AReA: Adaptive Receptive-Field Attention for Efficient High-Resolution Diffusion Transformers

Anonymous authors
Paper under double-blind review



Figure 1: **High-Resolution Generation.** AReA enables training-free and efficient high resolution generation across FLUX.1, Qwen-Image, FLUX.2, and Ideogram 4. Best viewed zoomed-in.

Abstract

Generating images beyond the training resolution of Diffusion Transformers (DiTs) remains challenging, as the number of tokens grows rapidly with resolution, full attention becomes expensive, and outputs often suffer from structural distortions and degraded detail. We identify uniform full attention as the common source of both problems. Attending to all tokens at every step causes quadratic cost and exposes the model to more tokens and longer relative distances than seen during training, pushing it out of distribution. We observe that attention's effective receptive field narrows during denoising, from global in early steps that form structure to local in later steps that refine details. We propose **AReA**, a training-free attention mechanism whose receptive field evolves from coarse to fine, following the denoising progression. In early steps, AReA applies cross-scale coarse attention over downsampled keys and values, capturing global context, while in later steps it applies local attention within progressively shrinking windows. Without additional parameters or training, experiments show that AReA consistently outperforms state-of-the-art training-free baselines across target resolutions and different models, while also being faster and more efficient.

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



Figure 2: **Failure Modes at High Resolution**. FLUX.1 at 4096 × 4096. Full attention keeps the model's intended subject but collapses the layout, while RoPE-based methods keep a layout but fragment, lose, or distort the subject. AReA preserves both, and renders fine details.

1 Introduction

Text-to-image (T2I) generation has witnessed remarkable progress driven by diffusion models (Ho et al., 2020; Dhariwal & Nichol, 2021), with Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Bao et al., 2023) pushing generation quality even further through highly scalable architectures. Despite these advancements, generating high-resolution images beyond a model's training resolution remains a significant challenge, often leading to noticeable degradation or even complete failure. While training at the target resolution seems to be the obvious remedy, it typically demands massive amounts of high-quality data and substantial computational resources. As a result, a growing body of research has shifted toward training-free approaches, seeking to harness the high-resolution capabilities already embedded in pre-trained diffusion models (Rajabi et al., 2026; Koh et al., 2026).

Existing training-free approaches largely follow two lines. The first adopts a two-scale paradigm, generating a low-resolution image and using it to guide high-resolution synthesis (Bu et al., 2026; Du et al., 2024a;b; Kim et al., 2025). While this preserves global structure, the output is limited by the low-resolution guidance rather than by the model's capacity. The second generates at the target resolution in a single end-to-end pass by modifying model components such as positional encodings or the denoising process itself (Zhao et al., 2025; Issachar et al., 2026; Rajabi et al., 2026). Although this direction seems promising, existing methods are often computationally expensive or tied to architecture-specific components such as Rotary Position Embeddings (RoPE) (Su et al., 2024), limiting their generalizability. Both approaches, however, often produce high-resolution images that remain low in quality and contain structural distortions (Figure 2). Therefore, efficient generation of high-quality, structurally coherent images across diverse high resolutions remains an open problem.

We argue that both the inefficiency and the artifacts share the same root in how tokens interact in attention. At high resolution, attention is still applied uniformly, so every token attends to every other token at every denoising step, regardless of what that step is actually resolving. No consideration is given to how far a token really needs to see; each is simply given a global receptive field, and this unrestricted freedom comes at a price. Token interactions grow quadratically with the number of tokens, making generation increasingly slow at high resolution, and they reach across distances beyond the training range, which pushes the model out of distribution. Uniform attention also ignores the coarse-to-fine nature of denoising (Xiao et al., 2026), in which early steps establish global structure and later steps refine local details. Thus, *the receptive field that attention needs varies across denoising steps* (Figure 3). Uniform global attention at every step is therefore unnecessary, and beyond the training range it is actively harmful. The model already has the capacity; what is needed is to align the attention receptive field with this coarse-to-fine progression.

Motivated by this insight, we introduce **AReA** (**A**daptive **Re**ceptive-Field **A**ttention), a training-free attention mechanism with a *coarse-to-fine receptive field* that evolves across denoising steps. As shown in Figure 4, AReA operates in two phases, cross-scale coarse attention in early steps and adaptive windowed fine attention in later steps. In the first phase, AReA attends to pooled keys and values placed on a coarse positional grid matched to the model's native training resolution, so every token still sees the whole image, but through a downsampled view that captures global context at low cost. In the second phase, AReA applies attention within local windows that progressively shrink as fine details emerge, so each token interacts mainly with its nearby tokens. Rather than tiling uniformly, AReA uses a Discrete Wavelet Transform (DWT) map of the current state to split windows over smooth regions into smaller ones, further reducing the attention cost. In this way,

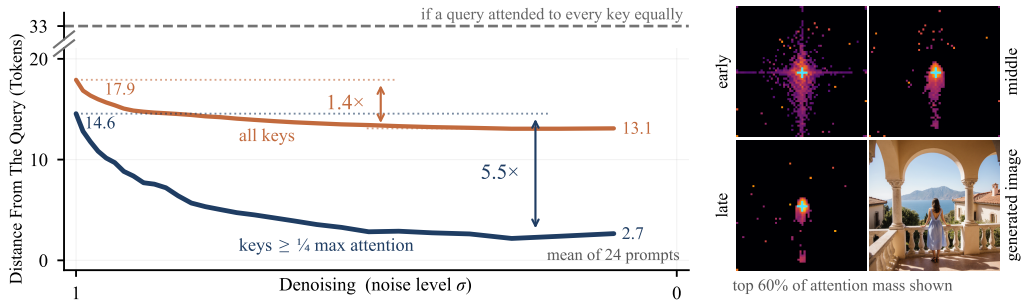

Figure 3: **Attention Locality as Denoising Progresses.** Vanilla FLUX.1 (1024×1024), averaged across heads, blocks, and 24 prompts. *Left*: attention-weighted query–key distance. Over all keys it shrinks only $1.4\times$, but over dominant keys ($\geq 1/4$ max attention) it shrinks $5.5\times$: late in denoising, the most important keys lie close to the query. *Right*: attention maps of a center query at early, middle, and late steps (top 60% of mass).

AReA avoids the quadratic cost of full-resolution global attention and suppresses long-range interactions when they are unnecessary, keeping the model close to its training regime (see Appendix G). AReA therefore needs no RoPE extrapolation and inherits none of its limitations, and it changes nothing about the model beyond what each query attends to.

Extensive experiments show that AReA consistently reduces generation time, with speedups that grow with resolution, reaching $3.2\times$ over full-resolution attention at 6144^2. Moreover, it improves structural coherence and fine-detail fidelity, outperforming baselines across resolution settings and generating rich, high-quality images up to 67 million pixels, as shown in Figure 1. AReA requires no retraining and no extra parameters, and integrates directly into existing DiT models.

2 Related Work

Training-free High-Resolution Image Synthesis Preserving both global structure and fine-grained detail remains an open challenge in high-resolution generation. To address this, recent work (Bar-Tal et al., 2023; Huang et al., 2024; Hwang et al., 2024; Jin et al., 2023) proposes training-free strategies for extending diffusion models beyond their training resolution. Early approaches explored cascaded generation (Ho et al., 2022), progressive upsampling (Gu et al., 2024), domain-specific pipelines (Ren et al., 2024), and latent-space super-resolution (Jeong et al., 2025). Recent methods have largely adopted multi-stage guidance pipelines, where generation is first performed at the model's native resolution and the resulting image or trajectory is then used to guide sampling at higher resolutions. DiffuseHigh (Kim et al., 2025) upsamples the base-resolution image and re-denoises it, using the Discrete Wavelet Transform to preserve its structure. I-Max (Du et al., 2024b) uses projected flows from the base-resolution trajectory to stabilize the formation of coarse structure. Similarly, HiFlow (Bu et al., 2026) constructs a virtual reference flow from base-resolution information to guide high-resolution sampling. ScaleDiff (Koh et al., 2026) follows a similar paradigm, refining the upsampled base-resolution image with SDEdit (Meng et al., 2021) while combining patch-level attention, latent frequency mixing, and structure guidance to balance global structure and fine detail. While effective, these methods inherently rely on base-resolution guidance and introduce additional complexity into the high-resolution denoising process.

Direct High-Resolution Inference More recent work aims to unlock the high-resolution capabilities already present in pretrained models, generating images directly at the target resolution by adjusting components such as attention and positional encodings at inference time. For example, ResDiT (Ma et al., 2026) applies global attention with scaled positional embeddings in parallel with patch-wise attention, then fuses their outputs through Gaussian-weighted blending and spectral fusion. Another line of work approaches the problem through the lens of RoPE length extrapolation, a problem that closely mirrors context extension in large language models (LLMs) (Peng et al., 2024; Ding et al., 2024). These methods typically formulate resolution extrapolation as a recalibration of RoPE's rotary frequencies so that positions beyond the training range remain within the distribution seen during training. DyPE (Issachar et al., 2026) introduces time-aware positional adjustments that evolve across diffusion timesteps. UltraImage (Zhao et al., 2025) mitigates repetitive artifacts by shifting the dominant frequency to align with the training resolution and employing entropy-guided

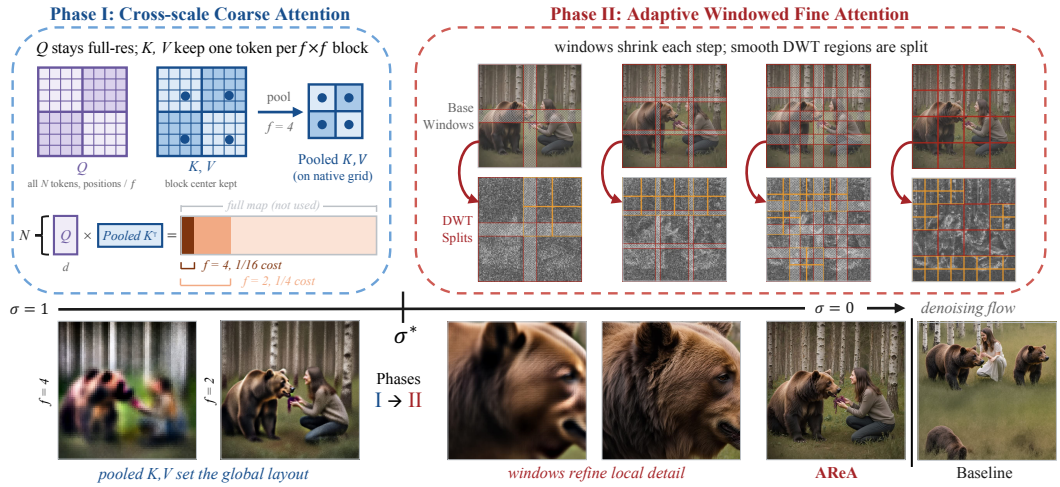

Figure 4: **Pipeline of AReA.** Denoising moves from coarse to fine across $\sigma^\star$. In Phase I, full-resolution queries attend to keys and values pooled to one token per $f \times f$ block on a native-resolution grid with a ramped f. In Phase II, attention runs in shrinking local windows, and windows over smooth regions of the Discrete Wavelet Transform (DWT) map are split to cut cost.

attention. SigMa (Zhao et al., 2026) uses a scale-adaptive sigmoid schedule to adjust positional modulation according to the target resolution. SEGA (Rajabi et al., 2026) analyzes the spectral energy of intermediate latents to dynamically adjust attention scaling, amplifying high-energy frequency bands while suppressing low-energy ones to preserve fine details without compromising global structure. While effective, these methods are tied to RoPE and inherit the limits of length extrapolation, restricting their generalizability and often causing artifacts under large resolution shifts. In contrast, AReA requires neither base-resolution guidance nor RoPE extrapolation. It adapts the attention receptive field across denoising steps, shifting from global to local attention as generation moves from coarse structure to fine detail. This keeps the model close to its training regime, remaining stable far beyond the training resolution while reducing attention cost.

3 Method

Adaptive Receptive-Field Attention (AReA) is a training-free method that schedules the attention receptive field by transitioning from a coarse, global view to fine, local windows, matching the level of detail emerging at each denoising step. Our approach is grounded in the observation that attention becomes increasingly local as denoising progresses (Figure 3). It is therefore natural to let the attention's receptive field adapt accordingly rather than remain fixed and uniform across all denoising steps. AReA achieves this in two phases (Figure 4). Cross-scale coarse attention first lets full-resolution queries attend to a downsampled view of the image through pooled keys and values, establishing the global layout at low cost, and adaptive windowed fine attention then refines local detail within shrinking windows. As a result, AReA never computes full attention at the target resolution. An important design choice is when to switch between the two phases. Switching too early leaves the composition unresolved, causing inconsistent structure across windows, while switching too late leaves little room for detail refinement and reduces the speedup. Since denoising progress varies across models and samplers, we define the switch by a noise level $\sigma^\star$ rather than a step count, chosen empirically as the point where the composition is established (Appendix A.3). Phase I thus covers the first S_c of the S steps, up to $\sigma^\star$, and Phase II the remaining $S_d = S - S_c$.

3.1 Preliminaries: Attention and Receptive Fields in DiTs

In standard DiTs, an image latent is divided into patches and flattened into a sequence of N tokens, $X \in \mathbb{R}^{N \times D}$. The self-attention layer projects these tokens into Queries (Q), Keys (K), and Values (V). RoPE encodes spatial information by rotating Q and K based on their 2D grid coordinates, enabling attention to capture relative distances. The attention output is then computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V, \tag{1}$$

where d is the per-head dimension. Every query attends to every key, giving each token a global receptive field at $O(N^2)$ cost. While this global reach is needed to establish the layout, it comes at a quadratic cost, much of which may be unnecessary for large N. Attention, however, is flexible in what each query attends to. Instead of computing the full $N \times N$ matrix, we can adapt a query's receptive field by varying its spatial coverage, resolution, or subset of keys and values it attends to. Decoupling queries from keys and values across scales or restricted regions reduces computation while preserving essential interactions. This principle of adaptive receptive fields forms the foundation of AReA's two-phase schedule.

3.2 PHASE I: CROSS-SCALE COARSE ATTENTION

The early denoising steps decide the global layout of the image, so each token needs a receptive field that covers the whole grid. Providing it at full target resolution, however, is problematic, as it incurs the quadratic cost we seek to avoid and exposes the model to unseen token counts and relative-position ranges. AReA addresses this via *cross-scale coarse attention*, where queries remain at the full target resolution while keys and values are spatially pooled by a factor f to form a downsampled global receptive field. This preserves positional geometry close to the model's native training resolution, enabling global attention through the downsampled representation at $1/f^2$ of the cost, which is sufficient for the low-frequency structure and overall composition formed in early diffusion steps. Since the query side is never reduced, the output remains at full resolution and requires no upsampling. Specifically, each full-resolution query attends to the N/f^2 pooled keys and values and produces one output token, preserving the original N-token resolution. The phase starts maximally coarse and progressively refines as the image's own frequency content grows; we describe the pooling strategy and refinement process next.

To construct the downsampled key and value grids, AReA partitions the spatial grid into $f \times f$ blocks and selects one representative token from each block, using the block center by default (see Appendix A.2 for alternatives). Pooling alone reduces the number of keys but not their distances, since the pooled keys would keep their original full-resolution coordinates. AReA therefore places the pooled keys on a coarse $(W/f) \times (H/f)$ grid and scales the full-resolution query coordinates by $1/f$ to match, so that both are measured in the same units. As a result, each $f \times f$ block of queries shares the position of its pooled key, keeping the relative distances encoded by RoPE close to the model's training range. It is worth noting that text tokens are never pooled, so each of the N queries attends to all $T + N/f^2$ keys.

Moreover, the choice of f involves a trade-off. A large f keeps the coarse grid at the model's native range but is too coarse to resolve details such as object boundaries, while a small f gives a finer grid beyond the native range, causing the model to duplicate subjects (Figure 8). We therefore introduce a *pooling ramp*, which decreases f linearly over Phase I following

$$\tilde{f}_t = f_0 + \frac{t-1}{S_c - 1}\left(f_{\min} - f_0\right), \qquad t = 1, \ldots, S_c, \tag{2}$$

decaying the factor from an initial f_0 down to a floor of $f_{\min} = 2$, with each value snapped to a valid grid divisor (Appendix A.1). As f decreases, the coarse grid expands, and this gradual increase in resolution smoothly bridges the gap between the initial global composition and the fine details targeted in the next phase. An overview of this process is shown in Figure 4.

3.3 PHASE II: ADAPTIVE WINDOWED FINE ATTENTION

Once the global layout is established, the remaining steps focus on refining fine details, which Phase I cannot provide, since pooling removes the local information needed at this stage. Fine details are inherently local, so at this stage, each token primarily depends on its surrounding neighborhood, making local-window attention sufficient. Windows are also where the speedups come from. Since attention scales quadratically with the number of tokens, processing multiple small windows is much cheaper than computing over the full grid. Moreover, as the window size decreases, the cost per window drops quadratically, while the number of windows grows only linearly, reducing the overall computational cost. However, windows are not always cheaper. A few large, overlapping windows, or simply too many windows, can cost more than a single full-attention step, so both the size and number of windows must be chosen carefully. The design goal is therefore to use the smallest windows that together remain cheaper than a single full-attention pass while

providing enough local context. However, starting with small windows too early can fragment the image. Although Phase I sets the global structure, neighboring regions must stay aligned during the detail phase. AReA begins with large windows for coherence, then gradually shrinks them as details emerge, keeping within-window relative positions near the model's training range.

Concretely, the window size w_t changes at every step. It starts at w_{start}, the largest window whose step is still cheaper than a full-attention pass, which is about 70% of the longer side of the patch grid, and shrinks each step until it reaches the floor $w_{\min}$, which is the model's native range,

$$w_t \;=\; w_{\text{start}} \left(\frac{w_{\min}}{w_{\text{start}}} \right)^{\frac{t-1}{S_d - 1}}, \qquad t = 1, \ldots, S_d, \tag{3}$$

where S_d is the number of steps in this phase. Both w_{start} and subsequent window sizes are checked against a cost budget on image-token attention, so that no windowed step exceeds a full-attention pass in that cost (see Appendix B). At step t, the $H \times W$ grid is covered by a base tiling of $n_H \times n_W$ windows, each of size $w_t \times w_t$, where $n_H = \lceil H/w_t \rceil$ and $n_W = \lceil W/w_t \rceil$ are the minimum numbers of windows needed to cover each axis. The windows are spaced evenly so that the overlap is distributed equally between neighboring windows. AReA then adapts this base tiling to the image content. A DWT map, computed from the current hidden states, measures the amount of fine detail in each region (see Appendix C). Windows over smooth, low-detail regions are split into four quadrants (visualized in Figure 4), where splitting is safe because a larger window over smooth content adds little information. In contrast, a window over a detailed region, such as a face, is left at its scheduled size, since splitting it would cut the structure it needs to see. This splitting saves computation, and we make sure that the adapted tiling does not cost more than the base tiling (see Appendix C.3).

When the window size does not evenly divide the grid, adjacent windows overlap. Outputs for shared tokens are blended using center-weighted averaging, which favors windows where the token lies closer to the center. Despite this blending, windowed attention remains prone to visible seams because boundary tokens lack context from the other side. AReA addresses this in two ways. First, window boundaries are randomly shifted across steps, preventing seams from forming at fixed locations. Second, each window's keys and values are extended with a narrow ring of neighboring tokens ($\delta_{\max}$), which we call the *Halo*, giving edge tokens access to context across the boundary (see Appendix D for details). Since the queries and outputs remain unchanged, the additional cost is much smaller than that of enlarging the window itself. Even with the coherent global structure established in Phase I, windowed attention can still duplicate small objects or local details, because each window cannot see beyond its boundaries and may reproduce content already present elsewhere in the image. AReA addresses this with a *global-KV bridge*. At every Phase II step, each window's KV set is extended with the keys and values of a small set of tokens from the rest of the grid. Tokens are selected based on their deviation from the global mean key, prioritizing distinctive content that should be shared across windows, with the selection spatially distributed across the grid to ensure broad spatial coverage. The number of bridge tokens scales with the window as $R = (\rho w_t)^2$, maintaining a fixed fraction of each window's keys (ρ) throughout the phase (see Appendix E).

Putting everything together, in Phase II the queries of each window of side w_t (or $w_t/2$ if split) attend to its own local tokens, the T text tokens, and E additional tokens from the Halo and the global-KV bridge. All tokens keep their original positional coordinates, and RoPE is applied without modification. This phase therefore provides fine-grained local attention with enough cross-window context to reduce seams and duplication.

4 Experiments

Setup. We evaluate AReA on FLUX.1 (Labs, Black Forest, 2024) and Qwen-Image (Wu et al., 2025). Since these models differ in their noise schedules and default numbers of denoising steps, we select the switching threshold $\sigma^\star$ separately for each model. Based on an empirical analysis (Appendix A.3), we set $\sigma^\star$ such that the first 8 steps for FLUX.1 and the first 20 steps for Qwen-Image use cross-scale coarse attention, while the remaining steps use adaptive windowed fine attention. Since AReA is a direct-inference method that generates images end-to-end, we compare it with other direct-inference methods, including DyPE (Issachar et al., 2026), UltraImage (Zhao et al., 2025), SigMa (Zhao et al., 2026), and SEGA (Rajabi et al., 2026). Comparisons with multi-stage methods on FLUX.1 are provided in Appendix F. All baselines are evaluated using their official implementations. Across all experiments, we set $\delta_{\max}$ to 8 for Halo and ρ to 0.5 for global-KV bridge. All experiments are conducted on NVIDIA H100 GPUs.

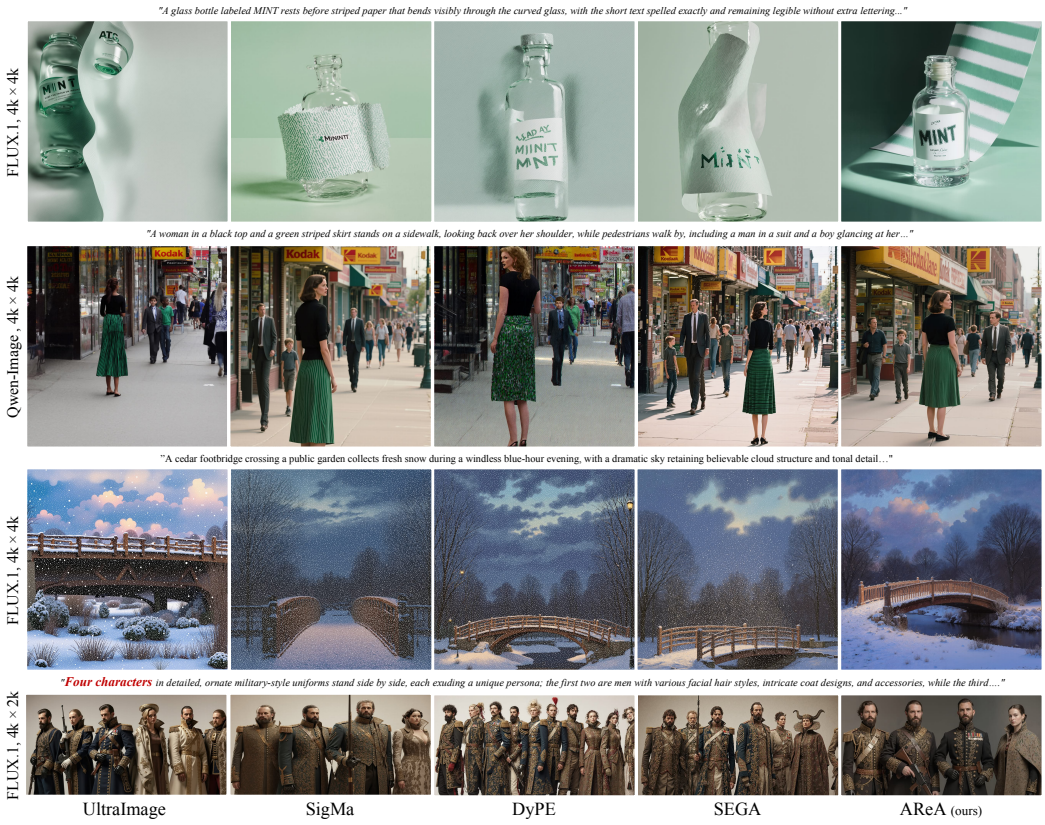

Figure 5: **Qualitative comparisons across resolutions.** AReA eliminates structural distortions and subject duplication seen in baseline methods, achieving superior fidelity and prompt alignment.

Evaluation. We evaluate on Aesthetic-4K (Zhang et al., 2025), which provides 195 prompts paired with high-resolution reference images. Inspired by SEGA (Rajabi et al., 2026) and JoLT (Koroglu et al., 2026), we further introduce **HRGen-1K**, a benchmark of 1,000 challenging and diverse LLM-generated prompts designed to rigorously evaluate high-resolution generation across complex scenes and compositions. Appendix H describes its construction and statistics. We conduct quantitative experiments at four high-resolution settings with varying aspect ratios: 6144×2048, 2048×4096, 4096×4096, and 5120×5120. We evaluate image fidelity using FID (Heusel et al., 2017), its patch-based variant FID_p computed on local crops, and Inception Score (IS). Perceptual quality is assessed using Aesthetic Score (Schuhmann, 2022) and the reference-free metrics MUSIQ (Ke et al., 2021), CLIP-IQA (Wang et al., 2023), and ARNIQA (Agnolucci et al., 2024). Semantic alignment is measured by CLIP Score (Radford et al., 2021), while joint alignment and human-preferred visual quality are assessed using ImageReward (IR) (Xu et al., 2023), PickScore (PS) (Kirstain et al., 2023), and HPSv2 (Wu et al., 2023). We also report latency as the average generation time per image in seconds.

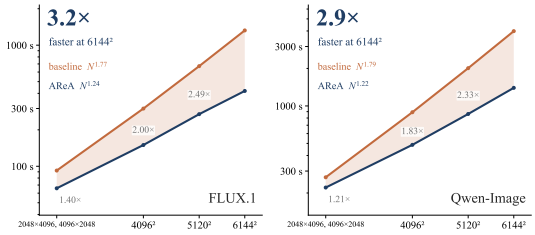

Figure 6: **Generation latency vs. resolution.** AReA is significantly faster than full-attention baselines across all resolutions, reaching $3.2\times$ and $2.9\times$ acceleration at 6144^2 on FLUX.1 and Qwen-Image, respectively.

4.1 State-of-the-art Comparison

Qualitative comparison. As shown in Figure 5, existing direct-inference methods often distort the global structure of subjects at high resolutions, frequently producing stretched objects and bod-

Table 1: Comparison of AReA against state-of-the-art baselines on FLUX.1 and Qwen-Image across different resolutions and aspect ratios on Aesthetic-4K. **Bold**: best; underline: second-best.

		FLUX.1							Qwen-Image						
Res.	**Method**	Latency(s)↓	ImageReward↑	Aesthetic↑	CLIP↑	PickScore↑	FID↓	FID_p ↓	Latency(s)↓	ImageReward↑	Aesthetic↑	CLIP↑	PickScore↑	FID↓	FID_p ↓
2048×4096	Base	97	-0.8696	5.91	26.49	20.41	171.4	67.4	274	-0.5186	5.96	27.25	20.83	167.6	**57.8**
	UltraImage	89	-0.1474	6.25	28.11	21.28	173.3	79.0	301	0.4448	5.79	28.19	21.07	161.7	62.6
	DyPE	97	0.5927	6.30	28.43	22.15	156.5	79.9	274	0.5769	6.12	28.71	21.64	159.0	74.7
	SigMa	103	0.6727	6.14	28.07	22.01	152.5	63.9	275	0.6155	6.22	28.87	22.03	152.5	63.0
	SEGA	98	0.7863	6.28	28.95	22.47	153.9	76.1	276	1.2467	6.42	29.89	23.10	146.1	69.6
	AReA	**66**	**1.0881**	**6.34**	**29.23**	**22.95**	**151.9**	**62.5**	**221**	**1.3260**	**6.44**	**30.07**	**23.26**	**144.5**	68.5
5144×2048	Base	185	-0.3800	5.82	25.15	20.73	173.5	83.6	544	0.1886	5.86	26.92	21.10	175.3	68.2
	UltraImage	183	0.4103	6.11	27.55	21.81	171.4	74.0	552	0.3310	5.71	27.59	20.80	173.4	63.1
	DyPE	185	0.8060	6.24	28.08	22.34	173.3	97.4	544	0.1442	5.60	26.60	20.47	181.7	100.6
	SigMa	191	0.4321	5.88	26.69	21.43	181.3	86.1	545	0.6295	6.04	28.43	21.75	161.6	65.1
	SEGA	188	0.5357	6.12	27.64	21.88	168.2	94.2	548	**1.3047**	6.33	**29.28**	**23.22**	160.8	73.4
	AReA	**107**	**1.0450**	**6.26**	**29.02**	**22.74**	**161.0**	**62.0**	**355**	1.2752	**6.34**	29.16	22.69	**159.6**	**62.2**
4096×4096	Base	314	-0.7177	5.80	25.72	20.34	179.8	112.0	902	-0.1313	5.95	27.20	20.84	167.9	77.2
	UltraImage	256	-0.1379	6.11	26.77	20.85	169.4	82.0	983	0.6911	5.84	28.64	21.43	155.3	65.1
	DyPE	314	1.1070	6.32	28.98	22.79	155.2	95.4	902	0.7669	6.05	28.70	21.62	165.1	82.1
	SigMa	322	1.0149	6.16	28.36	22.47	156.8	83.3	906	1.2614	6.26	29.40	22.66	152.2	60.8
	SEGA	319	1.2702	6.31	29.16	23.16	150.6	71.8	906	1.4548	6.47	**29.83**	**23.65**	147.4	65.6
	AReA	**151**	**1.3250**	**6.37**	**29.71**	**23.31**	**148.5**	**58.2**	**487**	**1.4847**	**6.49**	29.62	23.53	**147.3**	**55.3**
5120×5120	Base	685	-1.8620	5.15	19.32	18.68	252.7	212.5	2014	-0.6020	5.78	26.64	20.24	186.4	128.4
	UltraImage	546	-1.2421	5.60	22.50	19.47	206.1	165.8	2057	0.2061	5.64	27.13	20.73	181.5	98.3
	DyPE	685	0.8389	6.14	27.79	22.01	163.4	148.0	2014	-0.3734	5.38	25.06	19.86	202.8	178.9
	SigMa	704	0.5278	5.88	26.01	21.43	178.5	225.2	2019	0.5555	5.90	27.83	21.38	172.5	100.5
	SEGA	693	1.1143	6.25	29.16	22.86	155.7	85.0	2020	1.4192	6.42	**29.90**	**23.72**	149.2	63.4
	AReA	**269**	**1.2319**	**6.29**	**29.34**	**22.94**	**152.1**	**63.3**	**862**	**1.4540**	**6.50**	29.80	23.27	**146.7**	**55.8**



Figure 7: **Qualitative ablation on FLUX.1 at 4096 × 4096.** Without Phase I, windows diverge and disrupt global layout. Without Phase II, the image lacks fine texture and remains blurry. Mean pooling degrades high-frequency detail (compared in Table 4), while fixed windows without shrinking cause seams and content fragmentation compared to default AReA.

ies as well as struggling in generating texts. We attribute this to their RoPE modifications, which shift the model away from its training regime and disrupt spatial coherence across the image. In contrast, AReA preserves the intended global structure while generating fine details, without altering positional encodings. As a result, it also achieves stronger prompt alignment, particularly in composition and object counting. Further comparisons are in Appendix J.

Quantitative comparison. As reported in Tables 1 and 2, AReA outperforms direct-inference baselines, including SEGA, DyPE, SigMa, and UltraImage, on most metrics for both FLUX.1 and Qwen-Image. Across all resolutions, including non-square aspect ratios and up to 5120 × 5120, it consistently ranks first or second in both semantic alignment and perceptual quality. While most competing methods degrade noticeably as the number of tokens grows, AReA maintains high fidelity and structural coherence. These gains also come with improved efficiency. As shown in Figure 6 and Table 1, AReA achieves lower latency than all baselines at every resolution, with the advantage growing at larger resolutions where attention cost increasingly dominates.

Table 2: Comparison on FLUX.1 at 4096 × 4096 on HRGen-1K. **Bold**: best; underline: second-best.

Method	IR↑	Aes.↑	CLIP↑	Pick↑	HPS↑	IS↑
Base	-1.4649	5.12	23.36	19.04	0.232	7.20
UltraImage	-0.0277	5.77	28.26	20.46	0.269	10.17
DyPE	0.3280	5.83	29.35	20.94	0.266	10.93
SigMa	0.0627	5.63	27.91	20.54	0.258	10.70
SEGA	0.4422	5.82	29.63	21.18	0.270	11.99
AReA	**0.6001**	**5.87**	**29.89**	**21.44**	**0.275**	**13.90**

4.2 Ablation Studies

We conduct ablation studies at 4096 × 4096 on FLUX.1 using Aesthetic-4K, first examining the contribution of each phase and then their design choices. As shown in Table 3 and Figure 7, without

Table 3: Ablation study on FLUX.1 at 4096×4096 on Aesthetic-4K. Within each phase, the first row replaces the whole phase with full attention; the rows below change one component.

	Method	Latency(s)↓	ImageReward↑	Aesthetic↑	MUSIQ↑	PickScore↑	HPS↑	CLIP↑	FID↓
Phase I	No Phase I (full attention)	201	-0.41	5.85	38.06	20.73	0.264	26.63	171.42
	Flat $f = 2$	165	1.23	6.27	**46.67**	23.03	0.294	29.13	152.08
	Flat $f = 4$ (no ramp)	142	1.27	<u>6.38</u>	44.72	23.17	0.292	29.63	154.21
Phase II	No Phase II (full attention)	248	1.10	6.35	40.67	22.71	0.286	28.78	155.44
	Fixed windows ($w = 64$)	123	1.26	**6.43**	45.26	23.12	<u>0.296</u>	**29.78**	152.15
	w/o Global-KV bridge	133	**1.32**	6.25	<u>45.38</u>	23.25	0.294	29.11	161.25
	w/o Halo	145	**1.32**	6.34	45.28	<u>23.29</u>	<u>0.296</u>	29.64	<u>148.76</u>
	w/o DWT splits	187	<u>1.31</u>	6.35	45.06	23.22	0.285	29.53	149.63
AReA (full)		151	**1.32**	6.37	45.22	**23.31**	**0.297**	<u>29.71</u>	**148.50**

Phase I, the global structure is not established, and local windows evolve independently, producing inconsistent content across the image. Without Phase II, the output lacks fine details despite a well-formed low-frequency structure. Beyond improving efficiency, Phase II limits how far and over how many keys each token attends in later steps, avoiding diluted attention that blurs fine details.

We further evaluate design choices within each phase. In Phase I, keys and values are pooled within blocks so that queries attend to a downsampled representation of the image, with the initial scale factor f keeping this representation within the native resolution. As shown in Table 3, starting with f that yields a pooled representation beyond the native resolution degrades performance, as queries attend over distances unseen during training, pushing the model out of distribution. Moreover, since denoising progresses from coarse to fine, queries should attend to increasingly fine representations as sampling proceeds. Keeping this coarse scale fixed throughout Phase I passes only a weak representation of the image to Phase II, leaving a poorly formed global structure on which to build fine details. Accordingly, we start from f that matches the native resolution and decrease it linearly in discrete steps during Phase I, equivalently increasing the resolution of the pooled keys and values.

Table 4: Phase I pooling strategy on FLUX.1 (4096×4096, Aesthetic-4K). †Default.

Pooling	IR↑	Aes.↑	MUSIQ↑	Pick↑	HPS↑	CLIP↑	FID↓
Mean	1.23	6.30	44.93	22.84	0.289	29.28	152.78
Random	1.25	<u>6.35</u>	**45.59**	<u>23.22</u>	<u>0.293</u>	<u>29.37</u>	152.53
Medoid	<u>1.26</u>	6.31	44.04	23.19	<u>0.293</u>	29.19	<u>151.82</u>
Center†	**1.32**	**6.37**	<u>45.22</u>	**23.31**	**0.297**	**29.71**	**148.50**

Phase II schedules the size, number, and positions of windows across denoising steps. As shown in Figure 7, using a fixed window size of 64, matching the native resolution of FLUX.1, breaks the image into inconsistent regions. Such outputs are unusable, yet they still score well on metrics that are insensitive to layout errors, such as Aesthetic and CLIP; the failure is instead reflected in lower ImageReward and PickScore and a higher FID. Similarly, removing the global-KV bridge leads to duplicated and repeated subjects across the image (Figure 13), which leaves most metrics nearly unchanged but raises FID from 148.5 to 161.3. The Halo removes seams along window boundaries that are subtle and visible mainly in qualitative comparisons (Figure 13). Finally, guided by a DWT detail map, AReA splits windows over smooth regions into smaller ones, where this causes no harm, reducing latency while maintaining quality.

5 Conclusion

We presented AReA, an adaptive receptive-field attention mechanism that enables efficient and coherent end-to-end high-resolution generation with diffusion models, without any training and across different DiT backbones. Our key observation is that attention becomes increasingly local during denoising, so the receptive field that attention needs changes over time. AReA follows this coarse-to-fine progression in two phases. In early steps, it attends globally over pooled keys and values at a scale close to the native resolution to form the global structure, and in later steps, it attends within local windows that shrink and adapt to image content to recover fine details. This keeps token interactions close to the training regime without RoPE extrapolation, improving global consistency and fine-grained detail at resolutions far beyond training while reducing attention cost. In addition to the spatial redundancy addressed here, consecutive denoising steps also exhibit temporal redundancy, which we leave to future work for further acceleration.

AI USE STATEMENT

In this work, we used generative AI tools to generate the text-prompt dataset used for image-generation evaluation. We have not used generative AI tools to propose or refine hypotheses, develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, design or provide feedback on the research methodology or experiments, implement the proposed method, or interpret results, and translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for general code debugging and for polishing the paper text for grammar, syntax, and readability. We have reviewed all AI-assisted work: the authors manually inspected the generated prompts, reviewed and tested all debugging changes, and verified that polished text preserved the original technical content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. Arniqa: Learning distortion manifold for image quality assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 189–198, 2024.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22669–22679. IEEE, 2023.

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

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

Jiazi Bu, Pengyang Ling, Yujie Zhou, Pan Zhang, Tong Wu, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Hiflow: Training-free high-resolution image generation with flow-aligned guidance. *Advances in Neural Information Processing Systems*, 38:144317–144351, 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

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, 2024a.

Ruoyi Du, Dongyang Liu, Le Zhuo, Qin Qi, Hongsheng Li, Zhanyu Ma, and Peng Gao. I-max: Maximize the resolution potential of pre-trained rectified flow transformers with projected flow. *arXiv preprint arXiv:2410.07536*, 2024b.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 30234–30262, 2024.

Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, et al. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. In *European conference on computer vision*, pp. 39–55. Springer, 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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.

Linjiang Huang, Rongyao Fang, Aiping Zhang, Guanglu Song, Si Liu, Yu Liu, and Hongsheng Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. In *European conference on computer vision*, pp. 196–212. Springer, 2024.

Juno Hwang, Yong-Hyun Park, and Junghyo Jo. Upsample guidance: Scale up diffusion models without training. *arXiv preprint arXiv:2404.01709*, 2024.

Ideogram AI. Ideogram 4.0. `https://ideogram.ai/blog/ideogram-4.0/`, 2026.

Noam Issachar, Guy Yariv, Sagie Benaim, Yossi Adi, Dani Lischinski, and Raanan Fattal. Dype: Dynamic position extrapolation for ultra high resolution diffusion. In *Forty-third International Conference on Machine Learning*, 2026.

Jinho Jeong, Sangmin Han, Jinwoo Kim, and Seon Joo Kim. Latent space super-resolution for higher-resolution image generation with diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2355–2365. IEEE, 2025.

Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *Advances in Neural Information Processing Systems*, 36: 70847–70860, 2023.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5148–5157, 2021.

Younghyun Kim, Geunmin Hwang, Junyu Zhang, and Eunbyung Park. Diffusehigh: Training-free progressive high-resolution image synthesis through structure guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 4338–4346, 2025.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Sungho Koh, SeungJu Cha, Hyunwoo Oh, Kwanyoung Lee, and Dong-Jin Kim. Scalediff: Higher-resolution image synthesis via efficient and model-agnostic diffusion. *Advances in Neural Information Processing Systems*, 38:95571–95596, 2026.

Mathis Koroglu, Guillaume Jeanneret, Hugo Caselles-Dupré, Matthieu Cord, and Arnaud Dapogny. Jolt: Joint latent trajectories for context-guided high-resolution tiled generation. *arXiv preprint arXiv:2608.15395*, 2026.

Labs, Black Forest. Flux. *https://github. com/black-forest-labs/flux*, 2024.

Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36:50648–50660, 2023.

Zhihang Lin, Mingbao Lin, Meng Zhao, and Rongrong Ji. Accdiffusion: An accurate method for higher-resolution image generation. In *European Conference on Computer Vision*, pp. 38–53. Springer, 2024.

Songhua Liu, Weihao Yu, Zhenxiong Tan, and Xinchao Wang. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 2024.

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

Yiyang Ma, Feng Zhou, Xuedan Yin, Pu Cao, Yonghao Dang, and Jianqin Yin. Resdit: Evoking the intrinsic resolution scalability in diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40612–40621, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 31932–31951, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Javad Rajabi, Kimia Shaban, Koorosh Roohi, David B Lindell, and Babak Taati. Sega: Spectral-energy guided attention for resolution extrapolation in diffusion transformers. *arXiv preprint arXiv:2605.22668*, 2026.

Jingjing Ren, Wenbo Li, Haoyu Chen, Renjing Pei, Bin Shao, Yong Guo, Long Peng, Fenglong Song, and Lei Zhu. Ultrapixel: Advancing ultra high-resolution image synthesis to new peaks. *Advances in Neural Information Processing Systems*, 37:111131–111171, 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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Christoph Schuhmann. Improved Aesthetic Predictor: CLIP+MLP Aesthetic Score Predictor, 2022. URL `https://github.com/christophschuhmann/improved-aesthetic-predictor`. Accessed: 2025-06-06.

Shuwei Shi, Wenbo Li, Yuechen Zhang, Jingwen He, Biao Gong, and Yinqiang Zheng. Resmaster: Mastering high-resolution image generation via structural and fine-grained guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6887–6895, 2025.

Luigi Sigillo, Shengfeng He, and Danilo Comminiello. Latent wavelet diffusion for ultra high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2026, pp. 132796–132825, 2026.

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

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

Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 18885–18907, 2024.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 2555–2563, 2023.

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.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Howard Xiao, Brian Chao, Lior Yariv, and Gordon Wetzstein. Spectral progressive diffusion for efficient image and video generation. *arXiv preprint arXiv:2605.18736*, 2026.

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

Ruonan Yu, Songhua Liu, Zhenxiong Tan, and Xinchao Wang. Ultra-resolution adaptation with ease. *arXiv preprint arXiv:2503.16322*, 2025.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23464–23473. IEEE, 2025.

Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10188–10198. IEEE, 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.

Yuyao Zhang and Yu-Wing Tai. Ultraimagegen: Efficient ultra-high-resolution image generation with hierarchical local attention. In *European Conference on Computer Vision*, pp. 210–229. Springer, 2026.

Bingxuan Zhao, Qing Zhou, Yu Wang, Chuang Yang, and Qi Wang. σ: Sigmoid modulation for ultra high resolution diffusion. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=47JZSOkw5C.

Min Zhao, Bokai Yan, Xue Yang, Hongzhou Zhu, Jintao Zhang, Shilong Liu, Chongxuan Li, and Jun Zhu. Ultraimage: Rethinking resolution extrapolation in image diffusion transformers. *arXiv preprint arXiv:2512.04504*, 2025.

APPENDIX

A PHASE I: POOLING FACTOR AND FRAME EXTENT

In Phase I, we use full-resolution queries, and keys and values are pooled in $f \times f$ blocks with query coordinates divided by f, and steps cost $1/f^2$. This section covers the two choices it defers, which factor to run at on which step and which token represents a block.

Both turn on one quantity. The pooled grid is $(W/f_t) \times (H/f_t)$, so the coarse frame spans $\max(H, W)/f_t$ patches on its longer side, and that is the extent the model is asked to work at. At 4096^2 on FLUX.1 the patch grid is 256 and the native training extent is 64: a factor of 4 places the frame at exactly 64, and a factor of 2 places it at 128, twice native. In non-square cases, as we have two sides with different number of pixels (6144×2048) we choose the smallest factor that puts both sides inside the native range. The schedule therefore begins at or below the training extent and ends outside it, and both ends of that are deliberate.

A.1 WHY THE FACTOR RAMPS

Holding $f_t = f_0$ (in 4096×4096 f_0 is 4) for the whole phase never leaves the native extent. The layout it produces is in distribution and globally consistent, but it is decided on a 64×64 frame and stays there: the phase ends with structure settled only at that scale which is too coarse to settle details like border of objects, and this issue is handed to Phase II, where attention is local and cannot settle it either. Starting at $f_t = 2$ instead fails at the other end. Every step then runs at twice the native extent, so the layout is laid down at relative distances the model has never seen and arrives at Phase II already carrying duplicated subjects and degraded global structure. Nothing downstream repairs a layout; the detail phase only adds texture to whatever Phase I decided. The ramp takes the first and then moves to the second, once there is a layout for the finer frame to refine rather than to invent. Figure 8 shows both failure modes: fused, boundary-less subjects under a flat $f = 4$, and scattered, duplicated content under a flat $f = 2$.

Eq 2 gives the spacing, but its values must also divide the patch grid for the block pooling to be exact, which plain rounding does not guarantee. Each f_t is therefore snapped to the nearest grid divisor of both grid axes that does not exceed f_{t-1}, so the sequence is valid and never increases:

$$f_t = \underset{d \in \mathcal{D},\ d \le f_{t-1}}{\arg\min} \left| d - \tilde{f}_t \right|, \quad t = 2, \dots, S_c - 1, \qquad f_1 = f_0,\ \ f_{S_c} = f_{\min}, \tag{4}$$

$$\mathcal{D} = \{\, d \in \mathbb{N} : d \mid \gcd(H, W) \,\}, \qquad f_0 = \min\{\, d \in \mathcal{D},\ d \ge 2 : \max(H, W)/d \le P \,\}, \tag{5}$$

where P is the native extent in patches (64 for FLUX.1; we use the same value for Qwen-Image). At 4096^2 the divisors of 256 at or below 4 are $\{1, 2, 4\}$, so the schedule has only two rungs, $[4, 4, 4, 4, 2, 2, 2, 2]$: four steps at extent 64 followed by four at 128. Resolutions whose grid has more divisors give a genuine ramp, and at 6144^2 the plan is $[6, 6, 4, 4, 4, 3, 3, 2]$, an extent running $64 \to 96 \to 128 \to 192$. The floor is $f_{\min} = 2$ rather than 1 because an $f = 1$ pool is the identity, which would turn the last coarse steps into non-efficient plain full attention at the target resolution.

The cost runs opposite to the quality argument, which is worth stating plainly. Over the eight coarse steps at 4096^2 the ramp costs 1.25 full-attention passes instead of full 8 passes, against 0.50 for a flat $f_t = 4$ and 2.00 for a flat $f_t = 2$. The ramp is not the cheap option: it is two and a half times the cost of never refining, and it buys a settled layout at five eighths of the price of deciding that layout out of distribution.

A.2 WHICH TOKEN REPRESENTS A BLOCK

Phase I selects an actual token from each block. The immediate question is why, given that averaging within a block seems like the natural option, and it performs worst among the four variants in Table 4. The f^2 keys in a block share a substantial common component, so taking their average preserves that shared part while reducing the components that differentiate them by roughly a factor of f. As a result, the pooled keys become almost parallel to one another: the coarse map preserves mass but loses discrimination, which appears as blurred textures and washed-out fine detail (Figure 9). Using



Figure 8: **Effect of Phase I Pooling Schedule**. Holding $f = 4$ keeps the layout in distribution but settles it only on the 64×64 frame, so object boundaries stay unresolved and the sheep merge into a single blurred mass. Starting at $f = 2$ lays out the scene at twice the native extent, producing scattered and repeated subjects. Ramping $f = 4 \rightarrow 2$ first fixes the layout in distribution and then refines it on the finer frame, giving separated subjects and a coherent scene.



Figure 9: **Phase I Pooling Strategies**. Mean pooling averages away the components that distinguish keys within a block, blurring fine detail such as armor ornamentation and facial features. Medoid selection keeps a real token but favors the most typical one in each block, partially inheriting this loss. Random selection preserves more detail but depends on the seed. Center pooling (default) uses a fixed, content-free token per block and gives the sharpest and most coherent results.

a real key preserves its original magnitude and direction, and because K and V come from the same token, the key–value pair remains consistent.

The second issue is how to pick it. A content-based rule, such as the token closest to the block's mean (medoid), favours the most typical token in the block and discards the distinctive ones, so it partially inherits the detail loss of mean pooling. Content-free rules avoid this: random and centre selection pick by seed and by position, independent of what the block contains, and both preserve fine detail better than medoid (Table 4, Figure 9). A content-based index is also risky on a backbone that uses true classifier-free guidance, since the two forward passes rank each block independently: the selection becomes discontinuous, so a single flip can shift a pooled key by about its own norm, and guidance then magnifies that discrepancy by $(\text{scale} - 1)$.

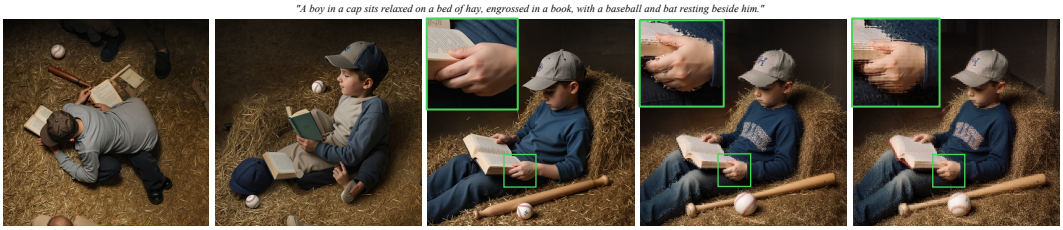

Figure 10: **Too Early and Too Late Phase Switch.** An early switch leaves the structure unsettled, degrading the second phase, while a late switch leaves residual noise with no room to add fine details.

Table 5: Ablation of the phase switch step s (Phase I $\rightarrow$ Phase II) on FLUX.1 at 4096×4096 resolution on 20 prompts from Aesthetic-4K. The noise level σ at the switch is given in parentheses.

	Too early	Stable					Too late				
Switch step s (σ)	2 (0.98)	4 (0.95)	6 (0.92)	**8 (0.89)**	10 (0.85)	12 (0.81)	14 (0.76)	16 (0.70)	18 (0.64)	20 (0.56)	22 (0.46)
Quality											
ImageReward↑	0.77	1.22	1.18	**1.25**	1.24	**1.25**	1.20	1.22	1.21	1.19	1.18
Aesthetic↑	6.03	6.20	6.22	**6.23**	6.15	6.16	6.21	6.20	6.18	6.16	6.08
CLIP-IQA↑	0.685	**0.712**	0.705	0.705	0.696	0.686	0.671	0.658	0.628	0.579	0.528
Alignment											
CLIP↑	29.00	**30.44**	30.08	30.34	30.21	30.16	30.18	30.25	30.10	30.05	29.81
PickScore↑	22.30	22.93	23.11	**23.24**	23.16	23.20	23.01	22.98	22.92	22.76	22.54

Between the two content-free rules, the block centre performs best (Table 4), and it is additionally static. It needs no per-step seed and cannot differ between two forward passes by construction, so one operator serves the distilled and the guided backbones with no record-and-replay path. Its one known cost, recorded rather than hidden: the centre of a block at $f = 4$ on a 256 grid lies on rows and columns $\equiv 2 \pmod 4$, so the phase reads the same 64 of 256 lines at every step it spends at that factor, and the first two and the last line of the canvas are never sampled there. Moving to $f = 2$ changes that lattice, a second and smaller reason not to stay at one factor.

A.3 When to switch phases

As an experiment, we evaluated different $\sigma^\star$ values by varying the switch step on FLUX.1 model. Results are provided in Table 5, and qualitative comparison is shown in Figure 10. As results show, switching too early leaves the structure undecided and windows are unable to shape high-level structure alone. Also, switching too late will add noise to the output and leave no room for windows to add fine details. There is a range of σ that outputs are stable, while the handoff σ of 0.89 seems the most robust option for the FLUX.1 model. The default $\sigma^\star$ for other backbones such as 0.86 for Qwen-Image, 0.94 for FLUX.2, and 0.91 for Ideogram 4 were chosen by the same procedure, and fall in the stable range of σ considering their different noise scheduling algorithm, distribution, and training strategies.

B Phase II: window attention cost and schedule

This section makes the cost model of the second phase explicit. Let H and W denote the height and width of the patch grid, so the canvas has HW image tokens. A window of side w contains $n = w^2$ image tokens. Attention inside a window is quadratic in the length of that window's joint sequence. We use this fact twice: here, to choose the start size and the schedule of window sizes over the second phase steps, and in Appendix C, to price the edits the DWT-adaptive tiler may make to that schedule.

B.1 Cost of a windowed step

A denoising step in the second phase covers the canvas with the base tiling of Section 3.3: the fewest full windows of side w that cover each axis,

$$N(w) \;=\; \left\lceil \frac{H}{w} \right\rceil \times \left\lceil \frac{W}{w} \right\rceil, \tag{6}$$

that is, $n_W \times n_H$ evaluated at $w = w_t$. Residual overlap is spread along each axis, but the *count* $N(w)$ depends only on w. Random jitter of interior boundaries does not change $N(w)$ or w, so it does not change the cost. If each window attends over its own w^2 image tokens, the number of query-key products in the step is $N(w) \times w^4$. One full-attention pass over the canvas costs $(WH)^2$ products. We therefore report the cost of a windowed step in units of one full-attention pass,

$$C(w) \;=\; N(w) \times \frac{w^4}{(WH)^2}. \tag{7}$$

Full attention is $C = 1$. Ignoring the overlap the ceilings introduce, $N(w) \approx WH/w^2$ and a step costs about $WH\,w^2$ products, so $C(w) \approx w^2/(WH)$: the map inside a window shrinks as w^4 while the number of windows grows only as w^{-2}, and halving the window quarters the step. Section B.2 is about the sizes at which the ceilings make that approximation fail.

Eq 7 prices the w^2 image tokens of a window and nothing else; the other two parts of the Phase II key set enter as follows. When E extra keys (the Halo ring and the global-KV bridge) are appended to every window, queries stay w^2 and keys become $w^2 + E$, which multiplies Eq 7 by $(1 + E/w^2)$. Text tokens are omitted: a window's joint sequence is $w^2 + T$ keys, but T is nearly constant across window sizes, so dropping it changes neither the shape of $C(w)$ nor the ranking among window sizes. It is reinstated in Appendix C, where a split multiplies the number of windows that each carry it.

B.2 Why the cost is not monotone

Eq 7 is not monotone in w, because $N(w)$ is a step function. Decreasing w by one token can add a whole extra row and column of windows, and the extra tiles can outweigh the cheaper per-window map. On a 256×256 grid, $w = 128$ uses two windows per axis and costs 0.250, while $w = 127$ uses three per axis and costs 0.545. Every size in $\{182, \ldots, 255\}$ is more expensive than full attention, because several near-canvas windows are each almost a full pass, and $w = 255$ costs 3.94. On a square canvas only the degenerate choice $w = \max(H, W)$, a single window, returns to $C = 1$; on a rectangular one it does not, since one window of that side costs $\max(H, W)^4/(HW)^2$.

A geometric shrink that ignored this lattice would therefore not stay cheaper than full attention, and the two sections below are the two places we intervene.

B.3 Choosing the start size

Section B.2 gives the rule directly. We open the detail phase with the largest window whose step is still cheaper than a full-attention pass,

$$w_{\text{start}} = \begin{cases} \max\{\, w \ge w_h : C(w) < 1,\; N(w) > 1 \,\} & \text{if non-empty,} \\ w_h & \text{otherwise,} \end{cases} \tag{8}$$

$$w_h = \max\big(w_{\min},\; \lfloor \max(H, W)/f_{S_c} \rfloor\big). \tag{9}$$

Both conditions are needed. The first is the point of the phase: a window phase that cost more than dense attention would buy nothing. The second excludes $w = \max(H, W)$, which is a single window covering the canvas, i.e. full attention itself, and so also sits at $C = 1$.

On a 256×256 grid this is $w_{\text{start}} = 181$. Everything from 182 to 255 costs more than a full pass, 181 costs 0.9996, and it tiles as two windows per axis, each covering just over 70% of an axis and half the canvas by area. The same rule gives 226 on a 320 grid and 271 on a 384 grid: the first step

from the second phase costs what one dense step costs, and Section B.4 makes every step after it cheaper. These values are for square canvases, where w_{start} is about 70% of the side. The search is bounded below by the handoff scale w_h, the longer side of the last Phase I grid, and falls back to it when no size qualifies, so Phase II then starts where Phase I left off. This happens on non-square canvases, where windows are clipped to the shorter axis and Eq 7 overestimates their cost.

This budget prices the image-token attention of Eq 7 only. The Halo and the global-KV bridge add keys on top of it (Appendices D and E), multiplying a step by $(1 + E/w^2)$. Because w_{start} is set at the budget limit, the initial steps of the phase would cost slightly more than a full pass once these keys are taken into account, with their additional cost being minimized by DWT splits (Figure 14).

B.4 The window size schedule

Over the S_d detail steps the window contracts from w_{start} to the floor $w_{\min}$, the model's native training extent in patches (64 for FLUX.1). Eq 3 is that contraction over the reals; the schedule the run uses is its integer form,

$$\tilde{w}_t \;=\; \max\Big(w_{\min},\; \Big\lfloor w_{\text{start}} \Big(\tfrac{w_{\min}}{w_{\text{start}}}\Big)^{\frac{t-1}{S_d-1}} \Big\rceil\Big), \qquad t = 1, \ldots, S_d, \tag{10}$$

with the endpoints pinned to w_{start} and $w_{\min}$.

The lattice of Eq 6 is not continuous either, so each rung is snapped down rather than rounded:

$$w_t \;=\; \max\big\{\, w \le \tilde{w}_t \;:\; C(w) \le C(w_{\text{start}}),\; w \le w_{t-1} - 1,\; w \ge w_{\min} + (S_d - t) \,\big\}, \qquad t \ge 2, \tag{11}$$

with $w_1 = w_{\text{start}}$. The three conditions are, in order: no rung costs more than the first one, consecutive sizes strictly decrease, so the window is genuinely dynamic rather than repeating, and rung t leaves room for the $S_d - t$ rungs still to place, so S_d distinct sizes fit and the last lands exactly on the floor. If no size satisfies all three, the cheapest admissible one is taken and the run reports it; if the span $[w_{\min}, w_{\text{start}}]$ cannot hold S_d distinct integers at all, Eq 10 is used unmodified.

The cap in Eq 11 is not redundant with Eq 8, because the cost is non-monotone *inside* the schedule as well: on the 256 grid the rung $w = 130$ costs 0.266 and the next ideal rung $w = 123$ costs 0.480, having crossed from four windows to nine. At the default start the ideal schedule happens to satisfy the cap at every rung, so Eq 11 returns Eq 10 unchanged; it binds when the start is forced higher, and it is what makes the guarantee hold for any start.

At 4096^2 ($H = W = 256$, $S_d = 20$, $w_{\min} = 64$) the schedule is

181, 171, 162, 154, 145, 138, 130, 123, 117, 111, 105, 99, 94, 89, 84, 80, 75, 71, 68, 64,

a constant ratio of 0.947, per-step drops falling from 10 to 4 tokens, and a total second-phase cost of 6.61 full-attention passes against the 20 that dense attention would spend over the same steps. No scheduled step exceeds the cost of attending over the full grid in the image-token attention that Eq 7 prices. As shown in Figure 11, the trend of window size as denoising proceeds aligns with how queries' attention with their most dominant keys gets more local (both attention curves are from Figure 3).

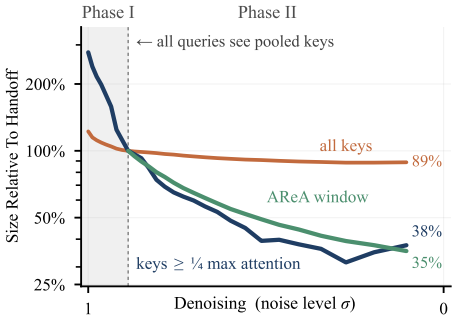

Figure 11: **Windows shrink with the Model's Attention.** AReA's window size at 4096^2 and the reach of the model's strongest attention at 1024^2, both relative to their value when Phase II begins. In Phase I, every query attends to pooled keys from the whole canvas. In Phase II, the window size closely tracks the reach of the model's strongest attention.

C CONTENT-ADAPTIVE WINDOW TILING

The schedule of Appendix B.4 fixes a window *size* per step, and Eq 6 turns it into a base tiling of $N(w_t)$ boxes. AReA then edits that cover using a wavelet detail map built from the current hidden states. The map is recomputed at every detail step, so the tiling can follow the image as it sharpens. Adaptation may move compute between regions; it may not spend more than the base tiling already costs.

C.1 THE DETAIL MAP

Let $x \in \mathbb{R}^{B\times W H\times D}$ be the image tokens of the current residual stream, arranged on the $W \times H$ patch grid. We first form a scalar field by taking the standard deviation across channels and averaging over the batch,

$$s_{r,c} \;=\; \frac{1}{B}\sum_{b=1}^{B} \mathrm{std}_d\big(x_{b,r,c,d}\big), \tag{12}$$

and subtract the spatial mean. Channel-wise standard deviation is high where tokens disagree across features, which is a cheap proxy for edges and texture; a channel mean averages a sharp channel against smooth ones and would blur those cues away before the transform sees them.

We then apply a separable Haar transform for $L = 3$ levels. On each 2×2 block of patches a, b, c, d (a, b the top row, c, d the bottom row) the low-pass and three detail bands are

$$\begin{aligned} A &= \tfrac{1}{4}(a+b+c+d),\\ D_v = \tfrac{1}{4}(a+b-c-d), \qquad D_h &= \tfrac{1}{4}(a-b+c-d), \qquad D_d = \tfrac{1}{4}(a-b-c+d), \end{aligned} \tag{13}$$

and the detail energy of that level is $|D_v| + |D_h| + |D_d|$. The cascade continues on A. Each level's detail is upsampled back to the full grid and accumulated with weight $2^{-\ell}$: the finest band localises an edge to a single patch, whereas a level-ℓ cell paints a $2^\ell \times 2^\ell$ patch square of uniform energy, so with equal weights those plateaus would dominate and drag the peaks onto smooth low-frequency blocks. The field is finally divided by its maximum. We write $E_{r,c}$ for this normalised map, and score each box ℓ of the uniform cover by $e_\ell = \mathrm{mean}(E|_\ell)$.

C.2 WHEN ADAPTATION MAY ACT

Percentile selection on $\{e_\ell\}$ is relative, so it would always mark some boxes as high and some as low, even on a map that is essentially uniform noise. We therefore gate adaptation on the coefficient of variation of E over a coarser tile grid of side $w_t/2$,

$$u \;=\; \frac{\mathrm{std}(\bar{E})}{\mathrm{mean}(\bar{E})}, \qquad \alpha \;=\; \mathrm{clip}\!\left(\frac{u-u_{\mathrm{lo}}}{u_{\mathrm{hi}}-u_{\mathrm{lo}}},\,0,\,1\right), \qquad u_{\mathrm{lo}} = 0.05,\; u_{\mathrm{hi}} = 0.25. \tag{14}$$

where $\bar{E}$, the map E averaged over a tile grid of side $w_t/2$. At $\alpha = 0$ the uniform cover is returned untouched. Tying the tile side to w_t rather than to a constant makes the gate track the schedule: early in the phase w_t is large, each tile averages fine noise away and the gate stays shut, while near the floor the same rule is sensitive to real structure. At most $\lfloor \alpha\, \phi\, N(w_t) \rfloor$ boxes may be edited, truncated rather than rounded up and with $\phi = 1$, so a step holding only a few large windows cannot rewrite its cover unless the map is strongly non-uniform, and the late many-box steps act the most. Figure 12 shows this on a full run: the gate leaves the base tiling untouched while windows are large (steps 10–14), and splits appear and multiply as windows approach the floor.

C.3 SPLITTING, AND WHAT IT COSTS

The base tiling remains the anchor and a box may change by at most one octave: a selected box is replaced by its four quadrants, which partition it exactly, and quadrants below $w_{\min}/2$ are not allowed. There is no recursion within a step.

Here the cost proxy includes the text: all windows attend to the same T text keys, and a split increases the number of windows that attend to them. Text tokens act only as keys, so a leaf ℓ with n_ℓ image tokens is charged

$$c(\ell) \;=\; n_\ell\big(T + n_\ell\big), \tag{15}$$

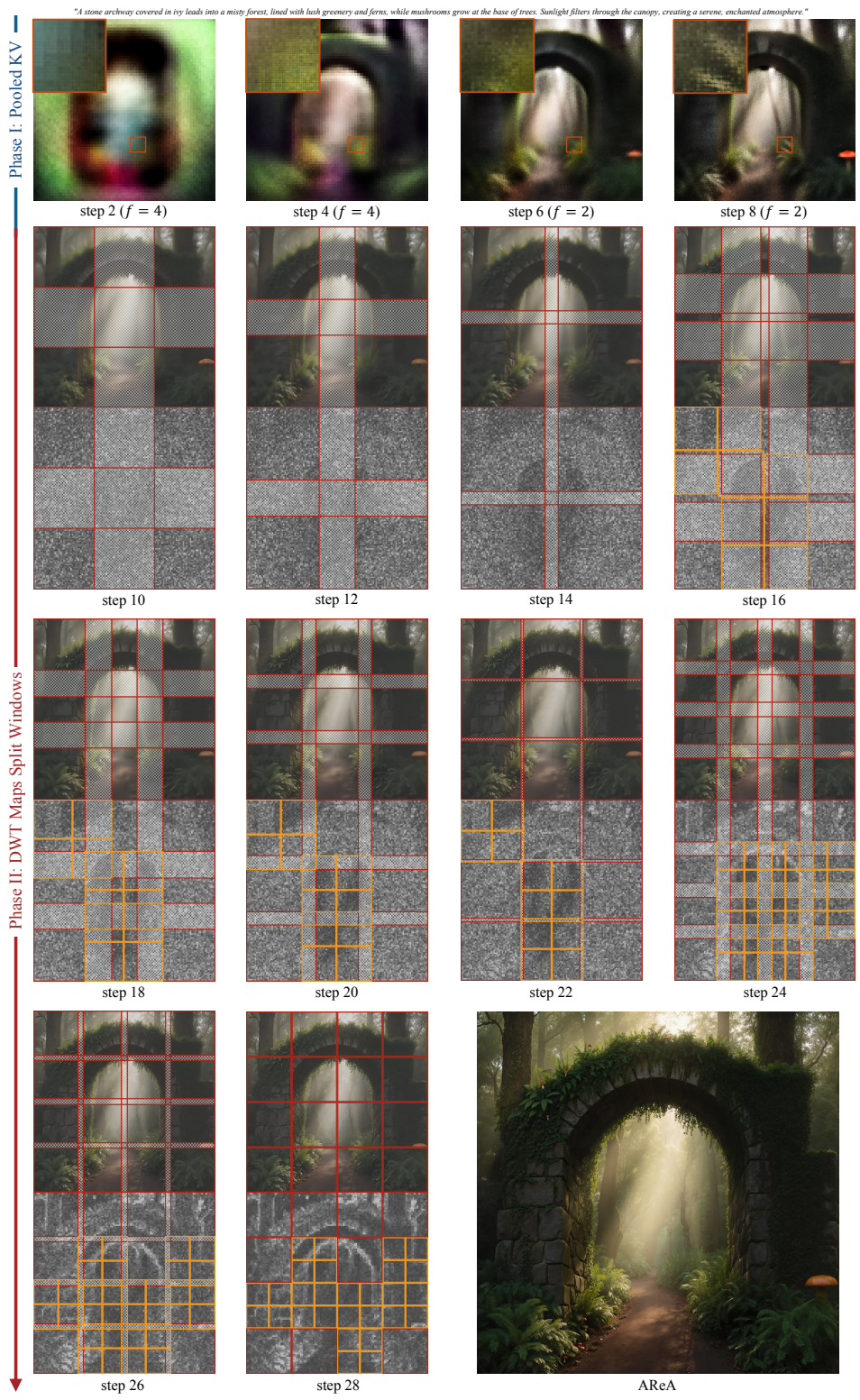

Figure 12: **AReA Across Denoising on FLUX.1 at 4096 × 4096**. *Top row*: Phase I steps with pooled keys and values (insets show the pixelation effect). *Other rows*: Phase II steps, each showing the current image (upper half) and its DWT detail map (lower half), with base windows in red, overlap regions hatched, and DWT splits in orange.

the budget for the step is the cost of the uniform cover, $B = \sum_{\ell \in \mathcal{U}_t} c(\ell)$ over its $N(w_t)$ boxes, and a candidate split is accepted only if the running total stays at most B. Splitting a parent with n image tokens into four quadrants changes the cost by

$$\Delta_{\text{split}} \;=\; 4 \cdot \tfrac{n}{4}\big(T + \tfrac{n}{4}\big) - n\big(T + n\big) \;=\; -\tfrac{3}{4}\, n^2. \tag{16}$$

The image-text term cancels: every image token still attends all T text keys, and the number of image tokens is unchanged. What moves is the image-image block, which falls from n^2 to $n^2/4$. Every split therefore returns three quarters of the parent's image-image products, so each admissible split is strictly cheaper and B acts as a guard rather than a binding constraint.

Which boxes split is the one choice left. Selection runs on $-e_\ell$: candidates are the boxes in the smoothest half of the cover (the $\tau = 0.5$ quantile of the inverted scores) that also sit strictly on the smooth side of the mean, the second condition guarding against quantile ties on sparse maps. We split the ***smoothest*** boxes rather than the most detailed ones. A flat region gains nothing from a wider receptive field and shows no seam when it is divided, while the split still returns three quarters of that box's attention cost, so textureless area is processed with cheaper maps and structure keeps the context that w_t already chose for it.

C.4 Deduplication and coverage

Adjacent parents in the uniform cover already overlap, so when two neighbours both split, their facing quadrants can cover each other almost entirely. Pairs of split leaves whose intersection over the smaller area is at least 0.8 are replaced by their bounding box, greedily and highest overlap first, before attention is run. That removes a near-duplicate SDPA call and stops the scatter-mean merge from double-counting those tokens. After these edits every image token is still covered by at least one window, and the step is no more expensive than the uniform tiling of side w_t that Appendix B.4 had already budgeted.

D A ring of neighbouring keys (Halo)

Windowed attention leaves a token at the rim of its window with no context on the far side of the boundary, which is why seams remain visible even after the center-weighted merge. The Halo answers this locally: at every Phase II step each window's key set is extended with the band of image tokens immediately surrounding its box (Figure 13, right).

Let a window of side w occupy a box on the $W \times H$ patch grid. For a ring of width δ the band is that box grown by δ on every side, minus the window itself and clipped at the canvas edge, so it contributes $(w + 2\delta)^2 - w^2$ keys. Only keys and values grow. The query set, the output rows and the coverage count the merge divides by are untouched, so both the coverage count and the blend weights are bit-identical to a run with no ring and only the attention output changes. That is what separates the Halo from simply enlarging the window: an enlarged window queries, denoises and merges its extra tokens, which moves the coverage and costs $(w + 2\delta)^4$, whereas here the cost is quadratic.

The width is not fixed. Each window redraws δ uniformly on $\{0, \ldots, \delta_{\max}\}$ at every step, with $\delta_{\max} = 8$ in all our runs. A ring of constant width would push every window's context break outward by the same amount rather than dissolving it, leaving a boundary in the same place on every step; redrawing it puts the break somewhere different for each window and each step. The draw is seeded on the run seed and the step index, so the two passes of a true-CFG model select the same band and the ring is never subtracted into the image as a prompt effect.

In the units of Eq 7 the ring multiplies a step by $\big((w + 2\delta)/w\big)^2$, and together with the R registers of Appendix E it makes up the E extra keys of that equation. Unlike the bridge, which holds a fixed share of each window's keys at every rung, a ring of fixed maximum width takes a growing share as the window contracts. The two are complementary, the ring being a dense band of immediate neighbours and the registers a sparse sample of the rest of the canvas.

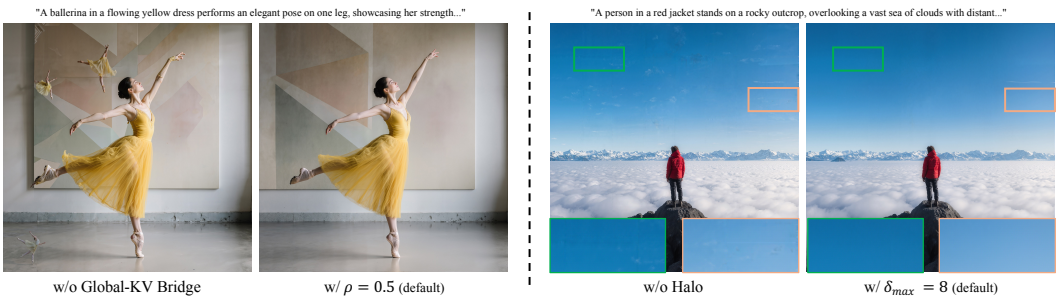

Figure 13: **Effect of the Global-KV Bridge and Halo**. *Left*: without the global-KV bridge, each window is blind to content elsewhere on the canvas and paints extra copies of the subject. With the bridge, windows see distinctive tokens from the rest of the grid and the duplicates disappear. *Right*: without the Halo, tokens at window edges lack context from across the boundary, leaving seams in smooth regions such as the sky.

E GLOBAL-KV BRIDGE

Phase I sets a coherent layout, but a Phase II window still attends only to tokens inside its box (and a thin local Halo). It therefore cannot see whether a small object already exists elsewhere on the canvas, and may paint a second copy as shown in Figure 13. The global-KV bridge gives every window a compact summary of the *complement* of that box, in the same softmax as its local keys.

E.1 WHAT IS APPENDED

Let a window contain $n = w^2$ image tokens. After the local rotary is applied, we append R extra keys and values, taken from image tokens *outside* the window, and leave the queries unchanged. The attention call is then over [text | window | registers], with n query rows and $n + R$ image key rows. Because the mix of local and global keys is inside one softmax, each query chooses its own balance. There is no separate global branch and no extra merge weight.

The R tokens are real tokens, not pooled averages. Keys carry a large shared component. Averaging many of them keeps that component and shrinks the rest, so the resulting vectors become nearly parallel and absorb attention mass without saying anything distinctive. We therefore gather K and V from the same selected token. K is taken after RoPE, so each register keeps the true canvas position of that token. V is never rotated. Sitting the registers at the origin, or giving them a second coarse coordinate frame, would be inconsistent with the single query rotary of one SDPA call. These registers sit at full-canvas distances from the query, beyond the extent the model was trained on. The resulting exposure is small and bounded: registers are a small share of each window's keys, and they account for part of AReA's remaining distance exposure. Offsets of this size are also not new to the model's attention: on FLUX.1, text keys sit at the origin of the positional grid, so every image query already attends to keys at canvas-scale offsets on every step, with or without the bridge.

Tokens already inside the querying window are excluded. A register from inside the box would only tell the window about itself, at lower resolution than it already has, and would waste a fraction $w/\max(W, H)$ of the budget.

E.2 WHICH TOKENS ARE SELECTED

Let k_i be the (unrotated) key of image token i, and let $\bar{k} = \frac{1}{HW}\sum_j k_j$ be the canvas-mean key. Each token receives an importance score

$$s_i = \left\|k_i - \bar{k}\right\|_2^2. \tag{17}$$

A large s_i means the token is unusual on this canvas, which is the content a distant window most needs to be told about if it is not to paint a second copy.

Taking the global top-R of $\{s_i\}$ is the wrong question. It returns a highlights reel: the budget can land on one striking object, and an empty region is never unusual, so the ranking never reports an absence. Both facts are what a window needs in order not to duplicate something. AReA therefore

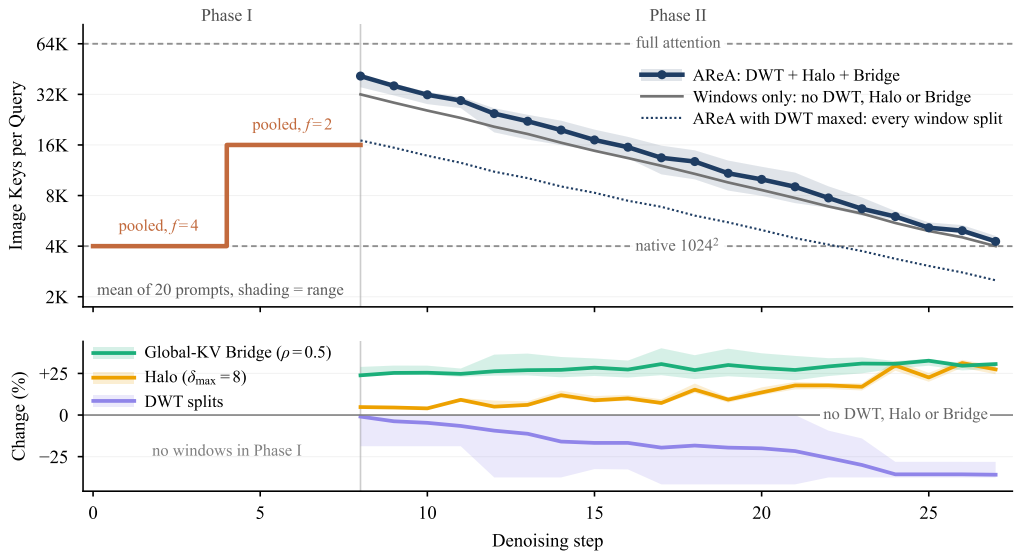

Figure 14: **Key Budget per Denoising Step**. Image keys attended per query on FLUX.1 at 4096×4096, mean of 20 prompts (shading: range). *Top*: in Phase I, pooled keys sit at the native 1024^2 extent for $f = 4$ and at four times that for $f = 2$. In Phase II, windows shrink from 32K keys to the native 4K, and full AReA stays close to plain windowing; no query attends to more than about 40K of the 64K image keys at any step. *Bottom*: change in keys per query relative to plain windows. The Global-KV Bridge adds a constant share, the fixed-width Halo takes a growing share as windows shrink, and DWT splits recover up to 35% late in the phase.

allocates the R slots over an 8×8 grid on the canvas, then picks by Eq 17 only inside each cell. Cell c receives a quota proportional to how many of its tokens lie outside the querying window. Leftover slots go to the cells with the most outside area. Every window still gets exactly R distinct registers, so the grouped SDPA call is unchanged.

E.3 How many registers

A constant R while w shrinks would make registers a growing share of each window's keys, i.e. weakest early (when duplication is decided) and strongest late (when only texture remains). Instead the count scales with the window,

$$R = (\rho\, w)^2, \tag{18}$$

so the register share of the image key set is constant, $R/(n+R) = \rho^2/(1+\rho^2)$, on every Phase II step. We use $\rho = 0.5$ on all models. In the units of Appendix B, the window-size budget is applied to $C(w)$ alone, so this factor sits on top of it.

Figure 14 summarizes how the three additions change the key budget over a full run. The bridge adds a constant quarter to each window's keys, as Eq 18 prescribes. The Halo adds little while windows are large and grows to about a quarter as they approach the floor. DWT splits offset both, removing up to a third of the keys late in the phase, where the gate opens most. The resulting key set per query stays within about a quarter of the plain windowed schedule, and well below the full canvas, at every step.

F Comparison with Multi-Stage Methods

Multi-stage methods such as HiFlow (Bu et al., 2026), I-Max (Du et al., 2024b), and ScaleDiff (Koh et al., 2026) first generate an image at the model's native resolution and then use it, or its trajectory, to guide sampling at the target resolution. The high-resolution output is therefore anchored to the base image: its layout is fixed at native resolution, and the target-resolution pass mainly adds local detail. In this sense, these methods are closer to generative super-resolution than to direct high-resolution synthesis, which AReA and the RoPE-based methods perform end-to-end at the target

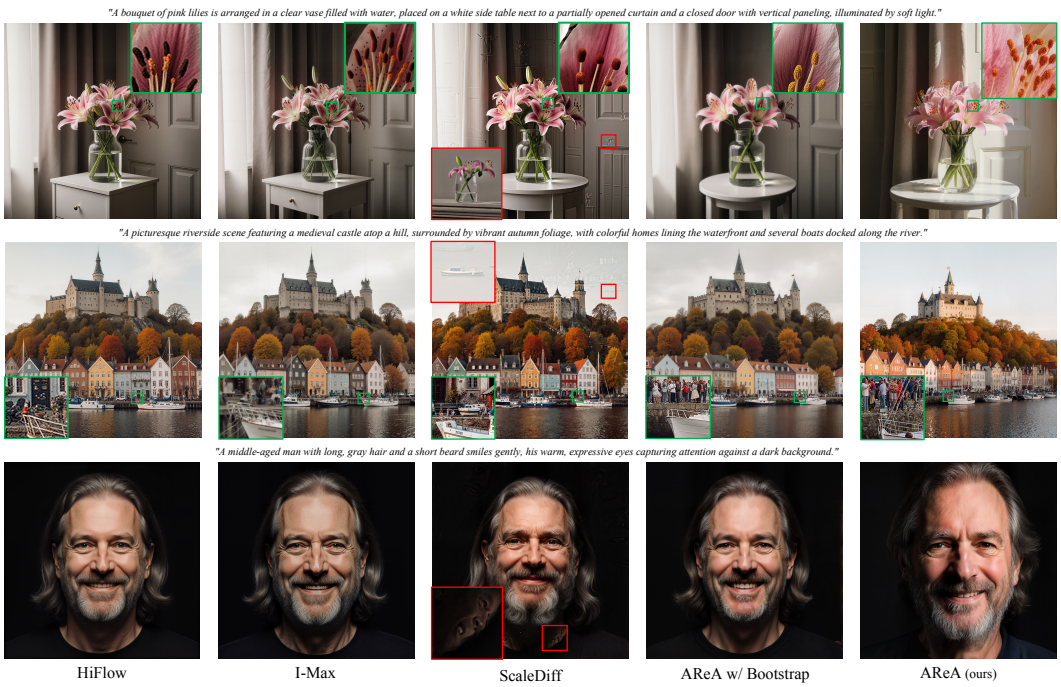

Figure 15: **Comparison with Multi-Stage Methods on FLUX.1 at 4096×4096.** Multi-stage methods and AReA-Bootstrap inherit the composition of the same base-resolution image and differ mainly in local detail. AReA with Phase I (default) sets its own layout at the target resolution and renders finer detail (green insets). ScaleDiff introduces repetitive content in low-detail regions (red insets).

Table 6: Comparison with multi-stage methods on FLUX.1 at 4096×4096 on Aesthetic-4K. **Bold**: best; underline: second-best.

Method	Latency (s)↓	ImageReward↑	Aesthetic↑	MUSIQ↑	CLIP-IQA↑	ARNIQA↑	CLIP↑	HPS↑	FID↓	FID_p ↓
HiFlow	182	1.1710	6.3147	38.674	0.6561	0.4811	29.3	0.2951	153.0	59.0
I-Max	440	1.1716	6.3043	41.086	0.6838	0.5013	29.2	0.2956	154.8	60.1
ScaleDiff	135	1.2339	6.2785	44.496	0.7066	0.5663	29.2	0.2935	152.5	73.7
AReA-Bootstrap	**126**	1.2658	6.3402	43.224	0.7027	0.5479	29.4	0.2955	152.0	61.2
AReA	151	**1.3250**	**6.3717**	**45.217**	**0.7132**	**0.5942**	**29.7**	**0.2969**	**148.5**	**58.2**

resolution. We therefore compare against them separately, on FLUX.1, the backbone used in their official implementations.

For a like-for-like comparison, we also evaluate AReA in a *bootstrap* mode that follows the same paradigm. Phase I is replaced by a base-resolution generation: the model produces a latent at its native resolution, which is upsampled to the target resolution's latent size without decoding and re-noised to the level at which Phase II begins ($\sigma^\star = 0.89$ on FLUX.1). The remaining 20 steps are then denoised with Phase II unchanged.

As shown in Table 6, AReA-Bootstrap is the fastest method and leads the multi-stage methods on ImageReward, Aesthetic, CLIP, and FID, while trailing ScaleDiff on the no-reference quality metrics. This shows that Phase II is an effective refiner even when the layout comes from a base image. Full AReA, however, achieves the best score on every quality, alignment, and fidelity metric. Figure 15 shows why: all multi-stage outputs, including AReA-Bootstrap, share nearly the same composition because they inherit the layout of the same base image, whereas AReA with Phase I forms its own layout at the target resolution and renders finer detail.

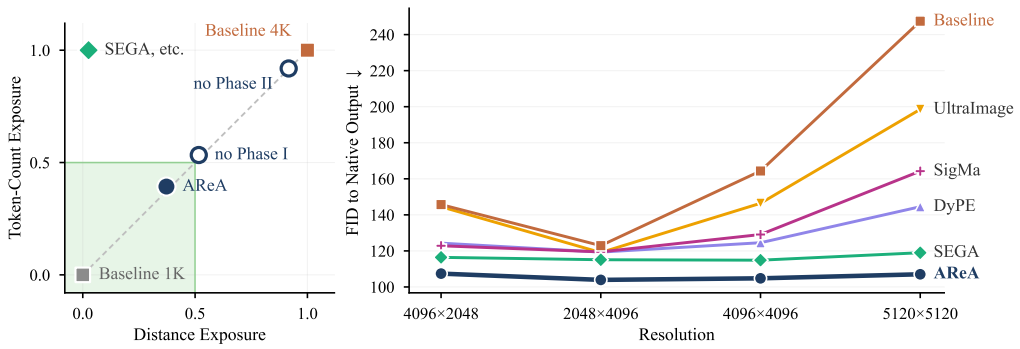

Figure 16: **Out-of-Distribution Exposure and Its Consequence on FLUX.1.** *Left*: distance and token-count exposure at 4096^2, averaged over denoising and scaled so that 0 is the model at its native 1024^2 and 1 is the vanilla model at 4096^2. RoPE-modifying methods (like SEGA) remove only the distance cause, while AReA reduces both. *Right*: FID of each method's outputs to the model's own native-resolution outputs; lower is closer to the model's intended and trained distribution. AReA stays lowest and flat as resolution grows.

G Out-of-Distribution Exposure

A high-resolution method is ultimately judged by how well it keeps the model inside the conditions it was trained on. This section describes how attention leaves those conditions when a diffusion transformer is run above its training resolution, measures how far each method goes, and shows what that costs in the output. The causes and AReA's response to them apply to any backbone with rotary position embeddings; we carry out the measurements on FLUX.1.

G.1 Two ways a large canvas leaves the training distribution

Consider a model trained on a $P \times P$ patch grid and run on a canvas k times larger per side. Its attention leaves the training distribution in two independent ways.

Distance. Query-key offsets grow to about kP patches per axis, while training never saw more than P. RoPE turns an offset into a rotation angle in each frequency band, so a large offset produces angles the model has never seen.

Token count. Every query attends to k^2 times as many keys as in training. Softmax spreads its mass over all of them, which flattens attention even when every offset is familiar.

Methods that adjust RoPE, such as SEGA, DyPE and UltraImage, rescale the rotary frequencies so that large offsets map back onto familiar angles. That addresses distance only. Every query still attends to every key on the canvas, so the token count is exactly that of the baseline. AReA acts on both. The Phase I pooled grid starts at the native size, and every Phase II window bounds both the offsets and the number of keys, and shrinks to the native size by the end of the phase.

G.2 Measuring exposure

The two causes can be separated only if each is measured on its own. For every method we therefore quantify, at each denoising step t, how much of its attention falls outside what the model saw in training along each cause. Let $\mathcal{A}_t$ be the set of query–key pairs the method attends to at step t. The *distance exposure* D_t is the fraction of pairs in $\mathcal{A}_t$ and rotary frequencies ω whose rotation angle $\omega\,\Delta$ lies outside the range $[-\omega(P-1), \omega(P-1)]$ covered on the training grid, where Δ is the offset between query and key; for a method that rescales RoPE, $\hat{\omega}$ is its rescaled frequency. The *token-count exposure* $K_t = \log_2(n_t/P^2)$ compares the number of keys n_t each query attends to with the number seen in training. Both follow from the attention pattern alone, so they can be computed exactly for any method without generating an image.

Steps do not contribute equally to the final image, so we average each quantity over the trajectory with weights proportional to the noise the step removes, $\sigma_t - \sigma_{t+1}$. We then express each average

relative to two references: the model at its native resolution, where $D = K = 0$ by construction, and the model at the target resolution with no method, which we set to 1. A value of 0.5 therefore means a method has removed half of that cause.

On FLUX.1 at 4096^2 ($P = 64$, $k = 4$), Figure 16 (left) places each method on the two axes. Rescaling the frequencies brings almost every rotation angle back into its trained range, but each query still attends to the full canvas, so the token count is exactly that of the baseline. AReA sits at $(0.37, 0.39)$, removing about 60% of each cause, and both phases are responsible. In Phase I the pooled grid begins at exactly the native size, so the early steps, which decide the layout, run in distribution on both axes. In Phase II each window limits both how far a query can look and how many keys it sees, and the windows shrink to the native size as denoising proceeds. The exposure that remains comes mainly from the larger windows early in Phase II and from the global-KV bridge, whose tokens keep their true canvas positions. Removing either phase moves AReA along the diagonal towards the baseline.

G.3 Measuring the consequence

If exposure causes artifacts, a method's outputs should drift away from what the model produces inside its training distribution. We measure that drift as the FID between each method's outputs and the same model's outputs at its native resolution with the same aspect ratio, on the same prompts and seed. The reference is the model's own output, not real photographs, so this FID asks how far a method pulls the model away from its own distribution, not how good the images are in general. Since FID resizes every image to 299^2, it reflects global statistics such as layout and colour rather than fine detail. Each resolution has its own reference set, so methods should be compared within a resolution, not across resolutions.

Figure 16 (right) shows the result on FLUX.1. AReA has the lowest FID at every resolution and stays nearly flat. The baseline drifts further as the canvas grows. The RoPE-based methods stay below the baseline but also drift upward at the two largest resolutions. SEGA, the method closest to native on the distance axis, stays above AReA at every resolution.

G.4 Why adjusting RoPE is not enough

On FLUX.1, SEGA essentially removes the distance cause, yet its outputs remain further from the model's own distribution than AReA's at every resolution, and the other RoPE-based methods drift further as the canvas grows. What they leave unchanged is the token count, which grows with the square of the side for any backbone ($16\times$ at 4096^2 and $25\times$ at 5120^2 for a model trained at 1024^2). AReA is the only method that reduces both causes, and its outputs stay closest to the model's own distribution at every resolution. This is consistent with the token count being a major source of the remaining out-of-distribution behaviour.

H HRGen-1K Benchmark

Existing evaluation sets for high-resolution generation are relatively small, such as Aesthetic-4K (Zhang et al., 2025) with 195 prompts, and cover a limited range of scene complexity. Inspired by SEGA (Rajabi et al., 2026) and JoLT (Koroglu et al., 2026), we introduce **HRGen-1K**, a benchmark of 1,000 curated prompts generated with GPT-5.6 Sol and Claude Opus 5. The prompts are designed to test the two properties that high-resolution generation must balance, a coherent global structure across a large canvas and fine detail at the local level, under diverse scenes, styles, and prompt formats. To this end, many prompts describe complex scenes with multiple interacting subjects, dense layouts, and unusual situations. Such prompts expose failures that simple single-subject prompts rarely reveal, such as duplicated subjects, distorted structures, and details that remain blurry or fragmented as the canvas grows.

Content and diversity. HRGen-1K spans a broad range of scene categories. Natural scenes cover landscapes, coastlines, and weather phenomena, while built environments range from architectural exteriors to detailed interiors. Human-centered prompts include portraits, fashion, street crowds, and people at work, and object-centered prompts feature food, still life, macro textures, and vehicles. The benchmark further includes wildlife, historical settings, narrative moments, imaginative worlds,

surreal concepts, and complex illustrated compositions. Many prompts pair a scene description with an explicit fine-detail constraint and a style directive, such as requiring individually resolved water droplets or species-accurate fur, while others are free-form descriptions. About 35% of the prompts request photographic styles and 21% request artistic styles such as illustration, painting, or paper craft. The remaining 43% leave the style unspecified, evaluating how well a model generates high-resolution images without any stylistic guidance.

Prompt length. HRGen-1K also covers a diverse range of prompt lengths, from 9 to 91 words with an average of 32.5 words (Figure 17(a)). Short and long prompts pose different challenges. Short prompts, which make up 12.1% of the benchmark, provide little information about the scene, so the model must infer plausible details on its own to fill a large canvas coherently. Long prompts, which make up 11.3%, describe many subjects and attributes at once, making it harder to follow every instruction and increasing the risk of omitted or mixed objects. All 1,000 prompts are unique, with a vocabulary of 4,867 distinct words.

Challenge attributes. Beyond scene diversity, HRGen-1K targets known failure modes of generative models (Figure 17(b)). A large share of prompts involve explicit spatial relations, human subjects whose anatomy must remain coherent, and specific weather or lighting conditions. The benchmark also includes 125 text-rendering prompts, ranging from single words on reflective or transparent surfaces to full sentences of up to 18 words that must preserve punctuation and numbers, as well as prompts with explicit object quantities, reflections, and transparency.

We hope HRGen-1K serves as a more challenging and comprehensive benchmark for evaluating future high-resolution generation methods. We will publicly release the full benchmark.

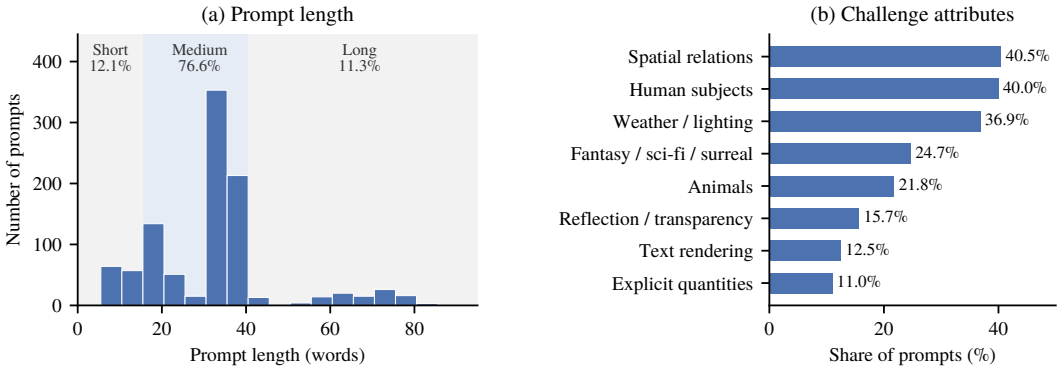

Figure 17: **Statistics of HRGen-1K**. *left*: Distribution of prompt lengths, grouped into short ($\leq$ 15 words), medium (16-40 words), and long ($>$ 40 words) prompts. *Right*: Share of prompts containing common attributes, estimated via keyword matching. Attributes are not mutually exclusive.

I Extended Related Work

In this section, we provide background on diffusion models and review prior work on high-resolution image generation in more detail, highlighting how our approach differs from existing methods.

Text-to-image (T2I) generation has advanced rapidly with diffusion models (Ho et al., 2020; Song et al., 2020), which produce high-quality images that closely follow textual prompts. A key step was latent diffusion (Rombach et al., 2022), which reduced computational cost and enabled models such as Stable Diffusion and SDXL (Podell et al., 2024). More recently, Rectified Flow (Liu et al., 2022) and Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have further improved quality and scalability, replacing the U-Net backbone (Ronneberger et al., 2015) with transformer architectures that capture global context and scale effectively with model and data size, enabling powerful text-to-image models such as FLUX.1 (Labs, Black Forest, 2024), FLUX.2 (Black Forest Labs, 2025), Qwen-Image (Wu et al., 2025) and Ideogram 4.0 (Ideogram AI, 2026).

High-resolution image generation aims to synthesize large, coherent images beyond a model's native training resolution. Naively applying pretrained diffusion models at higher resolutions often leads to repetitive patterns and structural distortions. Several works address this by training or fine-tuning on high-resolution data (Guo et al., 2024; Hoogeboom et al., 2023; Liu et al., 2024; Teng et al., 2024; Zhang et al., 2025). For example, URAE (Yu et al., 2025) proposed parameter-efficient adaptation, LWD (Sigillo et al., 2026) used wavelet-guided supervision to enhance fine details, UltraImageGen (Zhang & Tai, 2026) combined hierarchical local attention with low-resolution global guidance, and JoLT (Koroglu et al., 2026) jointly denoised low- and high-resolution latent trajectories to balance global coherence and local detail.

This has motivated training-free approaches that enable models trained at their native resolution to operate effectively at higher resolutions without additional data or fine-tuning. Instead of updating model weights, these methods (Bar-Tal et al., 2023; Huang et al., 2024; Hwang et al., 2024; Jin et al., 2023) modify the inference process, either by guiding high-resolution sampling with a base-resolution generation or by adjusting internal components such as attention and positional encoding. Some works (Du et al., 2024a; Lin et al., 2024) subdivide the target high-resolution images into overlapping trained-resolution patches, which are processed individually and then stitched together. MultiDiffusion (Bar-Tal et al., 2023) fuses diffusion paths over spatial windows to enable spatially controlled generation. DiffCollage (Zhang et al., 2023) similarly composes large content through parallel diffusion generation, while SyncDiffusion (Lee et al., 2023) synchronizes overlapping diffusion windows to decrease window incoherence for separate patches. However, the required overlap substantially increases computational cost and can lead to object repetition artifacts.

With the shift toward Diffusion Transformers, training-free high-resolution methods have largely adopted two-stage pipelines, in which a base-resolution generation guides sampling at the target resolution. Some methods (Kim et al., 2025; Hwang et al., 2024) upsample the base-resolution image and refine it with techniques such as SDEdit (Meng et al., 2021), relying on the model's ability to generate strong local details at higher resolutions. I-Max (Du et al., 2024b) treats the base-resolution flow as a projection of the high-resolution flow and exploits the linear interpolation property of rectified flows to construct dynamic guidance that steers high-resolution sampling. HiFlow (Bu et al., 2026) extends this idea by constructing a virtual reference flow from the full base-resolution trajectory, providing initialization, direction, and acceleration guidance that aligns high-resolution sampling with coarse structure while enhancing fine details. HiDiffusion (Zhang et al., 2024) resizes feature maps in the outer blocks of the U-Net so that deeper blocks operate at the training resolution, and replaces global self-attention with shifted-window attention to improve efficiency. ResMaster (Shi et al., 2025) uses a low-resolution reference to guide patch-wise high-resolution generation, aligning low-frequency components to preserve global structure while using reference-derived image and enriched text prompts to refine local details. DemoFusion (Du et al., 2024a) generates high-resolution images through progressive upscaling, using skip residuals from the base-resolution generation to preserve global structure and dilated sampling to improve global coherence across local patches. DiffuseHigh (Kim et al., 2025) upsamples the base-resolution image and re-denoises it, using the Discrete Wavelet Transform (DWT) to preserve its structure. ScaleDiff (Koh et al., 2026) upsamples the base-resolution image and refines it with SDEdit (Meng et al., 2021) to add local details at the target resolution. It combines Neighborhood Patch Attention with a frequency-mixing strategy that merges low-frequency content from latent-space upsampling with high-frequency content from RGB-space upsampling, and applies latent-space structure guidance to preserve global structure while recovering fine details.

While effective, these methods depend on base-resolution guidance and introduce additional complexity into the denoising process. Since the high-resolution output is anchored to the base image, its quality is bounded by that image, and any errors in the initial generation propagate to the final result. High-resolution generation is not simply a matter of producing more pixels, as the level of detail, structure, and composition must also scale with the resolution and remain coordinated across the image. As a result, many methods reach the target pixel count but fail to deliver a corresponding gain in detail fidelity, producing images that are larger but not truly richer in detail. Thus, matching the target pixel count alone does not necessarily yield high-quality images with faithful fine details.

To overcome the limitations of multi-stage approaches, more recent direct-inference methods generate high-resolution images end-to-end, aiming to unlock the capabilities already present in pre-trained models by modifying their internal behavior at inference time. This is typically achieved by adjusting the denoising process or key components such as positional encodings and attention, al-

lowing the model to operate beyond its training resolution without additional training. For example, ScaleCrafter (He et al., 2024) attributes repetition and structural artifacts at higher resolutions to the limited receptive field of convolutional kernels and addresses this by dilating convolutions in the U-Net at inference time, though this may distort structure. ResDiT (Ma et al., 2026) applies global attention with scaled positional embeddings in parallel with patch-wise attention, preserving the overall layout while recovering local details, and merges their outputs through Gaussian-weighted blending and spectral fusion to combine low-frequency structure with high-frequency detail.

Another line of work approaches the problem through the lens of RoPE length extrapolation. Modern DiTs encode spatial positions with Rotary Position Embeddings (RoPE) (Su et al., 2024), and generating at higher resolutions introduces positions beyond the training range, producing rotation angles the model has never encountered. This closely mirrors context extension in large language models (LLMs), where RoPE frequencies are rescaled to handle sequences longer than those seen during training (Peng et al., 2024; Ding et al., 2024). Building on these ideas, image-domain methods typically recalibrate RoPE's rotary frequencies along each spatial axis, so that positions at the target resolution remain within the distribution the model was trained on. DyPE (Issachar et al., 2026) introduces a step-dependent positional extrapolation schedule that follows the progression of denoising process, preserving low-frequency global structure early and gradually relaxing positional constraints to recover high-frequency details. UltraImage (Zhao et al., 2025) attributes repetition to mid-band RoPE components whose periods align with the training resolution and recursively adjusts them to span the target resolution; it further uses attention entropy to sharpen local patterns while preserving broad global attention. SigMa (Zhao et al., 2026) extends dynamic positional modulation with an axis-wise, scale-adaptive sigmoid schedule that adjusts both the timing and rate of the transition from structural formation to detail refinement. SEGA (Rajabi et al., 2026) analyzes the spatial-frequency energy of intermediate latents at each denoising step and maps it to per-dimension RoPE scaling, strengthening underrepresented frequency bands while relaxing already dominant ones and using spectral flatness to control the overall correction.

Despite their progress, these methods are tied to RoPE-based positional adjustments and inherit the constraints of length extrapolation. Since they modify architecture-specific components, they do not readily transfer to models with different positional encodings, and frequency recalibration alone cannot fully prevent out-of-distribution behavior at extreme scales, often resulting in artifacts and degraded detail fidelity at high resolutions. Moreover, these methods still apply full global attention over all tokens, leaving the quadratic cost of high-resolution generation unaddressed. As a result, balancing local detail, global coherence, and computational cost remains challenging, especially as resolution grows. In contrast, our method generates directly at the target resolution without base-resolution guidance or RoPE extrapolation. AReA adapts the attention receptive field across denoising steps, moving from global to local attention as generation progresses from coarse structure to fine detail. This keeps the model close to its training regime, allowing it to remain stable at resolutions far beyond the training range while reducing attention cost, without retraining or architecture-specific components.

J Additional Qualitative Comparisons

We provide further qualitative comparisons against direct-inference methods across backbones, resolutions, and aspect ratios; multi-stage methods are compared separately in Appendix F. On FLUX.1 at 4096×4096, Figure 18 shows prompts from HRGen-1K that stress composition, object counting, and text rendering, where existing methods distort scene geometry or lose the prompted content while AReA preserves both. On Qwen-Image, Figure 19 covers portrait (2048×4096) and landscape (6144×2048) canvases, where the failures of other methods grow with the aspect ratio, and Figure 20 shows results at 5120×5120. Finally, Figure 21 applies AReA to Ideogram 4 at 6144×6144 without any change to the method, illustrating that it transfers across backbones.



Figure 18: **Additional Comparison on FLUX.1 at 4096 × 4096 on HRGen-1K**. Existing methods warp scenes into symmetric, fisheye-like layouts, reduce object-centric prompts to zoomed-in texture, miscount objects, and garble rendered text. AReA preserves scene composition, object counts (highlighted in red), and legible text.



Figure 19: **Comparison on Qwen-Image at Non-Square Resolutions**. *Top*: 2048×4096. *Bottom*: 6144×2048. Full attention misplaces the subject on the tall canvas and duplicates it on the wide one. DyPE stretches the subject and introduces artifacts, and SEGA duplicates the subject in the wide setting. AReA keeps a single, well-proportioned subject in both aspect ratios.



Figure 20: **Comparison on Qwen-Image at 5120×5120.** Full attention blurs the scene and repeats subjects. SEGA keeps the layout but corrupts the details with high-frequency artifacts. AReA produces clean faces and the prompted number of subjects.

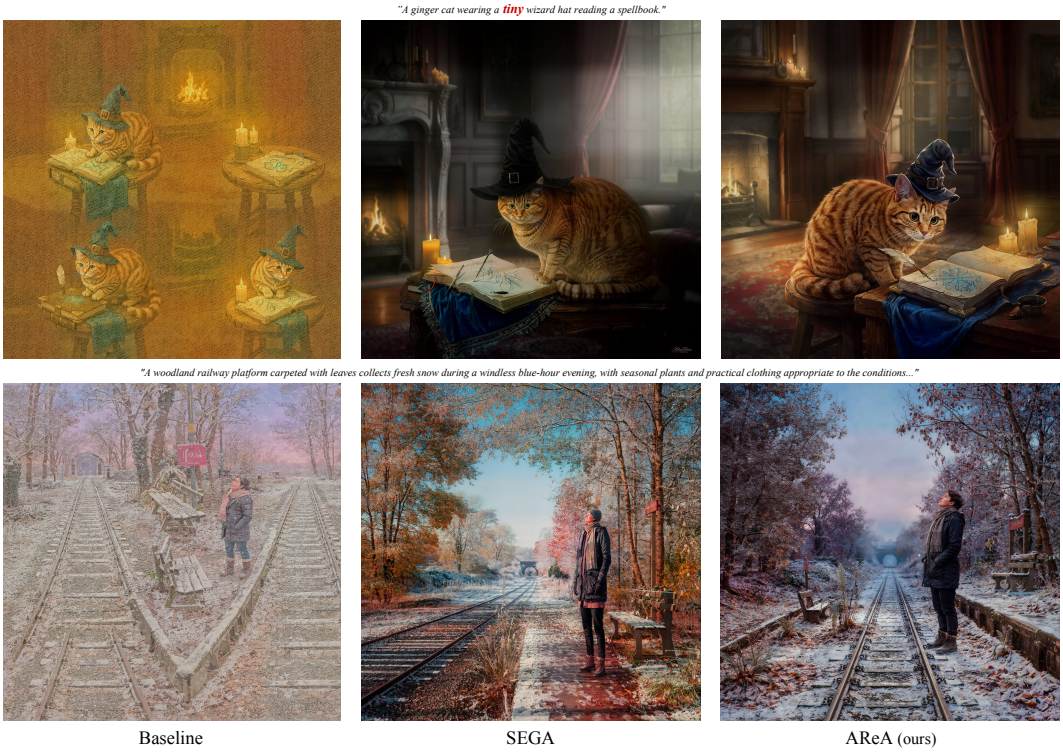

Figure 21: **Comparison on Ideogram 4 at 6144×6144.** Full attention completely collapses, while SEGA fails to keep the layout and add fine detail. AReA excels in both structure and fine detail, showing that it transfers to a different backbone without modification.