EFFICIENT AND FAITHFUL HIGH-RESOLUTION IMAGE EDITING WITH SPARSE CROSS-RESOLUTION DENOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-based image editors remain predominantly 1K systems, yet practical editing increasingly demands faithful 2K/4K outputs. Direct high-resolution adaptation preserves a unified editing trajectory but incurs dense target–source attention over rapidly growing token sequences; low-resolution editing followed by external refinement is cheaper but can separate detail restoration from the editing trajectory. We introduce *sparse cross-resolution denoising*, a native high-resolution editing paradigm that retains a frozen 1K stream as a semantic and structural anchor and trains a target-resolution stream to reconstruct local detail within the same Transformer. An asymmetric sparse topology preserves full 1K attention, sends spatially aligned noisy-target guidance from the low-resolution trajectory to high-resolution queries, and restricts high-resolution target–reference interaction to local Halo neighborhoods. Resolution-specific diffusion-state mapping and spatial YaRN adapt the editor to larger grids. A training-free progressive path further replaces an initial portion of target-resolution denoising with a 1K prefix and completes refinement in the same model. On VINS-4KEval, we achieves the highest reported metrics at both 2K and 4K; in a matched 4K forward benchmark, the sparse topology is $4.2\times$ faster overall and $25.3\times$ faster in attention than dense attention. Progressive inference further reduces end-to-end latency by 30.5% at 4K while retaining competitive quality.

1 INTRODUCTION

Diffusion models have advanced both image generation and image editing toward general-purpose visual systems. Modern editors can follow open-ended instructions while preserving source content (Brooks et al., 2023; Liu et al., 2025; Wu et al., 2025). As these systems move into applications that require readable text, fine textures, and precise local structure, editing quality must extend beyond standard 1K outputs to 2K and 4K resolutions. Although high-resolution generation has progressed rapidly, editing models and benchmarks remain largely centered at 1K or below, leaving reliable high-resolution editing an open problem (Chen et al., 2026b).

Existing diffusion-based approaches expose a quality–efficiency tension. Directly adapting an editor to high resolution keeps editing and detail synthesis within one denoising trajectory, but requires long target–source sequences. Under fixed latent downsampling and tokenization, scaling both image dimensions by k increases each image's spatial-token count by approximately k^2, making dense attention over the joint visual sequence increasingly expensive. As shown in Fig. 2, a dense 4K baseline already requires 17.33 seconds for a single Transformer forward, before this cost is repeated over the denoising trajectory. Editing at 1K and applying an external super-resolution model reduces the editor's cost, but separates detail recovery from the editing trajectory and may distort source-specific content, especially in unedited regions. These two routes leave open the problem of full-canvas 2K/4K editing that is both faithful and computationally practical.

Our central insight is that high-resolution editing requires different information at different spatial scales. A capable 1K editor already establishes the instruction-conditioned semantic transformation, global layout, and principal edit structure. The high-resolution grid is primarily needed to recover the fine structures, textures, and boundaries that cannot be represented at 1K. This motivates a low-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

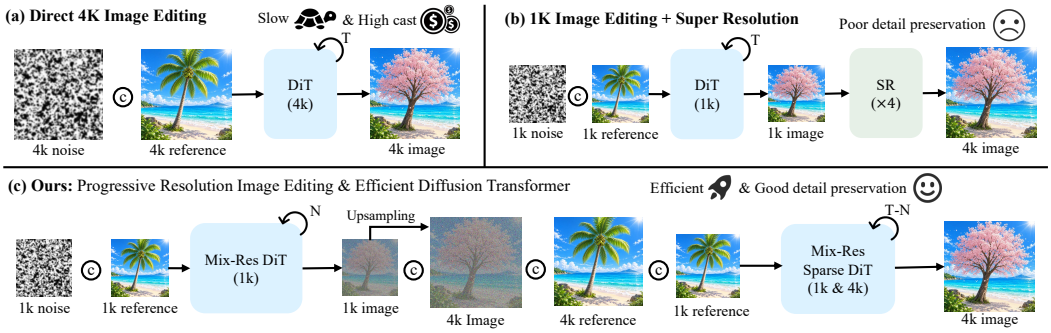


Figure 1: Three routes to high-resolution editing: dense native 4K editing, 1K editing with super-resolution, and our low-resolution-guided progressive framework.

resolution-guided high-resolution editing paradigm in which the 1K trajectory provides a reliable editing anchor and the high-resolution branch concentrates on detail generation, while the final high-resolution latent remains within the same editor.

Based on this principle, we introduce Sparse Cross-Resolution Denoising. A cross-resolution Transformer jointly processes text, low-resolution target–source tokens, and their high-resolution counterparts. The Full-Attention 1K Anchor preserves global editing behavior. Local High-Resolution Halo Attention models spatially aligned source details within bounded neighborhoods, while Spatially Aligned Low-to-High Target Guidance transfers the evolving 1K edit to high-resolution generation. Resolution-dependent timestep mapping and RoPE position adaptation make both spatial scales compatible with the pretrained editor (Peng et al., 2024). Together, these components preserve the global editing prior while restricting high-resolution interaction to the support needed for details.



Figure 2: Per-forward runtime across resolutions. Total time is shown by each bar and attention time by its upper segment. At 4K, sparse attention reduces total forward time $4.2\times$ and attention time $25.3\times$ versus dense.

To reduce repeated high-resolution computation over time, we further introduce Temporally Progressive Low-to-High-Resolution Editing. During early high-noise steps, both resolutions primarily establish coarse semantics and layout, while high-resolution textures and boundaries become more important later. The model therefore performs an initial segment of denoising at 1K, transfers the estimated clean state through RGB space, maps it to the corresponding high-resolution noise state, and completes the remaining trajectory with the same cross-resolution Transformer. This transition is internal to the editor rather than a separate super-resolution stage, so the final image remains jointly conditioned on the instruction, source image, and evolving 1K edit.

Our contributions are threefold:

1. To our knowledge, we are the first to formulate native high-resolution editing as low-resolution-guided target-resolution denoising, separating global edit formation from detail generation within one model.
2. We develop Sparse Cross-Resolution Denoising and Temporally Progressive Low-to-High-Resolution Editing to reduce spatial and temporal redundancy in high-resolution editing.
3. We demonstrate state-of-art (SOTA) 2K/4K editing quality in VINS-4KEval and a favorable efficiency–fidelity trade-off: the sparse topology is $4.2\times$ faster overall and $25.3\times$

faster in attention than dense 4K attention in a matched forward benchmark, while progressive inference further reduces complete 4K latency by 30.5%. We provide a solid approach to efficient and faithful high-resolution image editing.

2 Related Work

2.1 Diffusion-Based Image Editing

Instruction-based diffusion editing has progressed from direct text-conditioned manipulation to general-purpose source-conditioned generation. InstructPix2Pix established feed-forward instruction following, while MagicBrush strengthened human-annotated supervision and MGIE introduced multimodal reasoning for underspecified requests (Brooks et al., 2023; Zhang et al., 2023; Fu et al., 2024). Recent flow- and DiT-based editors, including Step1X-Edit, Qwen-Image-Edit, and FLUX.1 Kontext, further improve instruction fidelity and source consistency (Liu et al., 2025; Wu et al., 2025; Labs et al., 2025). Nevertheless, their training and evaluation remain concentrated around 1K (Chen et al., 2026b); directly expanding joint target–reference modeling to 2K/4K repeats global editing computation on much larger token grids. We therefore preserve the mature 1K editor as an editing anchor and adapt only target-resolution detail modeling.

2.2 Efficient High-Resolution Generation and Editing

High-resolution diffusion generation extends pretrained models through receptive-field correction, progressive multiscale synthesis, overlapping diffusion paths, and hierarchical or windowed computation (He et al., 2024; Du et al., 2024; Bar-Tal et al., 2023; Zhang et al., 2024; Crowson et al., 2024). These approaches establish that low-resolution priors remain useful beyond their training scale, but editing additionally requires instruction fidelity and source preservation.

High-resolution editing consequently follows two main routes. Cascaded systems apply generative super-resolution after a 1K edit (Chen et al., 2025a; Duan et al., 2025; Sun et al., 2025; Dong et al., 2025), reducing editor cost but moving detail reconstruction outside the editing trajectory. VINS-120K instead adapts the editor directly at 4K, retaining native denoising while incurring long target–reference sequences (Chen et al., 2026b). Complementary work targets computational redundancy rather than resolution scaling itself: RegionE accelerates instruction-based editing through adaptive region-wise denoising, while Sparse-vDiT exploits recurring structured attention sparsity in video diffusion Transformers (Chen et al., 2025b; 2026a). These studies demonstrate the value of selective computation, but do not address how a reliable low-resolution editing trajectory can guide native full-canvas reconstruction at 2K/4K. Our method fills this gap by jointly denoising a frozen 1K anchor and a sparse target-resolution stream through directed cross-resolution interactions.

3 Method

High-resolution image editing must preserve instruction fidelity and source details at larger spatial scales while controlling rapidly increasing computation. We address this challenge with a spatiotemporally efficient framework built on a pretrained 1K editor. Spatially, the native 1K branch retains instruction semantics, global layout, and the principal edit structure, while its evolving state guides a target-resolution branch that reconstructs local detail without repeating dense global modeling over the full high-resolution sequence. Temporally, we exploit the coarse-to-fine denoising process: early steps establish layout and object contours, whereas later steps refine target-scale textures and high-frequency details. A low-to-high progressive path therefore executes a 1K prefix, transfers the resulting state to the target resolution, and completes the trajectory with the same editor. Together, these designs enable efficient and faithful native high-resolution editing.

3.1 High-Resolution Image Editing

3.1.1 Flow-Matching Image Editing

Given a source image I_R and instruction c, the editor generates a target I_0 while preserving unrelated source content. The VAE encodes I_R as clean conditioning latent z_R, while the target latent starts

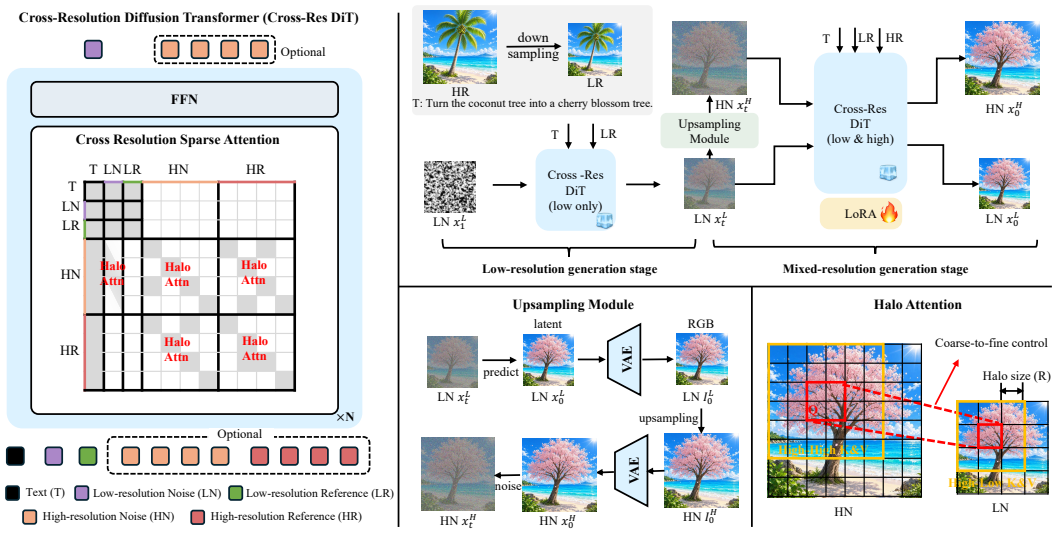


Figure 3: Overview of the proposed framework. The frozen 1K stream anchors global editing, sparse high-resolution attention reconstructs local detail, and progressive inference replaces an initial high-resolution prefix with 1K denoising.

from Gaussian noise x_1 (Rombach et al., 2022). Each Transformer block predicts $v_\theta(x_t, t; c, z_R)$ from noisy target, source, and text tokens; integrating this field from $t = 1$ to 0 and decoding the final latent yields the edit.

For training, encode I_0 as clean latent z_0, sample $t \in [0, 1]$ and $\epsilon \sim \mathcal{N}(0, I)$, and form the linear path

$$x_t = (1 - t)z_0 + t\epsilon, \qquad v^\star = \frac{\mathrm{d}x_t}{\mathrm{d}t} = \epsilon - z_0.$$

Here, $t = 0$ is the clean target and $t = 1$ is pure noise. The conditional velocity is learned from c and z_R by (Lipman et al., 2022)

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{z_0, z_R, c, \epsilon, t}\left[\|v_\theta(x_t, t; c, z_R) - (\epsilon - z_0)\|_2^2\right].$$

3.1.2 Computational Bottleneck at High Resolution

With fixed latent downsampling and tokenization, scaling both image dimensions by k increases spatial tokens by approximately k^2. VAE, projections, and FFNs grow with token count, whereas dense attention computes all query–key pairs (Vaswani et al., 2017; Crowson et al., 2024). Memory-efficient exact attention reduces score materialization but retains dense pairwise arithmetic (Dao et al., 2022). Figure 2 isolates the attention overhead: dense attention takes 15.17 s of the 17.33-s 4K forward. Our sparse topology reduces these to 0.60 s and 4.13 s, respectively, yielding 25.3× attention and 4.2× total-forward speedups. As every denoising step repeats this cost, attention is the primary per-step target for acceleration.

3.2 High-Resolution Capability Adaptation

A pretrained 1K editor encounters two resolution-dependent mismatches when applied to a larger latent grid. The same nominal timestep corresponds to a different effective noise state at a different spatial scale, while the enlarged grid contains coordinates beyond the positional range observed during pretraining. We address these mismatches with resolution-dependent timestep mapping and spatial YaRN, establishing the diffusion-state and positional representations required at the target resolution.

Resolution-dependent timestep mapping. At higher resolutions, increased spatial redundancy preserves more recoverable signal under the same nominal noise level. Directly reusing a sched-

ule designed for a lower resolution can therefore make high-resolution states effectively cleaner and change the denoising difficulty encountered by the model. Existing high-resolution diffusion and rectified-flow models address this issue with resolution-dependent timestep shifts (Chen, 2023; Hoogeboom et al., 2023; Esser et al., 2024). Given a raw timestep u, we map it to a resolution-specific shifted state:

$$s = f(u; \mu) = \frac{e^{\mu} u}{1 + (e^{\mu} - 1)u},$$

where μ controls the target-resolution mapping. The resulting state s constructs the noisy input and its temporal conditioning, changing the effective time parameterization without modifying the token topology or network architecture. The mapping parameters are reported in the experimental settings.

RoPE position adaptation. Applying a pretrained editor to a larger spatial grid requires RoPE to represent positions beyond the range observed during pretraining (Su et al., 2024). Insufficient adaptation can cause semantic drift or local repetition in high-resolution editing (Chen et al., 2026b). Uniform interpolation compresses phase differences across frequencies, so we use YaRN's frequency-aware transition between interpolation and extrapolation (Peng et al., 2024).

For target-grid coordinate i, the pretrained-frame coordinate is $p_i = i/n$, where n is the target-to-pretrained scale. For rotary dimension j, the native inverse frequency $\omega_j = \theta^{-2j/d}$ is adjusted by YaRN as

$$\widetilde{\omega}_j = \omega_j[n - (n-1)\gamma_j], \qquad \phi_j(i) = p_i \widetilde{\omega}_j,$$

where $\gamma_j \in [0, 1]$ is the frequency ramp. This aligns low-frequency phases while preserving finer high-frequency variation; other axes retain native RoPE.

3.3 Sparse Cross-Resolution Denoising

The resolution adaptations above enable the pretrained editor to operate on the target-resolution grid, but they do not resolve the efficiency bottleneck caused by dense full attention. This raises a central question: how can we reduce the computational cost of target-resolution editing while preserving editing responsiveness and source-image fidelity? We decompose high-resolution editing into two complementary modeling processes: the low-resolution branch provides an instruction-conditioned global editing prior, while the high-resolution branch supplements target-scale detail under this guidance. Based on this division of labor, we introduce Sparse Cross-Resolution Denoising. Here, cross-resolution denotes joint denoising of the 1K and target-resolution states within one Transformer, with explicitly constrained information exchange rather than a post-hoc super-resolution bridge. This design allows the high-resolution branch to recover local structures, textures, and boundaries without repeating the complete global editing computation on the target-resolution grid.

At each denoising step, our cross-resolution Transformer block jointly processes five token groups from a mixed-resolution token sequence,

$$\mathcal{X} = [\mathbf{T}_{\text{text}}, \mathbf{T}_{L_N}, \mathbf{T}_{L_R}, \mathbf{T}_{H_N}, \mathbf{T}_{H_R}],$$

where $\mathbf{T}_{\text{text}}$ denotes text tokens; $\mathbf{T}_{L_N}$ and $\mathbf{T}_{L_R}$ denote the low-resolution noisy-target and clean-source tokens; and $\mathbf{T}_{H_N}$ and $\mathbf{T}_{H_R}$ denote their target-resolution counterparts. This joint sequence is the basis of the attention topology below.

Full-attention 1K anchor. The low-resolution branch retains the pretrained editor's native attention pattern on the 1K grid, preserving instruction understanding, global layout, and the principal edit structure. Low-resolution image queries do not read high-resolution image keys, so the high-resolution branch cannot alter the global 1K trajectory through attention. The global editing decision is therefore maintained at the smaller spatial scale and serves as the anchor for subsequent target-resolution modeling.

Local High-Resolution Halo Attention. Conditioned on the low-resolution editing anchor, the high-resolution branch establishes local target–source attention on its native grid to recover target-scale structures and textures. Each low-resolution parent cell corresponds to an $n \times n$ high-resolution region, which is expanded by h latent-token units on each side to form a Halo support. Before boundary clipping, the support size for each high-resolution segment is

$$K_H = (n + 2h)^2$$

This support covers the high-resolution noisy-target and reference tokens, allowing the high-resolution branch to use spatially aligned source details and exchange local context across neighboring parent cells. Because the attention support is defined by fixed local geometry on the latent-token grid, the high-resolution branch avoids global dense attention over the complete target–source sequence and requires no additional user-provided edit mask.

Spatially Aligned Low-to-High Target Guidance. Local Halo attention defines how the high-resolution branch generates details on the target grid, but local high-resolution tokens alone do not contain the global editing state established by the low-resolution branch. We therefore introduce a spatially aligned pathway $L_N \to H_N$ from the low-resolution noisy target to the high-resolution noisy target, injecting the evolving low-resolution editing trajectory into the local high-resolution generation process. Since the high-resolution grid is n times larger along each spatial axis, every high-resolution location (y, x) maps to a low-resolution parent cell

$$\pi(y, x) = \left(\left\lfloor \frac{y}{n} \right\rfloor, \left\lfloor \frac{x}{n} \right\rfloor \right).$$

Each high-resolution noisy-target query reads a low-resolution noisy-target neighborhood of radius R centered at $\pi(y, x)$. Under Chebyshev distance, this support contains at most

$$K_L = (2R + 1)^2$$

low-resolution keys. The semantic information flows from the evolving low-resolution edit to the high-resolution target, while the attention operation is implemented by high-resolution queries reading low-resolution keys. This bounded pathway aligns local target-resolution generation with the global editing trajectory without introducing dense cross-resolution interaction.

For efficient execution, we arrange high-resolution tokens according to their low-resolution parent cells, keeping tokens from the same parent cell contiguous in the sequence. This ordering exposes the local Halo and cross-resolution supports as regular neighborhoods, which a customized sparse-attention kernel evaluates directly without constructing a dense attention mask or score matrix. The resulting topology adds only the token-wise projection and FFN computation of the relatively small low-resolution branch, while substantially reducing high-resolution attention computation and end-to-end latency. It preserves the 1K global editing interaction and concentrates target-resolution computation on local structure and detail generation; the full support-count derivation is deferred to the supplementary material.

3.4 Temporally Progressive Low-to-High-Resolution Editing

Direct joint denoising, in which both resolution branches follow the complete paired schedule, is the standard inference path. We also consider a training-free progressive path motivated by the coarse-to-fine nature of denoising: the early high-noise trajectory can be formed at 1K, while target-scale refinement is reserved for later high-resolution steps. For a transition index $1 \le K < N$, the model first runs an initial 1K denoising segment and then enters the same cross-resolution denoising model for the remaining trajectory.

At the transition, the low-resolution state x_L^K and its velocity prediction provide a clean estimate

$$\hat{x}_{0,L}^K = x_L^K - s_L^K v_{\theta,L}^K.$$

We transfer this estimate to the target resolution through RGB/VAE space:

$$\hat{I}_L^K = D_{\mathrm{VAE}}\big(\hat{x}_{0,L}^K\big), \qquad \widetilde{I}_H^K = \mathcal{R}_{\mathrm{RGB}}\Big(\hat{I}_L^K\Big), \hat{x}_{0,H}^K = E_{\mathrm{VAE}}\Big(\widetilde{I}_H^K\Big).$$

The re-encoded target-resolution state is then re-noised with independent Gaussian noise at the corresponding high-resolution mapped timestep:

$$x_H^K = (1 - s_H^K)\hat{x}_{0,H}^K + s_H^K \epsilon_H, \qquad \epsilon_H \sim \mathcal{N}(0, I).$$

The initialized high-resolution branch then joins the low-resolution branch for the remaining paired steps. Unlike an external super-resolution cascade, the RGB-resized image is only an initialization; the same trained editor performs the subsequent target-resolution denoising with the instruction, source condition, and low-resolution editing state. Thus, K directly controls the quality–efficiency trade-off: increasing K reduces expensive high-resolution computation but leaves fewer steps for target-resolution refinement. Latent interpolation is evaluated separately as a transition ablation.

Table 1: Main 2K/4K comparison. Bold and underlined mark the best and second-best quality per resolution; time is not ranked. Shaded areas indicate inference w/o super resolution.

Method	HaarPSI↑	mPSNR↑	mSSIM↑	mMSE↓	mLPIPS↓	Time↓
2K Resolution						
FaithDiff	0.3840	21.7671	0.6900	0.01700	0.3101	29.43
DiT4SR	0.3761	21.3047	0.6966	0.01757	0.3130	52.91
PiSA-SR	0.3651	20.7503	0.6647	0.01843	0.3177	16.63
TSD-SR	0.3600	20.1663	0.6468	0.02035	0.3338	16.89
VINS	<u>0.4164</u>	22.2907	0.7065	0.01693	**0.2486**	100.82
Ours (R=3)	**0.4235**	**22.5668**	**0.7780**	<u>0.01616</u>	<u>0.2976</u>	108.95
Ours (Progressive, R=3, K=12)	0.4137	<u>22.4258</u>	<u>0.7733</u>	**0.01554**	0.3165	88.19
4K Resolution						
FaithDiff	0.3630	21.5728	0.6578	0.01741	0.3079	87.04
DiT4SR	0.3512	21.3398	0.6806	0.01786	0.3137	317.73
PiSA-SR	0.3421	21.0066	0.6610	0.01817	0.3149	146.18
TSD-SR	0.3352	20.5414	0.6329	0.01962	<u>0.3161</u>	62.46
VINS	0.2426	12.6890	0.4598	0.07081	0.4185	997.21
Ours (R=3)	**0.4067**	**22.2891**	**0.7621**	**0.01648**	**0.2934**	309.69
Ours (Progressive, R=3, K=12)	<u>0.3775</u>	<u>21.6970</u>	<u>0.7369</u>	<u>0.01664</u>	0.3190	215.31

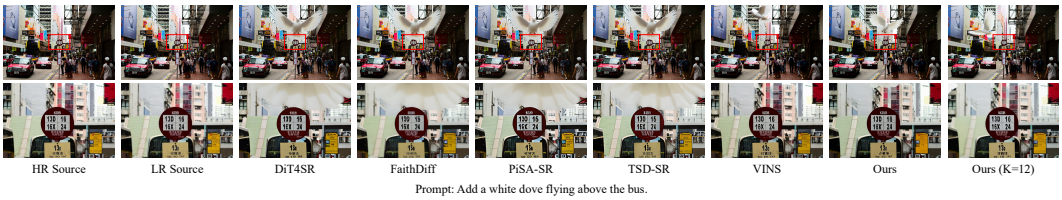


Figure 4: 2K outputs for "Add a white dove flying above the bus." Direct and progressive predictions are compared with edit–super-resolution cascades and VINS; crops show the edit and nearby text detail.

4 Experiments

This section evaluates source-faithful editing quality and target-resolution efficiency at 2K and 4K. We report the shared evaluation protocol, compare against representative edit–super-resolution cascades and native high-resolution adaptation, and isolate each component of the proposed framework. Detailed mask construction and implementation settings are deferred to the supplementary material.

4.1 Experimental Settings and Evaluation Protocol

We use FLUX.2 Klein Base 4B as the editing backbone, the full VINS-120K training split (131,010 examples), and VINS-4KEval (509 instructions across 12 edit categories). All methods share the source images, instructions, seeds, and target resolutions. Local edits use a shared source-only edit mask; global edits are excluded from region-restricted scores. We compare FaithDiff, DiT4SR, PiSA-SR, TSD-SR, and training-free VINS.

Every route starts from the same 1K SFT checkpoint. The proposed model freezes the $L = 1\text{K}$ branch and adapts $H \in \{2\text{K}, 4\text{K}\}$ with low-rank residuals (Hu et al., 2021). Default direct and progressive configurations are used throughout; adapter, topology, schedule, and optimizer details are given in the supplement.

We report HaarPSI, keep-region mPSNR/mSSIM/mMSE, edit-region mLPIPS, and end-to-end time. Quality values are macro averages over edit types; timing excludes one-time model loading and includes the external SR stage where applicable.

Table 2: 2K synchronized Halo ablation. Increasing R expands low- resolution support and the matched high-resolution Halo; time is not ranked.

Support	h	Haar↑	PSNR↑	SSIM↑	MSE↓	LPIPS↓	Time↓
$R = 1$	2	0.4234	22.4356	0.7774	0.01591	0.2966	94.42
$R = 2$	4	0.4213	22.4387	0.7732	0.01650	0.2977	99.64
$R = 3$	6	0.4235	22.5668	0.7780	0.01616	0.2976	108.95

Table 3: Progressive transition ablation at $R = 3$. K is the initial 1K-prefix length; Direct denotes joint denoising. Time is end-to-end latency in seconds.

2K Resolution

Transition	Haar↑	PSNR↑	SSIM↑	MSE↓	LPIPS↓	Time↓
Direct	0.4235	22.5668	0.7780	0.01616	0.2976	108.95
$K = 8$	0.4173	22.4866	0.7768	0.01551	0.3132	102.32
$K = 12$	0.4137	22.4258	0.7733	0.01554	0.3165	88.19
$K = 16$	0.4097	22.4041	0.7693	0.01563	0.3157	74.84

4K Resolution

Transition	Haar↑	PSNR↑	SSIM↑	MSE↓	LPIPS↓	Time↓
Direct	0.4067	22.2891	0.7621	0.01648	0.2934	309.69
$K = 8$	0.3888	21.8677	0.7470	0.01651	0.3118	257.22
$K = 12$	0.3775	21.6970	0.7369	0.01664	0.3190	215.31
$K = 16$	0.3682	21.5672	0.7299	0.01678	0.3205	174.12

4.2 Main Results

Table 1 shows that the direct model provides the strongest overall quality among the retained methods: it leads on the principal structural metrics at both resolutions and on 4K mMSE and edit-region mLPIPS. The progressive variant remains competitive while reducing complete latency by 19.1% at 2K and 30.5% at 4K.

The direct model is also substantially more efficient than dense target- resolution attention: in the matched 4K forward benchmark, it is $4.2\times$ faster overall and $25.3\times$ faster in attention. Thus, the quality gains do not come from accepting the full dense cost; they result from retaining the 1K editing anchor while restricting target-resolution interaction.

4.3 Ablation Studies

Halo-supported local attention. Table 2 varies the synchronized support while keeping the checkpoint, timestep mapping, topology, and evaluation protocol fixed. At 2K, increasing R from 1 to 3 improves the full-image perceptual and structural metrics, but the gain is not uniform: $R = 1$ gives the lowest non-edited-region mMSE and edit-region mLPIPS, while $R = 3$ incurs the largest latency. The time column is an efficiency diagnostic rather than a quality ranking. This exposes the intended trade-off between local context and computation rather than suggesting that the largest support is always optimal. Figure 5 provides a matched qualitative $R = 0$ diagnostic: removing Halo support weakens cross-window continuity and local texture around the edit boundary.

Temporally progressive low-to-high editing. Table 3 evaluates the transition index K at both target resolutions. A larger K replaces more target-resolution denoising with the 1K prefix and leaves fewer steps for high-resolution refinement. At 2K, $K = 8$ provides the strongest progressive quality among the evaluated transitions, whereas the 4K results favor the longer $K = 12$ prefix among the reported progressive settings. The table uses corrected complete-generation timing for $K = 8, 12, 16$; representative transition outputs are provided in the supplement.

Low-resolution anchor and low-to-high guidance. We next remove the complete 1K editing anchor while retaining the high-resolution sparse stream. This controlled qualitative diagnostic removes the evolving low-resolution target state, including the spatially aligned low-to-high guidance, so the high-resolution branch must infer instruction-conditioned layout and object placement from bounded local interactions alone. Figure 6 shows this structural failure mode. The edge-wise study that separates $L_N \to H_N$ guidance from low-reference cross-links is reported in the supplement.

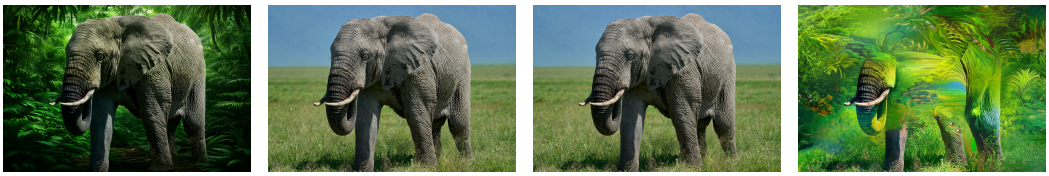
RGB/VAE versus latent lifting. We compare RGB resize followed by target-resolution VAE encoding with direct latent interpolation. Subsequent denoising is identical, isolating the transition operator; Fig. 7 shows upsampling in RGB space works better in information rentention.



HR Source | LR Source | R=3 (Ours) | R=0 (w/o Halo)

Prompt: Change the indoor setting to an outdoor garden.

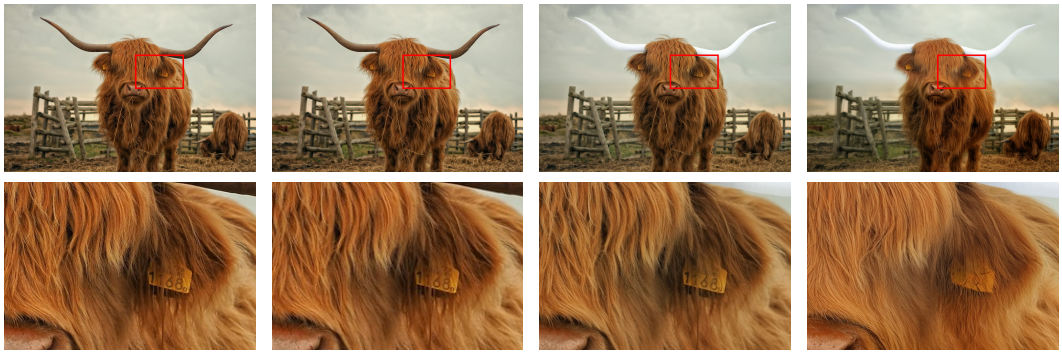
Figure 5: 2K Halo ablation under a shared instruction. $R = 3$ is compared with no Halo ($R = 0, h = 0$), highlighting cross-window continuity and texture.



HR Source | LR Source | Cross-Res (Ours) | Sparse High-Res only

Prompt: Transform the backdrop into a tropical rainforest.

Figure 6: 2K anchor ablation. The sparse high-resolution-only variant removes the 1K stream; the full model retains instruction-driven scene layout.



HR Source | LR Source | w/ RGB | w/ latent

Prompt: Turn the cow's horns white.

Figure 7: 2K lifting ablation at $R = 3$. RGB/VAE and latent variants share the same progressive denoising and differ only in the lifting operator.

5 Conclusion

We presented sparse cross-resolution denoising for efficient and faithful native high-resolution image editing. The framework treats the 1K editor as an editing anchor rather than recomputing global instruction-conditioned reasoning densely on the target-resolution grid. A frozen low-resolution stream preserves the established edit trajectory, while spatially aligned low-to-high guidance and local Halo attention allow a trainable high-resolution stream to reconstruct target-scale structures and textures with bounded interaction. Resolution-specific diffusion states and spatial position adaptation make the two grids compatible with the pretrained editor. The optional progressive path further shifts computation toward the later high-resolution refinement stage and provides an explicit quality–latency control through its transition index. Experiments at 2K and 4K show that this combination offers a practical alternative to dense native editing and edit–then–refine cascades.

AI USE STATEMENT

Generative AI tools were used only to assist with academic writing tasks, including language polishing, grammar correction, restructuring of paragraphs, and improving clarity of captions and supplementary explanations. They were not used to generate experimental data, formulate mathematical claims, implement the proposed method, run experiments, or interpret experimental results. All AI-assisted text was reviewed and revised by the authors, and the authors take responsibility for the final manuscript, claims, figures, and released artifacts.

ETHICS STATEMENT

This work does not involve human subjects, new personal-data collection, or human-participant experiments. We evaluate image editing models on an existing benchmark and report model behavior, source-image preservation, and computational cost. The released evaluation protocol should be used in accordance with the licenses and terms of the underlying benchmark and model checkpoints.

REPRODUCIBILITY STATEMENT

The main paper specifies the model formulation, token topology, resolution mapping, training objective, and evaluation metrics. The supplementary material provides the resolved training and inference settings, pseudocode for permutation, sparse attention, and progressive inference, the complete evaluation-mask construction protocol, kernel implementation details, and additional ablations and visualizations. The released run records include the resolved checkpoint, seed, resolution, topology, support radius, transition index, timing boundary, and per-sample metrics.

REFERENCES

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

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

Junyang Chen, Jinshan Pan, and Jiangxin Dong. Faithdiff: Unleashing diffusion priors for faithful image super-resolution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28188–28197. IEEE, 2025a.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Peng Ye, Bangyin Xiang, Zhibo Wang, Wei Cheng, Gang Yu, and Tao Chen. Regione: Adaptive region-aware generation for efficient image editing. *arXiv preprint arXiv:2510.25590*, 2025b.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Wei Cheng, Gang Yu, and Tao Chen. Sparse-vdit: Unleashing the power of sparse attention to accelerate video diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2957–2965, 2026a.

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Zhizhou Chen, Shanyan Guan, Zhanxin Gao, En Ci, Yanhao Ge, Wei Li, Zhenyu Zhang, Jian Yang, and Ying Tai. Vins-120k: Ultra high-resolution image editing with a large-scale dataset. *arXiv preprint arXiv:2605.23518*, 2026b.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. *arXiv preprint arXiv:2401.11605*, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Linwei Dong, Qingnan Fan, Yihong Guo, Zhonghao Wang, Qi Zhang, Jinwei Chen, Yawei Luo, and Changqing Zou. Tsd-sr: One-step diffusion with target score distillation for real-world image super-resolution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23174–23184. IEEE, 2025.

Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6159–6168. IEEE, 2024.

Zheng-Peng Duan, Jiawei Zhang, Xin Jin, Ziheng Zhang, Zheng Xiong, Dongqing Zou, Jimmy S Ren, Chunle Guo, and Chongyi Li. Dit4sr: Taming diffusion transformer for real-world image super-resolution. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18948–18958. IEEE, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54820–54833, 2024.

Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *International conference on learning representations*, volume 2024, pp. 37664–37684, 2024.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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.

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

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2333–2343. IEEE, 2025.

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

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.

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.

Shen Zhang, Zhaowei Chen, Zhenyu Zhao, Yuhao Chen, Yao Tang, and Jiajun Liang. Hidiffusion: Unlocking higher-resolution creativity and efficiency in pretrained diffusion models. In *European Conference on Computer Vision*, pp. 145–161. Springer, 2024.

Efficient and Faithful High-Resolution Image Editing with Sparse Cross-Resolution Denoising

Supplementary Material

This supplementary material is organized as Sections A–H. It reports the resolved implementation contract and evaluation protocol used by the anonymous submission. Numerical results are intentionally not repeated here. Settings that are not recoverable from the checked-in configuration or logs are identified as unresolved rather than filled with inferred values.

A Training and Inference Settings for FLUX.2 Klein Base 4B

A.1 Model initialization and data

The model is initialized from FLUX.2 Klein Base 4B. The low-resolution editor is first fine-tuned at 1K on the VINS-120K training split, whose resolved training manifest contains 131,010 examples. The resulting 1K editor is merged into the base Transformer and then frozen for the high-resolution adaptation. The high-resolution stage does not re-train the text encoder or replace the timestep embedding network. It trains a high-resolution-only LoRA branch while retaining the merged 1K editor as the editing anchor.

The target resolutions are $H \in \{2\mathrm{K}, 4\mathrm{K}\}$, with $L = 1\mathrm{K}$ and scale factor $n = H/L$. Thus, a low-resolution parent cell corresponds to an $n \times n$ group of high-resolution latent tokens. The native VAE encodings of the target and reference images are formed at their respective resolutions; the high-resolution noisy latent is not created by upsampling a low-resolution noisy latent.

A.2 Trainable parameters and optimization

The high-resolution adaptation uses rank-32 LoRA modules on attention and FFN projections. The 1K LoRA is frozen because its effect has already been absorbed into the merged 1K checkpoint. The checked inference logs also record `train_io_embedding_lora=0` and `separate_qkv_lora=0`; the final reported recipe therefore does not add a separate input/output embedding LoRA or a separate QKV adapter bank. Only the high-resolution residual path is updated in the high-resolution stage.

The local FLUX.2 ablation protocol resolves the optimizer as AdamW with learning rate 2×10^{-5}, weight decay 0.01, and 100 warm-up steps. The first 2K screening stage uses a matched 2,000-step budget; the same data order, seed policy, adapter rank, optimizer family, and inference contract are retained when comparing high-resolution variants. A final training run should archive the resolved checkpoint, sampler, data split, and step count together with the evaluation manifest.

Table 4: Resolved settings used by the anonymous submission. A dash denotes that the field is not an independent model parameter.

Item	Setting
Base model	FLUX.2 Klein Base 4B
Low-resolution editor	1K SFT checkpoint, merged and frozen
Training data	Full VINS-120K split; 131,010 examples
High-resolution branch	Rank-32 LoRA on attention and FFN projections
Optimizer	AdamW; learning rate 2×10^{-5}; weight decay 0.01
Warm-up	100 optimizer steps
High-resolution targets	2K and 4K
Low-to-high scale	$n = 2$ for 2K; $n = 4$ for 4K
RoPE	YaRN spatial adaptation
Timestep conditioning	Paired low/high branch conditioning (B2 contract)
Sparse support	$R = 3$, synchronized Halo $h = Rn$ in the submission recipe
IO/QKV adapters	IO embedding LoRA disabled; separate QKV LoRA disabled

A.3 Noise schedule and conditioning

The training sampler draws a shared raw $u \in [0, 1]$ and applies the resolution-dependent flow-matching shift

$$s(u; \mu) = \frac{\exp(\mu)u}{1 + (\exp(\mu) - 1)u}.$$

The low-resolution branch uses the 1K anchor schedule, while the high-resolution branch uses the paired high-resolution state in the B2 conditioning contract. The high branch receives its own branch timestep in the branch-conditioned Transformer blocks; reference rows remain clean conditioning inputs. In the production FLUX.2 protocol the low anchor uses $\mu_L = 1.15$, and the 2K high branch uses the checked paired value $\mu_H = 1.380517$. The 4K protocol uses $\mu_H = 2.302585$ in the resolved B2 recipe.

The training loss is the flow-matching velocity regression on native resolution latents. For a clean latent z_0, independent Gaussian noise ϵ, and branch state s,

$$x_s = (1 - s)z_0 + s\epsilon, \qquad v^\star = \epsilon - z_0.$$

The high-resolution branch is optimized against $v^\star_H$, while the frozen low-resolution branch provides the editing anchor. Any auxiliary cross-resolution consistency term must be recorded as part of the training configuration; it is not silently assumed in the results reported by the main paper.

A.4 Direct and progressive inference

Direct inference uses 28 denoising steps, classifier-free guidance scale 4.0, a batch size of one per generated image, and the full joint low–high trajectory. The checked production logs report `rope=yarn`, `resolution_mode=joint`, `kv_cache=0`, and one warm-up forward before timing. Timing boundaries exclude one-time model loading; the progressive timing boundary includes the complete 1K prefix and the resolution transition.

Progressive inference uses transition index $K = 12$ in the main configuration. It executes the first K steps with the native 1K editor, forms a clean estimate, lifts it to the target latent grid with the default latent-bilinear operator, and re-noises it at the corresponding high-resolution state. The remaining $N - K$ steps use the same trained cross-resolution model as direct inference. RGB resize followed by target-resolution VAE re-encoding is evaluated separately as a lifting ablation; it is not the default progressive path.

B Pseudocode for Training, Permutation, Sparse Support, and Progressive Inference

B.1 High-resolution training

Algorithm 1: high-resolution branch training.

```
Input: source/target images I_R, I_0; instruction c; frozen 1K editor L
       and trainable high-resolution LoRA parameters theta_H
for each training example do
    z_L^R, z_L^0 <- VAE_encode(I_R at 1K), VAE_encode(I_0 at 1K)
    z_H^R, z_H^0 <- VAE_encode(I_R at H),   VAE_encode(I_0 at H)
    u <- Uniform(0, 1); s_L <- shift(u; mu_L); s_H <- shift(u; mu_H)
    eps_L, eps_H <- independent standard Gaussian noise
    x_L <- (1 - s_L) * z_L^0 + s_L * eps_L
    x_H <- (1 - s_H) * z_H^0 + s_H * eps_H
    X <- pack(text(c), x_L, z_L^R, x_H, z_H^R)
    X_H <- high-resolution segment of X
    X_H <- parent_cell_permutation(X_H, n)
    X <- replace high-resolution segment in X with X_H
    v_L, v_H <- sparse_cross_resolution_transformer(X; theta_H)
    loss <- mean_square(v_H, eps_H - z_H^0)
```

```
    update theta_H using AdamW(loss)
end for
Return: frozen merged 1K editor and high-resolution LoRA checkpoint
```

The low-resolution stream is present in the forward pass so that the high-resolution branch can read the evolving editing state. Freezing the merged 1K branch prevents high-resolution adaptation from changing the native 1K editing prior.

B.2 PARENT-CELL PERMUTATION

Algorithm 2: window-major parent-cell permutation.

```
Input: low grid (H_L, W_L), high grid (H_H, W_H), scale n = H_H/H_L
for each high token coordinate (y, x) do
    parent_y <- floor(y / n); parent_x <- floor(x / n)
    local_y  <- y mod n;       local_x  <- x mod n
    key <- (parent_y, parent_x, local_y, local_x)
    append original index to bucket[key]
end for
perm <- concatenate buckets in parent-cell row-major and local row-major order
X_perm <- gather(X_high, perm)
meta_perm <- gather(token_metadata, perm)
Return X_perm, meta_perm, inverse_perm
```

The permutation makes all n^2 high-resolution tokens belonging to the same low-resolution parent cell contiguous. It changes storage order only; the inverse permutation restores the canonical sequence before decoding or writing diagnostics.

B.3 SPARSE ATTENTION SUPPORT

Algorithm 3: support construction for one Transformer block.

```
Input: token metadata; parent radius R; high Halo width h;
       declared topology gates for the selected run
for each query q do
    if q is text or a low-resolution image token then
        keys(q) <- native low/text support under the declared gates
    else if q is H_N then
        p <- parent_cell(q)
        K_hi <- H_N and H_R tokens in the local Halo around p
        K_lo <- L_N tokens in parent neighborhood Chebyshev_radius(R)
        K_txt <- text tokens
        keys(q) <- K_hi union K_lo union K_txt
    else if q is H_R then
        keys(q) <- local high-resolution reference support
                       plus gate-dependent text/high support
    end if
    emit CSR row offsets and exact key indices for q
end for
apply one normalized sparse softmax over each emitted row
Return: sparse attention output and row metadata
```

The high-resolution query rows are formed from exact spatial indices rather than a dense mask. Boundary cells are clipped to the valid image grid. Training uses the Flex/reference mask; CSR/Triton is an inference backend for the same connectivity plan when enabled. Under the declared `no_low_ref` gate, high queries do not read low-reference keys, but they retain the selected local low-noisy-target support.

B.4 PROGRESSIVE LOW-TO-HIGH INFERENCE

Algorithm 4: training-free progressive inference.

```
Input: source I_R, instruction c, total steps N, transition K
z_L^R <- VAE_encode(I_R at 1K)
z_H^R <- VAE_encode(I_R at target resolution H)
x_L <- initialize_noise(1K)
for i = 0 ... K-1 do
    v_L <- frozen_1K_editor(x_L, z_L^R, c, s_L[i])
    x_L <- Euler_step(x_L, v_L, s_L[i] -> s_L[i+1])
end for
x0_L <- clean_estimate(x_L, v_L, s_L[K])
x0_H <- latent_bilinear_lift(x0_L, target resolution H)
x_H  <- (1 - s_H[K]) * x0_H + s_H[K] * Gaussian_noise()
for i = K ... N-1 do
    v_L, v_H <- cross_resolution_model(
        x_L, x_H, z_L^R, z_H^R, c, s_L[i], s_H[i])
    x_L, x_H <- paired_Euler_step(x_L, x_H, v_L, v_H)
end for
Return VAE_decode(x_H)
```

C GROUNDINGDINO+SAM2 EVALUATION MASKS AND METRIC PROTOCOL

C.1 EVALUATION CONTRACT

The evaluation set is VINS-4KEval with 509 instructions spanning 12 edit categories. A mask is generated before method comparison from the source image and the instruction-derived edit specification only. Generated outputs are never passed to mask construction. The same frozen mask is attached to every method, resolution route, and transition setting for a case.

The checked evaluation stack uses GroundingDINO-Tiny, converted to the Transformers split-QKV format with strict state-dict checks, and SAM2.1 Hiera Tiny. GroundingDINO runs in FP32; SAM2 uses the model dtype on the selected device. The fixed detector contract is a box threshold of 0.25, a text threshold of 0.20, and at most four detections per query. The final mask is dilated with an elliptical structuring element whose radius is 1.5% of the shorter source-image dimension.

C.2 INSTRUCTION PARSING AND QUERY ROUTING

The deterministic edit parser records

$(\texttt{mask_scope}, \texttt{source_query}, \texttt{output_query}, \texttt{anchor_query}, \texttt{anchor_relation}, \texttt{parse_status})$.

Part-level phrases are kept at part granularity when possible. For example, “eyes,” “beak,” and “horns” are queried directly rather than replacing them with the full parent object. Subject replacement stores both the source and requested object phrases, while addition uses a source-only spatial anchor because the requested object is absent from the source.

C.3 MASK CONSTRUCTION

For local deletion, replacement, and attribute edits, GroundingDINO detects the source phrase in the source image only. SAM2 receives the detected boxes and returns one or more masks; the union is the source edit mask M_e, followed by the fixed dilation. If the SAM2 backend is disabled for a diagnostic, the detector boxes are used as a coarse fallback and the backend is recorded as `grounding_box`; it is not silently merged with SAM2 masks.

For subject addition, the parser extracts a relation such as “above,” “near,” “left,” or “in the background.” GroundingDINO/SAM2 localize the source anchor, and a deterministic relation-aware region around that anchor is used as M_e. If no concrete anchor is detected, the parser’s position

prior supplies a deterministic foreground/background/left/right fallback. The generated image is never used to find the added object.

For background edits, the evaluator detects a category-aware source foreground query (for example, `animal.`, `person.`, `vehicle.`, or `object.`) and segments it with SAM2. Let M_f be the dilated foreground mask. The editable region is

$$M_e = 1 - M_f, \qquad M_k = M_f,$$

so the changed background is scored as editable while the source foreground is evaluated for preservation. Global tone, style, camera, and personalized generation edits use $M_e = \mathbf{1}$ and are marked `global_not_applicable` for local keep-region metrics.

C.4 Resolution normalization and per-image metrics

Every output is converted to the common target canvas before mask application. The mask is resized with nearest-neighbor interpolation. Let I_s be the source, I_o the output, M_e the editable region, and $M_k = 1 - M_e$ the keep region.

Non-edited-region preservation. For the keep region, the evaluator reports

$$\mathrm{mMSE} = \frac{1}{|M_k|} \sum_{p \in M_k} \|I_s(p) - I_o(p)\|_2^2 ,$$

$$\mathrm{mPSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(M_k)}.$$

SSIM is computed after replacing the masked edit region with the same neutral value in both images. The keep-region mask is eroded by the SSIM window radius before averaging the SSIM map; exact erosion and invalid-mask handling follow the evaluator configuration used for the reported table.

Edit-region perceptual distance. The reported main-table protocol computes mLPIPS on the editable region. Both images are composited against the same neutral background outside M_e, then LPIPS is computed on the composite pair:

$$\mathrm{mLPIPS} = \mathrm{LPIPS}(\mathrm{comp}(I_s, M_e), \mathrm{comp}(I_o, M_e)) .$$

The evaluator records the mask region used for each metric so that keep-region and edit-region perceptual scores are not mixed.

Full-image structural score. HaarPSI is computed on the full normalized source/output pair and is not restricted to M_e or M_k. The main comparison reports HaarPSI, keep-region mPSNR/mSSIM/mMSE, and edit-region mLPIPS.

C.5 Aggregation, validity, and audit

Per-sample JSON/TSV records store case index, method, edit type, mask scope, mask status, coverage, metric-set name, normalization resolution, and metric values. Summary tables macro-average over edit types so categories with more examples do not dominate the reported means. Global edits contribute to full-image and no-reference metrics but not to local masked-preservation averages. Detector edit-success flags for additions, deletions, and replacements are auxiliary detector proxies, not human-verified semantic accuracy.

The mask audit retains source queries, detected boxes, confidence scores, labels, mask coverage, and a visual sheet containing the source, the shared binary mask, and method outputs with that mask overlaid. Suspicious part-level masks require a logged manual override or exclusion reason. OCR exact-match and CER are not part of the implemented metric run and are therefore not reported as available metrics.

D Low–High Topology and Observed Ablation

D.1 Information-flow statement

The notation $A \to B$ in this section describes information flow, not the direction of the query–key matrix. The only direct cross-resolution guidance retained by the observed ablation is

$$L_N \longrightarrow H_N.$$

Its implementation is a high-resolution noisy-target query reading low-resolution noisy-target keys and values:

$$\mathrm{Attn}_{L_N \to H_N}\left(Q_{H_N}, K_{L_N}, V_{L_N}\right).$$

Thus the low-resolution editing state informs high-resolution target generation while preserving the correct query–key direction.

D.2 What is retained

The low-resolution branch keeps its native global editing interaction over text, L_N, and L_R. High-resolution noisy-target queries use a spatially aligned neighborhood of L_N keys, where the parent cell of a high-resolution query is

$$\pi(y, x) = \left(\lfloor y/n \rfloor, \lfloor x/n \rfloor\right).$$

The support is bounded by a Chebyshev radius R, giving at most

$$K_L = (2R+1)^2$$

low-resolution target keys before image-boundary clipping. High-resolution target/reference tokens interact locally through the Halo support on the high-resolution grid. These are local same-resolution source/detail interactions, not additional cross-resolution reference bridges.

D.3 What is not retained

Under the declared `no_low_ref` gates, high-resolution queries do not read low-reference keys. The retained H_N-query support still contains the selected local L_N neighborhood. Other direct high-to-low reference paths are disabled only when the corresponding resolved topology gate is off; this statement is therefore recipe-dependent. The low-resolution reference remains available to the native low-resolution editor, and the high-resolution reference remains available through local high-resolution support.

D.4 Interpretation of the ablation

The final topology retains $L_N \to H_N$ as the direct cross-resolution guidance path and removes low-reference bridge edges under `no_low_ref`. This is a tested topology choice; a general claim that every alternative reference-fusion design is ineffective requires a separately matched ablation.

E Sparse Kernel Implementation

E.1 Sequence layout and metadata

The canonical packed sequence is

$$[\mathbf{T}_{\text{text}}, \mathbf{T}_{L_N}, \mathbf{T}_{L_R}, \mathbf{T}_{H_N}, \mathbf{T}_{H_R}].$$

Before the high-resolution kernel is invoked, the high-resolution portion is reordered by parent cell. The metadata carried through the permutation contains segment identity, parent-cell coordinates, high latent-token coordinates, and inverse-permutation indices. This lets the kernel construct exact row supports without materializing a dense $(S \times S)$ mask.

E.2 PRODUCTION CSR PATH

As an example of the checked 2K inference path, the logs use `low_flash_sdpa+hi_csr_triton`. Low-resolution and text rows use PyTorch Flash SDPA. High-resolution rows use a Triton CSR kernel with paired high-resolution target/reference rows. The logged main configuration is

$$R = 3, \qquad h = 6 \text{ (2K)}, \qquad \texttt{variant=pair}, \qquad \texttt{block_n=64},$$

with backend-specific launch parameters stored in the run log. The row descriptor records valid key indices and offsets; it does not materialize masked scores for invalid keys. At image boundaries, row lengths are clipped and the kernel uses the valid prefix of the CSR row. CSR/Triton is an inference backend; the training reference is the exact Flex connectivity plan.

The high-resolution row support is the union of local high-resolution target/reference keys, the aligned L_N neighborhood for H_N queries, and the permitted text keys. The row is normalized as one attention distribution over this union. Consequently, the implementation preserves the semantic coupling between high-resolution detail, low-resolution edit guidance, and text conditioning.

E.3 CORRECTNESS AND RUNTIME SAFEGUARDS

The implementation keeps a dense/Flex reference path for connectivity checks. A new permutation or CSR builder is accepted only when the densified support agrees with the intended topology and the inverse permutation restores token order. Boundary cases are tested separately from interior parent cells because their valid key counts differ. The production logs also record one warm-up forward, disabled KV caching, model loading time, generation time, save time, and peak allocated/reserved memory so that the timing boundary can be reconstructed.

F COMPLEXITY DERIVATION

F.1 NOTATION

Let $P_L = H_L W_L$ and $P_H = H_H W_H = n^2 P_L$ be the low- and high-resolution image-token counts per image stream. Let T be the text token count, d the head dimension, R the low-resolution support radius, and h the high-resolution Halo width. The high-resolution local support before boundary clipping contains

$$K_{\text{cell}} = (n + 2h)^2$$

tokens per image stream. The aligned low-resolution target support contains

$$K_{\text{low}} = (2R + 1)^2$$

keys.

F.2 DENSE ATTENTION

If all text, low-resolution, and high-resolution target/reference tokens participate in global attention, the joint sequence length is

$$S_{\text{dense}} = T + 2P_L + 2P_H.$$

The score and value-mixing work of one dense attention layer is

$$\mathcal{C}_{\text{dense}} = \Theta\big(S_{\text{dense}}^2 d\big), \qquad \mathcal{M}_{\text{dense}} = \Theta\big(S_{\text{dense}}^2\big)$$

if the score matrix is materialized. Flash-style kernels reduce score storage, but the dense pairwise arithmetic remains quadratic in S_{dense}.

F.3 SPARSE CROSS-RESOLUTION ATTENTION

The low-resolution/text part retains native global interaction:

$$\mathcal{C}_L = \Theta\big((T + 2P_L)^2 d\big).$$

For high-resolution query rows, each query sees a finite support and, for H_N rows, an additional low-resolution target neighborhood. Writing Q_H for the number of high-resolution query rows gives the conservative upper bound

$$\mathcal{C}_H = O\big(Q_H \left[T + 2(n+2h)^2 + (2R+1)^2\right] d\big) .$$

For the common paired target/reference layout Q_H is proportional to $2P_H$. Therefore,

$$\mathcal{C}_{\text{sparse}} = O\big((T+2P_L)^2 d \;+\; Q_H \left[T + 2(n+2h)^2 + (2R+1)^2\right] d\big) .$$

The corresponding sparse score storage is

$$\mathcal{M}_{\text{sparse}} = O\big((T+2P_L)^2 \;+\; Q_H \left[T + 2(n+2h)^2 + (2R+1)^2\right]\big)$$

for an implementation that stores low/text scores densely and high rows as CSR. A fused kernel can reduce the constant factor and avoid materializing the full score tensor.

F.4 Scaling with resolution

With $P_H = n^2 P_L$, dense high-resolution visual interactions grow as $\Theta(n^4 P_L^2 d)$. The sparse high-resolution term retains the explicit support factor

$$O\big(n^2 P_L \left[(n+2h)^2 + (2R+1)^2 + T\right] d\big) .$$

For each fixed 2K/4K run, this is a finite-support edge count. Under the synchronized rule $h = Rn$, the support itself scales with n; therefore the sparse term should not be interpreted as resolution-independent linear scaling. The parent-cell permutation does not change the edge count; it improves memory locality and exposes regular CSR rows.

F.5 End-to-end denoising cost

For direct inference with N denoising steps,

$$\mathcal{C}_{\text{direct}} = \Theta(N\,[\mathcal{C}_{\text{sparse}} \;+\; \mathcal{C}_{\text{FFN}}]) \;+\; \mathcal{C}_{\text{VAE}}.$$

For progressive inference with transition K, only $N-K$ steps use the high-resolution sparse branch:

$$\mathcal{C}_{\text{progressive}} = \Theta(K\,\mathcal{C}_{L\text{-only}} \;+\; (N-K)\,[\mathcal{C}_{\text{sparse}} \;+\; \mathcal{C}_{\text{FFN}}]) + \mathcal{C}_{\text{decode/resize/encode}}.$$

The final term is a single transition cost, not a repeated external super-resolution stage.

G Limitations and Future Work

G.1 Current limitations

The evaluation masks are automatic estimates because VINS-4KEval does not provide pixel-level edit annotations. GroundingDINO can select a complete object for a part-level request, and source-only addition regions are necessarily conservative when the instruction lacks a concrete spatial anchor. Mask validity is audited, but the mask is not a substitute for human semantic judgment.

The current topology assumes that a 1K editor is a reliable semantic anchor and that high-resolution detail can be generated from local target/reference support plus bounded $L_N \rightarrow H_N$ guidance. Highly nonlocal edits, large geometric rearrangements, and instructions whose meaning depends on fine target-scale text can violate this division of labor. The sparse kernel also has boundary-dependent row lengths and can lose efficiency if the support distribution produces poorly packed CSR buckets.

The progressive path is training-free but not training-aware. Its default latent-bilinear transition changes the state distribution seen by the high-resolution branch relative to the native joint trajectory. The current model does not explicitly train on these transition states; RGB/VAE lifting is evaluated separately as a transition ablation.

G.2 FUTURE WORK: TRAINING-AWARE PROGRESSIVE INFERENCE

A direct extension is to expose the progressive transition during training. At a sampled transition K, the low-resolution prefix would produce a clean estimate, the selected latent or RGB/VAE lift would initialize the high-resolution state, and the high-resolution branch would be trained on the resulting post-transition state distribution. A mixture of direct joint trajectories and transition trajectories could preserve direct-path quality while reducing the train–test gap for progressive inference. The training objective could also weight transition positions by the number of remaining high-resolution steps, allowing the model to learn where high-resolution detail is most sensitive to the lift.

Additional directions include learned but source-only mask uncertainty for evaluation, resolution-adaptive CSR bucketing, a fused neighborhood kernel that preserves the single-softmax union of high, low, and text keys, and explicit human evaluation of part-level edits and small text.

H Additional Visualizations and Reserved Panels

H.1 2K Resolution



Figure 8: More comparison results under 2K resolution.

H.2 4K Resolution



Figure 9: More comparison results under 4K resolution.