This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Layer- and Timestep-Adaptive Differentiable Token Compression Ratios for Efficient Diffusion Transformers

Haoran You[1,2,*], Connelly Barnes[2], Yuqian Zhou[2], Yan Kang[2],
Zhenbang Du[1], Wei Zhou[1], Lingzhi Zhang[2], Yotam Nitzan[2], Xiaoyang Liu[2],
Zhe Lin[2], Eli Shechtman[2], Sohrab Amirghodsi[2], Yingyan (Celine) Lin[1]

[1]Georgia Institute of Technology [2]Adobe Research

Abstract

*Diffusion Transformers (DiTs) have achieved state-of-the-art (SOTA) image generation quality but suffer from high latency and memory inefficiency, making them difficult to deploy on resource-constrained devices. One major efficiency bottleneck is that existing DiTs apply equal computation across all regions of an image. However, not all image tokens are equally important, and certain localized areas require more computation, such as objects. To address this, we propose **DiffCR**, a dynamic DiT inference framework with differentiable compression ratios, which automatically learns to dynamically route computation across layers and timesteps for each image token, resulting in efficient DiTs. Specifically, DiffCR integrates three features: (1) A token-level routing scheme where each DiT layer includes a router that is fine-tuned jointly with model weights to predict token importance scores. In this way, unimportant tokens bypass the entire layer's computation; (2) A layer-wise differentiable ratio mechanism where different DiT layers automatically learn varying compression ratios from a zero initialization, resulting in large compression ratios in redundant layers while others remain less compressed or even uncompressed; (3) A timestep-wise differentiable ratio mechanism where each denoising timestep learns its own compression ratio. The resulting pattern shows higher ratios for noisier timesteps and lower ratios as the image becomes clearer. Extensive experiments on text-to-image and inpainting tasks show that DiffCR effectively captures dynamism across token, layer, and timestep axes, achieving superior trade-offs between generation quality and efficiency compared to prior works. The project website is available here.*

1. Introduction

Diffusion models have recently demonstrated outstanding performance in image generation, with architectures evolving from U-Nets [13, 28, 32, 39] to Transformers [1, 3, 4, 27]. Among these, Diffusion Transformers (DiTs) [1, 27] stand out for their superior scalability. However, diffusion models, particularly DiTs, are hampered by substantial computational and memory demands, which limits their efficiency in generation and deployment. For instance, generating a 1024px image with full context on a single A100 GPU can take 19.48 seconds and require >40GB GPU memory [26, 28]. One major efficiency bottleneck in most DiTs stems from the uniform application of computation across all image regions, despite varying levels of complexity in different areas [26, 30, 55]. Such an efficiency bottleneck suggests an ideal DiT inference framework could have adaptive and dynamic computation across three key axes in DiTs: token, layer, and timestep.

Various techniques have been proposed to address the efficiency bottleneck along the three aforementioned key axes: (1) token merging [2], pruning [43], and downsampling [37]; (2) layer [17] or channel [9] pruning; and (3) few-step distillation and sampling [21, 33, 44, 54]. While promising, the techniques (1-2) for the most part rely on heuristics, such as heuristic rules for token importance and channel and layer pruning rules. Moreover, compression ratios are often uniform across layers or adjusted empirically based on prior experience. In addition, most approaches focus on a single efficiency axis, overlooking the compounded effect of combining optimizations across all three.

To achieve a unified and learnable dynamic DiT inference framework with differentiable compression ratios across layers and timesteps, three key challenges must be tackled: (1) *Token Perspective:* Developing a learnable token importance metric that adapts to visual content, as not all tokens are equally important. (2) *Layer Perspective:* Designing mechanisms to autonomously learn adaptive compression ratios for each layer, optimizing processing efficiency, since not all layers contribute equally. (3) *Timestep Perspective:* Developing methods to learn and apply compression ratio patterns effectively across timesteps, as not all timesteps are equally important. We make the following

*Work done while interning at Adobe.

contributions to address these three challenges:

- We propose a dynamic DiT inference framework with differentiable compression ratios, dubbed **DiffCR**, which automatically learns an optimal dynamic computation across layers and timesteps for each image token, resulting in efficient DiT models for content generation tasks.
- ***Enabler 1***: We adopt a token-level routing scheme inspired by the mixture-of-depth (MoD) [31], which automatically learns token importance scores. Each DiT layer includes a lightweight router that is fine-tuned jointly with the model weights. Based on the compression ratio, less important tokens bypass computation in the entire layer. To the best of our knowledge, we are the first to apply MoD to the vision domain. Our routing analysis reveals that token importance varies across layers and timesteps.
- ***Enabler 2***: Based on our analysis, we introduce a novel DiffCR module that enables the token routing scheme to be differentiable with respect to compression ratios, allowing the model to learn adaptive compression ratios for each layer starting with zero initialization. Redundant layers learn higher compression ratios, while important layers remain less compressed or entirely uncompressed.
- ***Enabler 3***: We further present a timestep-wise differentiable ratio mechanism, enabling each layer and denoising timestep to learn its own compression ratio. This results in a pattern where noisier timesteps adopt higher compression ratios, while clearer stages maintain lower ratios.

Our extensive experiments on both image inpainting and text-to-image (T2I) tasks consistently demonstrate that DiffCR achieves a superior trade-off between generation quality and efficiency, with an average FID reduction of 8.51 while maintaining comparable latency and memory usage, compared to the most competitive baseline.

2. Related Work

2.1. Diffusion Models

Diffusion models [13, 38] have demonstrated superior performance over prior SOTA generative adversarial networks (GANs) in image synthesis tasks [7]. Early diffusion models primarily utilized U-Net architectures. Subsequent work introduced several improvements, such as advanced sampling [16, 21, 39] and classifier-free guidance [12]. Although effective, these models suffered from high generation latency due to processing directly in pixel space, thus limiting their practical applications. The introduction of Latent Diffusion Models (LDMs) [32] marked a significant advancement by encoding pixel space into a more compact latent space through training a Variational Auto-Encoder (VAE). This reduced the computational cost of the diffusion process, paving the way for widely used models like Stable Diffusion Models (SDMs) [28]. More recently, researchers have explored Transformer [41] architectures for diffusion, leading to the development of DiTs [1, 27], which employ a pure Transformer backbone and exhibit improved scalability. Our DiffCR proposes a novel dynamic DiT inference framework with differentiable compression ratios and is compatible with all recent DiT models.

2.2. Efficient Diffusion and DiT Models

DiTs [27] are resource-intensive due to the transformer architecture, with the attention module exhibiting quadratic complexity relative to the number of tokens. Previous work has mainly focused on optimizing DiTs' deployment efficiency along three dimensions: token, layer, and timestep. For tokens, researchers have introduced techniques like token merging [2] to merge similar tokens, token pruning [43] or image resolution downsampling [37] to remove redundant tokens, and LazyDiffusion [26], which is specialized for the inpainting task and bypasses generating background tokens. For layers, methods such as layer [17] and channel [9] pruning, as well as intermediate feature caching [20, 23, 52], have been proposed to skip redundant computations. For timesteps, strategies include distillation to reduce the required number of timesteps, which has been explored for UNets [15, 21, 33, 34, 53, 54] although there is no reason to believe these techniques cannot apply to Transformers, and asymmetric sampling, which has been applied to Transformer architectures and allocates more samples to undersampled stages and fewer to stages that have already converged [29, 44]. Additionally, to accelerate diffusion T2I models, more specialized techniques have been introduced [3, 4]. In contrast, our proposed DiffCR is a learnable and unified dynamic DiT inference framework with differentiable compression ratios across layers and timesteps, exploring the compounded effects of compression across all three axes. However, we do not explore few step distillation (e.g. [53]) in this paper, since it is an orthogonal acceleration method that is complementary to ours.

2.3. Dynamic Inference

Model compression [6] offers a static approach to improving inference efficiency, while dynamic inference [31, 46, 47, 50, 56] enables adaptive compression based on the input, layer, or other conditions. For example, early exiting methods [14, 22, 40] predict the optimal point for early termination within intermediate layers, allowing the model to exit before completing all computations. Dynamic layer-skipping methods [46, 47, 50] selectively execute subsets of layers for each input, often utilizing a gating network to make decisions on the fly. At a finer granularity, researchers have also explored channel skipping [9, 24] and mixture-of-depths (MoD) approaches [31], which select specific subsets of layers for individual tokens rather than processing the entire input uniformly. In contrast, our DiffCR is the first to introduce a unified dynamic DiT inference framework that optimizes across three axes: token, layer, and

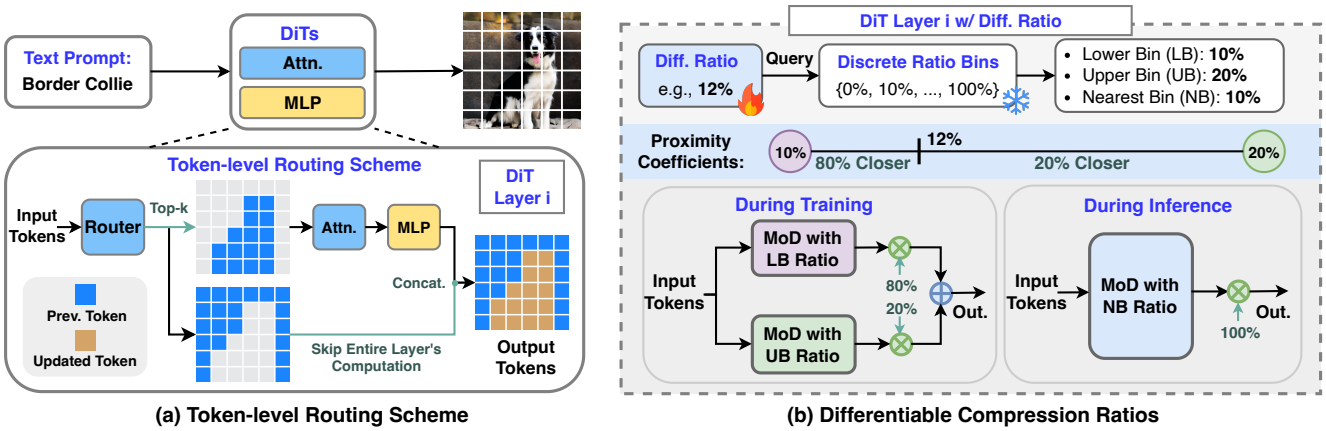


Figure 1. Overview of the proposed DiffCR framework: (a) token-level routing scheme and (b) differentiable compression ratios.

timestep. It also enables differentiable compression ratios that are fine-tuned jointly with the network, enhancing adaptability and efficiency.

3. The Proposed DiffCR Framework

In this section, we present the proposed DiffCR framework. First, we provide an overview of the method. Then, we detail the three enablers: (1) the token-level routing scheme for DiTs in Sec. 3.2; (2) the layer-wise differentiable compression ratio scheme in Sec. 3.3; and (3) the timestep-wise differentiable compression ratio scheme in Sec. 3.4.

3.1. Overview of DiffCR

Motivated by the need for unified and dynamic compression during DiT inference, DiffCR introduces a token-level routing scheme to dynamically learn the importance of each token on the fly. As illustrated in Fig. 1 (a), similar to previous mixture-of-depths (MoD) work [31] for NLP tasks, each DiT layer incorporates a lightweight router using a single linear layer to predict the importance of each token based on the input image/noise and text embedding. This allows us to bypass computations for less important tokens in each layer and to directly forward their activations to the layer's outputs. Consequently, each token is processed by only a selective subset of layers. Visualization of these routers' predictions reveals that different layers or timesteps favor varying compression ratios—for instance, some layers prioritize generating objects, while others focus on backgrounds—highlighting the need for adaptive compression across layers and timesteps. To achieve such dynamic compression, DiffCR incorporates a differentiable compression ratio scheme, as shown in Fig. 1 (b). This scheme includes a learnable scalar parameter that represents a continuous compression ratio, and predefined discrete ratio bins as proxy ratios. The scalar queries the bins to identify lower and upper bin ratios, creating two separate paths with distinct compression ratios. The final output is a linear combination of these two paths, weighted by the proximity of the learned ratio to each bin. We apply a mean-squared error (MSE) loss to ensure that the average learned ratio across layers or timesteps converges to the target ratio. By doing so, DiffCR learns the adaptive compression ratios in a differentiable manner, resulting in efficient and dynamic mixture-of-depths DiTs.

3.2. Enabler 1: Token-level Routing Scheme

Motivation. We are motivated by the varying computational demands across tokens, as many of them require fewer layers for efficient processing. We start from the same token-level routing scheme as MoD [31]. We remove from MoD two features that were specialized for the acausal NLP task: specifically, we remove the auxiliary loss and auxiliary MLP predictor from Section 3.5 of their paper. To the best of our knowledge, our paper is the first application of MoD to the vision domain, so we next review the routing mechanism, perform some visualizations, and report insights for vision tasks.

Token-level Routing. DiTs process noise and conditional text embeddings as inputs, aiming to denoise and generate images in an end-to-end manner. To predict token importance, we employ a simple yet effective token-level routing scheme from MoD [31]. As illustrated in Fig. 1 (a), each DiT layer incorporates a lightweight router composed of a single linear layer with a sigmoid activation function, predicting each token's importance on a scale from 0 to 1. After passing through the routers, we select the top-k most important tokens for this layer's processing, while the activations of other tokens are cached and concatenated with the layer outputs, bypassing the entire layer computation, including both attention and MLPs. To enable gradient flow to the router's weights during joint fine-tuning with pretrained DiT models, the same as MoD [31], we rescale the top-k token output activations by multiplying them with the router's predictions. This rescaling ensures that gradients are propa-

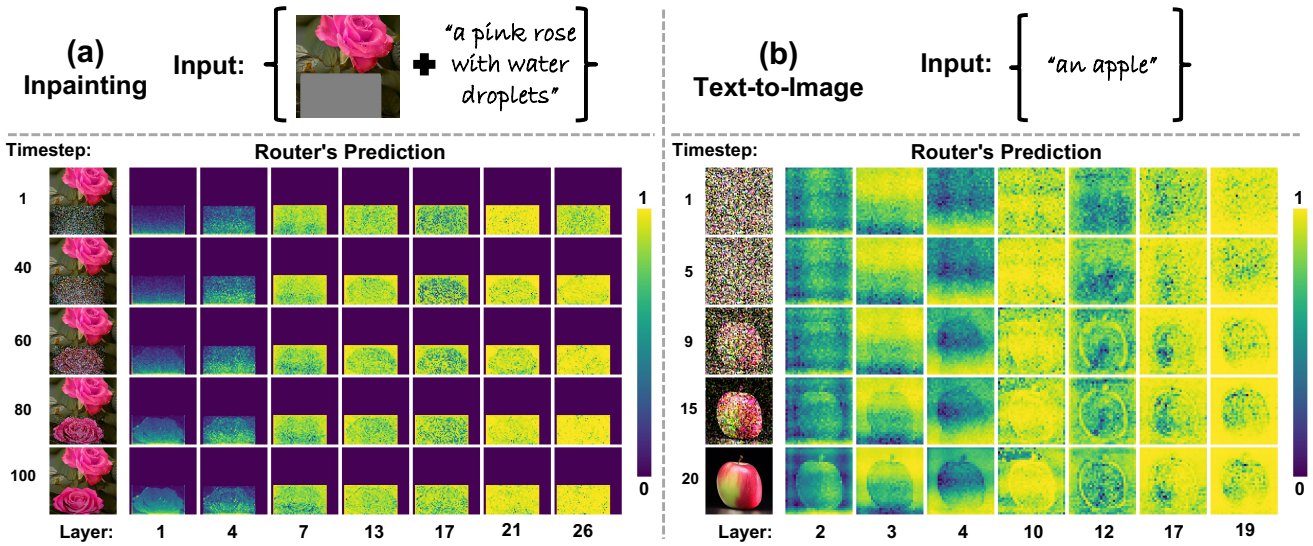


Figure 2. Visualization of the router's predictions: (a) For inpainting tasks, where inputs are masked images with text prompts, we follow the previous SOTA method Lazy-Diffusion [26] to generate only the masked area rather than the entire image; (b) For text-to-image (T2I) tasks, where inputs are noise and text prompts, we follow PixArt-Σ [4] for generation. Each visualization includes the router's prediction map with values ranging from 0 to 1. The generated image at each corresponding timestep is shown on the left, while the router's prediction maps across various layers and timesteps are displayed on the right. More visualizations are provided in the supplementary materials.

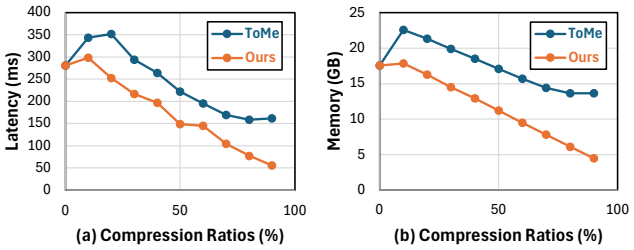


Figure 3. Comparison of latency and memory savings between our DiffCR router and the previous token merging method (ToMe) [2] when applied to ViT-XL/2 [8, 26] on an A100 GPU.

gated effectively to the router during backpropagation. The value of k is determined based on compression ratios; we empirically find that 20% or 30% compression offers an optimal trade-off between latency/memory efficiency and minimal degradation in generation quality. Unlike previous token merging techniques [2], where computational savings are not directly proportional to the merge ratios due to the extra overhead, as shown in Fig. 3, the MoD in our approach yields more reductions in actual latency and memory usage due to negligible overhead.

Router Visualization and Insights. To test the router's effectiveness, we visualize its predictions in Fig. 2. Our observations reveal that: (1) *The router effectively captures semantic information*, clearly delineating object shapes and achieving an attention-like effect with significantly reduced computational costs; (2) *The predicted token importance varies across layers and timesteps*. For instance, some layers prioritize object generation, while others emphasize background areas. Additionally, as timesteps progress, the router increasingly captures the semantic contours of objects, underscoring the need for dynamic token importance estimation; (3) *The optimal compression ratio differs across layers and timesteps*. For example, certain layers designate all tokens as high-importance, showing minimal redundancy, whereas other layers selectively prune object or background tokens with distinct shapes, requiring varying compression ratios. Similar variance is observed across timesteps. In the current MoD, a fixed global compression rate is applied equally to each layer and timestep, rather than adapting to its individual significance. Uniform pruning risks over-pruning critical layers or timesteps while leaving redundant ones less compressed. This motivates us to apply adaptive and dynamic compression ratios across both layers and timesteps.

3.3. Enabler 2: Layer-wise Differentiable Ratio

Motivation. Recognizing that different layers prioritize different objects or background elements and thus benefit from distinct compression ratios, we propose a novel layer-wise differentiable compression ratio mechanism. This approach automatically learns each layer's compression ratio from a zero initialization in a differentiable manner, adapting to the varying redundancy levels across layers.

Design Choice. Before designing DiffCR, we address a key choice: a discrete proxy or a continuous ratio representation. Previous work [5] uses a discrete proxy with multiple compression ratio candidates and learnable probabilities, but this approach poses three challenges for MoD: (1) it lacks effective initialization, as the final ratio relies on the product of candidates and probabilities, making it

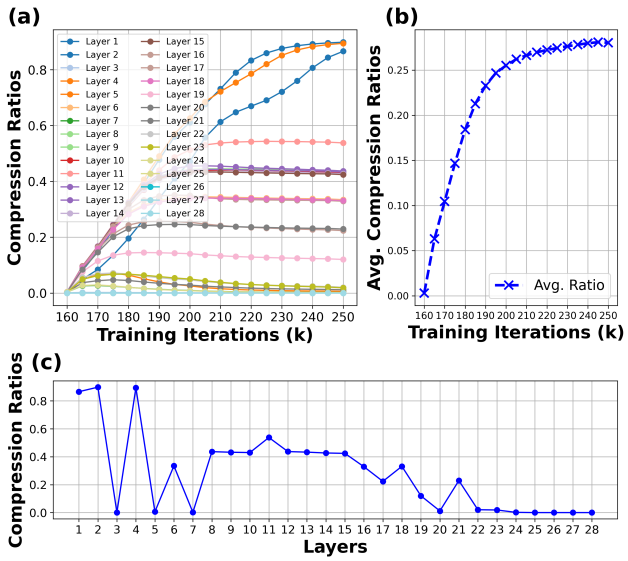


Figure 4. Visualization of the compression ratio trajectory during fine-tuning: (a) Trajectories for each of the 28 layers in DiT models; (b) Average ratio trajectory across all layers; and (c) The final learned ratio distribution across 28 layers.

difficult to initialize all ratios at zero; (2) MoD requires a differentiable and learnable router, incompatible with discrete proxies that need multiple sets of top-k tokens; and (3) it introduces numerous learnable parameters, complicating training and interpretability, and leading to hard-to-interpret ratio distributions. In contrast, we represent each layer's compression ratio with a single continuous scalar parameter, reducing the number of parameters to 28 for DiT's 28 layers [3] and directly representing the compression ratio.

Layer-wise DiffCR. As illustrated in Fig. 1 (b), we assign each layer a single learnable parameter and introduce discrete MoD compression ratio bins at 10% intervals, ranging from 0% to 100%. During training, the learnable MoD ratio queries the nearest two discrete bins to retrieve the lower and upper bin ratios. For example, a 22% learnable ratio would correspond to 20% as the lower bin and 30% as the upper bin. We then apply a forward pass through the DiT layer (with MoD routers) with each of these bin ratios, producing two output branches. The final output is a weighted linear combination of these branches, where the weights are determined by the proximity of the learnable ratio to each bin—e.g., for 12%, the output would be 80% weighted towards the 10% branch and 20% towards the 20% branch. Although this approach doubles the cost of a forward pass during training, we simply select the nearest bin as the final compression ratio during inference, eliminating any overhead. To ensure that the ratio converges to our target value, we incorporate an additional MSE loss between the current learned average ratios across all layers in the batch and the target ratio, which is a hyperparameter.

Ratio Trajectory Analysis. We visualize the training

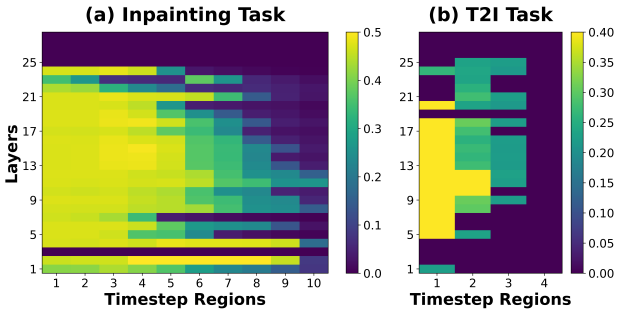


Figure 5. Visualization of the learned ratio patterns across both timesteps and layers for the (a) inpainting task and (b) T2I task.

trajectory of compression ratios for all layers during fine-tuning of a LazyDiffusion model on an inpainting task in Fig. 4 (a-c). The visualization reveals that: (1) Each layer learns its unique compression ratio, with redundant layers achieving higher compression and critical layers remaining less or entirely uncompressed; (2) The average ratio across layers gradually converges to the target ratio. In this example, with a target of 30%, the final achieved average ratio is approximately 29%, indicating a minor gap. Notably, a trade-off exists between convergence speed and generation quality: a higher MSE loss coefficient for the ratio accelerates convergence but may degrade quality due to overly rapid compression, while a smaller coefficient promotes gradual convergence and maintains quality, albeit with slower training. In practice, we set the coefficient to 0.3 to balance speed and quality effectively; (3) The middle layers exhibit greater redundancy, while the later layers generally have less redundancy and often cannot be compressed. The early layers show variable redundancy levels.

3.4. Enabler 3: Timestep-wise Differentiable Ratio

Motivation. In addition to layer-wise ratio variances, we also observe that the model exhibits varying levels of redundancy across timesteps. This motivates us to explore an approach for timestep-wise compression ratios as well.

Timestep-wise Differentiable Ratio. On top of the layer-wise DiffCR, we introduce learnable parameters specific to different timestep regions. For the image inpainting task, following the previous SOTA LazyDiffusion [26], we use 1,000 training timesteps and 100 sampling timesteps, which we evenly divide into 10 regions. We assign 10 learnable ratios per layer accordingly, resulting in a total of 280 learnable parameters. Similarly, for T2I tasks, following PixArt-Σ [4] with 20 timesteps, we divide them into 4 regions, assigning 4 learnable ratios per layer, yielding 112 learnable parameters. The same as before, we apply an MSE loss between the averaged learned ratios within the batch and the target ratio to ensure convergence.

Ratio Pattern Analysis. We visualize the learned compression ratio patterns across both timesteps and layers in Fig. 5. For both image inpainting and T2I tasks, we

consistently observe that noisy timesteps (corresponding to earlier sampling timesteps or later training timesteps) exhibit higher redundancy and allow for higher compression, whereas timesteps where images become clearer (corresponding to later sampling timesteps or earlier training timesteps) show less redundancy. This learned pattern aligns with previous empirical findings [44], which suggest that high-noise timesteps are associated with convergence regions containing easier samples, allowing for fewer sampling timesteps, while low-noise timesteps involve harder samples and require more frequent sampling.

4. Experiments

4.1. Experiment Settings

Tasks, Datasets, and Models. *Tasks & Datasets.* We evaluate our DiffCR on two representative image generation tasks using corresponding benchmark datasets: (1) an image inpainting task on an internal dataset of 220 million high-quality images, covering diverse objects and scenes. Masks and text prompts are generated following [26, 51]; and (2) a T2I task on the LAION-5B dataset [36], restricted to image samples with high aesthetic scores, English text, and a minimum text similarity score of 0.24. *Models.* We integrate our proposed DiffCR approach with SOTA models. For the inpainting task, we use Lazy Diffusion (an adapted PixArt-α model with an additional ViT encoder) to generate images at 1024×1024 resolution. For the T2I task, we use PixArt-Σ to generate images at 512×512 resolution.

Training and Sampling Setting. For the inpainting task, we fine-tune the model parameters until convergence using the AdamW optimizer [18] with a learning rate of 10^{-4} and weight decay of 3×10^{-2}. For sampling, images are generated using IDDPM [25] with 100 timesteps and a CFG factor of 4.5. For the T2I task, we fine-tune the model using a LoRA adapter with a rank of 32 until both training and validation losses converge, and the MSE loss between the current and target compression ratios drops to approximately zero for DiffCR models. During training, we calculate the diffusion loss with IDDPM [25] over 1K timesteps. For sampling, we generate images using DPM-solver [19] with 20 timesteps and a CFG factor of 4.5. All training is conducted on a cluster of 8×A100-80GB GPUs.

Baselines and Evaluation Metrics. *Baselines.* For both the T2I and inpainting tasks, we compare the proposed DiffCR against SOTA baselines, including ToMe [2], AT-EDM [43], and our adapted MoD with uniform MoD compression ratio. For the inpainting task, we also compare against *RegenerateCrop*, which generates a tight square crop around the masked region, similar to popular software frameworks [42, 48], and *RegenerateImage*, which generates the entire image, as commonly done in the literature [28, 32, 45, 51]. *Evaluation Metrics.* We assess the generated image quality using FID scores [11], text-image alignment using CLIP scores [10], and efficiency through inference FLOPs, latency, and memory usage, all measured on an A100 GPU. For inpainting and T2I models, we evaluate on 10K images from LAION-400M [35] or LAION-5B [36], excluding training samples, respectively.

4.2. DiffCR over SOTA Baselines

Text-to-Image. To assess the effectiveness of our proposed DiffCR, we apply our proposed DiffCR to the general text-to-image task and compare it with previous token merging [2] and pruning [43] baselines. Specifically, we apply these compression methods on PixArt-Σ, a SOTA publicly accessible T2I model known for its high-resolution image generation quality and efficiency tradeoffs. As shown in Tab. 1, PixArt-Σ with DiffCR significantly improves generation quality, achieving 57.83 and 241.11 FID reductions over ToMe [2] and AT-EDM [43], respectively, with comparable or even lower latency (↓8.59%∼20.15%) and memory usage (↓-2.71%∼0.72%). Also, under similar latency compared to ToMe [2] with 20% compression ratio, DiffCR achieves 335.23 FID reductions. Moreover, PixArt-Σ with DiffCR also achieves comparable image generation quality to uncompressed PixArt-Σ, while delivering 20.68% and 8.33% latency and memory savings. Note that we compare with fine-tuned PixArt-Σ on the LAION datasets for a fair comparison. This set of experiments demonstrates the effectiveness of DiffCR for general T2I tasks.

Image Inpainting. We further extend DiffCR to the inpainting task. Specifically, we apply it on top of the SOTA Lazy Diffusion (LD) [26], which uses a DiT decoder to generate only the masked areas rather than the entire image, leveraging a separate ViT encoder to capture the global context of the input masked images. We compare our DiffCR approach against two types of baselines: (1) RegenerateImage and RegenerateCrop, and (2) LD with previous token merging [2] or pruning [43] techniques. As shown in Tab. 2, our DiffCR consistently outperforms all baselines in terms of accuracy-efficiency tradeoffs. For example, LD with DiffCR achieves FID reductions of 47.35 and 189.93 compared to LD with ToME [2] or AT-EDM [43], while achieving similar or up to 23.61% and 13.63% higher latency and memory savings. Also, under similar memory usage compared to ToMe [2] with 30% compression ratio, LD with DiffCR achieve 265.94 FID reduction while delivering up to 21.54% latency savings. Moreover, compared to RegenerateImage, our method achieves 73.51%/60.26% FLOPs and latency savings when inpainting 256^2 mask sizes within 1024^2 images. Notably, like Lazy Diffusion, our method's complexity scales with mask size, while RegenerateImage generates based on full image resolution, making it less efficient for smaller mask sizes. In comparison to RegenerateCrop, our method achieves significantly higher image generation quality (+41.01 FID) while also delivering 22.96%

Table 1. Quantitative comparison of DiffCR with other baselines on the T2I task. All experiments are fine-tuned from the pre-trained PixArt-Σ [4] on the LAION-5B [36] dataset. C.R. denotes compression ratios. We report FID (↓) and CLIP Score (↑) on 10K images (excluding training samples) as quality metrics and measure FLOPs (↓), latency (↓), and memory (↓) on an A100 GPU as efficiency metrics under batch sizes of 16 and 128. Memory is averaged across all layers. "-L" and "-LT" indicate layer-wise or layer/timestep-wise DiffCR. "TF" denotes training-free methods.

Methods	DiT C.R.	Quality		DiT Efficiency (512^2; BS = 16)			DiT Efficiency (512^2; BS = 128)		
		FID	CLIP Score	FLOPs (G)	Lat. (s)	Mem. (GB)	FLOPs (G)	Lat. (s)	Mem. (GB)
PixArt-Σ [4]	0%	151.0	0.173	17361.7	225.38	1.798	138893.9	1784.64	13.107
PixArt-Σ (Fine-tuned)	0%	11.93	0.242	17361.7	225.38	1.798	138893.9	1784.64	13.107
PixArt-Σ w/ ToMe (TF) [2]	10%	68.51	0.211	16391.4	224.04	1.674	131131.6	1772.84	12.087
PixArt-Σ w/ ToMe (TF) [2]	20%	345.91	0.122	15421.2	213.79	1.546	123369.2	1681.83	11.078
PixArt-Σ w/ AT-EDM (TF) [43]	20%	251.79	0.129	15132.7	196.88	1.617	121061.2	1548.80	11.870
PixArt-Σ w/ **MoD**	20%	22.78	0.207	13949.3	178.89	1.659	111594.2	1402.96	11.987
PixArt-Σ w/ **DiffCR-L**	20%	12.28	0.232	13967.1	180.84	1.662	111737.0	1425.30	12.015
PixArt-Σ w/ **DiffCR-LT**	20%	10.68	0.238	13957.2	179.71	1.664	111657.3	1415.63	12.021

Table 2. Quantitative comparison of DiffCR with other baselines on the inpainting task. Scores for SDXL [28] are provided for reference only and are not directly comparable. C.R. denotes compression ratios. We report FID (↓) [11] and CLIP Score (↑) [10] on 10K images from LAION-400M [35] as quality metrics, and measure FLOPs (↓), latency (↓), and memory usage (↓) on an A100 GPU as efficiency metrics for two inpainting mask sizes (512^2 and 256^2 within 1024^2 images). "-L" and "-LT" indicate layer-wise or layer/timestep-wise DiffCR. "TF" denotes training-free methods.

Methods	ViT En. C.R.	DiT De. C.R.	Quality		DiT Efficiency (512^2 within 1024^2)			DiT Efficiency (256^2 within 1024^2)		
			FID	CLIP Score	FLOPs (G)	Lat. (s)	Mem. (GB)	FLOPs (G)	Lat. (s)	Mem. (GB)
SDXL [28]	N/A	0%	6.37	0.2112	5979.5	66.09	OOM	5979.5	66.09	OOM
RegenerateImage	N/A	0%	9.53	0.1942	809.8	13.11	OOM	809.8	13.11	OOM
RegenerateCrop	N/A	0%	54.43	0.1737	267.3	4.30	45.00	199.2	4.19	13.66
Lazy Diffusion (LD) [26]	0%	0%	10.90	0.1882	1085.1	17.13	45.00	285.8	5.68	13.66
LD w/ Model Pruning	50%	30%	27.36	0.1796	775.5	15.33	45.00	204.5	5.37	13.66
LD w/ ToMe (TF) [2]	50%	10%	60.77	0.1756	979.0	16.87	40.14	259.7	6.82	13.01
LD w/ ToMe (TF) [2]	50%	30%	279.36	0.1496	765.7	16.82	32.61	206.7	6.64	11.14
LD w/ AT-EDM (TF) [43]	50%	30%	203.35	0.1459	751.4	15.49	32.94	202.8	6.11	11.91
LD w/ **MoD**	50%	30%	18.34	0.1850	764.6	13.02	32.58	205.1	4.83	11.11
LD w/ **DiffCR-L**	50%	30%	13.53	0.1839	822.9	13.71	34.67	220.4	5.30	11.30
LD w/ **DiffCR-LT**	50%	30%	13.42	0.1845	804.9	13.89	34.88	214.5	5.21	11.52

memory savings. Note that all memory measurements are taken with a batch size of 128. This set of experiments validates the effectiveness of our DiffCR when applied to image inpainting tasks.

4.3. Ablation Studies of DiffCR

We conduct ablation studies on DiffCR, analyzing the contributions of the three enablers described in Sec. 3. As shown in Tabs. 1 and 2, we report the performance of LD or PixArt-Σ with MoD routers (Sec. 3.2), DiffCR-L (Sec. 3.3), DiffCR-LT (Sec. 3.4) for T2I and inpainting tasks, respectively. The results consistently show that all components of our DiffCR contribute to the final performance. Specifically, MoD alone achieves average FID reductions of 323.13 and 229.01 compared to ToMe [2] and AT-EDM [43] for the T2I task, with comparable or even lower latency and memory usage. DiffCR-L and DiffCR-LT further enhance the generation quality, achieving additional FID reductions of 4.81/4.92 for the inpainting task and 10.5/12.1 for the T2I task.

Also, a key benefit of our DiffCR is that during fine-tuning, the average compression ratios across all layers gradually converge to the target ratio, producing a series of "by-product" models with a range of compression ratios. As shown in Fig. 7, we visualize the model trajectory with corresponding FID scores and compression ratios for both inpainting and T2I tasks. The observations indicate that, for T2I, the FID gradually increases with compression ratio, achieving the desired results. In contrast, for inpaint-

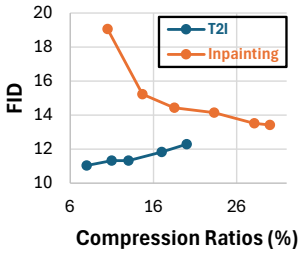


Figure 7. Model trajectories.

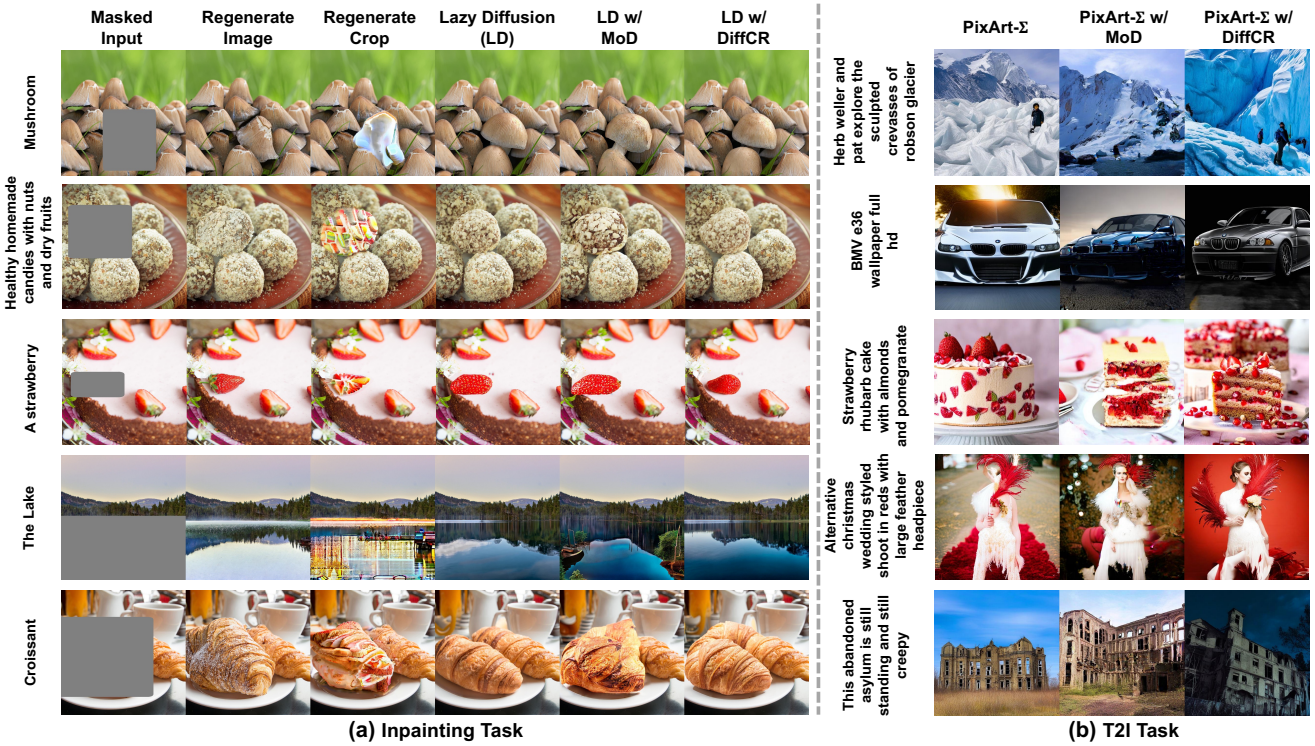


Figure 6. Visual comparisons of our DiffCR with previous uncompressed models and SOTA compression methods: (a) Inpainting tasks, where DiffCR is applied to the LD models [26], and (b) T2I tasks, where DiffCR is applied to PixArt-Σ [4].

ing, the FID gradually decreases. This difference arises because, compared to T2I, inpainting tasks and LD models are more sensitive to pruning and require longer fine-tuning to boost the generation quality. This is also reflected in Tabs. 2 and 1, where FID increases post-pruning for inpainting, while it even decreases for T2I.

4.4. Qualitative Visual Examples

Visual Examples. We select challenging input prompts to evaluate the qualitative results of our proposed DiffCR. As shown in Fig. 6, the examples demonstrate that DiffCR achieves comparable or even superior generation quality compared to the RegenerateCrop baseline and even uncompressed LD or PixArt-Σ for inpainting and T2I tasks, respectively. Note that ToMe and AT-EDM are omitted here due to their poor generation quality when applied to DiTs, even at a mere 10% compression ratio.

Human Preference Scores. We use a computer vision model to estimate likely human preferences and assess the models' ability to generate high-quality, contextually relevant images. Specifically, we generated 2K samples for the T2I task and used HPSv2 [49] to evaluate human preferences for images generated by different methods. As shown in Tab. 3, for T2I, we apply all compression methods to PixArt-Σ [4]. DiffCR achieves a higher human preference score of 4.685/0.847 compared to previous compression methods, ToMe [2] and vanilla MoD [31], respectively.

Table 3. Human preference score (HPS) (↑) comparison of the proposed DiffCR with other baselines on the T2I task.

Methods	DiT C.R.	HPS Score
PixArt-Σ (Fine-tuned)	0%	22.582
PixArt-Σ w/ ToMe	20%	16.742
PixArt-Σ w/ MoD	20%	20.580
PixArt-Σ w/ DiffCR	20%	21.427

5. Conclusion

In this work, we present DiffCR, a dynamic DiT inference framework with differentiable compression ratios that adaptively routes computation across tokens, layers, and timesteps, resulting in efficient DiT models. Specifically, DiffCR incorporates a token-level routing scheme based on MoD that dynamically learns the importance scores for each token, alongside a novel module that makes MoD differentiable with respect to compression ratios, enabling the model to learn adaptive compression ratios for each layer and timestep. Redundant layers and timesteps learn higher compression ratios, while critical layers and timesteps remain minimally compressed or uncompressed. Extensive experiments on both image inpainting and text-to-image (T2I) tasks consistently demonstrate DiffCR's superior trade-off between image generation quality and efficiency compared to other compression works.

Acknowledgment

The work is supported in part by an internship at Adobe and in part by CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023. 1, 2

[2] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023. 1, 2, 4, 6, 7, 8

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 1, 2, 5

[4] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 1, 2, 4, 5, 7, 8

[5] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17164–17174, 2023. 4

[6] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020. 2

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 4

[9] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*, 2023. 1, 2

[10] 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, 7

[11] 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, 7

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[14] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 2

[15] Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. *ECCV 2024*, 2024. 2

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 2

[17] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. *arXiv preprint arXiv:2305.15798*, 2023. 1, 2

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*, 2019. 6

[19] 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. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 6

[20] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *arXiv preprint arXiv:2406.01733*, 2024. 2

[21] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 1, 2

[22] Taehong Moon, Moonseok Choi, EungGu Yun, Jongmin Yoon, Gayoung Lee, and Juho Lee. Early exiting for accelerated inference in diffusion models. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. 2

[23] Giovane CM Moura, John Heidemann, Ricardo de O Schmidt, and Wes Hardaker. Cache me if you can: Effects of dns time-to-live. In *Proceedings of the Internet Measurement Conference*, pages 101–115, 2019. 2

[24] Ravi Teja Mullapudi, William R Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2018. 2

[25] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings*

of the 38th International Conference on Machine Learning, pages 8162–8171. PMLR, 2021. 6

[26] Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michaël Gharbi. Lazy diffusion transformer for interactive image editing. *arXiv preprint arXiv:2404.12382*, 2024. 1, 2, 4, 5, 6, 7, 8

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 2

[28] 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. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 6, 7

[29] Yifan Pu, Zhuofan Xia, Jiayi Guo, Dongchen Han, Qixiu Li, Duo Li, Yuhui Yuan, Ji Li, Yizeng Han, Shiji Song, et al. Efficient diffusion transformer with step-wise dynamic attention mediators. *arXiv preprint arXiv:2408.05710*, 2024. 2

[30] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1

[31] David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. 2, 3, 8

[32] 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, 2, 6

[33] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 1, 2

[34] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2025. 2

[35] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6, 7

[36] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 6, 7

[37] Ethan Smith, Nayan Saxena, and Aninda Saha. Todo: Token downsampling for efficient generation of high-resolution images. *arXiv preprint arXiv:2402.13573*, 2024. 1, 2

[38] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2

[40] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE, 2016. 2

[41] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2

[42] 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

[43] Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16080–16089, 2024. 1, 2, 6, 7

[44] Kai Wang, Yukun Zhou, Mingjia Shi, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. *arXiv preprint arXiv:2405.17403*, 2024. 1, 2, 6

[45] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18359–18369, 2023. 6

[46] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 409–424, 2018. 2

[47] Yue Wang, Jianghao Shen, Ting-Kuei Hu, Pengfei Xu, Tan Nguyen, Richard Baraniuk, Zhangyang Wang, and Yingyan Lin. Dual dynamic inference: Enabling more efficient, adaptive, and controllable deep inference. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):623–633, 2020. 2

[48] Stable Diffusion WebUI. https://github.com/AUTOMATIC1111/stable-diffusion-webui, 2023. Accessed: 2024-11-10. 6

[49] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 8

[50] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8817–8826, 2018. 2

[51] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 6

[52] 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. 2

[53] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024. 2

[54] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 1, 2

[55] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022. 1

[56] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. *arXiv preprint arXiv:2410.03456*, 2024. 2