
- Examples: "Change the background to an autumn park with falling leaves";
"Apply golden-hour sunset lighting with long shadows"; "Transform the image
into a watercolor painting style while keeping the scene layout"; "Add a light
snowfall and change the ground to snow-covered"; "Shift the overall color tone
to cold cyan-blue".
- Changes are cumulative: each new prompt can introduce a new layer on top of
previous ones. They do NOT need to be reversible.
Strategy B: Radical Overhaul (20% cases) - Only use this if the image is NOT a
portrait/close-up of a person/animal (i.e., suitable for scenery/architecture
abstraction).
- Remove or heavily obscure the original main subject and apply a drastic
style/environment shift.
- Examples: "Remove the car and turn the scene into an abstract cyberpunk
cityscape"; "Replace the building with a floating island in a fantasy sky
realm".
- If the image is a creature/portrait, fall back to Strategy A automatically.
Universal Constraints:
1. The 10 prompts must form a logically coherent chain. Adjacent turns should
feel like natural next steps in a creative workflow, not random jumps.
2. Each prompt must be self-contained and specific: include exact colors,
styles, lighting descriptors, weather, or environmental details.
3. Do NOT use pronouns like "it" or "that". Refer to subjects explicitly
(e.g., "the girl", "the building", "the mountain").
4. Do NOT issue identity-preserving prompts like "keep unchanged". Every
prompt must actively edit the image.
5. Output format: a single JSON array of exactly 10 strings. No markdown, no
explanation, no code block fences. Just the raw JSON array.
```

Example Data Point

```
"image": "017.png",
"category": "creature",
"cycle": [
"Change the character's hair color from red to platinum blonde.",
"Restore the character's hair color to its original vibrant red.",
"Change the character's eye color to a deep emerald green.",
"Revert the character's eye color to its original amber orange.",
"Change the character's white dress to a dark charcoal grey.",
"Restore the character's dress to its original white color.",
"Change the halo above the character's head to a glowing purple ring.",
"Revert the halo to its original thin golden ring.",
"Change the gun held by the character to a bright gold finish.",
"Restore the gun to its original silver and black finish."
],
"long_chain": [
"Change the background behind the red-haired girl with the halo to a bustling
```

```
futuristic marketplace with holographic signs and stalls.",
"Apply a warm, golden-hour sunlight filter over the entire scene, casting soft
shadows on the girl's white outfit and the marketplace stalls.",
"Add a gentle snowfall that accumulates on the girl's red hair, shoulders, and
the large gun she is holding.",
"Transform the artistic style of the image into a detailed pencil sketch
with cross-hatching shading, while keeping the snow and marketplace elements
visible.",
"Change the background environment to a dark, gothic cathedral interior with
tall arched windows.",
"Illuminate the cathedral scene with beams of colored light streaming through
the windows, highlighting the girl's face and golden halo.",
"Replace the pencil sketch style with a watercolor painting style, using soft,
bleeding edges and vibrant washes of color for the girl and the cathedral
interior.",
"Add floating cherry blossom petals swirling around the girl in the cathedral,
catching the light from the windows.",
"Shift the overall color tone to a monochromatic red and black palette, making
the girl's red hair prominent against the dark surroundings while her white
outfit remains bright.",
"Transform the background into a deep space nebula with swirling purple and
blue gas clouds, placing the girl as if she is floating in zero gravity with
bubbles rising around her."
]
```

F.5 Salient Object vs. Clear Object

We apply a Qwen-3.6-Plus [28] VLM to decide on whether an input image has a clear main object or not. The prompt for decision is provided below.

Object Taxonomy

```
Role: You are an objective image taxonomy annotator for a computer vision
benchmark.
Task: Given a single input image, assign it to exactly one of the following
two categories.

Category definitions:
1. salient_object:
The image is primarily scene-dominant or layout-dominant. The visual semantics
depend strongly on the broader environment, background, or spatial composition
rather than a single foreground object. Typical examples include architecture,
landscapes, street scenes, interiors, and other large-scale scenes.

2. clear_object:
The image contains one clearly identifiable main subject that dominates visual
attention. The semantics are mainly determined by this foreground subject
rather than the surrounding environment. Typical examples include a single
creature, person, product, or other salient object with a relatively simple
supporting background.

Instructions:
- Use only the provided image.
- Choose exactly one label: "salient_object" or "clear_object".
- Prefer "clear_object" only when one principal foreground subject is
unambiguously dominant.
- Output must be strict JSON.

Required JSON format:
"label": "salient_object", "reason": "short explanation"
```

F.6 VLM Metrics Prompts

We provide the prompts for VLM metrics here.

Common prompt. Common prompt is appended for every editing judge. It tells the VLM which round is being evaluated and provides the executed instruction prefix.

Common Prompt

```
Mode: mode
Evaluate the edited image after round round_idx.
Instructions executed up to this round:
Round 1: ...
Round 2: ...
...
Return strict JSON only.
```

Instruction progress. The prompt for the instruction progress agent measures how much of the intended cumulative edit state is visibly achieved after applying the instructions up to the requested round.

Instruction Progress

```
Role: You are a careful judge for iterative image editing.
Task: Compare the original image and the current edited image. Score only how
much of the intended cumulative edit state has been achieved after applying the
instructions up to the requested round.

Important reasoning rule:
- The edit sequence is cumulative and later instructions may overwrite earlier
ones.
- Do NOT penalize the edited image for failing to preserve an earlier state
that should have been replaced by a later instruction.
- Judge the target state after applying the instructions in order up to the
requested round.
- Be reasonably forgiving to approximate but clearly visible progress. Do not
be overly strict.

Scoring:
- Score each subcriterion independently from 0 to 10.
- Use the full range when needed. Do not default to the same pattern of
near-perfect scores unless the visual evidence truly supports it.
- Higher means the visible result better matches the intended cumulative state.

Score anchor guidance for each 0-10 subcriterion:
- 0-2: absent or clearly wrong
- 3-4: weak / barely present
- 5-6: partial but recognizable
- 7-8: strong but imperfect
- 9-10: nearly complete or excellent

Output must be strict JSON:
"subscores": "cumulative_target_match": 8, "major_edit_completion": 7,
"attribute_specificity": 9, "spatial_layout_correctness": 8, "reason":
"short explanation"
```

State consistency. The prompt for the state consistency agent measures whether the edited image forms one coherent final state implied by the executed instructions, without obvious contradictions between old and new attributes.

State Consistency

```
Role: You are a visual consistency judge for iterative image editing.
Task: Compare the original image and the current edited image. Evaluate
whether the edited image looks like one coherent visual state after applying
the instructions up to the requested round.

Important reasoning rule:
- The instruction sequence is cumulative and overwrite-aware.
- Later instructions may replace earlier edits. This is expected.
- Penalize contradictions only when the image simultaneously retains
incompatible old/new states, or when the current result does not settle into
a coherent target state.
- Be reasonably tolerant of imperfect execution if the overall intended state
is still clear.

Scoring:
- Score each subcriterion independently from 0 to 5.
- Use the full range when needed. Do not default to the same pattern of high
scores unless the evidence strongly supports it.
- Higher means the image expresses a more coherent overwrite-aware target
state.

Score anchor guidance for each 0-5 subcriterion:
- 0: failed
- 1: very poor
- 2: weak
- 3: acceptable
- 4: strong
- 5: excellent

Output must be strict JSON:
"subscores": "contradiction_free_state": 4, "global_coherence": 4,
"overwrite_consistency": 5, "state_clarity": 4, "reason": "short
explanation"
```

Visual quality. The prompt for the visual quality agent measures perceptual image quality, artifact level, and overall visual coherence of the edited image.

Visual Quality

```
Role: You are a perceptual quality judge for generated images.
Task: Evaluate only the visual quality of the edited image, while using the
original image as reference context when helpful.

Judge:
- visual coherence
- artifact severity
- object realism and boundary cleanliness
- texture plausibility
- overall perceptual quality

Important rule:
- Do not over-penalize mild imperfections.
- Reserve very low scores for severe corruption or obvious generation failure.
- The purpose is discrimination, not harsh filtering.

Scoring:
- Score each subcriterion independently from 0 to 5.
- Use the full range when needed and avoid reusing the same near-perfect
```

```
pattern by default.

Output must be strict JSON:
"subscores": "artifact_control": 4, "boundary_cleanliness": 4,
"texture_material_quality": 3, "overall_perceptual_quality": 4, "reason":
"short explanation"
```

Content preservation. The prompt for the content preservation agent measures how well non-target content, identity, and layout are preserved when they are not supposed to change.

Content Preservation

```
Role: You are a preservation judge for iterative image editing.
Task: Compare the original image and the current edited image. Evaluate
whether the image preserves important content that should remain stable while
still allowing the requested edits.

Judge:
- preservation of subject identity when identity should remain
- preservation of scene structure/layout when not explicitly replaced
- avoidance of unnecessary collateral changes

Important reasoning rule:
- If the instruction explicitly requests broad scene replacement, do not
over-penalize large intended changes.
- Penalize only unnecessary drift beyond the requested edits.
- Be reasonably forgiving when the main editable content is correct and only
minor collateral changes occur.

Scoring:
- Score each subcriterion independently from 0 to 5.
- Use the full range when needed. Avoid defaulting to the same pattern of high
scores unless preservation is clearly strong.

Output must be strict JSON:
"subscores": "identity_preservation": 4, "layout_preservation": 4,
"collateral_change_control": 3, "edit_locality": 4, "reason": "short
explanation"
```

F.7 Parameters

In this section, we summarize the critical parameters used in our practical implementation of VAE-LFA, including the number of latent channels, the window size of the low-pass filter, and the momentum coefficients.

Channels. We apply VAE-LFA on VAE latents with shape $\mathbf{z} \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of latent channels and (H, W) denotes the spatial resolution of each latent feature map. All statistics used by VAE-LFA are computed independently for each channel over the spatial dimensions. In our FLUX.2 experiments, the VAE latent has $C = 32$ channels.

Low-Pass filter. We use channel-wise average pooling as the low-pass filter. Specifically, we use replicate padding and apply a 9×9 average-pooling filter with stride 1, pad 4, to obtain the low-frequency component.

Momentum anchor. We maintain an exponential moving average of low-frequency statistics from previous pre-alignment latents. The momentum is applied separately to the channel-wise mean and log-standard-deviation. In our default setting, we use $\alpha_\mu = 0.95$ and $\alpha_\sigma = 0.85$. Specifically, the updates of mean momentum and log-standard-deviation momentum satisfy:

$$\mathbf{m}_\mu^{(k)} = 0.95\mathbf{m}_\mu^{(k-1)} + (1-0.95)\mu(\boldsymbol{\ell}^{(k)}), \quad \mathbf{m}_{\log\sigma}^{(k)} = 0.85\mathbf{m}_{\log\sigma}^{(k-1)} + (1-0.85)\log\sigma(\boldsymbol{\ell}^{(k)}).$$

The initialization satisfies:

$$\mathbf{m}_{\mu}^{(0)} = \mu(\boldsymbol{\ell}^{(0)}), \quad \mathbf{m}_{\log\sigma}^{(0)} = \log\sigma(\boldsymbol{\ell}^{(0)}),$$

where $\boldsymbol{\ell}^{(0)}$ is the low-frequency component of the initial latent.

Black-box model configurations. For all black-box API experiments, we use the official image-editing endpoints with fixed decoding parameters whenever exposed by the API. We fix the random seed to 42 for all editable API calls. When an API does not expose a particular sampling parameter, we keep the default setting unchanged and use the same setting for the base editor and the +LFA variant. We observe good reproducibility across repeated calls under the fixed seed, and all reported black-box results are obtained under this deterministic protocol.

F.8 Hardware

All white-box experiments in this paper are performed on a single NVIDIA RTX A6000 Ada GPU with 48GB VRAM, costing about 40 hours of compute time. All white-box experiments are possible with CPU offload on a single GPU with 24GB VRAM, while all black-box experiments are possible with CPU offload on a single GPU with 12GB VRAM.

G License and Terms of Use for Existing Assets

We list the existing assets used in this work and their respective licenses or terms of service.

Models and APIs.

- **FLUX.2 Klein** [18]: Used under FLUX Non-Commercial License v2.1.
- **FLUX.1 Kontext** [19]: Used under FLUX.1 [dev] Non-Commercial License.
- **Stable Diffusion 3 (SD3)** [4]: Used under Stability AI Community License.
- **Qwen Image 2.0** [38]: Evaluated via API under Alibaba's License and Terms of Service.
- **Seedream 4.0** [30]: Evaluated via API under ByteDance/Seed terms of service.
- **SD-VAE-ft-ema** [29]: Used under MIT License.
- **SANA AutoencoderDC** [40]: Used under Apache License 2.0.
- **DINOv3** [33]: Used under Meta's license terms (see project repository).
- **Qwen-3.6-Plus** [28]: Used via API under Alibaba Cloud's Terms of Service.

Models are verified to be accessible using official APIs up to Oct.20, 2026.

Evaluation Dataset. Our 120-image evaluation set was curated from publicly available web images and author-collected photographs. No identifiable personal data or sensitive content is included. The dataset will be released under CC BY 4.0 for research purposes.

Code and Libraries. Our code builds on diffusers [35] (Apache 2.0), and numpy/scipy (BSD).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by our component-wise latent frequency analysis in Sec. 3 and by the quantitative and qualitative experiments in Sec. 5. We state the scope of the method as a training-free, plug-and-play stabilization module for multi-turn DiT-based image editing.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Discussion and Limitations paragraph after Sec. 6, including cases with extreme prompts where low-frequency alignment may moderately suppress intended large global changes. We also discuss possible future directions such as dynamic prompt-strength adaptation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical formulation in Sec. 3.1 and Appendix C is used to define multi-turn editing bias and motivate the empirical analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the experimental protocols, evaluated models, dataset construction, metrics, prompt-generation procedure, and implementation details in Sec. 5 and Appendix F. The proposed method is training-free, and its key hyperparameters, including the low-pass filter and EMA coefficients, are specified in Appendix F.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our dataset and code for evaluating both white-box and black-box models is provided in the supplementary materials. All sub-folders are paired with step-by-step README files that guide result reproduction.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental details (data splits, hyperparameters, system prompts, etc.) are provided in Sec. 5 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report non-parametric percentile bootstrap 95% confidence intervals for paired round-10 improvements on the no-op and cycle protocols in Appendix D.1. Since each base editor and its +LFA variant are evaluated on identical images and instruction

sequences, the intervals are computed over paired metric differences rather than independent samples. We use 10,000 bootstrap resamples and do not assume normality of the paired differences; all reported intervals are strictly positive.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide compute resource details in Appendix F.8.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both potential positive and negative societal impacts in Appendix A.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new pretrained image generator or a high-risk scraped dataset. VAE-LFA is a training-free stabilization module evaluated on existing editors; safeguards for generation misuse are expected to be enforced by the underlying editing systems and API providers.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the existing models, datasets, and metrics used in the experiments and provide a list of licenses and terms of uses accordingly in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce an evaluation set of 120 images and prompt sequences for no-op, cycle, and long-chain multi-turn editing. We document the data taxonomy, prompt-generation procedure, and evaluation protocols in Sec. 5 and Appendix F, and provide the dataset in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This research does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research, crowdsourcing, or collection of private personal data, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use Qwen-3.6-Plus as part of the experimental pipeline for generating cycle and long-chain editing prompts, assigning object taxonomy labels, and computing VLM-based evaluation scores. The corresponding prompts and scoring protocol are provided in Appendix F.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.