

Region-Adaptive Sampling for Diffusion Transformers

Ziming Liu*

National University of Singapore

liuziming@comp.nus.edu.sg

Yifan Yang*

Microsoft Research

yifanyang@microsoft.com

Chengruidong Zhang

Microsoft Research

chengzhang@microsoft.com

Yiqi Zhang

National University of Singapore

yiqi.zhang@comp.nus.edu.sg

Lili Qiu

Microsoft Research

liliqiu@microsoft.com

Yang You

National University of Singapore

youy@comp.nus.edu.sg

Yuqing Yang

Microsoft Research

yuqyang@microsoft.com

Abstract

*Diffusion models (DMs) have achieved state-of-the-art performance across diverse generative tasks, yet their dependence on sequential forward passes fundamentally limits real-time efficiency. Existing acceleration methods primarily focus on reducing the number of sampling steps or reusing intermediate features. Leveraging the inherent flexibility of Diffusion Transformers (DiTs) in handling variable token counts, we introduce **RAS**, a training-free sampling strategy that dynamically adjusts update ratios across image regions based on model attention. Our key insight is that DiTs progressively focus on semantically meaningful regions, and such focused areas exhibit strong temporal continuity between consecutive steps. Building on this observation, RAS updates only these focused regions while reusing cached noise elsewhere, with focus maps inferred from the previous step’s output. Experiments on Stable Diffusion 3 and Lumina-Next-T2I demonstrate up to $2.36\times$ and $2.51\times$ speedups, respectively, with negligible quality degradation—highlighting a practical pathway toward real-time diffusion transformer generation.*

1. Introduction

Diffusion models (DMs) [8, 20, 44, 45] have emerged as highly effective probabilistic generative models, producing high-quality data across a wide range of domains. They have been successfully applied to image synthesis [7, 38], image super-resolution [14, 27, 57], image-to-image translation [26, 41, 50], image editing [24, 58], inpainting [32],

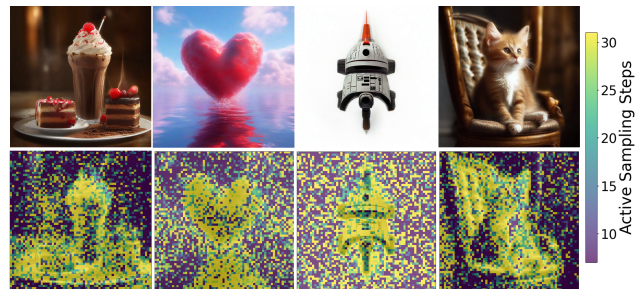


Figure 1. The main subject and the regions with more details are brushed for more steps than other regions in RAS. Each block represents a patchified latent token.

video synthesis [3, 9], text-to-3D generation [36], and even planning tasks [22]. Despite their impressive generative capabilities, sampling from DMs requires solving a stochastic or ordinary differential equation (SDE/ODE) [17, 37] in reverse time, which entails multiple sequential forward passes through a large neural network. This inherent sequential dependency significantly limits their real-time applicability.

Considerable effort has been devoted to accelerating the sampling process in diffusion models (DMs) by reducing the number of sampling steps. Existing approaches can be broadly categorized into training-based methods, such as progressive distillation [42], consistency models [46], and rectified flow [1, 29, 30], and training-free methods, including DPM-Solver [31], AYS [40], DeepCache [55], and Delta-DiT [5]. However, these techniques process all regions of an image uniformly during sampling, regardless of regional content complexity or semantic importance.

Intuitively, different regions within an image exhibit varying levels of structural and semantic complexity: intri-

*Equal contribution

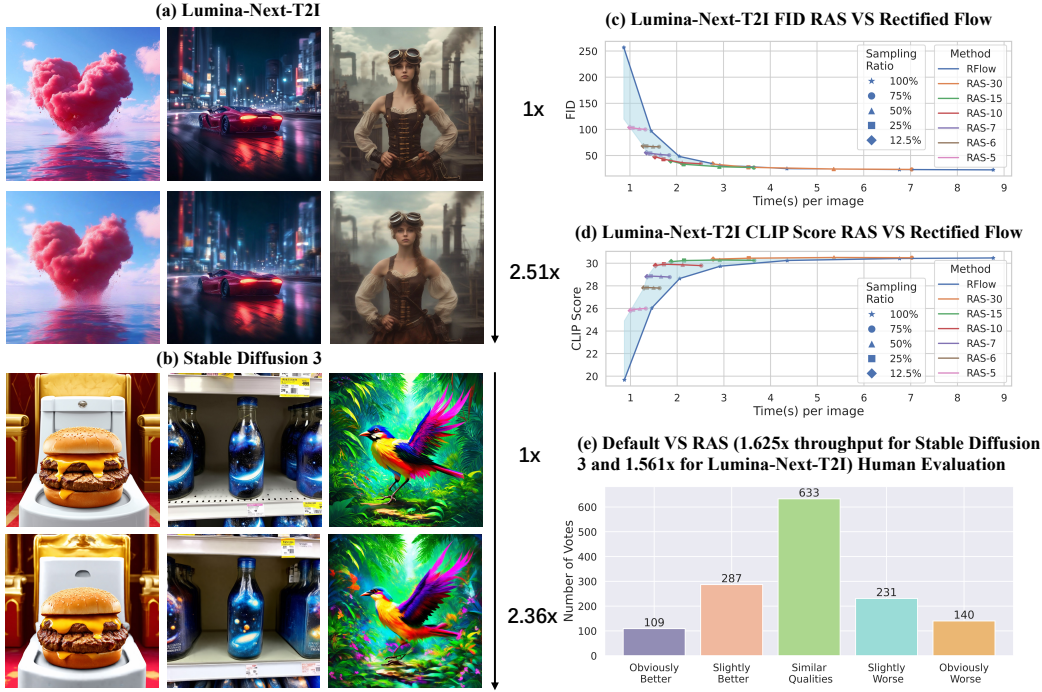


Figure 2. (a)(b) Acceleration results on Lumina-Next-T2I and Stable Diffusion 3 using 30 and 28 timesteps, respectively. (c)(d) Multiple configurations of RAS outperform rectified flow in both visual fidelity and text alignment; “RAS-X” denotes RAS with X total sampling steps. (e) RAS achieves comparable human evaluation scores to the default models while providing $\sim 1.6\times$ higher throughput.

cate foreground details may require more refinement steps to preserve fidelity, while smooth or repetitive background areas could tolerate more aggressive step reduction without noticeable quality loss. This observation motivates the development of a more flexible sampling strategy that dynamically adjusts the update ratio across regions, achieving faster generation while maintaining perceptual quality.

This direction represents a natural progression in the evolution of diffusion models. From DDPM [20] to Stable Diffusion XL [35], most existing models have been built upon U-Net backbones [39], whose convolutional structures impose uniform spatial processing due to their fixed-size grid inputs. In contrast, the emergence of Diffusion Transformers (DiTs) [34] and the growing adoption of fully transformer-based generative architectures [48] enable flexible tokenization and non-uniform computation across spatial regions. Leveraging this property, we are inspired to design a novel sampling strategy that adaptively allocates different sampling steps to different image regions, paving the way for more efficient and semantically aware diffusion generation.

To evaluate the feasibility of this idea, we visualized the intermediate diffusion outputs at various sampling steps (Figure 4). Two clear patterns emerged: (1) regions of focus exhibit strong continuity across adjacent steps in later stages, and (2) at each step, the model predominantly con-

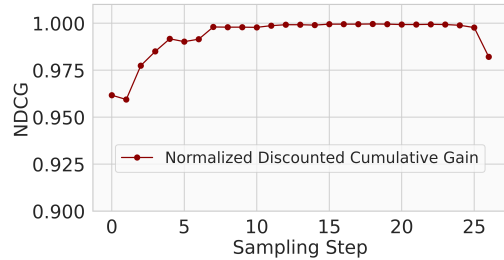


Figure 3. Normalized Discounted Cumulative Gain (NDCG) [23, 51] between adjacent sampling steps. The consistently high NDCG values throughout the diffusion process indicate strong similarity in the ranking of focused tokens (ranging from 0 to 1), reflecting temporal continuity of model attention.

centrates on semantically meaningful areas of the image. This behavior resembles an artist refining a canvas through thousands of careful strokes, where each iteration selectively enhances specific regions. Consequently, regions that receive little attention at a given step could potentially be skipped during computation in Diffusion Transformers (DiTs), enabling the model to concentrate computational resources on regions of interest.

We further validated this hypothesis by ranking tokens at each step using our proposed *output-noise metric*, which

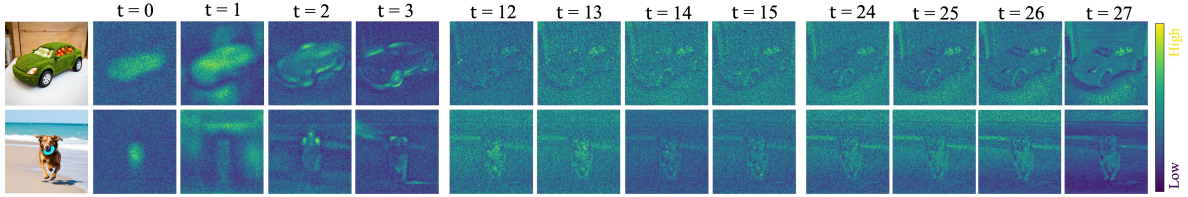


Figure 4. Visualization of predicted noise across diffusion steps. The DiT model gradually focuses on semantically meaningful regions at each timestep, and the attention shift between steps exhibits strong temporal continuity.

identifies the areas that the model focuses on most. We then measured the ranking similarity between consecutive steps using the NDCG metric (Figure 3), revealing high temporal consistency in attention across steps. These findings motivate a sampling strategy that adaptively allocates different update ratios to regions based on their attention persistence, paving the way for efficient yet high-quality diffusion sampling.

As illustrated in Figure 5, our method leverages the output noise from the previous step to identify the model’s primary focus for the current step, referred to as the *fast-update regions*. Only these regions are forwarded through the Diffusion Transformer (DiT) for denoising, while the remaining *slow-update regions* reuse the cached noise from the previous step. This design introduces regional variability in the number of effective sampling steps: regions of interest are updated more frequently, whereas less critical areas retain their previous noise estimates, thereby reducing overall computation.

For each input X_t , a fast-update rate determines the proportion of tokens selected for refinement. The updated fast-region noise is then combined with the preserved slow-region noise to construct the next-step input X_{t-1} . To maintain global semantic and visual consistency, features from slow-update regions are retained as reference keys and values for subsequent attention computations. Although fast-region selection is dynamic and recalculated at every step to prioritize semantically significant areas, we periodically perform full-image updates to prevent cumulative approximation errors. In summary, we propose **RAS**, the first diffusion sampling strategy that enables *regionally adaptive sampling ratios*. Unlike spatially uniform samplers, our approach allocates DiT’s computation to regions of current importance, achieving a better balance between efficiency and fidelity. As shown in Figure 2(c)(d), **RAS** substantially reduces inference cost with minimal FID degradation, while surpassing uniform baselines in FVD under equivalent budgets. Furthermore, Figure 2(a)(b) demonstrates that with large-scale models such as Lumina-Next-T2I [13] and Stable Diffusion 3 [11], our adaptive fast-region updating yields more than $2\times$ acceleration with negligible perceptual quality loss.

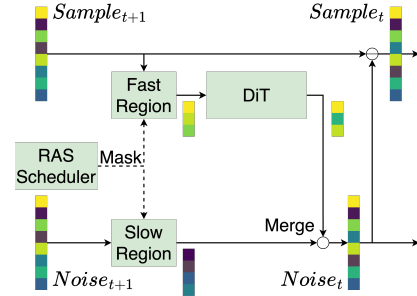


Figure 5. Overview of the **RAS** framework. At each diffusion step, only the *fast-update* regions identified from the previous output are forwarded through the model, while the remaining regions reuse cached noise. This selective update mechanism enables region-adaptive computation and efficient denoising.

2. Related Work

2.1. Diffusion Models: From U-Net to Transformer

Diffusion models (DMs) [8, 20, 44, 45] have demonstrated remarkable generative capabilities, often surpassing generative adversarial networks (GANs) [16] across diverse downstream tasks. Early variants such as DDPMs [20] and Stable Diffusion XL [35] primarily adopted convolutional U-Net backbones [39]. However, convolutional architectures inherently preserve strict spatial locality and resolution for operations like pooling, which constrains their ability to exploit redundancy in latent representations and complicates pruning or selective computation.

This limitation has been addressed by the advent of Diffusion Transformers (DiTs) [34], now employed in state-of-the-art generative models including Stable Diffusion 3 [10], Lumina-T2X [13], and PixArt- Σ [4]. In contrast to U-Nets, DiTs adopt a fully Transformer-based architecture [48] enhanced with adaptive layer normalization for conditional prompts, thereby eliminating convolution entirely. Positional information is encoded via embeddings, making latent tokens spatially independent and enabling flexible token-level manipulation. This property allows us to exploit redundancy (Section 1) by selectively updating the most relevant tokens at each sampling step while reusing cached

noise predictions for others.

2.2. Efficient Diffusion Model Inference

Reducing the high computational cost of diffusion model inference has been an active area of research. A common line of work focuses on decreasing the number of sampling steps required for generation. Several methods achieve this through additional training, such as progressive distillation [42], consistency models [46], and rectified flow [1, 29, 30]. Progressive distillation iteratively applies a specialized distillation process that transfers a pre-trained deterministic sampler into one capable of using substantially fewer diffusion steps. Latent Consistency Models (LCMs), in contrast, directly predict the clean image instead of the noise component during inference, allowing high-quality image synthesis with drastically reduced iterations. Among these, rectified flow has been particularly influential and is adopted in models such as Stable Diffusion 3 [10]. It learns an ordinary differential equation (ODE) trajectory that follows a straight path between the standard normal prior and the target data distribution. Such rectified trajectories effectively shorten the transport distance between distributions, thereby lowering the number of required sampling steps and improving inference efficiency.

Training-free methods have also been introduced to reduce either the number of sampling steps or the computational cost per step. For instance, DeepCache [55], designed for U-Net-based diffusion models, caches and reuses intermediate features across adjacent stages to skip redundant down- and up-sampling operations. However, such approaches uniformly process all image regions, disregarding the heterogeneous complexity present across different spatial areas, which leads to suboptimal efficiency.

As discussed in Section 1, image regions often vary significantly in structural and semantic complexity. To exploit this heterogeneity, we propose *RAS*, which adapts the computational allocation dynamically based on region-specific characteristics. Unlike existing methods that uniformly update all regions, *RAS* selectively refines the model’s regions of focus while reusing cached noise for less salient areas. Notably, *RAS* is orthogonal to prior acceleration techniques, including step-reduction methods and module-level optimizations such as DiTFastAttn [56] and Δ -DiT [5], and can be seamlessly combined with them for further efficiency gains.

3. Methodology

3.1. Overview

In this section, we introduce the overall design of *RAS* and describe the techniques used to exploit inter-timestep token correlations and the regional token attention mechanism introduced in Section 1. Our framework is built upon three

t	The current timestep
N	The noise output of the DiT model
\tilde{N}	The cached noise output from the previous timestep
\hat{N}	The estimated full-length noise calculated with N and \tilde{N}
S	The unpathified image sample
x	The pathified input of the DiT model
M	Mask generated to drop certain tokens in the input
D	The number of times the tokens in a patch being dropped

Table 1. Notation summary.

key components: (1) Motivated by the regional characteristics observed during DiT inference, we design an end-to-end pipeline that dynamically eliminates DiT computation for selected tokens at each timestep; (2) To capture temporal continuity across consecutive timesteps, we propose a simple yet effective method for identifying *fast-update regions* that require refinement in subsequent steps; and (3) Based on our observations of consistent spatial distribution patterns, we introduce several scheduling optimization techniques to further enhance the fidelity of the generated results.

3.2. Region-Adaptive Sampling

Region-Aware DiT Inference with *RAS*. Building on the insight that only certain regions are critical at each timestep, we design the *RAS* inference pipeline for Diffusion Transformers (DiTs). In U-Net-based diffusion models such as SDXL [35], token positions must remain fixed to preserve spatial structure. In contrast, the architecture of DiT, where positional information is injected via embeddings such as RoPE [47], allows masking or reordering of latent tokens without disrupting positional encoding. This flexibility enables us to selectively determine which regions are actively processed by the model.

At the end of each timestep, the current latent sample is reconstructed by merging the newly generated outputs for active tokens with the cached noise from inactive tokens. Formally, the noise sequence is restored by integrating the model output for the fast-update regions with the previously cached noise for the slow-update regions. This mechanism allows important tokens to move toward the updated direction determined at the current timestep, while less critical tokens maintain their previous trajectories.

To facilitate this process, we compute a region-wise metric R to identify fast-update regions based on the model’s output noise. We then update the drop count D to record how frequently each token has been skipped, and generate a binary mask M that governs computation in the subsequent step. Using M , the noise of slow-update regions is cached, while the tokens in fast-update regions are patchified and passed through the DiT model. Since modules

such as LayerNorm [2] and MLP operate independently across tokens, their computation remains consistent even when the token sequence is incomplete. For the attention module [48], we further introduce a caching mechanism to accelerate repeated key–value lookups, as detailed in a later section. Overall, *RAS* dynamically detects regions of focus and reduces DiT’s computational workload by at least the same proportion as the user-defined sampling ratio.

Region Identification. At each timestep, the DiT model receives the current timestep embedding, latent sample, and prompt embedding to predict the noise guiding the sample toward the clean image. To quantify the refinement requirement of each token, we analyze the model’s noise output and observe that the *standard deviation* of predicted noise effectively distinguishes different semantic regions. Empirically, the main subject (fast-update regions) exhibits noticeably lower noise variance than the background (slow-update regions), likely reflecting the uneven information density after the addition of Gaussian noise. Using the standard deviation as a region-selection metric yields robust results, highlighting semantically meaningful regions, maintaining image quality, and producing distinct contrasts between focused and background areas.

Temporal Token Continuity. Given the strong similarity between latent samples across adjacent timesteps, we hypothesize that tokens deemed important at the current timestep are likely to remain important in subsequent ones. Conversely, less-focused tokens can often be safely dropped with minimal perceptual impact. Before presenting the final formulation of our importance metric, we first introduce a mechanism to prevent the persistent exclusion of certain regions.

Starvation Prevention. During the diffusion process, the primary subject regions typically require more frequent refinement than background areas. However, repeatedly skipping background tokens may lead to excessive blurring or noise accumulation in the final output. To mitigate this issue, we track the frequency with which each token is dropped and incorporate this *drop count* as a scaling factor within our region-selection metric. This adjustment ensures that even low-importance tokens are periodically revisited, preventing starvation and maintaining global consistency throughout sampling.

Since the Diffusion Transformer (DiT) processes patchified latent tokens, we compute our metric at the patch level by averaging the per-token scores within each patch. Combining the above factors, our final region-selection metric is defined as:

$$R_t = \text{mean}_{\text{patch}}(\text{std}(\hat{N}_t)) \cdot \exp(k \cdot D_{\text{patch}}), \quad (1)$$

where \hat{N}_t denotes the predicted noise at timestep t , D_{patch} is the number of times the tokens in a patch have been dropped, and k is a scaling coefficient that controls the con-

trast between fast-update and slow-update regions. A higher k value enforces more aggressive recovery of long-inactive patches, balancing efficiency with image quality.

Key–Value Caching for Attention. The self-attention mechanism computes relationships between all tokens by using their queries, keys, and values. In *RAS*, the attention of active tokens can in principle be computed using only other active tokens. However, our metric R_t determines active and inactive regions solely based on noise statistics, without explicitly accounting for their contributions to the attention context. Naively excluding inactive tokens would therefore distort attention outputs and degrade generation quality.

To preserve contextual integrity while maintaining efficiency, we introduce a *key–value caching* mechanism. During each step, the complete key and value tensors are cached, and only the portions corresponding to active tokens are updated. This approach leverages the temporal consistency between adjacent timesteps: since token embeddings evolve smoothly, previously cached representations for inactive regions remain strong approximations of their true values. The attention output for the active tokens can thus be estimated as:

$$O_a = \text{softmax}\left(\frac{Q_a[K_a, \tilde{K}_i]^\top}{\sqrt{d}}\right)[V_a, \tilde{V}_i], \quad (2)$$

where Q_a , K_a , and V_a denote the query, key, and value matrices of the active tokens, while \tilde{K}_i and \tilde{V}_i represent the cached keys and values of the inactive tokens. This formulation approximates the full-attention output with minimal computation and negligible quality degradation. Please refer to the appendix for more details.

3.3. Scheduling Optimization

Dynamic Sampling Ratio. As shown in Figure 3, correlations between adjacent timesteps are relatively weak during the early diffusion stages but gradually strengthen as the process stabilizes, consistent with the patterns observed in Figure 4. This suggests that applying selective sampling too early may damage the structural foundation of the generated image. To address this, we employ a *dynamic sampling schedule*: the initial few steps (e.g., the first 4 out of 28) are executed with a full 100% sampling ratio to preserve the global image outline, after which the ratio is gradually reduced as the generation stabilizes. This adaptive design balances quality and efficiency, enabling substantial computational savings while minimizing any degradation in fine-grained details.

Accumulated Error Resetting. Because *RAS* focuses on regions of interest that persist across adjacent sampling steps, tokens in less-attended regions may remain inactive for extended periods. Without intervention, these regions

can accumulate stale denoising directions, leading to noticeable discrepancies between the latent produced by *RAS* and that from the original full sampling process. To mitigate this, we introduce *dense steps*, periodic full updates inserted into the diffusion process to reset accumulated errors. For example, in a 30-step sampling schedule where *RAS* begins at step 4, we designate steps 12 and 20 as dense steps. During these steps, all regions are reprocessed by the model, allowing corrections to any drift that may have developed in inactive areas. This periodic reset mechanism ensures long-term stability and keeps the denoising trajectory aligned with the intended generative path.

3.4. Implementation

Kernel Fusing. To reduce redundant computation, we employ key–value caching in self-attention and update only the active tokens during selective sampling steps. These partial updates correspond to a scatter operation on active-token indices. Instead of launching a separate scatter kernel, we fuse this operation into the preceding GeMM kernel, enabling in-place updates and avoiding extra memory copies. Following the principle in PIT [59], where permutation-invariant transformations can be integrated into GPU I/O stages with negligible overhead, we embed the scatter in the GeMM epilogue, effectively merging linear projection and token reindexing into a single efficient operation. This fusion eliminates synchronization and I/O overhead, yielding notable latency reductions during *RAS* inference.

4. Experiments

4.1. Experiment Setup

Models, Datasets, Metrics, and Baselines. We evaluate *RAS* on two state-of-the-art text-to-image diffusion models: *Stable Diffusion 3* [10] and *Lumina-Next-T2I* [13]. Experiments are conducted using 10,000 randomly sampled caption–image pairs from the MS-COCO 2017 dataset [28]. To evaluate both visual fidelity and text–image alignment, we adopt three standard metrics: (1) **Fréchet Inception Distance (FID)** [19], which measures overall image realism; (2) **Sliding FID (sFID)** [19], which provides a localized perceptual quality assessment; and (3) **CLIP Score** [18], which quantifies semantic consistency between generated images and their corresponding text prompts. For human evaluation, we have also included ImageReward [54], PickScore [25], and hpsv2 [53] as benchmarks. For baseline comparison, we evaluate against a suite of widely adopted rectified-flow and flow-matching–based acceleration methods [1, 6, 10, 12, 29, 30], all of which uniformly reduce the number of diffusion timesteps across the entire image. We implement *RAS* under multiple configurations with varying total timestep counts to assess trade-offs between efficiency and image quality, and compare the re-

Method	Steps	Sample Ratio	Image/s \uparrow	FID \downarrow	sFID \downarrow	CLIP \uparrow score
SD3						
RFlow	5	100%	1.43	39.70	22.34	29.84
RAS	7	25.0%	1.45	31.99	21.70	30.64
RAS	7	12.5%	1.48	32.86	22.10	30.55
RAS	6	25.0%	1.52	33.24	21.51	30.38
RAS	6	12.5%	1.57	33.81	21.62	30.33
RFlow	4	100%	1.79	61.92	27.42	28.45
RAS	5	25.0%	1.94	51.92	25.67	29.06
RAS	5	12.5%	1.99	53.24	26.04	28.94
Lumina						
RFlow	7	100%	0.49	48.19	38.60	28.65
RAS	10	25.0%	0.59	45.67	32.36	29.82
RAS	10	12.5%	0.65	47.34	32.69	29.75
RFlow	5	100%	0.69	96.53	59.26	26.03
RAS	7	25.0%	0.70	53.93	39.80	28.85
RAS	7	12.5%	0.74	54.62	40.23	28.83
RAS	6	25.0%	0.75	67.16	46.46	27.85
RAS	6	12.5%	0.78	67.88	45.88	27.83

Table 2. Pareto Improvements of rectified flow with *RAS* on COCO Val2014 1024 \times 1024. Full experiment results are available in the Supplementary Material.

sults against the original full-sampling implementations under equivalent throughput settings.

Code Implementation. We implement *RAS* in PyTorch [33], leveraging the *diffusers* library [49] and its `FlowMatchEulerDiscreteScheduler` for inference scheduling. Evaluation metrics are computed using publicly available implementations from GitHub repositories, including FID [43], sFID [21], and CLIP Score [60]. All experiments are conducted on four servers, each equipped with eight NVIDIA A100 (40 GB) GPUs, while latency and throughput benchmarks are measured on a single NVIDIA A100 (80 GB) GPU.

4.2. Generation Benchmarks

We conduct a comparative evaluation between *RAS* and rectified flow baselines, which uniformly reduce the number of timesteps for all tokens during inference. To comprehensively assess performance, we experiment with multiple configurations of inference timesteps and selective sampling ratios. The results can be interpreted from two complementary perspectives.

Pushing the Efficiency Frontier. From the first perspective, *RAS* extends the efficiency frontier by further reducing inference cost at each fixed number of timesteps offered by rectified flow. As shown in Figure 2(c)(d), we generate 10,000 images using dense inference across timestep settings ranging from 3 to 30, and subsequently apply *RAS* with varying average sampling ratios over the selective-sampling stages. The results demonstrate that *RAS* substantially decreases inference time with only marginal impact on generation quality metrics. For example, applying

Stable Diffusion 3			
Method	Steps	Memory (GB)	Speedup
RFlow	28	19.21 (1x)	1x
RAS-50%	28	20.36 (1.06x)	1.62x
RAS-12.5%	28	20.36 (1.06x)	2.44x

Lumina-Next-T2I			
Method	Steps	Memory (GB)	Speedup
RFlow	30	10.30 (1x)	1x
RAS-50%	30	10.73 (1.04x)	1.56x
RAS-12.5%	30	10.73 (1.04x)	2.70x

Table 3. Memory Consumption of RAS.

RAS with a 25% sampling ratio over 30 timesteps achieves a $2.25\times$ increase in throughput, with only a **22.12%** rise in FID, a **26.22%** rise in sFID, and a negligible **0.065%** drop in CLIP score. Moreover, the efficiency gains achieved through *RAS* are obtained at a lower quality cost than those incurred by simply reducing the total number of sampling steps. Specifically, the rate of quality degradation when decreasing *RAS*'s sampling ratio is significantly smaller than that observed when performing dense inference with fewer timesteps, particularly in low-step regimes (< 10 steps). These findings highlight *RAS* as a practical and effective strategy for improving inference efficiency while maintaining visual fidelity and prompt alignment.

Pareto Improvements over Uniform Sampling. We observe that *RAS* frequently yields *Pareto improvements* over rectified-flow baselines. To demonstrate this, we sorted the results from Stable Diffusion 3 and Lumina-Next-T2I by throughput and compared different configurations of *RAS* with their closest baseline counterparts in Table 2. Across nearly all cases, *RAS* achieves higher throughput while simultaneously improving FID, sFID, and CLIP scores compared with dense rectified-flow inference. These results indicate that for any given throughput level, *RAS* provides configurations that offer both superior speed and quality, effectively expanding the Pareto frontier for balancing efficiency, fidelity, and text-image alignment.

4.3. Memory Consumption

Since *RAS* introduces caching of intermediate noise estimates as well as attention keys and values during inference, we quantify its additional memory overhead in Table 3. *RAS* incurs only a modest memory increase, approximately 6% for Stable Diffusion 3 and 4% for Lumina-Next-T2I, relative to baseline inference. This overhead is acceptable given the substantial acceleration gains achieved. Moreover, the additional memory consumption remains stable across different sampling ratios, as the full set of activations is cached for reuse throughout the inference process.

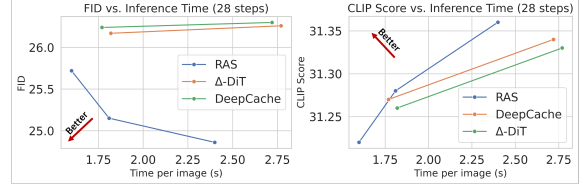


Figure 6. Comparison on Stable Diffusion 3 of *RAS* with DeepCache and Δ -DiT, utilizing different cache interval.

4.4. Comparison with Layer-Wise Methods

Although orthogonal in principle, we compare *RAS* against representative layer-wise cache-based acceleration methods for a more comprehensive analysis, using a subset of 5,000 images from MS-COCO. We manually adapt DeepCache [55] for DiT models by reusing intermediate features and reimplement Δ -DiT [5] according to its official description. As shown in Figure 6, *RAS* achieves greater speedups while also yielding improved FID and CLIP scores. These results further highlight the advantage of region-adaptive token selection over layer-wise caching strategies in fully Transformer-based diffusion models.

4.5. Detailed Prompts, Object Positions, and Counts

To evaluate the robustness of *RAS* under highly detailed prompts, particularly when users specify precise object counts, spatial arrangements, or relationships, we conduct experiments on the ParaImage-3000 [52] and GenEval [15] datasets. In the evaluations, *RAS* maintains comparable overall performance to baseline inference and achieves Pareto improvements across multiple evaluation dimensions, confirming that region-adaptive sampling preserves compositional and positional accuracy even under complex generation constraints. Due to the space limit, please refer to the quantitative result tables in the supplementary material.

4.6. Human Evaluation

To further assess perceptual quality, we conduct a human evaluation to determine whether *RAS* improves efficiency without compromising visual fidelity. We select 14 prompts from the official papers and blogs of Stable Diffusion 3 and Lumina-Next-T2I, generating two images per prompt, one using dense inference and the other using *RAS* with the same random seed and timestep schedule. During the selective sampling phase, *RAS* employs a 50% average sampling ratio. We recruit 100 participants from 18 universities and companies to perform pairwise comparisons between outputs. Participants consistently report that images generated with *RAS* are visually indistinguishable from or slightly preferred over those from dense inference, reinforcing the effectiveness of our approach in preserving perceptual quality while significantly improving inference throughput.

Method	FID ↓	sFID ↓	CLIP score ↑
Default	35.81	18.41	30.13
Static Sampling Freq.	37.92	19.11	29.98
Random Dropping	43.19	22.23	29.65
W/O Error Reset	46.10	24.85	30.41

Method	Timesteps	FID ↓	sFID ↓	CLIP score ↑
Default	28	24.30	26.26	31.34
W/O	28	31.36	20.19	31.29
Default	10	35.81	18.41	30.13
W/O	10	32.33	20.21	30.27

Reset ID	FID ↓	sFID ↓	CLIP ↑
5	27.04	19.03	31.33
8	24.60	17.24	31.31
11	25.80	16.67	31.17
7,11	24.58	15.82	31.31

Method	Steps	FID ↓	sFID ↓	CLIP ↑
Default	10	35.81	18.41	30.13
W/O	10	39.87	19.75	29.84
Default	14	26.48	18.14	31.18
W/O	14	26.58	17.96	31.11

Table 4. Ablation Study on Stable Diffusion 3. All techniques including dynamic sampling ratio, region identifying, error reset, key & value recovery are necessary for high quality generation.

As shown in Figure 2(e), 45.21% of 1,400 votes judged the two images to be of comparable quality, while 28.29% favored the dense inference results and 26.50% preferred outputs generated by *RAS*. These results indicate that *RAS* achieves substantial throughput gains: $1.625\times$ on Stable Diffusion 3 and $1.561\times$ on Lumina-Next-T2I, with negligible degradation in perceived image quality.

Furthermore, we evaluate *RAS* on three human-preference-aligned metrics: ImageReward [54], PickScore [25], and HPSv2 [53]. As shown in Table 5, *RAS* maintains high performance across all benchmarks while delivering significantly faster generation, reaffirming that our region-adaptive sampling achieves human-aligned quality at markedly lower computational cost.

4.7. Ablation Study

Token Drop Scheduling. As reported in Table 4(a), we evaluate the impact of the scheduling strategies introduced in Section 3, including (1) dynamic sampling-ratio scheduling, (2) selective caching of dropped tokens, and (3) insertion of dense steps to reset accumulated errors. All experiments are conducted on Stable Diffusion 3 using 10 timesteps and an average sampling ratio of 12.5%. Results show that each component contributes positively to image quality, with the combination providing the best overall balance between efficiency and fidelity.

Key-Value Caching. As shown in Table 4(b), reusing keys and values from the previous timestep is critical for maintaining generation quality, particularly when using longer sampling schedules. While discarding the keys and values of inactive tokens slightly improves throughput, it substantially distorts the attention distribution of active to-

Method	Steps	Time(s)	SpeedUp ↑	Img. Rew. ↑	PickScore ↑	hpsv2 ↑
RFlow	30	8.77	1	0.37	21.88	0.26
RFlow	15	4.36	2.01	0.13	21.45	0.24
RAS-25%	30	3.89	2.26	0.13	21.45	0.22
RAS-75%	15	3.72	2.35	0.05	21.34	0.24
RFlow	10	2.92	3	-0.20	20.94	0.21
RAS-25%	15	2.31	3.78	-0.18	20.98	0.21
RFlow	7	2.05	4.27	-0.75	20.24	0.19
RAS-25%	10	1.70	5.15	-0.43	20.54	0.19
RAS-12.5%	10	1.54	5.68	-0.54	20.34	0.18

Table 5. Benchmarks for evaluating the human preference on Lumina-Next-T2X. RAS-X% stands for RAS with X% tokens activated each step. RAS provides Pareto improvements in multiple settings.

kens. Importantly, a low ranking in our region metric does not imply negligible influence in the attention mechanism—demonstrating the necessity of caching to preserve cross-token contextual integrity.

Error-Resetting Schedule. Table 4(c) presents results on different error-resetting schedules using 14 timesteps on Stable Diffusion 3. We find that inserting a dense-step reset near the midpoint of the selective-sampling phase (e.g., between steps 4–13) yields the optimal trade-off between image quality and computation. Adding additional dense steps offers only marginal improvement while introducing nontrivial time overhead.

Starvation Prevention. Finally, Table 4(d) validates the effectiveness of our starvation-prevention mechanism. By tracking token drop frequency and scaling their reactivation probability, this component prevents persistent omission of background regions and stabilizes long-horizon sampling—achieving quality improvements with negligible computational overhead.

5. Conclusion

In this work, we investigate the spatial heterogeneity in diffusion generation, observing that different image regions demand varying refinement levels during denoising and exhibit strong temporal continuity across steps. Leveraging these insights, we propose *RAS*, a training-free sampling strategy that dynamically allocates computation based on regional attention—prioritizing semantically important areas while reusing cached predictions for less critical ones.

Extensive experiments and user studies show that *RAS* achieves significant acceleration with minimal perceptual loss, consistently surpassing uniform sampling baselines. These results underscore the promise of region-adaptive and temporally aware strategies for efficient diffusion transformers and real-time generative modeling.

References

- [1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 1, 4, 6
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 5
- [3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 1
- [4] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. 3
- [5] Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Δ -dit: A training-free acceleration method tailored for diffusion transformers. *arXiv preprint arXiv:2406.01125*, 2024. 1, 4, 7
- [6] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space, 2023. 6
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1
- [8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 1, 3
- [9] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023. 1
- [10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 3, 4, 6
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 3
- [12] Johannes S Fischer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan A Baumann, and Björn Ommer. Boosting latent diffusion with flow matching. *arXiv preprint arXiv:2312.07360*, 2023. 6
- [13] Peng Gao, Le Zhuo, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, Chen Lin, Rongjie Huang, Shijie Geng, Renrui Zhang, Junlin Xi, Wenqi Shao, Zhengkai Jiang, Tianshuo Yang, Weicai Ye, He Tong, Jingwen He, Yu Qiao, and Hongsheng Li. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers, 2024. 3, 6
- [14] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yan-jing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023. 1
- [15] Dhruva Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023. 7, 1
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3
- [17] Philip Hartman. *Ordinary differential equations*. SIAM, 2002. 1
- [18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6
- [19] 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. 6
- [20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 1, 2, 3
- [21] Tao Hu. pytorch-fid-with-sfid. <https://github.com/dongzhuoyao/pytorch-fid-with-sfid>, 2022. 6
- [22] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1
- [23] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 41–48, New York, NY, USA, 2000. Association for Computing Machinery. 2
- [24] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 1
- [25] 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, 2023. 6, 8
- [26] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdtm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1952–1961, 2023. 1
- [27] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single

- image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 1
- [28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6
- [29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1, 4, 6
- [30] 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. 1, 4, 6
- [31] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 1
- [32] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 1
- [33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019. 6
- [34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. 2, 3
- [35] 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, 2023. 2, 3, 4
- [36] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1
- [37] Philip E Protter and Philip E Protter. *Stochastic differential equations*. Springer, 2005. 1
- [38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1
- [39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2, 3
- [40] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *Forty-first International Conference on Machine Learning*, 2023. 1
- [41] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 1
- [42] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*. 1, 4
- [43] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. <https://github.com/mseitzer/pytorch-fid>, 2020. Version 0.3.0. 6
- [44] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, page 2256–2265. JMLR.org, 2015. 1, 3
- [45] Yang Song and Stefano Ermon. *Generative modeling by estimating gradients of the data distribution*. Curran Associates Inc., Red Hook, NY, USA, 2019. 1, 3
- [46] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1, 4
- [47] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4
- [48] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2, 3, 5
- [49] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022. 6
- [50] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 1
- [51] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of ndcg type ranking measures, 2013. 2
- [52] Weijia Wu, Zhuang Li, Yefei He, Mike Zheng Shou, Chunhua Shen, Lele Cheng, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Paragraph-to-image generation with information-enriched diffusion model, 2023. 7, 1
- [53] 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, 2023. 6, 8
- [54] 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, 2023. 6, 8

- [55] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. Deepcache: Principled cache for mobile deep vision. In *Proceedings of the 24th annual international conference on mobile computing and networking*, pages 129–144, 2018. 1, 4, 7
- [56] Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models, 2024. 4
- [57] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36, 2024. 1
- [58] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023. 1
- [59] Ningxin Zheng, Huiqiang Jiang, Quanlu Zhang, Zhenhua Han, Lingxiao Ma, Yuqing Yang, Fan Yang, Chengruidong Zhang, Lili Qiu, Mao Yang, et al. Pit: Optimization of dynamic sparse deep learning models via permutation invariant transformation. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 331–347, 2023. 6
- [60] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. <https://github.com/taited/clip-score>, 2023. Version 0.1.1. 6