AViTS: Adaptive Spatiotemporal Token Selection for Efficient Dynamic-Resolution Generation

Haoran Qin[1,3⋆], Zhengan Yan[1⋆], Shikang Zheng[1⋆], Xiaobing Tu[2], Jiacheng Liu[1], Yuqi Lin[1,4], Chang Zou[1], JinShan Liu[5], Peiliang Cai[1], Xiantao Zhang[2], Jinkui Ren[2], and Linfeng Zhang[1†]

[1] Shanghai Jiao Tong University, China
[2] Terminal Intelligent Computing Division, Alibaba Cloud, China
[3] Shandong University, China
[4] Jilin University, China
[5] Xi'an Jiaotong University, China

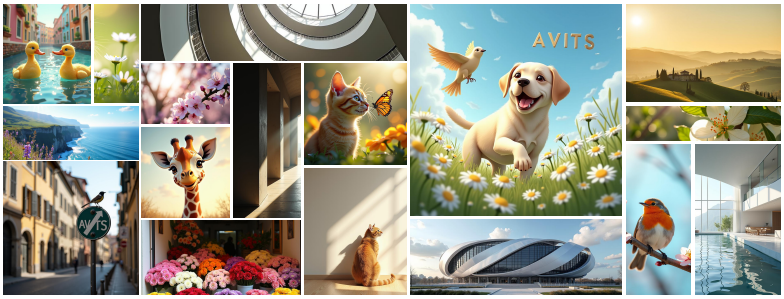

Fig. 1: Images sampled by FLUX.1-dev with AViTS with 6.34× acceleration.

Abstract. Diffusion Transformers (DiTs) achieve high-quality generation but are costly due to iterative sampling. Dynamic-resolution sampling reduces early-stage cost by denoising at low resolution; however, uniformly upsampling all latent tokens at resolution transitions incurs redundant computation and may degrade fine-detail consistency. Existing partial upsampling strategies typically rely on local latent structure cues or single-step statistics, making it difficult to jointly capture token–text semantic relevance and token-wise representation dynamics across diffusion steps. We propose **AViTS**, an adaptive spatiotemporal token selection framework for dynamic-resolution DiTs. AViTS models **spatial importance** via latent–text attention and **temporal importance** via token-level feature variation across diffusion timesteps, and fuses them to enable **spatiotemporal importance-aware selective upsampling**: it prioritizes resolution refinement for critical tokens while deferring less important ones, thereby reducing redundant high-resolution computation and improving the quality–efficiency trade-off. AViTS achieves up to **6.34**× on **FLUX** and nearly **9**× FLOPs reduction on **Qwen-Image-Edit** and **FLUX.1-Kontext-dev**, orthogonal to distillation, quantization, and feature caching, and reaching **14.76**× with distilled models. Code: `https://github.com/QHR69/AViTS`.

Keywords: Spatiotemporal token selection · Diffusion Acceleration

⋆ Equal contribution. † Corresponding author.

1 Introduction

Diffusion models have become a dominant paradigm for image and video generation and editing. Among them, Diffusion Transformers (DiTs) stand out for their scalability and strong performance in high-fidelity conditional generation and editing. However, DiT inference remains costly due to iterative sampling: generating or editing a single sample typically requires dozens of denoising steps, each passing through a large Transformer backbone, resulting in high latency. This challenge is further amplified at high resolutions, where the increased token count leads to quadratic growth in self-attention cost and memory footprint, hindering real-time deployment and use in resource-constrained settings.

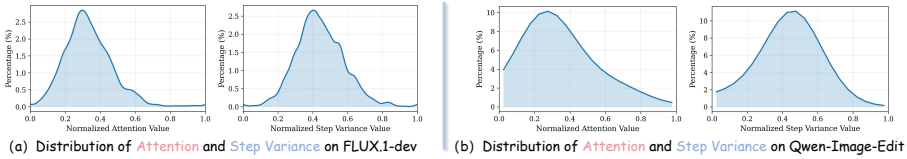

Fig. 2: Token-importance distributions. (a) Distributions of attention and step variance on FLUX.1-dev. (b) Distributions of attention and step variance on Qwen-Image-Edit.

To accelerate inference, existing approaches mainly follow two directions including reducing the number of sampling steps and reducing the computational cost of each step. Methods in the first category reduce the number of sampling steps through techniques such as distillation and consistency training. Methods in the second category reduce the per step computational cost through techniques such as sparse computation and feature caching [28, 29]. Recently, dynamic resolution sampling has emerged as a promising direction for spatial acceleration in DiTs. It performs denoising at a reduced resolution during early stages to save computation and then gradually transitions to higher resolution to recover fine details. Nevertheless, during resolution transitions many dynamic resolution strategies still upsample all latent tokens and perform high resolution computation uniformly. This process introduces substantial redundant computation and may also cause artifacts and inconsistency across stages. The recent development of partial upsampling further raises an important question. Under a limited high resolution computation budget how should latent tokens be prioritized so that computation focuses on the most critical content and achieves a better balance between efficiency and quality. However, existing partial upsampling methods usually determine priorities using internal latent cues such as spatial structure or local single step statistics through heuristic rules. These approaches do not explicitly model text conditioned semantic alignment or cross timestep representation dynamics [8, 27]. As a result they may fail to reliably identify and prioritize the truly critical tokens.

To characterize where token importance arises in low-resolution denoising, we conduct statistical analysis and visualization from two perspectives: seman-

tic alignment and cross-step dynamics. Specifically, we quantify semantic alignment by aggregating token–text cross-attention responses at low resolution, and quantify cross-step dynamics by measuring the magnitude of token representation changes across multiple sampling steps (step variance) in the DiT backbone. As shown in Fig. 2, both the attention-based spatial importance and the step-variance-based temporal importance exhibit pronounced long-tailed, highly non-uniform distributions: only a small fraction of tokens strongly correlate with the text condition, and only a small subset continue to evolve during denoising and remain more sensitive to subsequent updates. This pattern holds consistently on FLUX.1-dev (Fig. 2(a)) and Qwen-Image-Edit (Fig. 2(b)). The corresponding heatmaps further qualitatively suggest that attention tends to cover instruction-relevant semantic regions, whereas higher step variance often concentrates on regions requiring fine-grained modeling or modification (Fig. 3). These observations indicate that jointly modeling semantic alignment and cross-step dynamics enables more effective allocation of a limited high-resolution budget.

Fig. 3: Heatmaps of token importance on two text-to-image examples. Attention-based selection covers instruction-relevant semantic regions, whereas step-variance-based selection focuses on regions requiring fine-grained synthesis.

Motivated by this insight, we propose AViTS (Adaptive Spatiotemporal Token Selection), a spatiotemporal information-driven token selection framework for efficient dynamic-resolution generation. AViTS models token importance along two complementary dimensions: it estimates spatial importance via latent–text attention interactions to capture semantic alignment strength, and estimates temporal importance via token-wise feature variation across diffusion timesteps (step variance) to capture evolution and stability. AViTS then fuses these signals into a unified spatiotemporal importance score and performs importance-aware selective upsampling: it upsamples high-importance tokens earlier while deferring low-importance tokens, reducing redundant high-resolution computation while preserving critical semantics and details. Extensive experiments demonstrate that AViTS achieves a strong efficiency–quality trade-off across generation and editing (Fig. 1): it provides up to 6.34× acceleration on FLUX, nearly 9× FLOPs reduction on Qwen-Image-Edit and FLUX.1-Kontext-dev, and up to 14.76× when combined with distillation, while remaining compatible with quantization and feature caching. Our contributions are three-fold:

- **Spatiotemporal Heterogeneity Analysis.** From the perspectives of spatial semantic alignment and temporal cross-step dynamics, we reveal and validate pronounced spatiotemporal heterogeneity of latent-token importance during

low-resolution denoising, providing empirical grounding for prioritization in resolution transitions.
- **Spatiotemporal Importance Modeling.** We introduce **AViTS**, formulating upsampling prioritization as a spatiotemporal importance estimation problem jointly determined by semantic alignment and cross-step dynamics, and enabling spatiotemporal-importance-aware selective upsampling.
- **Broad Effectiveness And Composable Acceleration.** AViTS substantially reduces inference cost while preserving quality across multiple models and tasks, and composes effectively with distillation, quantization, and feature caching for further acceleration.

2 Related Works

Diffusion models have become a dominant paradigm for high-quality visual generation [21]. While early methods were built on U-Net backbones [18], recent systems increasingly adopt Diffusion Transformers (DiTs) for improved scalability [17]. However, the iterative denoising process incurs substantial inference cost, motivating extensive research on diffusion acceleration.

2.1 Model Compression–Based Acceleration

Model compression reduces inference cost by simplifying the network structure or numerical representation. Representative approaches include structural pruning [6], numerical quantization [11, 20], knowledge distillation, and token reduction strategies [2]. These techniques lower computation and memory usage with minimal modification to the inference pipeline. However, maintaining generation quality usually requires additional fine-tuning, and aggressive compression may degrade generalization under complex editing scenarios.

2.2 Temporal Acceleration

Another major direction reduces the number of denoising steps. Deterministic samplers such as DDIM [22] and higher-order solvers (e.g., DPM-Solver) improve the efficiency–quality trade-off by controlling numerical errors. Alternative formulations, including Rectified Flow, progressive distillation, and consistency models, further shorten denoising trajectories. Complementary training-free approaches exploit temporal redundancy by caching or forecasting intermediate features across timesteps [3, 14, 16]. Although effective, many existing caching methods rely on heuristic temporal assumptions and lack principled modeling of token-level dynamics.

2.3 Spatial Acceleration

Spatial acceleration reduces computation by lowering latent resolution, which in DiTs corresponds to decreasing the number of spatial tokens. Sparse attention patterns [4] provide limited gains, while coarse-to-fine strategies perform

early denoising at reduced resolution. Cascaded diffusion frameworks [7] follow this paradigm but require retraining. Recent training-free methods selectively upsample tokens during sampling. For instance, RALU [8] uses edge detection to identify important tokens, while Fresco [27] selects tokens based on inter-channel variance. However, these low-level heuristics are weakly aligned with editing semantics, often leading to unstable token selection and inconsistent denoising trajectories. Bottleneck Sampling [23] further adjusts resolution during inference but may introduce trajectory distortion and artifacts after upsampling.

3 Method

3.1 Preliminaries

Setup and token representation. We work with Diffusion Transformers (DiTs) that operate in latent space. An image of resolution $H \times W$ is encoded by a VAE into a compact spatial map and packed into a sequence of M non-overlapping patch tokens. The image token sequence is $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{M} \in \mathbb{R}^{M \times D}$, where $\mathbf{z}_i \in \mathbb{R}^D$ is the feature of the i-th patch at spatial coordinate $\mathbf{p}_i = (r_i, c_i)$. The text prompt is encoded into L tokens $\mathbf{C} = \{\mathbf{c}_j\}_{j=1}^{L} \in \mathbb{R}^{L \times D}$. In models with joint image–text attention, $\mathbf{Z}$ and $\mathbf{C}$ interact in a shared attention mechanism, producing cross-modal attention that we use for spatial importance in Sec. 3.3.

Flow matching. We adopt the flow-matching formulation. A velocity network $v_\theta(\mathbf{x}_t, t, \mathbf{c})$ is trained to regress the conditional velocity. At inference, we integrate the learned ODE from t=1 to t=0 via Euler steps:

$$\mathbf{x}_{t_{i+1}} \;=\; \mathbf{x}_{t_i} + v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})\, \Delta t_i, \qquad \Delta t_i = t_{i+1} - t_i < 0. \tag{1}$$

This update is used at each denoising step in our pipeline.

3.2 Framework Overview

AViTS organizes inference into three stages of progressively increasing spatial resolution (Figure 4(a)). Stage 1 performs low-resolution denoising and includes a feature-collection phase in its latter steps. This design differs from prior dynamic-resolution methods that lack an explicit multi-step, multimodal collection phase.

Stage 1: Low-resolution denoising and feature collection. We initialize with noise $\mathbf{Z}_{t_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at the target resolution $H \times W$, spatially downsample by factor f=2, and reduce the token count to $M' = M/f^2$. The velocity network v_θ performs N_1 Euler denoising steps via Eq. (1) to establish coarse global structure at low computational cost. Over the subsequent N_T denoising steps at the same reduced resolution, we record two complementary signals: the per-step latent snapshots $\{\mathbf{Z}^{(t_k)}\}_{k=1}^{N_T}$, which capture the temporal evolution of each token, and the cross-modal attention maps at each step, which encode the alignment between image tokens and the text condition. These signals feed the spatiotemporal importance estimator detailed in Sec. 3.3.

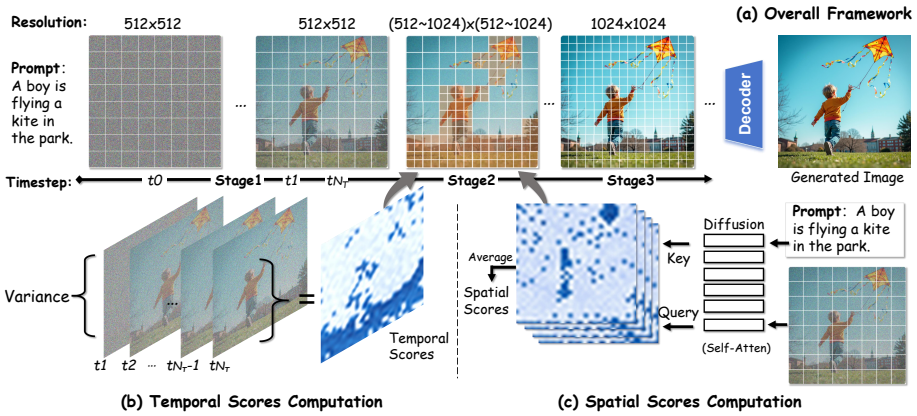

Fig. 4: Overview of AViTS and spatiotemporal importance estimation. (a) Three-stage dynamic-resolution sampling: Stage 1 low-res denoising with signal collection over the last N_T steps, Stage 2 selective upsampling of top-K tokens, and Stage 3 full-res refinement. **(b)** Temporal importance from token-wise step variance across the collected snapshots. **(c)** Spatial importance from aggregated latent–text cross-attention (averaged over heads/blocks and collection steps); fused scores guide prioritization.

Stage 2: Selective upsampling. Using the collected signals, AViTS computes a scalar importance score S_i for each of the M' tokens and selects the top-K subset $\mathcal{S}$ with $K = \lfloor \rho M' \rfloor$, $\rho \in (0, 1)$. Tokens in $\mathcal{S}$ are expanded via orthogonal upsampling; tokens in $\bar{\mathcal{S}}$ remain at the current resolution. We re-inject coordinate-bound noise into the mixed-resolution sequence and perform N_2 denoising steps.

Stage 3: Full-resolution refinement. The remaining tokens $\bar{\mathcal{S}}$ are expanded to complete the M-token sequence. After a final coordinate-bound noise re-injection and reordering by spatial coordinate, N_3 denoising steps recover fine details.

3.3 Spatiotemporal Token Importance Estimation

A central question in dynamic-resolution sampling is which tokens should be prioritized for upsampling under a limited budget. Existing partial upsampling methods typically assign priorities based on internal latent cues, such as spatial structure (e.g., edge detection on a single decoded frame) or per-channel or single-step feature statistics, without explicitly modeling text-conditioned semantic alignment and cross-timestep representation dynamics.

Figure 5 shows token importance heatmaps from our method, RALU [8], and Fresco [27] on an editing task. Ours concentrates on instruction-relevant editing regions; RALU emphasizes edges; Fresco yields more diffuse, less discriminative patterns. See the caption for a detailed analysis. These observations motivate us to formalize token importance as a function of both the text condition and the temporal trajectory of latent tokens.

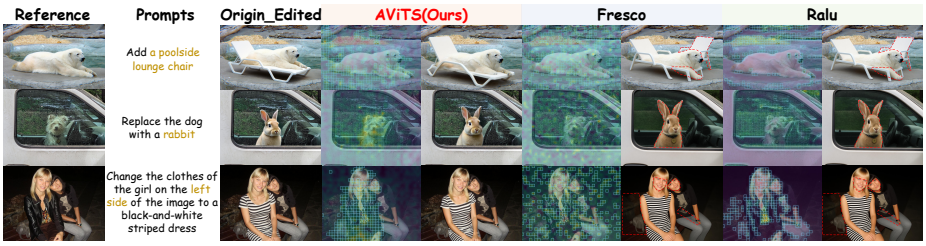

Fig. 5: Token-importance maps for editing. RALU/Fresco tend to allocate importance to irrelevant regions (edges or diffuse background), which leads to **semantic misalignment** (Row 1), degraded **color/style consistency** (Row 2), and **inconsistency in non-edited regions** (Row 3), whereas AViTS concentrates tokens on the instruction-relevant edit areas.

We decompose importance into a spatial component A_i (latent–text alignment, Sec. 3.3) and a temporal component V_i (cross-step dynamics, Sec. 3.3):

$$p_i \;=\; g\Big(\mathbf{z}_i,\; \mathbf{c},\; \big\{\mathbf{z}_i^{(t_k)}\big\}_{k=1}^{N_T}\Big) \;=\; g_{\text{sp}}\Big(\big\{\alpha_{i,j}^{(h)}\big\}\Big) \;+\; g_{\text{tmp}}\Big(\big\{\mathbf{z}_i^{(t_k)}\big\}_{k=1}^{N_T}\Big), \tag{2}$$

where $\{\alpha_{i,j}^{(h)}\}$ are the cross-modal attention weights and $\{\mathbf{z}_i^{(t_k)}\}$ are the latent snapshots recorded during the latter steps of Stage 1. We detail each component.

Spatial Importance via Latent–Text Alignment Tokens strongly aligned with the text condition carry conditional semantics and benefit from early high-resolution processing. We measure this via the cross-modal attention from image tokens to text tokens.

Let $\alpha_{i,j}^{(h)}$ be the attention weight from image token $\mathbf{z}_i$ to text token $\mathbf{c}_j$ in head h. The spatial importance of token i is

$$A_i \;=\; \frac{1}{|\mathcal{H}|} \sum_{h \,\in\, \mathcal{H}} \sum_{j=1}^{L} \alpha_{i,j}^{(h)}. \tag{3}$$

The computation of $\{\alpha_{i,j}^{(h)}\}$ depends on whether the backbone uses a multimodal large language model (MLLM) or a DiT with joint image–text blocks.

MLLM-based extraction. Models built on MLLMs (e.g., Qwen-Image-Edit) encode image patches and text in a single unified sequence. The attention matrices at the MLLM layers directly provide an image-to-text submatrix $\boldsymbol{\alpha}^{(h)} \in \mathbb{R}^{M' \times L}$ per head h. We extract this submatrix from a selected layer, aggregate over heads, and sum over the text dimension via Eq. (3) to obtain A_i.

Joint-attention extraction. Models without an MLLM (e.g., FLUX-family) use *double-stream* blocks that compute joint self-attention over the concatenated sequence $[\mathbf{C}; \mathbf{Z}]$. The joint query and key are

$$\mathbf{Q} \;=\; \big[\mathbf{Q}_{\text{txt}} \,\|\, \mathbf{Q}_{\text{img}}\big], \qquad \mathbf{K} \;=\; \big[\mathbf{K}_{\text{txt}} \,\|\, \mathbf{K}_{\text{img}}\big], \tag{4}$$

where $\mathbf{Q}_{\text{img}}, \mathbf{K}_{\text{img}}$ and $\mathbf{Q}_{\text{txt}}, \mathbf{K}_{\text{txt}}$ are the image and text projections. The full attention matrix (before softmax) is $\mathbf{P} = \mathbf{Q}\mathbf{K}^\top/\sqrt{d_h}$. Because fused attention kernels do not expose weights, we register a forward hook on each double block to obtain $\mathbf{Q}$, $\mathbf{K}$, apply the same QK-normalisation as the model, and reconstruct

$$\boldsymbol{\alpha}^{(h)} \;=\; \big[\mathrm{softmax}\big(\tilde{\mathbf{Q}}^{(h)}\tilde{\mathbf{K}}^{(h)\top}/\sqrt{d_h}\big)\big]_{\text{img}\to\text{txt}}, \tag{5}$$

where $\tilde{\mathbf{Q}}^{(h)}$, $\tilde{\mathbf{K}}^{(h)}$ are ℓ_2-normalised and the subscript selects rows for image queries and columns for text keys. We aggregate scores across selected blocks and the N_T collection steps via Eq. (3).

Temporal Importance via Cross-Step Feature Dynamics The spatial score A_i does not capture how actively a token evolves. A token whose feature changes substantially across denoising steps is still under construction and benefits from early high-resolution refinement.

We stack the N_T latent snapshots $\{\mathbf{Z}^{(t_k)}\}_{k=1}^{N_T}$ and compute, for each token i and channel d, the unbiased sample variance across steps. Let $\bar{z}_{i,d}$ denote the temporal mean:

$$\bar{z}_{i,d} \;=\; \frac{1}{N_T}\sum_{k=1}^{N_T} z_{i,d}^{(t_k)}. \tag{6}$$

The temporal importance of token i is the mean over channels of these variances:

$$V_i \;=\; \frac{1}{D}\sum_{d=1}^{D}\frac{1}{N_T-1}\sum_{k=1}^{N_T}\big(z_{i,d}^{(t_k)} - \bar{z}_{i,d}\big)^2. \tag{7}$$

Because V_i measures cross-step intra-latent dynamics, it is orthogonal to the cross-modal signal A_i.

Joint Score and Token Selection Both scores are normalised to $[0,1]$ via min–max: $\hat{A}_i = (A_i - \min_i A_i)/(\max_i A_i - \min_i A_i + \varepsilon)$ and analogously $\hat{V}_i$. The joint importance score is

$$S_i \;=\; \alpha\,\hat{A}_i \;+\; (1-\alpha)\,\hat{V}_i, \qquad \alpha \in [0,1]. \tag{8}$$

We select the top-K tokens:

$$\mathcal{S} \;=\; \mathrm{TopK}\big(\{S_i\}_{i=1}^{M'},\; K{=}\lfloor \rho M' \rfloor\big). \tag{9}$$

A small Gaussian perturbation (on the order of 10^{-6}) is applied before sorting to break ties. Orthogonal upsampling and coordinate-bound noise re-injection follow prior practice to preserve cross-stage consistency.

4 Experiment

4.1 Experiment Settings

Model Configurations. We conduct experiments on three diffusion-based models: the text-to-image model FLUX.1-dev [9] and two representative diffusion-based editing models, **Qwen-Image-Edit** [25] and **FLUX.1-Kontext-dev** [10]. All experiments are conducted on NVIDIA A800 GPUs for FLUX.1-dev, and on NVIDIA H20 GPUs for Qwen-Image-Edit and FLUX.1-Kontext-dev.

We compare AViTS with a diverse set of acceleration baselines, attention sparsification methods such as SpargeAttention [26], feature caching approaches including TeaCache [12], token reuse and forecasting methods such as ToCa [28], DuCa [29], FORA [19], FreqCa [13], and TaylorSeer [15], as well as spatial resolution scheduling strategies including Bottleneck Sampling [23], RALU [8], and Fresco [27]. More details are provided in the supplementary materials.

Datasets and Metrics. Experiments use two representative benchmarks for text-to-image generation and instruction-based image editing. For generation evaluation, we adopt the **DrawBench** benchmark, where the generated samples are assessed using **ImageReward** and **CLIP Score** to measure image quality and text–image semantic alignment. These metrics jointly evaluate whether the generated images faithfully reflect the input prompt while maintaining high perceptual realism. For image editing evaluation, we use the **GEdit** benchmark, which evaluates instruction-driven editing fidelity and alignment to target modifications under textual and visual guidance. Editing quality is measured using **Semantic Consistency (SC)**, **Perceptual Quality (PQ)**, and **Overall Score (OS)**, which together reflect semantic correctness, visual realism, and overall editing performance. Computational efficiency is evaluated by the **number of function evaluations (NFE)**, **latency**, **speedup**, and **FLOPs**.

4.2 Results on Text-to-Image Generation

As shown in Table 1, AViTS consistently improves the efficiency–quality trade-off on FLUX.1-dev across a wide range of acceleration regimes. Under the moderate acceleration setting with NFE=30, AViTS reduces latency to **8.21**s (**3.14**×) with **2.65**× FLOPs reduction, achieving the lowest latency among competing methods such as Bottleneck Sampling, RALU, and Fresco. Notably, AViTS also delivers the best generation quality, obtaining the highest ImageReward (**1.0104**) and CLIP score (**32.476**), surpassing the original FLUX baseline despite using significantly fewer effective computations.

When the sampling budget is further reduced to NFE=18, AViTS continues to demonstrate strong robustness under more aggressive acceleration, reaching **4.73**s latency with **5.45**× speedup and **4.80**× FLOPs reduction while maintaining high visual quality (**0.9959** ImageReward and **32.361** CLIP score). These results indicate that the proposed token prioritization strategy effectively preserves semantically important structures even when the available computation

Table 1: Quantitative comparison of text-to-image results on FLUX.1-dev.

Method	NFE	Acceleration				Image Reward ↑	CLIP Score ↑
		Latency (s) ↓	Speed ↑	FLOPs(T) ↓	Speed ↑		
FLUX.1-dev	50	25.78	1.00×	3719.50	1.00×	0.9719 (+0.00%)	32.325 (+0.00%)
60% steps	30	16.63	1.55×	2231.70	1.67×	0.9646 (-0.75%)	32.232 (-0.28%)
SpargeAttention	50	15.44	1.67×	2150.02	1.73×	0.9795 (+0.78%)	32.312 (-0.04%)
TeaCache ($l = 0.25$)	50	14.09	1.83×	1937.24	1.92×	0.9442 (-2.85%)	32.167 (-0.49%)
TaylorSeer ($\mathcal{N} = 3$)	50	9.88	2.61×	1320.07	2.82×	0.9861 (+1.46%)	32.313 (-0.04%)
Bottleneck Sampling	30	11.31	2.28×	1582.77	2.35×	0.9721 (+0.02%)	32.141 (-0.57%)
RALU	30	11.02	2.34×	1499.79	2.48×	0.9626 (-0.96%)	32.118 (-0.64%)
Fresco	30	9.17	2.81×	**1295.99**	**2.87**×	0.9801 (+0.84%)	32.125 (-0.62%)
AViTS	30	**8.21**	**3.14**×	1401.03	2.65×	**1.0104** (+3.96%)	**32.476** (+0.47%)
36% steps	18	9.88	2.61×	1339.02	2.77×	0.9553 (-1.71%)	32.114 (-0.65%)
ToCa ($\mathcal{N} = 6$)	50	13.15	1.96×	924.30	4.02×	0.9702 (-0.17%)	32.083 (-0.75%)
DuCa ($\mathcal{N} = 5$)	50	8.19	3.15×	978.76	3.80×	0.9855 (+1.40%)	32.241 (-0.26%)
TeaCache ($l = 0.8$)	50	6.63	3.89×	892.35	4.17×	0.8805 (-9.40%)	31.827 (-1.54%)
TaylorSeer ($\mathcal{N} = 4$)	50	9.21	2.80×	967.91	3.84×	0.9857 (+1.42%)	**32.413** (+0.27%)
RALU	18	6.33	4.07×	904.98	4.11×	0.9481 (-2.45%)	32.074 (-0.78%)
Fresco	18	5.72	4.51×	788.03	4.72×	0.9861 (+1.46%)	31.970 (-1.10%)
AViTS	18	**4.73**	**5.45**×	**774.74**	**4.80**×	**0.9959** (+2.47%)	32.361 (+0.11%)
ToCa ($\mathcal{N} = 10$)	50	7.93	3.25×	714.66	5.20×	0.7055 (-27.41%)	31.808 (-1.60%)
DuCa ($\mathcal{N} = 9$)	50	7.26	3.55×	690.25	5.39×	0.8182 (-15.81%)	31.759 (-1.75%)
TeaCache ($l = 1.6$)	50	3.78	6.82×	520.54	7.15×	0.6423 (-33.91%)	31.656 (-2.07%)
TaylorSeer ($\mathcal{N} = 9$)	50	4.85	5.32×	596.07	6.24×	0.8562 (-11.90%)	31.653 (-2.08%)
FLUX.1-schnell	8	4.21	6.12×	595.12	6.25×	0.9097 (-6.40%)	**33.837** (+4.68%)
RALU	10	3.72	6.92×	540.62	6.88×	0.9289 (-4.42%)	32.113 (-0.66%)
Fresco	11	3.43	7.52×	486.85	7.64×	0.9366 (-3.63%)	31.897 (-1.32%)
AViTS	14	3.65	7.06×	586.56	6.34×	**0.9723** (+0.04%)	32.352 (+0.08%)
AViTS	11	3.10	8.32×	486.85	7.64×	0.9423 (-3.05%)	31.959 (-1.13%)
AViTS	9	**2.64**	**9.78**×	**380.30**	**9.78**×	0.9201 (-5.33%)	32.169 (-0.48%)

is significantly reduced. In contrast, spatial scheduling methods such as RALU and Fresco show noticeable quality degradation at similar speeds, suggesting that heuristic spatial upsampling strategies struggle to maintain consistent generation fidelity under stronger acceleration. AViTS maintains stable generation quality while achieving up to **9.78**× latency acceleration, demonstrating a more

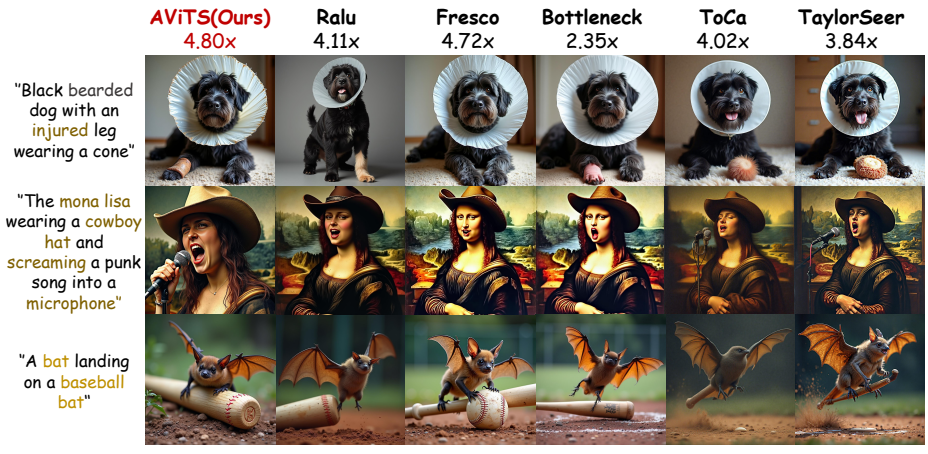

Fig. 6: Visualization of the image generated by different methods on FLUX.1-dev. AViTS achieves the best semantic fidelity and the fastest speed (4.80×), surpassing all dynamic-resolution and feature-caching baselines.

Table 2: Quantitative evaluation results of image editing on the GEdit-bench with **FLUX.1-Kontext-dev**.

Method	NFE	Acceleration				GEdit-EN(FULL)		
		Latency (s) ↓	Speed ↑	FLOPs (T) ↓	Speed ↑	SC ↑	PQ ↑	OS ↑
100% steps	50	50.20	1.00×	8299.54	1.00×	**6.80**	7.26	**6.51**
60% steps	30	32.23	1.56×	4979.72	1.67×	6.54	**7.28**	6.25
20% steps	10	**10.47**	**4.79**×	**1659.91**	**5.00**×	6.60	7.18	6.28
SpargeAttention	50	46.05	1.09×	5603.68	1.48×	6.46	**7.25**	6.21
RALU	30	23.34	2.15×	2730.53	3.04×	7.06	7.20	6.70
Bottleneck	30	18.25	2.75×	2727.10	3.04×	6.46	6.63	6.08
Fresco	50	20.48	2.45×	2361.22	3.51×	7.02	7.12	6.65
AViTS	30	**15.63**	**3.21**×	**2139.98**	**3.88**×	**7.08**	7.12	**6.71**
RALU	18	15.30	3.28×	2123.74	3.91×	7.05	7.17	6.69
Bottleneck	18	15.21	3.30×	2274.58	3.65×	6.86	6.77	6.43
ToCa ($\mathcal{N}=8$)	50	29.56	1.70×	1841.35	4.51×	6.43	7.25	6.12
TaylorSeer ($\mathcal{N}=6$)	50	13.95	3.60×	1660.95	5.00×	6.47	**7.29**	6.17
Fresco	50	17.73	2.83×	1878.83	4.42×	7.01	7.15	6.66
AViTS	18	**11.51**	**4.36**×	**1544.45**	**5.37**×	**7.07**	7.12	**6.70**
ToCa ($\mathcal{N}=12$)	50	20.72	2.42×	1359.61	6.10×	6.39	6.91	6.04
TaylorSeer ($\mathcal{N}=9$)	50	12.05	4.17×	1329.02	6.24×	6.40	**6.99**	6.07
AViTS	11	**7.25**	**6.92**×	**937.92**	**8.85**×	**7.10**	6.98	**6.57**

effective allocation of high-resolution computation to semantically important tokens and enabling a better efficiency–quality trade-off across acceleration regimes (see Fig. 6 for qualitative comparisons).

4.3 Results on Image Editing

Results on FLUX.1-Kontext-dev

Moderate acceleration regime. As shown in Table 2, in this regime around **3**×, AViTS achieves a strong balance between efficiency and editing quality. Compared with RALU which reaches **2.15**× speedup, AViTS improves the acceleration to **3.21**× while maintaining better editing performance. On GEdit-EN, AViTS obtains the highest semantic consistency with SC **7.08** and the best overall score of **6.71**. These results indicate that AViTS improves efficiency while preserving reliable semantic alignment.

High and extreme acceleration regimes. Under higher acceleration around **4**×, AViTS continues to maintain stable editing quality compared with existing methods. TaylorSeer reaches **3.60**× speedup but its overall score decreases to 6.17. In contrast, AViTS achieves **4.36**× speedup with stronger editing performance (OS **6.70**), indicating better allocation of high-resolution computation to important regions. This suggests importance-aware refinement preserves critical structures even under limited computation.

When the acceleration further increases beyond **5**×, AViTS achieves **6.92**× speedup and **8.85**× FLOPs reduction while maintaining strong editing quality

Table 3: Quantitative evaluation results of image editing on the GEdit-bench with **Qwen-Image-Edit**.

Method	NFE	Acceleration				GEdit-CN(FULL)			GEdit-EN(FULL)		
		Latency (s) ↓	Speed ↑	FLOPs (T) ↓	Speed ↑	SC ↑	PQ ↑	OS ↑	SC ↑	PQ ↑	OS ↑
100% steps	50	284.51	1.00×	28219.71	1.00×	7.68	7.51	7.41	**7.82**	**7.54**	**7.54**
60% steps	30	172.43	1.65×	16931.83	1.67×	**7.70**	**7.53**	**7.44**	7.77	7.52	7.47
20% steps	**10**	**58.66**	**4.85**×	**5638.18**	**5.01**×	7.65	7.42	7.35	7.73	7.46	7.44
SpargeAttention	50	231.30	1.23×	16846.78	1.67×	7.87	7.57	7.56	7.81	7.53	7.50
Bottleneck	30	109.38	2.60×	9954.36	2.83×	7.44	**7.59**	7.28	7.62	7.40	7.24
RALU	30	105.87	2.69×	9286.52	3.04×	7.83	7.58	7.55	7.83	7.52	7.52
ToCa ($\mathcal{N}=6$)	50	172.43	1.65×	7850.13	3.59×	**7.89**	7.50	**7.57**	7.89	7.46	7.54
AViTS	30	**87.54**	**3.25**×	**7581.38**	**3.72**×	**7.89**	7.54	**7.57**	**7.95**	**7.54**	**7.62**
RALU	18	66.94	4.25×	6200.73	4.55×	7.89	**7.56**	**7.60**	7.82	**7.52**	7.51
Bottleneck	18	85.95	3.31×	8090.07	3.48×	7.75	7.52	7.45	7.70	7.46	7.39
FORA ($\mathcal{N}=5$)	50	**63.15**	**4.51**×	5643.13	5.00×	7.60	7.31	7.25	7.62	7.34	7.28
DuCa ($\mathcal{N}=7$)	50	69.54	4.09×	5699.89	4.95×	7.73	7.44	7.44	7.80	7.40	7.45
TaylorSeer ($\mathcal{N}=6$)	50	65.66	4.33×	5643.13	5.00×	7.53	7.40	7.25	7.60	7.37	7.30
AViTS	**18**	63.36	4.49×	**5382.48**	**5.24**×	**7.91**	7.53	7.58	**7.87**	**7.52**	**7.55**
FORA ($\mathcal{N}=7$)	50	52.20	5.45×	4515.74	6.25×	7.42	7.13	7.06	7.43	7.19	7.06
FreqCa ($\mathcal{N}=9$)	50	51.09	5.57×	4514.48	6.25×	7.62	7.18	7.27	7.66	7.12	7.21
TaylorSeer ($\mathcal{N}=9$)	50	53.92	5.28×	4515.74	6.25×	6.61	6.65	6.31	6.67	6.63	6.31
AViTS	**11**	**40.93**	**6.95**×	**3262.43**	**8.65**×	**7.90**	**7.47**	**7.57**	**7.80**	**7.48**	**7.48**

with OS **6.57**. Despite the substantial reduction in computation, the editing results remain stable across different prompts and image contents. This observation indicates that AViTS effectively preserves key semantic structures during the denoising process while avoiding unnecessary high-resolution computation.

Results on Qwen-Image-Edit

Moderate acceleration regime. As shown in Table 3, around 3×, AViTS balances efficiency and editing quality. Compared with RALU which reaches 2.69× speedup, AViTS increases the acceleration to 3.25× while maintaining better

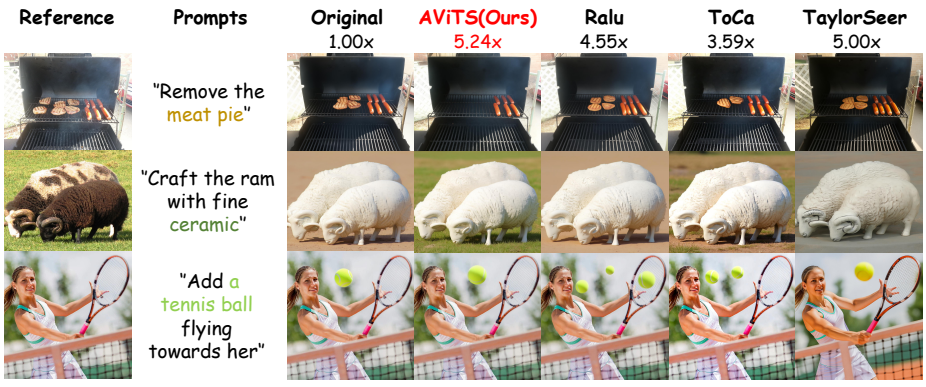

Fig. 7: Qualitative comparison on GEdit-Bench with Qwen-Image-Edit. AViTS achieves superior semantic preservation and visual quality at a higher acceleration ratio (**5.24**×) compared to state-of-the-art baselines.

Table 4: Quantitative comparison of text-to-image generation with other acceleration methods.

Method	NFE	Acceleration				CLIP-IQA ↑	CLIP Score ↑
		Latency(s) ↓	Speed ↑	FLOPs(T) ↓	Speed ↑		
FLUX.1-dev [9]	50	25.78	1.00×	3719.50	1.00×	**0.8494** (+0.00%)	**33.402** (+0.00%)
FreqCa(N=4) [13]	50	8.38	3.08×	1116.32	3.33×	0.8472 (-0.26%)	32.255 (-3.43%)
AViTS+Feature Caching	18	**3.02**	**8.54**×	**412.26**	**9.02**×	0.8217 (-3.26%)	32.47 (-2.79%)
FLUX.1-lite-8B [24]	28	9.16	2.81×	1291.49	2.88×	**0.8617** (+1.45%)	**32.52** (-2.64%)
AViTS+Model Distillation	14	**3.09**	**8.34**×	**438.76**	**8.48**×	0.8484 (-0.12%)	31.87 (-4.59%)
FLUX.1-schnell [1]	8	4.20	6.14×	595.12	6.25×	0.8587 (+1.09%)	33.69 (+0.86%)
FLUX.1-schnell [1]	6	3.49	7.39×	439.34	8.47×	0.8703 (+2.46%)	**33.75** (+1.04%)
AViTS+Step Distillation	8	2.55	10.11×	362.35	10.26×	**0.8784** (+3.41%)	31.78 (-4.86%)
AViTS+Step Distillation	6	**1.76**	**14.65**×	**252.00**	**14.76**×	0.8709 (+2.53%)	32.20 (-3.60%)
FLUX.1-dev-int8 [5]	50	14.01	1.84×	1888.07	1.97×	0.8498 (+0.05%)	**33.51** (+0.32%)
AViTS+Quantization	30	5.28	4.88×	752.52	4.94×	0.8542 (+0.57%)	32.25 (-3.45%)
AViTS+Quantization	18	**2.90**	**8.89**×	**413.40**	**9.00**×	**0.8723** (+2.70%)	32.08 (-3.96%)

editing performance. On GEdit-EN, AViTS improves the overall score from 7.52 to **7.62** and achieves the highest semantic consistency with SC **7.95**. These results indicate that AViTS can further reduce computation while preserving semantic fidelity and perceptual quality (see Fig. 7).

High and extreme acceleration regimes. Under higher acceleration around 4 to 5×, AViTS continues to maintain stable editing quality compared with existing acceleration methods. FORA achieves a similar speedup of 4.51× but its editing performance drops to OS 7.25 on GEdit-CN and 7.28 on GEdit-EN. In contrast, AViTS achieves 4.49× speedup while maintaining stronger semantic alignment with OS **7.58** and **7.55**. When the acceleration further increases beyond 5×, methods such as FreqCa reach 5.57× speedup but suffer from clear quality degradation. AViTS achieves 6.95× speedup and 8.65× FLOPs reduction while maintaining strong editing performance with OS **7.57** and **7.48**. These results demonstrate that AViTS scales robustly under aggressive acceleration while preserving stable editing quality.

4.4 Compatibility with Acceleration Methods

Table 4 together with Figure 8 shows that AViTS is orthogonal to other acceleration axes and can be combined with different compression strategies. Without additional training, AViTS combined with feature caching achieves up to **9**× acceleration at reduced NFE while maintaining coherent and high-fidelity generations. AViTS also remains effective on compressed trajectories. When paired with step distillation, it reaches up to **14.76**× acceleration while preserving plausible structures and key visual details. Moreover, AViTS composes naturally with quantization such as FLUX.1-dev-int8, providing around **9**× speedup while maintaining or slightly improving perceptual quality measured by **CLIP-IQA**. Overall, these results demonstrate that AViTS serves as a plug-and-play token prioritization module that consistently improves the efficiency–quality trade-off when combined with caching, distillation, and quantization.

Table 5: Quantitative comparison of text-to-image gen. on FLUX.1-dev.

Method	NFE	Acceleration		Image Reward ↑	CLIP Score ↑
		Latency(s) ↓	Speed ↑		
FLUX.1-dev	50	25.78	1.00×	0.9719 (+0.00%)	32.325 (+0.00%)
Random	30	5.25	4.91×	0.9257 (-4.75%)	31.488 (-2.59%)
Evenly	30	5.25	4.91×	0.9689 (-0.31%)	32.012 (-0.97%)
Edge detection	30	5.76	4.48×	0.9512 (-2.13%)	32.074 (-0.78%)
Fresco	30	5.72	4.51×	0.9861 (+1.46%)	31.970 (-1.10%)
AViTS (only attention)	30	4.72	5.46×	0.9875 (+1.60%)	32.251 (-0.23%)
AViTS (only step-variance)	30	**4.67**	**5.52**×	0.9846 (+1.31%)	32.202 (-0.38%)
AViTS (mix)	30	4.73	5.45×	**0.9959** (+2.47%)	**32.361** (+0.11%)

Fig. 8: AViTS composes with feature caching and step distillation for further speedup.

4.5 Ablation Study

We conduct an ablation on FLUX.1-dev under the *same* compute budget (NFE = 30) and the *same* upsampling ratio ρ (Table 5), isolating the effect of *which* tokens are prioritized. The training-free allocations, including *Random* and *Evenly*, achieve similar speedups ($\sim$4.9×) but cause clear drops in ImageReward and CLIP Score. Low-level heuristics such as *Edge detection* and *Fresco* remain suboptimal under the same ρ, showing limited alignment with text semantics and reduced CLIP Score. In contrast, our proposed importance cues are both effective. Using **Attention** or **Step-variance** alone already reaches $\sim$5.5× acceleration with better quality retention, where step-variance is slightly faster while attention better preserves conditional semantics. Finally, **AViTS (mix)** achieves the best trade-off at fixed ρ, reaching 5.45× speedup with the highest ImageReward (0.9959) and CLIP Score (32.361), validating the complementarity between semantic alignment and cross-step dynamics.

5 Conclusion

We propose **AViTS**, a spatiotemporal token selection framework for dynamic-resolution DiTs. AViTS fuses *latent–text attention* (spatial importance) and *cross-step feature variation* (temporal importance) to prioritize critical tokens for early refinement, reducing redundant high-resolution computation while preserving quality. Experiments on DrawBench and GEdit show strong speedups with stable fidelity on FLUX.1-dev, FLUX.1-Kontext-dev, and Qwen-Image-Edit, and AViTS composes well with distillation, quantization, and feature caching.

Acknowledgements

This paper was partially sponsored by Terminal Intelligent Computing Division - Alibaba Cloud.

References

1. Black Forest Labs: Flux.1-schnell. https://huggingface.co/black-forest-labs/FLUX.1-schnell (2024), hugging Face model card
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4599–4603 (2023)
3. Cai, P., Liu, J., Xu, H., Wang, X., Zou, C., Zhang, L.: Lesa: Learnable stage-aware predictors for diffusion model acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 43300–43309 (2026)
4. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)
5. Diffusers: Flux.1-dev-torchao-int8. https://huggingface.co/diffusers/FLUX.1-dev-torchao-int8 (2024), hugging Face model card; quantized checkpoint derived from black-forest-labs/FLUX.1-dev
6. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. arXiv preprint arXiv:2305.10924 (2023)
7. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. Journal of Machine Learning Research **23**(47), 1–33 (2022)
8. Jeong, W., Lee, K., Seo, H., Chun, S.Y.: Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers. arXiv preprint arXiv:2507.08422 (2025)
9. Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024)
10. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), https://arxiv.org/abs/2506.15742
11. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17535–17545 (2023)
12. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It's time to cache for video diffusion model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7353–7363 (2025)
13. Liu, J., Cai, P., Zhou, Q., Lin, Y., Kong, D., Huang, B., Pan, Y., Xu, H., Zou, C., Tang, J., Zheng, S., Zhang, L.: Freqca: Accelerating diffusion models via frequency-aware caching (2025), https://arxiv.org/abs/2510.08669
14. Liu, J., Wang, X., Lin, Y., Wang, Z., Wang, P., Cai, P., Zhou, Q., Yan, Z., Yan, Z., Shi, Z., et al.: A survey on cache methods in diffusion models: Toward efficient multi-modal generation. arXiv preprint arXiv:2510.19755 (2025)
15. Liu, J., Zou, C., Lyu, Y., Chen, J., Zhang, L.: From reusing to forecasting: Accelerating diffusion models with taylorseers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15853–15863 (2025)

16. Liu, J., Zou, C., Lyu, Y., Li, K., Wang, S., Zhang, L.: Speca: Accelerating diffusion transformers with speculative feature caching. In: Proceedings of the 33rd ACM International Conference on Multimedia (2025)
17. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
19. Selvaraju, P., Ding, T., Chen, T., Zharkov, I., Liang, L.: Fora: Fast-forward caching in diffusion transformer acceleration. arXiv preprint arXiv:2407.01425 (2024)
20. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1972–1981 (2023)
21. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. pmlr (2015)
22. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
23. Tian, Y., Xia, X., Ren, Y., Lin, S., Wang, X., Xiao, X., Tong, Y., Yang, L., Cui, B.: Training-free diffusion acceleration with bottleneck sampling. arXiv preprint arXiv:2503.18940 (2025)
24. Verdú, D., Martín, J.: Flux.1 lite: Distilling flux1.dev for efficient text-to-image generation (2024)
25. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
26. Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., Chen, J.: Spargeattn: Accurate sparse attention accelerating any model inference. In: Proceedings of the International Conference on Machine Learning (ICML) (2025)
27. Zheng, S., Chen, G., He, L., Liu, J., Lin, Y., Zou, C., Zhang, L.: From sketch to fresco: Efficient diffusion transformer with progressive resolution. arXiv preprint arXiv:2601.07462 (2026)
28. Zou, C., Liu, X., Liu, T., Huang, S., Zhang, L.: Accelerating diffusion transformers with token-wise feature caching. arXiv preprint arXiv:2410.05317 (2024)
29. Zou, C., Zhang, E., Guo, R., Xu, H., He, C., Hu, X., Zhang, L.: Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching. arXiv preprint arXiv:2412.18911 (2024)