ONE MODEL, MANY BUDGETS: ELASTIC LATENT INTERFACES FOR DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion transformers (DiTs) achieve high generative quality but lock FLOPs to image resolution, hindering principled latency-quality trade-offs, and allocate computation uniformly across input spatial tokens, resulting in wasteful resource allocation to unimportant regions. We introduce Elastic Latent Interface Transformer (ELIT), a drop-in, DiT-compatible mechanism that decouples input image size from compute. Our approach inserts a latent interface, a learnable variable-length token sequence on which standard transformer blocks can operate. The latent interface works through lightweight Read and Write cross-attention layers that move information between spatial tokens and latents to prioritize the most important input regions. Additionally, by training with random dropping of tail latents, this module learns to produce importance-ordered representations with earlier latents capturing global structure while later latents contain information to refine details. At inference, the number of latents can be dynamically adjusted to match time or compute constraints by focusing its capacity on "hard" regions. Our proposed approach is deliberately minimal, leaving the rectified flow objective and the DiT stack unchanged, and adding only two cross-attention layers. Across datasets and architectures (DiT, U-ViT, HDiT, MM-DiT), ELIT delivers consistent gains. On ImageNet-1K 512px, ELIT delivers an average of 35.3% and 39.6% improvement in FID and FDD scores over baselines.

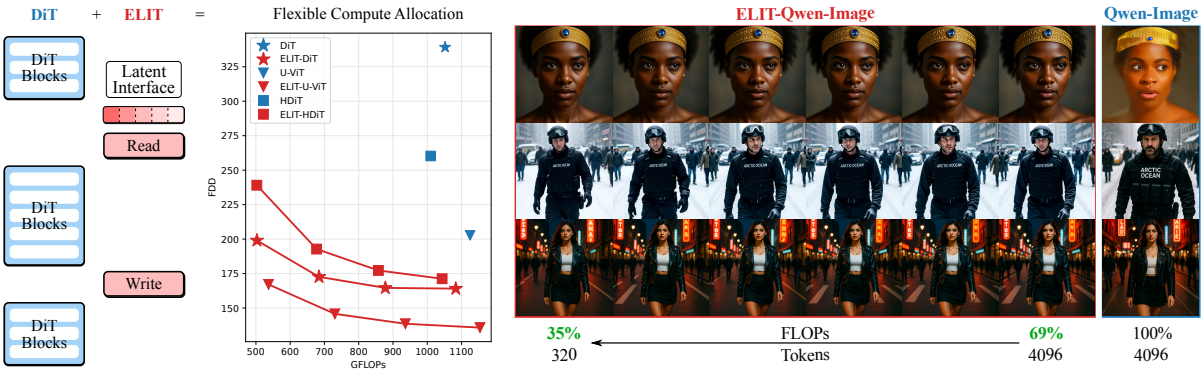

Figure 1: **Flexible compute allocation with ELIT.** Starting from a vanilla DiT, we add a variable-length set of latent tokens—the *latent interface*—and two lightweight cross-attention layers, Read and Write, enabling higher quality at adjustable per-step compute. At inference, the number of latent tokens is a user-controlled knob that yields a smooth quality–FLOPs trade-off across DiT, U-ViT, HDiT, and MM-DiT backbones.

1 INTRODUCTION

Recent years have seen dramatic progress in image and video generation. Owing to the simplicity of their design, architectures centered on Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have scaled reliably and delivered state-of-the-art fidelity (Wu et al., 2025; Wan et al., 2025). Compute has been the primary determinant of generation quality, but continued scaling has inflated training and inference costs. This raises a central question: do DiTs utilize available computation effectively? We argue that such costs are amplified by the *rigidity* in the DiT design. First, a DiT typically commits to a per-step computational cost that is a fixed function of the input resolution, without accounting for latency and compute constraints. Second, we found that DiT allocates computation uniformly across image regions. In a controlled experiment, we probe the ability of a DiT to use extra compute to improve generation quality. As expected, quality improves on standard images when we increase the number of tokens by reducing the patchification size. However, when we increase the number of tokens by padding encoded image patches with zero-valued patches, we find that DiT fails to leverage the extra computation to improve generation quality. These observations suggest that compute is spent *uniformly* across image tokens. This is suboptimal since visual information in images is uneven: some regions are easy, others might require more

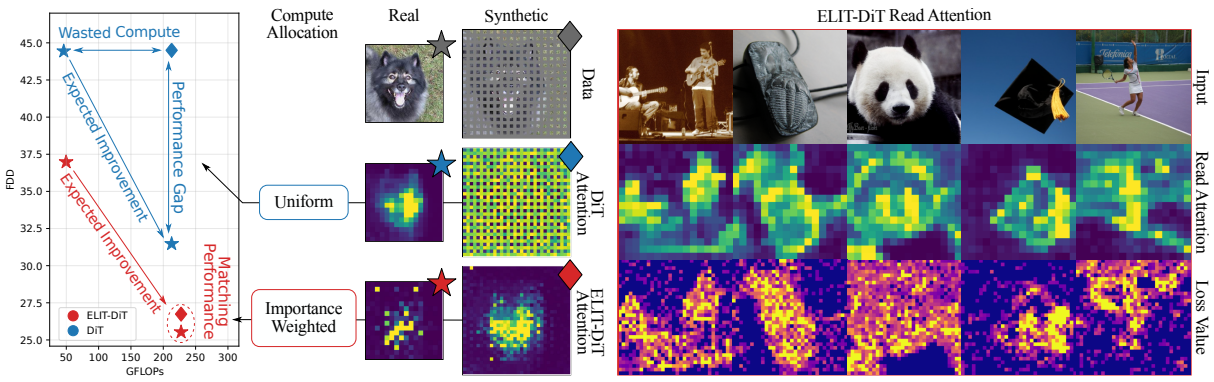

Figure 2: **Adaptive computation.** We test whether DiT and ELIT-DiT can reallocate compute across image regions by training on synthetic inputs formed by zero-padding real images, artificially increasing the token count (◆). We compare its performance to baselines trained on real data using patch size 2×2 and patch size 1×1 (★). Vanilla DiT does not improve: attention in zeroed regions targets other zeroed regions (see "DiT Attention"), so extra tokens raise cost without benefits. In contrast, ELIT-DiT uses the Read layer to pull informative spatial tokens into the latent interface (see "Read Attention"), effectively filtering out zeroed areas (see "ELIT-DiT Attention"). Consequently, it leverages the larger token budget and matches the real-data baseline at equal FLOPs.

work. In this context, *learning* how to allocate computation across tokens through a flexible architecture holds the potential to dynamically reduce unnecessary computation.

Important previous work has sought forms of flexibility to make generative models more efficient. Adaptive generators allow a *single* model to operate at different budgets, but still spread computation uniformly across tokens (Anagnostidis et al., 2025; Yu et al., 2019) or suffer from high complexity (Zhao et al., 2025).

Masking-based methods improve training speed by skipping tokens but disable token dropping for inference to avoid unrecoverable information loss (Gao et al., 2023; Krause et al., 2025). Some training-free acceleration approaches reduce computation for the least informative tokens, but compute redistribution does not benefit training (Liu et al., 2025c; Yang et al., 2025; Jeong et al., 2025). A complementary thread moves flexibility in the autoencoder by learning variable-length representations but stop short of endowing a *generative* model with an internal flexible representation (Bachmann et al., 2025; Li et al., 2025). Finally, RINs (Jabri et al., 2023; Chen & Li, 2023) learn to distribute computation non-uniformly across input tokens through a set of latent tokens, but keep their inference budget fixed and depart significantly from the DiT architecture, hindering adoption.

Building on the previous observations, we propose Elastic Latent Interface Transformer (ELIT) (see Figure 1), a minimal, DiT-compatible mechanism for representation-and-compute adaptivity. We introduce two lightweight cross-attention layers, *Read* and *Write*, that equip DiT-like architectures with a set of latent tokens we refer to as *latent interface*. These latent tokens act as a variable-size surface onto which to distribute input information in a flexible and learnable manner based on the difficulty of each region. *Read* pulls information from input tokens, which we will refer to as *spatial*, into the latent interface, prioritizing difficult input regions. *Write* broadcasts the updated latent state back to spatial tokens. Importantly, the number of latent tokens is user-controlled and sets the per-step compute budget. This capability is achieved during training, by randomly dropping a set of latent tail tokens, so that the model learns an *importance-ordered* latent representation: the first tokens capture globally useful information, while later tokens refine details. No changes to the training objective are necessary.

We provide a thorough analysis of ELIT, giving insights into its main design choices. ELIT successfully redistributes compute non-uniformly across input tokens across varying base architectures. Our latent interface consistently improves over a fixed-grid model with ImageNet-1k 512px FDD (Fréchet Distance on DINOv2 (Oquab et al., 2024) features), improving 58.0% for DiT, 34.0% for U-ViT, 37.4% for HDiT. ELIT allows for graceful compute-quality tradeoffs by selecting the number of latent tokens at inference time, regularly achieving better tradeoffs than sampling steps reduction while being compatible with training-free acceleration techniques (Liu et al., 2025b). Additionally, variable compute enables autoguidance (Karras et al., 2024) out of the box, which reduces inference cost by $\approx 33\%$ without affecting the generation quality.

In summary, this simple addition yields a framework capable of: (I) **Adaptive computation.** Compute is concentrated where it matters rather than spread uniformly across input regions. (II) **Variable test-time compute.** A single set of weights serves a spectrum of latency-quality points by selecting the number of latent tokens. (III) **Improved sampling.** Variable compute enables autoguidance (Karras et al., 2024) out of the box. (IV) **Drop-in training.** We keep the vanilla rectified-flow objective, showing our method's applicability to DiT, U-ViT, HDiT and MMDiT. Implementation amounts to adding the Read and Write layers and latent tokens sampling during training.

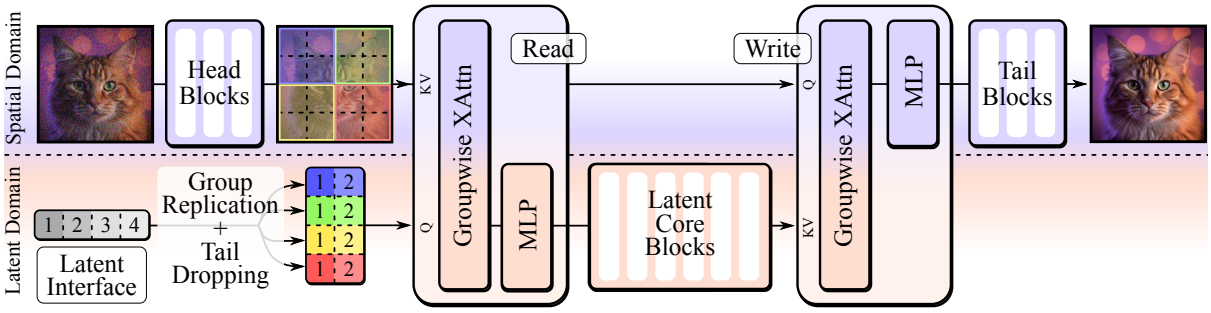

Figure 3: **Architecture of ELIT.** We equip a DiT-like generator with a variable-length set of latent tokens—the *latent interface*—and lightweight Read and Write cross-attention layers. A short spatial head processes the patchified input; Read pulls information from spatial tokens into the latent interface, creating a latent domain where core blocks operate. After latent processing, Write broadcasts the updated latent state back to spatial tokens, and a small spatial tail produces the output. Cross-attention is grouped: spatial tokens and latents are partitioned into corresponding groups that attend only within-group. During training, we drop latent tail tokens, making the interface importance-ordered. At inference, the number of latents is a user-controlled knob setting compute budget.

2 Related Work

Adaptive generators for inference budget control. Supernetwork designs showed that a single set of weights can support many sub-networks, allowing test-time accuracy–efficiency trade-offs without retraining (Yu et al., 2019; Cai et al., 2020). Other works jointly train transformer models with multiple patchification sizes to operate under different compute budgets (Anagnostidis et al., 2025; Liu et al., 2025a). (Zhao et al., 2025) uses learnable routers to dynamically alter network width and drop MLP input tokens. These methods differ in *where* adaptivity lives but share the goal of a *single* model that gracefully scales compute at inference time. Our method adopts a simple variable-length latent interface that improves model convergence during training while enabling granular control over the inference budget by selecting the amount of latent tokens during inference.

Token dropping for training speedup. A second line of work accelerates model training by skipping tokens. MaskDiT (Zheng et al., 2024) reduces training time by restructuring the DiT architecture as an encoder-decoder model where the encoder input tokens are randomly dropped and subject to an auxiliary reconstruction objective. MDTv2 (Gao et al., 2023) similarly leverages masked latent modeling to strengthen synthesis while training on partial inputs. TREAD (Krause et al., 2025) randomly selects a set of tokens that skip computation of all blocks from a predefined start to an end DiT block index. Due to the destructive nature of token dropping, which eliminates information for the corresponding spatial locations, such methods typically rely on auxiliary losses (Zheng et al., 2024), full-token finetuning (Zheng et al., 2024; Krause et al., 2025) and adopt full-token inference (Zheng et al., 2024; Gao et al., 2023; Krause et al., 2025), limiting acceleration during training. Our method speeds up convergence, enables variable amounts of tokens to be selected at inference time to control the inference budget, and does not necessitate any auxiliary loss or finetuning procedure.

Latent interfaces. Latent tokens have been used in varied ways within neural networks. Neural Turing Machines (Graves et al., 2014) use them as a memory to store and retrieve information, Perceiver (Jaegle et al., 2021) condenses a high-dimensional input into latent vectors through cross attention operations. In the context of generative models, RINs (Jabri et al., 2023) and FITs (Chen & Li, 2023) introduced a series of interleaved read and write operations based on cross attention within an ad-hoc transformer architecture, allowing efficient generation of high-dimensional visual modalities by focusing computation on the smaller set of latent tokens. (Menapace et al., 2024) demonstrated their scalability to high-resolution text-to-video generation. Despite the ability of these methods in efficiently handling high-dimensional inputs, we posit that they may not have seen widespread adoption due to their architectural departure from the DiT architecture and the requirement of LAMB (You et al., 2020) optimization. The same idea was recently brought to autoencoders by TiTok (Yu et al., 2024), where latent tokens are concatenated with input tokens before a standard ViT encoder and later retained as the sole bottleneck. Subsequent work extend this idea to variable-length token sets by adopting the "Tail dropping" technique (Koike-Akino & Wang, 2020) (e.g. (Bachmann et al., 2025; Li et al., 2025; Yan et al., 2025)). Our work brings for the first time within generative models the representational benefits of variable-length latent interfaces and integrates them within existing architectures (i.e. DiT) with the only addition of a single Read and Write layer.

3 Method

We propose Elastic Latent Interface Transformer (ELIT) (see Figure 3), a novel approach that enables flexible compute allocation both across spatial locations and subsequent model calls in DiT-like transformers. The core

component is a variable-length set of latent tokens—the *latent interface*—where most transformer blocks operate. Two lightweight cross-attention layers, Read and Write, pass information between domains: Read pulls content from spatial tokens into the latent interface, prioritizing harder regions, while Write broadcasts the updated latent state back to the spatial domain. Unlike the spatial domain, where model FLOPs are a fixed function of input resolution, the latent interface is trained with random tail token dropping, making it resizable. The length of this latent interface is set by the user during inference to dynamically adjust the FLOPs for each model call.

Section 3.1 reviews Flow Matching, Section 3.2 presents a motivating experiment, Section 3.3 details the architecture, and Section 3.4 describes training and sampling.

3.1 Preliminaries: Flow Matching

We train our generators with the Rectified Flow (RF) variant of flow matching (Liu et al., 2023; Lipman et al., 2023), which learns a deterministic velocity field connecting a noise distribution p_n to the data distribution p_d. Let $\mathbf{X}_1 \sim p_d$ and $\mathbf{X}_0 \sim p_n = \mathcal{N}(0, \mathbf{I})$. A linear path is defined as $\mathbf{X}_t = (1-t)\,\mathbf{X}_0 + t\,\mathbf{X}_1, t \in [0,1]$, whose ground-truth velocity is constant along the path: $v_t = \frac{d\mathbf{X}_t}{dt} = \mathbf{X}_1 - \mathbf{X}_0$. A neural network $\mathcal{G}(\cdot)$ predicts the velocity from a noised input and time and is optimized as follows:

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{t\sim p_t,\, \mathbf{X}_1\sim p_d,\, \mathbf{X}_0\sim p_n} \left\| \mathcal{G}(\mathbf{X}_t,\, t) - (\mathbf{X}_1 - \mathbf{X}_0) \right\|_2^2, \tag{1}$$

where p_t is a training distribution over t chosen as a logit-normal (Esser et al., 2024). At inference, samples are obtained by integrating the learned ODE from $\mathbf{X}_0$ to $\mathbf{X}_1$ with a standard solver.

3.2 Uniform Spatial Computation in Diffusion Transformers

Standard DiTs operate in the spatial domain where an input $\mathbf{X}_t$ at time t is patchified by a linear projection layer into N tokens and processed by B transformer blocks. Each block output is connected to the previous block through a residual connection at the same spatial location before being projected by an output layer to a velocity estimate $\hat{v}$. This structure maintains a fixed mapping between tokens s in intermediate transformer blocks and the corresponding spatial location of $\mathbf{X}_t$, resulting in uniform compute distribution across spatial locations in $\mathbf{X}_t$.

We probe this behavior through an experiment presented in Figure 2. We first train a DiT-B/2 (i.e., using 2×2 patchification) and a corresponding DiT-B/1 (i.e., using 1×1 patchification) on ImageNet-1k. As expected, performance improves due to the fourfold amount of tokens in DiT-B/1. We then introduce a synthetic ImageNet-1k dataset version, constructed by padding encoded images with zeros, making the number of tokens four times larger. We train a DiT-B/2 on this dataset, named DiT-B/2-Synth, without imposing loss on padded regions to ensure zero importance is assigned to them. We also employ learnable positional encodings instead of standard Rotary Position Embedding in these experiments to avoid biasing image patches toward attending to neighboring zero regions. For evaluation, we remove padded regions before decoding to recover the real image content. This baseline matches the number of tokens and training FLOPs of DiT-B/1. Thus, if compute were used effectively, it should match DiT-B/1 performance. Instead, as shown in Figure 2, DiT-B/2-Synth matches the original DiT-B/2 by FID, indicating no benefit from the extra compute introduced.

Visualizations of attention maps in Figure 2 reveal that in DiT-B/2-Synth, zeroed-out tokens primarily attend to each other instead of important image regions, effectively wasting compute as no loss is imposed on them. We conclude that DiT cannot reallocate computation from easy to hard regions. While demonstrated in a synthetic setting, the same inefficiency likely arises in natural images, where different spatial regions have diverse difficulty (lower or higher loss value) and would benefit from an architecture capable of distributing compute accordingly.

3.3 Elastic Latent Interface Transformer

From spatial tokens to a variable latent interface. To allow flexible distribution of computation in DiTs, we introduce a minimal change that eliminates the fixed mapping between tokens and their image patches, as shown in Figure 3. We create a latent domain by instantiating a *latent interface* of K tokens. To map the original spatial domain to the new latent domain, we use a lightweight *Read* cross-attention layer, following Chen & Li (2023). The Read layer enables the model to select how many latent tokens to allocate for each spatial region of $\mathbf{X}_t$, based on their difficulty, forming a compact latent domain on which most transformer blocks operate. As the majority of computation is performed in the latent domain, compute is concentrated on salient regions. Finally, a lightweight *Write* cross-attention layer maps the latent updates back to the spatial grid, allowing the model to write predictions back to their locations and retain direct access to input details.

Architecture. Earlier work has shown that early and late transformer blocks in diffusion models exhibit different specializations compared to the intermediate blocks (Chen et al., 2024a;b; Wang et al., 2025). Therefore, we split the DiT into three segments:

1. **Initial spatial head** (B_{in} blocks). Processes input tokens $s \in \mathbb{R}^{N\times d}$ to produce a refined spatial representation to transfer to the latent interface. This avoids reading from raw patches and provides local mixing.

2. **Latent core** (B_{core} blocks). Variable-length latent sequence $l \in \mathbb{R}^{K\times d}$ drives most computation. We insert a Read layer $\mathcal{R}$ that pulls information from spatial tokens into l, then process l with standard transformer blocks in the latent domain, and finally insert a Write layer $\mathcal{W}$ that broadcasts updated latent information back to spatial tokens.
3. **Final spatial tail** (B_{out} blocks). A small number of spatial blocks completes processing of the written features and project them to the output velocity. This head restores fine spatial detail, noise information, and aligns features to the prediction space of $\hat{v}$.

Read and Write layers. Let $s \in \mathbb{R}^{N\times d}$ be the current spatial tokens after the spatial head and $l \in \mathbb{R}^{K\times d}$ a learnable set of initial latent tokens. The Read layer updates the latent interface via cross-attention from spatial tokens, producing output latents $l_O \in \mathbb{R}^{K\times d}$ as follows:

$$l_{CA} = l \; + \mathrm{CA}\big(\text{Queries} = l;\; \text{Keys, Values} = s\big), \quad l_O = l_{CA} \; + \mathrm{MLP}(l_{CA}). \tag{2}$$

Conversely, the Write layer updates the spatial representation with the results of latent computations, producing updated spatial tokens $s_O \in \mathbb{R}^{N\times d}$ and is fully symmetric. We adopt pre-norm, and use adaLN-Zero (Peebles & Xie, 2023) modulation for Read to keep the interface timestep-aware. To improve stability, we apply QK normalization inside cross-attention operations. No hidden dimensionality expansion is applied to the MLP blocks to reduce the computational overhead.

Grouped cross-attention. To reduce the cost of Read and Write operations, we partition spatial tokens into G non-overlapping groups (e.g., a regular 2D/3D grid for images/videos), and latents are partitioned accordingly in groups of $J = K/G$ latent tokens each. Cross attention operations attend *within* corresponding groups only. This turns the cross-attention cost from $\mathcal{O}(NK)$ into $\mathcal{O}(NK/G)$ (Chen & Li, 2023).

Latent interface initialization. Before the first Read, each group's latents are initialized from a set of learnable positional embeddings, which is reused across groups. This encodes positional information *within* each group. Sharing the initial latents across groups avoids hard-coding the interface to a specific resolution, increasing spatial resolution changes G and N but not the number of learnable latents J. We use RoPE to encode spatial information *across* groups, with the same group-level RoPE phase applied to its latents to anchor them to the group location.

3.4 Elastic Computation with ELIT

Spatial compute redistribution. When applied on DiT (ELIT-DiT), our architecture enables spatial compute redistribution. When repeating the experiment of Section 3.2, ELIT-DiT-B/2-Synth repurposes the compute from zeroed regions to enhance generation in real regions, matching the quality of the baseline with equal compute trained only on the original ImageNet-1k ELIT-DiT-B/1 (see Figure 2). Attention maps of the read operations averaged over all latent tokens confirm this behavior, showing that ELIT-DiT builds its latent representation by focusing on the most informative spatial regions with the highest flow-matching loss.

Multi-budget elastic latent interface. To address the rigidity of DiT with respect to the fixed inference compute at a given resolution, we aim to build a multi-budget model that supports variable inference budgets. Since each latent token summarizes its group information via the read operation, a subset of tokens can still predict for the entire group, enabling budget-adaptive inference. Thus, we train an importance-ordered latent interface, where earlier tokens within each group capture globally useful information and later tokens refine details, so that any prefix of $\tilde{J} <= J$ tokens of the group tokens yields a valid interface associated with reduced computation (see Section D). We enforce this with a simple random prefix-keeping scheme as in Rippel et al. (2014).

At training time, we randomly sample $\tilde{J} \sim \mathrm{Uniform}\{J_{\min}, \ldots, J_{\max}\}$ once per training iteration, defining the training budget for the current iteration. The same value of $\tilde{J}$ is used across all groups. In every Read/Write and latent-core block, we keep only the first $\tilde{J}$ latents of each group and drop the subsequent tail. This process creates a consistent hierarchy where head latents are seen (and trained on) more often, forcing the model into storing the most important information in earlier latents. The generator is trained end-to-end only with the standard RF loss in Equation (1).

Asymmetric compute for improved guidance. Classifier-free guidance (CFG) (Ho & Salimans, 2021) is a cornerstone technique in diffusion model sampling. Given a conditioning signal c and guidance scale λ, CFG redefines the velocity prediction $\hat{v}_{\text{CFG}} = (\lambda+1)\mathcal{G}(\mathbf{X}_t \mid c) - \lambda\mathcal{G}(\mathbf{X}_t \mid \emptyset)$. While improving the quality, this comes at the cost of duplicating the number of model invocations. Recently, AutoGuidance (AG) (Karras et al., 2024) was proposed to improve guidance by replacing the unconditional model with a weak version of itself. While producing consistent improvements, it relies on the availability of a weaker model version, ideally producing artifacts that are similar to the main model (Chen et al., 2025). Weak models are separately trained or obtained through handcrafted model corruptions (Chen et al., 2025; Hong, 2024; Hyung et al., 2025). However, in our multi-budget framework, such model is natively available by varying the inference budget defined by $\tilde{J}$. Thus, we evaluate the main term with budget $\tilde{J}$ and the guidance term with a smaller budget $\tilde{J}_{\text{w}} \leq \tilde{J}$. We find, however, that AG degrades metrics that reward class alignment such as Inception Score. We thus opt to drop the class condition

Table 1: **Comparative performance on ImageNet-1K at 256px and 512px resolutions.** We evaluate FID↓, FDD↓, and IS↑ without (–G) and with 0.25 CFG (+G). TFLOPs (TF) indicate single training iteration TFLOPs. Superscripts show percentage of improvement of ELIT MultiBudget (MB) relative to the baseline.

Model	Params	ImageNet 256×256							ImageNet 512×512						
		TF	FID_{50K} ↓		FDD_{50K} ↓		IS↑		TF	FID_{50K} ↓		FDD_{50K} ↓		IS↑	
		@256	–G	+G	–G	+G	–G	+G	@512	–G	+G	–G	+G	–G	+G
DiT-XL	675M	182	13.0	5.7	346.3	232.9	66.2	115.3	806	18.8	9.5	339.2	233.6	53.0	86.4
⌞ ELIT	698M	188	8.2	**3.8**	**200.2**	**124.5**	93.0	160.1	831	11.1	4.9	175.6	106.1	80.0	134.1
⌞ *MB*	698M	190	**7.8**$^{-40\%}$	**3.8**$^{-33\%}$	203.7$^{-41\%}$	128.6$^{-45\%}$	**99.0**$^{+50\%}$	**167.6**$^{+45\%}$	804	**10.1**$^{-46\%}$	**4.5**$^{-53\%}$	**164.1**$^{-52\%}$	**98.2**$^{-58\%}$	**88.8**$^{+68\%}$	**147.0**$^{+70\%}$
UViT-XL	707M	196	8.3	3.8	220.2	138.0	84.4	145.1	861	11.6	5.3	202.7	125.9	72.5	117.2
⌞ ELIT	730M	202	7.5	3.8	203.8	**130.0**	95.2	159.7	886	8.9	4.2	155.3	94.9	85.8	141.0
⌞ *MB*	730M	204	**7.1**$^{-14\%}$	**3.7**$^{-3\%}$	**203.2**$^{-8\%}$	130.6$^{-5\%}$	**100.3**$^{+19\%}$	**168.2**$^{+16\%}$	858	**7.7**$^{-34\%}$	**3.8**$^{-28\%}$	**135.8**$^{-33\%}$	**83.1**$^{-34\%}$	**98.0**$^{+35\%}$	**159.3**$^{+36\%}$
HDiT-XL	1.4B	182	12.8	5.6	361.6	247.0	68.7	119.7	776	13.0	6.0	260.3	170.5	69.4	114.2
⌞ ELIT	1.4B	188	**9.4**	**4.6**	272.2	**184.4**	89.5	150.2	801	10.1	**4.6**	**164.1**	112.0	88.8	141.0
⌞ *MB*	1.4B	191	**9.4**$^{-27\%}$	**4.6**$^{-18\%}$	**271.8**$^{-25\%}$	185.0$^{-25\%}$	**92.3**$^{+34\%}$	**155.7**$^{+30\%}$	791	**9.6**$^{-26\%}$	**4.6**$^{-23\%}$	171.2$^{-34\%}$	**106.8**$^{-37\%}$	**94.7**$^{+36\%}$	**154.6**$^{+35\%}$

from the guidance term, combining the power of AG and CFG, and name the resulting guidance mechanism cheap classifier-free guidance (CCFG). Consequently, the guidance mechanisms are defined as:

$$\hat{v}_{\text{AG}} = (\lambda+1)\mathcal{G}(\mathbf{X}_t \mid c;\tilde{J}) - \lambda\mathcal{G}(\mathbf{X}_t \mid c;\tilde{J}_{\text{w}}), \quad \hat{v}_{\text{CCFG}} = (\lambda+1)\mathcal{G}(\mathbf{X}_t \mid c;\tilde{J}) - \lambda\mathcal{G}(\mathbf{X}_t \mid \emptyset;\tilde{J}_{\text{w}}). \tag{3}$$

This results in both improved quality and a reduced cost without any retraining or handcrafted model corruptions.

4 Experiments

4.1 Experimental setup

We demonstrate ELIT's broad applicability by evaluating it on several popular diffusion backbones: DiT (Peebles & Xie, 2023), U-ViT (Bao et al., 2023), and HDiT (Crowson et al., 2024). To ensure a fair comparison and evaluate architectural advantages in isolation, all baselines are built with the same base transformer blocks that follow standard DiT block design and adopt the same rectified flow framework. Furthermore, we integrate several improvements across all baselines, including RoPE (Su et al., 2024), and QK normalization (Esser et al., 2024).

Training details. We conduct conditional image and video generation experiments on ImageNet-1k (Deng et al., 2009) and Kinetics-700 (Carreira et al., 2019), respectively. We train on 256px and 512px resolutions for ImageNet-1k experiments and use 29 frames, at 24 fps and 256px resolution for Kinetics-700 experiments. We use the FLUX (Labs, 2024) autoencoder for images and the CogVideo (Yang et al., 2024) autoencoder for videos. Our main experiments are based on DiT-XL/2, while ablation studies use a DiT-B/2 model. Unless noted, we use a batch size of 256, learning rate 1×10^{-4} with 10k warmup steps, gradient clipping at 1.0, Adam (Kingma & Ba, 2015), and EMA with $\beta = 0.9999$. Unless noted, all image experiments are trained for 500k steps, while video experiments are trained for 200k steps.

Evaluation metrics and protocol. We follow the evaluation protocol of Haji-Ali et al. (2025). For images, we report FID (Heusel et al., 2017), FDD (Fréchet Distance on DINOv2 (Oquab et al., 2024) features), and Inception Score (IS) (Salimans et al., 2016). For video, we report FID, FDD, and FVD (Unterthiner et al., 2019). Statistics are computed over 50k samples for main image experiments, while 10k samples are used for all other experiments. We use an Euler sampler with 40 steps unless otherwise noted. We use FLOPs to measure the amount of computation in all experiments and show in Figure 9 the relationship between FLOPs and forward time.

4.2 Comparison to baselines

Baseline selection and details. DiT (Peebles & Xie, 2023) is selected as the base architecture for our experiments. We also consider two derived variants: U-ViT and HDiT. U-ViT (Bao et al., 2023) adds long skip connections akin to U-Net (Ronneberger et al., 2015). We add our proposed read/write operations while keeping the U-ViT design untouched to obtain ELIT-U-ViT. HDiT (Crowson et al., 2024) reduces tokens in intermediate blocks via PixelUnshuffle/PixelShuffle. We use a single 2×2 down-/up-sampling operation at blocks 8 and 20, respectively and double the bottleneck hidden dimensionality to match vanilla DiT FLOPs at the cost of an increased parameter count. We apply ELIT by placing read/write operations and apply downsampling in the latent space by aggregating tokens within the same group, obtaining ELIT-HDiT. Further details are provided in Section A.

Results. To disentangle the effect of dynamic compute redistribution from multi-budget training, we evaluate two variants for each baseline: ELIT, trained with a single budget matching the baselines configurations, and ELIT-MB, trained in a multi-budget setup following the tail-token dropping strategy from Section 3.4. This yields 64 budgets at 512px and 16 budgets at 256px. To account for the compute saved at shorter token lengths, we increase the batch size to 484, keeping training FLOPs comparable. We report per-iteration FLOPs for all baselines.

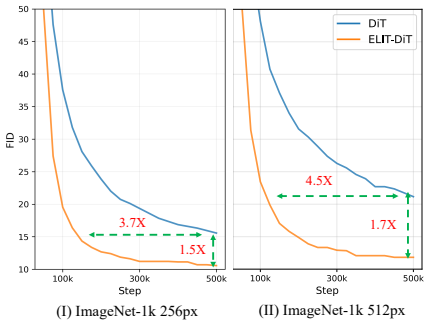

Figure 4: **Training convergence.** ELIT-DiT—with simple read/write operations—significantly accelerates convergence, achieving 3.7× speedup on ImageNet-1K 256px and 4.5× on ImageNet-1K 512px.

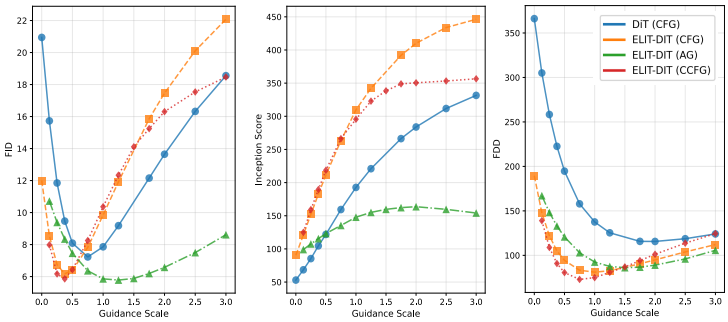

Figure 5: **Guidance strategies.** ELIT enables autoguidance out of the box by providing a well-aligned weaker model that runs at $\approx$ 35% of the cost for the unconditional path. When paired with classifier-free guidance (CFG), denoted as cheap CFG (CCFG), it reduces overall generation cost by $\approx$ 33% while improving quality. Compared to DiT, ELIT-DiT achieves a 19% better best FID.

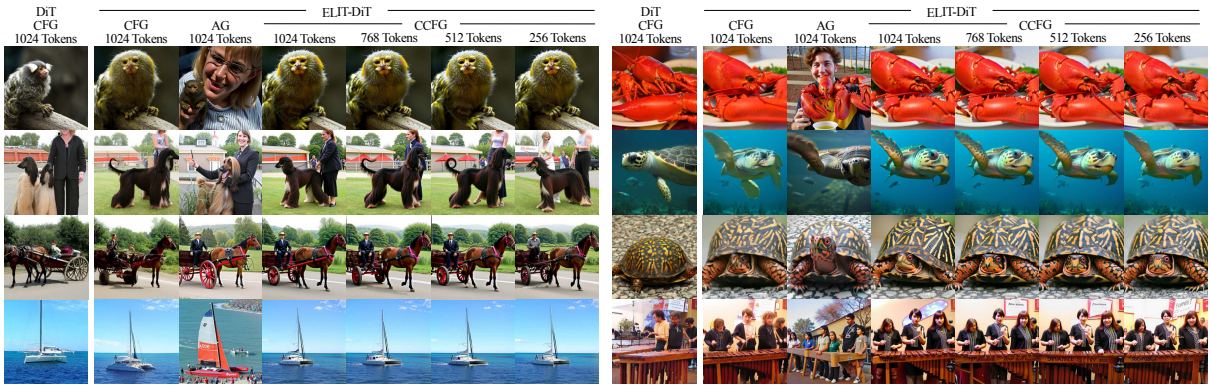

Figure 6: **Qualitative assessment of ELIT on ImageNet-1K 512px.** We compare DiT against ELIT-DiT and ablate over different guidance settings and number of latent tokens. ELIT improves structural details while allowing per-step selection of inference budget through latent token dropping and giving access to autoguidance (AG) and cheap classifier free guidance (CFG) for improved sampling quality and cost. See full results in Section E.

As shown in Table 1, ELIT, despite its simplicity, improves over all baselines under similar training FLOPs. In the multi-budget setting, ELIT-MB delivers further gains, achieving sizable improvements over DiT, U-ViT, and HDiT across all metrics, with FDD reductions of 40%, 14%, and 27%, respectively. We attribute these gains to the importance ordering, which leads to better token semantics while benefiting from the larger effective batch size. The improvements are even more pronounced at 512px, where FDD decreases by 53%, 28%, and 23% for DiT, U-ViT, and HDiT, respectively, suggesting that our method scales favorably with higher resolution, where pixel redundancy is greater and dynamic compute redistribution is more beneficial. We report in Figure 4 the convergence speed of ELIT-DiT relative to DiT at both resolutions, showing faster convergence. Finally, Figure 5 compares performance across CFG values and confirms the advantage of our method under the optimal CFG scale. Finally, Figure 6 qualitatively shows the improvements of our method over DiT. More qualitative examples are provided in Section E.

Furthermore, we validate the performance of ELIT in class-conditional video generation and report the results in Table 2 where we apply grouping of tokens in spatial and temporal dimension. ELIT-DiT shows favorable results over DiT across all metrics.

4.3 Elastic Inference Capabilities

We analyze the ability of the model to perform inference at varied inference budgets using the number of retained latent tokens per group after tail dropping $\tilde{J}$ as a knob to control the inference budget.

Sampling steps trade-off. We compare our approach for reducing inference compute against naively lowering the number of sampling steps. As shown in Figure 7, our method delivers a superior compute–quality trade-off to varying the step count. Notably, for every inference FLOP target, the optimal combination of number of steps and tokens count varies, underscoring the value of models that support a continuum of inference budgets. We

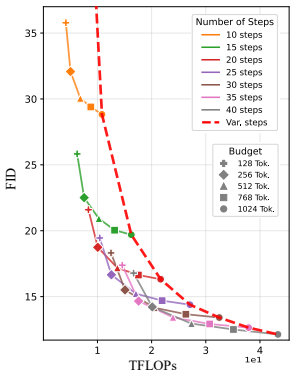

Figure 7: **ELIT compute-quality tradeoff.** Varying inference budget in ELIT-DiT give better quality–compute tradeoff than reducing sampling steps.

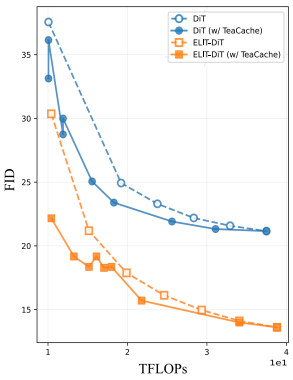

Figure 8: **ELIT with TeaCache.** Our method benefits from TeaCache similarly to DiT, yielding comparable improvements at different inference FLOPs.

Table 2: **Comparative performance on Kinetics-700 256px.** We report metrics without (–G) and with 0.25 CFG (+G).

Baseline	FDD_{10K} ↓		FID_{10K} ↓		FVD_{10K} ↓	
	–G	+G	–G	+G	–G	+G
DiT-XL	371.5	309.1	14.0	11.3	135.9	100.5
⌞ ELIT	**277.4**	**222.0**	**13.3**	**10.7**	**116.5**	**90.5**

Table 3: **Architectural ablations on DiT-B/2.** Using cross-attn in Read/Write is superior to alternatives. Increasing the model capacity is only beneficial in Write and FFN.

Baseline	FID_{10K}↓	FDD_{10K}↓	IS↑
ELIT-DIT	26.53	531.8	45.95
Qformer Read	30.49	589.9	41.10
Self-Attn Read	28.38	602.5	40.12
Self-Attn Read/Write	29.46	631.1	38.49
↑ Read Capacity	27.45	540.7	45.40
↑ Write Capacity	25.23	516.9	47.59
↑ FFN Capacity	**24.80**	**507.7**	**48.22**

also demonstrate compatibility with TeaCache (Liu et al., 2025b) in Figure 8, where our method attains gains comparable to the baseline when TeaCache is applied.

Efficient model guidance. In Figure 5, we compare classifier-free guidance (CFG) with autoguidance (AG) and cheap classifier-free guidance (CCFG). Figure 6 and Section E, show qualitative examples of such guidance strategies. AG achieves comparable performance to CFG while using $\approx 33\%$ fewer FLOPs. Combining AG with CFG by dropping the class condition in the guidance term (*i.e.* CCFG) gives the best results in all metrics and delivers the same $\approx 33\%$ inference speedup compared to using CFG.

4.4 Large Scale Multi-Budget Model

We evaluate the applicability of ELIT to large-scale generative models by applying it on top of Qwen-Image (Wu et al., 2025), which is based on a 20B MM-DiT backbone. We insert the Read and Write layers respectively after block 8 and 52. Due to the asymmetric nature of MM-DiTs and small number of text tokens (≈ 300 on average) versus 4096 image tokens at 1024px, we apply ELIT to the large image tokens stream only. Rather than outperforming the original model, a task which would require access to large-scale curated image datasets and post-training procedures matching to the original ones Wu et al. (2025), the experiment aims to demonstrate that ELIT enables stable training and multi-budget inference for large-scale MM-DiT at high resolution. Therefore, we fine-tune from Qwen-Image in a distillation setting. Specifically, we fine-tune for 60k steps at 512px resolution, using a combination of RF loss and a distillation loss scaled to a similar magnitude. We then fine-tune for an additional 60k steps at 1024px. We train on a combination of real image data and synthetic data generated from FLUX.1-Schnell (Labs, 2024) and SDXL (Podell et al., 2023).

Table 4: Qwen-Image DPG-Bench Results.

Model	Tokens	FLOPs	Entity	Relation	Attribute	Other	Global	Avg.
Qwen-Image		1×	90.51	92.21	91.03	91.34	91.70	91.27
ELIT-Qwen-Image:								
⌞100% Tokens	4096	0.688×	90.30	92.18	88.97	90.34	89.18	90.45
⌞50% Tokens	2048	0.494×	90.15	89.94	89.05	90.09	89.06	89.81
⌞25% Tokens	1024	0.409×	89.31	91.87	89.71	88.28	84.79	89.79
⌞12.5% Tokens	512	0.369×	91.20	90.35	88.77	89.94	79.84	88.02

We perform inference using the Euler sampler with 40 steps and CFG of 5.0 for the original method, while we use the faster CCFG with same weight to further accelerate the inference. Additional details are reported in Section E. Figure 1 and Figure 15 show qualitative results, where ELIT-Qwen-Image cuts sampling FLOPs by up to 63%, achieving $\approx 2.7\times$ sampling speedup while gracefully trading off speed for quality. When quantitatively evaluated on DPG-Bench (Hu et al., 2024), ELIT-Qwen-Image performance remains high across different inference budgets, with the average score ranging from 90.45 to 88.02 at the lowest budget. We include original Qwen-Image results using the same sampling parameters for completeness. With respect to the original Qwen-Image, we observe an initial score gap of 0.82 average score points in our model.

4.5 Ablations

Read/Write strategies. In Table 3, we compare alternative Read/Write designs and find that a single cross-attention Read layer outperforms both a Q-Former–style Read layer (Li et al., 2023) and full self-attention. Additionally, stacking two cross-attention layers in the Read yields no measurable gain, suggesting one layer suffices.

Table 5: **Ablations.** (a) Read/Write group size. (b) Blocks allocation to head-latent core-tail blocks. (c) Strategies for achieving variable budget inference. (d) Batching strategy for multibudget training.

(a) Group Size Ablation

Group Size	Groups	$FID_{10K}\downarrow$	$FDD_{10K}\downarrow$	IS↑
ImageNet-1K 256px (16×16 tokens)				
1×1	256	29.94	638.8	38.39
2×2	64	**25.48**	546.8	45.66
4×4	16	26.53	**531.8**	**45.95**
8×8	4	27.73	552.5	44.64
16×16	1	30.03	599.1	43.44
ImageNet-1K 512px (32×32 tokens)				
1×1	1024	41.67	701.6	27.23
2×2	256	34.50	604.0	33.99
4×4	64	31.60	540.1	37.85
8×8	16	**30.86**	**524.6**	**39.48**
16×16	4	31.93	545.7	38.24

(b) Blocks Allocation Ablation

Block Alloc.	$FID_{10K}\downarrow$	$FDD_{10K}\downarrow$	IS↑
DiT-B/2			
0-12-0	33.84	706.4	34.49
1-10-1	28.55	557.5	41.82
2-8-2	26.53	531.8	45.95
3-6-3	25.37	**531.0**	46.19
4-4-4	**25.34**	560.1	**46.40**
5-2-5	26.95	612.1	43.15
DiT-XL/2			
0-28-0	13.53	333.2	76.07
2-24-2	12.33	239.9	86.87
4-20-4	11.14	**229.6**	93.20
6-16-6	10.84	234.8	93.59
8-12-8	**10.44**	237.3	**95.16**
10-8-10	10.80	250.1	90.11

(c) Variable Budget Strategy

Model	$FID_{10K}\downarrow$	$FDD_{10K}\downarrow$	IS↑
DiT	39.0	779.3	29.2
⌞ DiT + var. patch size	57.36	991.2	20.34
⌞ 25% Tokens	85.25	1181.9	13.06
ELIT-DiT + rand. drop	27.0	540.1	46.3
⌞ 25% Tokens	38.6	718.0	34.5
ELIT-DiT + tail drop	**26.6**	**536.8**	**47.2**
⌞ 25% Tokens	**36.3**	**682.1**	**36.4**

(d) Batching Strategy

Model	$FID_{10K}\downarrow$	$FDD_{10K}\downarrow$	IS↑
(i) variable batch size	**26.15**	537.07	**48.45**
(ii) constant batch size	26.65	**536.83**	47.18

However, adding a second cross-attention layer in the Write or expanding the FFN hidden dimension by $\times 4$ (as in the DiT block) offers improvements at the cost of additional FLOPs. To keep overhead at a minimum, we adopt a single Read/Write layer and keep the reduced FFN width.

Group sizes. The latent group size controls how flexibly the interface can attend over spatial tokens, with larger groups enabling more opportunities for non-uniform compute. As shown in Table 5 (a), dividing the image into 16 groups performs best across 256px and 512px resolutions. Groups of 1×1 force a rigid one-to-one, spatially aligned mapping, while 16×16 spans the full 256px image and underperforms. We hypothesize that using >1 groups provides useful coarse spatial regularization, while still permitting intra-group compute redistribution.

Blocks allocation. In Table 5 (b) We vary $(B_{\text{in}}, B_{\text{core}}, B_{\text{out}})$ for DiT-B/2 and DiT-XL/2, i.e., the spatial head, latent core, and spatial tail blocks count. Optimal performance occurs when $\approx 67\%$ and $\approx 71\%$ of blocks are in the latent core, respectively for DiT-B and DiT-XL, with the remaining split between head and tail. We chose $4-20-4$ for our main experiments on DiT-XL.

Alternative variable-budget strategies. We evaluate other approaches for variable-budget inference in Table 5 (c). Following Anagnostidis et al. (2025); Liu et al. (2025a), we train a DiT with two patchification layers (2×2 and 4×4) sampled uniformly during training, and set the batch size to 48 to match the baseline's training FLOPs. In our experiments, this multi-patchification setup underperforms the standard DiT. We also replace our proposed tail-dropping policy with random latent-token dropping and observe a consistent performance drop.

Training Strategy. At each training step, we sample a compute budget by selecting the number of latent tokens to retain per group, yielding variable per-iteration FLOPs which is on average lower than the baseline. To match baseline training compute, we compare two equalization strategies: (i) varying batch size with the sampled budget with smaller budgets using larger batch sizes (ii) using a constant batch size chosen to match vanilla setup compute in expectation. As shown in Table 5 (d), both behave similarly so we adopt the simpler constant batch size option.

5 Discussion

Limitations. While ELIT demonstrates encouraging results, showing significant improvements in class-conditioned generation for both video and images across resolutions and backbone sizes, we have not yet validated its benefits in a large-scale, from-scratch text-to-image training setting. We fine-tuned Qwen-Image (20B parameters) in a distillation setup to demonstrate finetuning stability and effectiveness of multi-budget inference at this scale. However, corresponding training-time gains remain unverified. Furthermore, we observe that the proposed CCFG saturates images more quickly than CFG and therefore requires lower guidance scales.

Conclusions. We propose ELIT, a novel framework to improve the quality of diffusion transformers and allow a single model to perform inference at different computational budgets. Our framework is minimal, requiring the only addition of two Read and Write layers based on cross attention, a set of latent tokens, and tail token dropping. ELIT consistently improves performance on ImageNet-1k and Kinetics-700 across various base architectures (DiT, U-ViT, HDiT) and resolutions (256px and 512px). Beyond visual quality, a single model that can naturally adjust training and inference compute brings numerous benefits such as cheap CFG and efficient quality-compute tradeoff. Promising future directions include exploration of progressive training budget schedulers, and inference budget schedulers that allocate different budgets across sampling steps, following prior evidence that early sampling steps require less compute (Jin et al., 2024; Anagnostidis et al., 2025).

6 Reproducibility Statement

We strive to make the presented results reproducible by comprehensively describing method details in Section 3 and experimental setup in detail in Section 4.1 and Section A. All main experiments are conducted on publicly available academic datasets (ImageNet-1k (Deng et al., 2009) and Kinetics-700 (Carreira et al., 2019) for ease of reproduction.

References

Sotiris Anagnostidis, Gregor Bachmann, Yeongmin Kim, Jonas Kohler, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Albert Pumarola, Ali Thabet, and Edgar Schönfeld. Flexidit: Your diffusion transformer can easily generate high-quality samples with less compute. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, June 2025.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *International Conference on Machine Learning (ICML)*, 2025.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-All: Train One Network and Specialize it for Efficient Deployment. In *International Conference on Learning Representations (ICLR)*, 2020.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A Short Note on the Kinetics-700 Human Action Dataset. *arXiv*, 2019.

Chubin Chen, Jiashu Zhu, Xiaokun Feng, Nisha Huang, Meiqi Wu, Fangyuan Mao, Jiahong Wu, Xiangxiang Chu, and Xiu Li. S^2-Guidance: Stochastic Self Guidance for Training-Free Enhancement of Diffusion Models. *arXiv*, 2025.

Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Tianlong Chen, and Cheng Yu. Accelerating vision diffusion transformers with skip branches. *arXiv e-prints*, pp. arXiv–2411, 2024a.

Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Delta-dit: A training-free acceleration method tailored for diffusion transformers. *arXiv preprint arXiv:2406.01125*, 2024b.

Ting Chen and Lala Li. FIT: Far-reaching Interleaved Transformers. *arXiv:2305.12689*, 2023.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *International Conference on Machine Learning (ICML)*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked Diffusion Transformer is a Strong Image Synthesizer. In *International Conference on Computer Vision (ICCV)*, October 2023.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. *arXiv*, 2014.

Moayed Haji-Ali, Willi Menapace, Ivan Skorokhodov, Arpit Sahni, Sergey Tulyakov, Vicente Ordonez, and Aliaksandr Siarohin. Improving progressive generation with decomposable flow matching. *arXiv preprint arXiv:2506.19839*, 2025.

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. In *NIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Susung Hong. Smoothed Energy Guidance: Guiding Diffusion Models with Reduced Energy Curvature of Attention. In *Neural Information Processing Systems (NeurIPS)*, 2024.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv*, 2024.

Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal Skip Guidance for Enhanced Video Diffusion Sampling. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable Adaptive Computation for Iterative Generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14569–14589. PMLR, 23–29 Jul 2023.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General Perception with Iterative Attention. In *International Conference on Machine Learning (ICML)*, 2021.

Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Upsample What Matters: Region-Adaptive Latent Sampling for Accelerated Diffusion Transformers. *arXiv*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv*, 2024.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a Diffusion Model with a Bad Version of Itself. In *Neural Information Processing Systems (NIPS)*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Toshiaki Koike-Akino and Ye Wang. Stochastic Bottleneck: Rateless Auto-Encoder for Flexible Dimensionality Reduction. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2735–2740, 2020. doi: 10.1109/ISIT44484.2020.9174523.

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. TREAD: Token Routing for Efficient Architecture-agnostic Diffusion Training. *arXiv*, 2025.

Black Forest Labs. Flux, 2024. URL `https://bfl.ai/`.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

Yan Li, Changyao Tian, Renqiu Xia, Ning Liao, Weiwei Guo, Junchi Yan, Hongsheng Li, Jifeng Dai, Hao Li, and Xue Yang. Learning Adaptive and Temporally Causal Video Tokenization in a 1D Latent Space. *arXiv*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Dongyang Liu, Shicheng Li, Yutong Liu, Zhen Li, Kai Wang, Xinyue Li, Qi Qin, Yufei Liu, Yi Xin, Zhongyu Li, Bin Fu, Chenyang Si, Yuewen Cao, Conghui He, Ziwei Liu, Yu Qiao, Qibin Hou, Hongsheng Li, and Peng Gao. Lumina-Video: Efficient and Flexible Video Generation with Multi-scale Next-DiT. *arXiv*, 2025a.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep Embedding Tells: It's Time to Cache for Video Diffusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363, June 2025b.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-Adaptive Sampling for Diffusion Transformers. *arXiv*, 2025c.

Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, and Sergey Tulyakov. Snap Video: Scaled Spatiotemporal Transformers for Text-to-Video Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7038–7048, June 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

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*, 2023.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning Ordered Representations with Nested Dropout. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1746–1754, Bejing, China, 22–24 Jun 2014. PMLR.

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*, pp. 234–241, Cham, 2015. Springer International Publishing.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. 2019.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and Advanced Large-Scale Video Generative Models, 2025.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled Diffusion Transformer. *arXiv preprint arXiv:2504.05741*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image Technical Report. *arXiv*, 2025.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. ElasticTok: Adaptive Tokenization for Image and Video. In *International Conference on Learning Representations (ICLR)*, 2025.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, et al. Sparse VideoGen2: Accelerate Video Generation with Sparse Attention via Semantic-Aware Permutation. *arXiv*, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. *arXiv*, 2024.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020.

Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An Image is Worth 32 Tokens for Reconstruction and Generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 128940–128966. Curran Associates, Inc., 2024.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Hao Luo, Yibing Song, Gao Huang, Fan Wang, and Yang You. DyDiT++: Dynamic Diffusion Transformers for Efficient Visual Generation. *arXiv*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast Training of Diffusion Models with Masked Transformers. *Transactions on Machine Learning Research (TMLR)*, 2024.

Appendix

Appendix Contents

A Baseline Details

DiT setup. We follow the standard DiT block design and incorporate recent improvements including QK normalization and rotary position embeddings (RoPE). Training hyperparameters match those of Peebles & Xie (2023): batch size $= 256$, 12 transformer blocks for DiT-B and 28 for DiT-XL. We use patch size of 2×2 for all experiments. We train all baseline using rectified flow matching objective and use logit-normal distribution for sampling the timesteps.

U-ViT setup. U-ViT mirrors DiT but adds U-Net–style residual (skip) connections. To isolate architectural effects, we use the same transformer blocks and training hyperparameters as DiT, differing only in the inclusion of these residual connections.

HDiT setup. HDiT follows DiT but applies PixelShuffle/PixelUnshuffle to reduce the token count while increasing channel dimensionality. We adopt this token–channel trade-off on the same transformer blocks as baselines. We use a single downsampling/upsampling operation after blocks 6 and 22. We also exclude local attention and instead use full self-attention. We train with the same hyperparameters as the other baselines.

Qwen-Image setup. We add ELIT *Read/Write* layers at blocks 8 and 52 of the 60-layer Qwen-Image backbone. Training uses a weighted sum of RF and distillation losses. The distillation term is scaled by $20\times$ to match the magnitude of the RF loss. We train for 60k steps at 512px with a global batch size of 1536, followed by 60k steps at 1024px with a global batch of 384. We sample timesteps from a logit-normal distribution and use time shifting of 2.22 during training and 2.0 during inference, following (Wu et al., 2025). We do not apply any timestep-aware loss re-weighting. The training dataset is a combination of internal real images with synthetic samples generated by FLUX.1-Schnell and Stable Diffusion-XL (with 50/50 ratio). We found that the model converges quickly, but we observe a style bias toward the synthetic data (reduced detail and more saturation relative to original Qwen-Image). For sampling, we use the Euler ODE sampler with 40 steps and use CFG value of 6.0

TeaCache setup. TeaCache proposes two strategies for deciding when to reuse (cache) the previous step's prediction: (1) using *timestep-modulated tensor relative error* between current and previous step to predict the accumulative error of caching the current step. (2) using *timestep-embedding relative error*, which measures the relative change of the timestep embedding itself across steps.

The original paper reports that strategy (1) generally works better. In text-to-image models (e.g., FLUX Labs (2024)), input tensors are modulated by the timestep embedding, providing access to the timestep-modulated input tensor. In our class-conditional image and video setting, those tensors are additionally modulated by the class signal, preventing access to timestep-modulated tensor. Empirically, on DiT for class-conditional ImageNet, we found that using class-timestep modulated input tensor following strategy (1) does not provide good estimate for the caching error and leads to degraded quality, underperforming the second strategy. Consequently, we adopt the timestep-embedding relative error (strategy 2) for all TeaCache experiments in this work.

Table 6: **Compute-quality tradeoff efficiency of baselines on ImageNet-1K 512px.** $\rho = (\text{Metric Ratio})/(\text{FLOPs Ratio})$ indicates the model degradation with respect to change in FLOPs between the low- and high-compute variants.

Baseline	Params	TFLOPs	$\text{FID}_{50K}\downarrow$ ($\rho\downarrow$)		$\text{FDD}_{50K}\downarrow$ ($\rho\downarrow$)		IS$\uparrow$ ($\rho\downarrow$)	
			–G	+G	–G	+G	–G	+G
DiT	675M	806	18.8 (1.00)	9.5 (1.00)	339.2 (1.00)	233.6 (1.00)	53.0 (1.00)	86.4 (1.00)
⌞ Patch size 2x4	675M	377	22.5 (0.56)	12.3 (0.61)	434.0 (0.60)	317.9 (0.74)	45.7 (0.54)	73.8 (0.55)
HDiT	1.4B	776	13.0 (1.00)	6.0 (1.00)	260.3 (1.00)	170.5 (1.00)	69.4 (1.00)	114.2 (1.00)
⌞ Smaller backbone	703M	392	22.2 (0.85)	11.5 (0.96)	435.2 (0.83)	315.4 (0.93)	48.8 (0.71)	80.0 (0.71)
ELIT-DiT	698M	831	11.1 (1.00)	4.9 (1.00)	175.6 (1.00)	106.1 (1.00)	80.0 (1.00)	134.1 (1.00)
⌞ 25% Tok.	698M	386	12.5 (**0.52**)	5.7 (**0.54**)	217.7 (**0.57**)	137.8 (**0.60**)	75.7 (**0.49**)	124.5 (**0.50**)

B Method Details

Adapting ELIT to baselines. Aside from adding the Read/Write operations, we leave each baseline's architecture and training unchanged. Unless noted, we place the *Read* at block 4 and the *Write* at block 24 for XL-size models across all baseliness (DiT, U-ViT, HDiT), as motivated by our ablations in Table 5.

Multi-budget training setup.

We use 16 spatial groups per image in all main experiments. On ImageNet-1K, each group contains 16 latent tokens at 256px and 64 at 512px. Unless otherwise noted, during training we set $J_{\max}$ to the per-group maximum (64 at 512px; 16 at 256px) and $J_{\min}$ to 1 for 256px and 4 for 512px, yielding 16 distinct inference budgets at 256px and 60 at 512px. At each training iteration, $\tilde{J}$ is sampled once and broadcast to all GPUs, ensuring synchronized compute with no added overhead. To account for the reduced compute, we increase the batch size from 256 (baselines) to 384 to match training FLOPs.

Kinetics-700 setup. We train at 256px on 29 frames sampled at 24 fps. The encoder produces 8 latent frames of shape $8\times32\times32$. We use a patch size of $1\times2\times2$, this yields 2,048 tokens. We use a group size of $2\times4\times4$, giving 64 groups per video. Kinetics-700 is trained with a single compute budget without multi-budget training.

Inference setup. We use the Euler ODE sampler with 40 steps for all experiments. Image experiments are evaluated on the ImageNet-1K validation split, and video experiments on the Kinetics-700 validation split.

C Compute-quality Tradeoff Efficiency

Increasing the training image resolution scales the required compute quadratically, making higher resolution training expensive. To control the compute while keeping model configuration the same, DiT proposed to increase the patch size to cut token count, while HDiT inserts a downsampling stage that reduces tokens but increases parameters count. We instead propose to cap the number of latent tokens per group during training, reducing training compute while keeping both patch size and model size constant.

To evaluate compute–quality trade-offs, we train low/high-compute variants for each baseline: DiT (larger patch size for the low variant), HDiT (model size matching other baselines), and ELIT-DiT (fewer latent tokens). Intuitively, given a similar reduction in compute between the two versions, the architecture with least performance degradation is more desirable.

To measure this, we define a degradation metric $\rho = ((\text{Metric Ratio})/\text{FLOPs Ratio})$, where "Metric Ratio" represents metric degradation caused by the low-compute model and "FLOPs Ratio" represents the corresponding reduction in FLOPs. As shown in Table 6, not only our method outperforms baselines at similar training compute, but also shows consistently lower ρ indicating it can more efficiently make use of its comupte if constrained, a capability we attribute to the latent interface's focus on the most important information in the input.

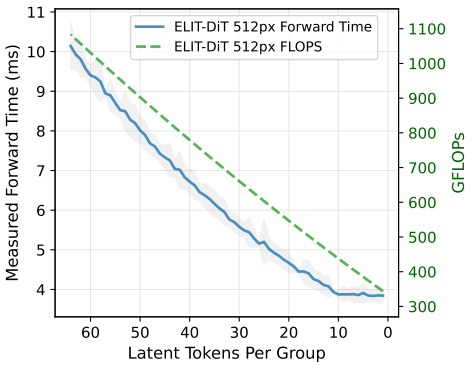

Figure 9: Lowering inference budget by using fewer latent tokens per group yields correlated reductions in forward time and FLOPs.

D Compute Analysis of ELIT

We analyze the theoretical computation requirement for ELIT-DiT in comparison with standard DiT design. Figure 10 (left) shows the relation between main architecture hyperparameters and FLOPs for the blocks employed by our architecture. When the number of core blocks

	Spat. Blocks	Lat. Blocks	Read	Write
Attn. Proj.	$8Nd^2$	$8JGd^2$	$d^2(4N+4JG)$	$d^2(4N+4JG)$
Attn. Mat.	$2N^2d$	$2J^2G^2d$	$2JNd$	$2JNd$
FF	$16Nd^2$	$16JGd^2$	$4JGd^2$	$4Nd^2$

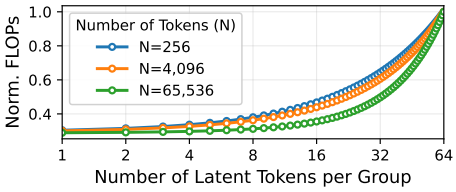

Figure 10: (left) FLOPs for spatial blocks, latent blocks, and Read/Write layers as a function of input tokens N, groups count G, latent tokens per group J, and hidden size d. (right) Relationship between latent tokens per group and model FLOPs for a DiT-XL with 8 spatial blocks, 20 latent core blocks, and $N/64$ groups, varying input tokens N and latent tokens per group $\tilde{J}$. FLOPs are shown relative to 64 tokens per group.

is large with respect to spatial blocks, computation is focused on the latent core blocks and the Read and Write operations' cost is minimal with respect to the model cost. Figure 10 (right) exemplifies the case of a DiT-XL/2 architecture for varying input sequence lengths. The latent interface is particularly effective at reducing FLOPs with large sequence lengths (*e.g.* training on higher resolutions) due to the dominant self attention cost that is quadratically reduced with $\tilde{J}$.

FLOPs vs latency in ELIT. Figure 9 reports FLOPs and wall-clock forward time for ELIT–DiT on ImageNet-1k at 512px as we vary the number of latent tokens per group. Forward time drops monotonically with token count and closely follow the FLOPs reduction, showing that budget control yields real speedups. At higher budgets, the correlation weakens slightly due to fixed overheads (e.g., I/O and kernel launch), but the overall trend remains strongly aligned.

E Additional Results

Compute-quality tradeoff.

To verify the advantage of our method over simply reducing the number of sampling steps, we show in Figure 11 that our multi-budget model achieves a more favorable quality–compute tradeoff compared to varying the number of sampling steps.

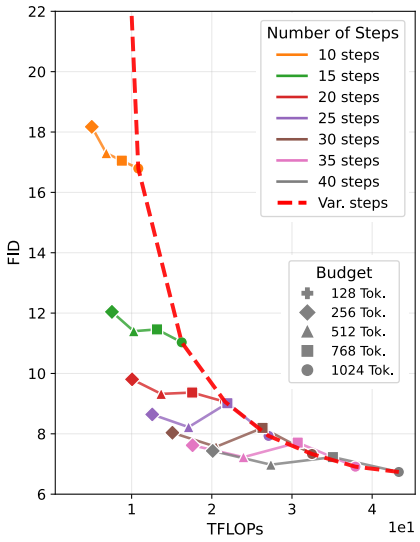

Figure 11: When tested with CFG 0.25, ELIT provides better quality-compute tradeoff than reducing the number of sampling steps.

Comparison to baselines. We show in Figure 14 additional qualitative results comparing our method to baselines on ImageNet-1K 512px. ELIT variants show less structural artifacts while allowing for per-step selection of inference budget and enabling auto-guidance and cheap classifier-free guidance out of the box for cheaper and higher quality sampling.

Varying inference budget. In Figure 12, we evaluate the effects of varying the number of tokens in the latent interface for ELIT-DiT trained on ImageNet-1K 512px. As the model FLOPs decrease with the number of latent tokens, the model is able to preserve image structure while changing less noticeable details.

Comparison of guidance methods. We qualitatively evaluate the effects of classifier-free guidance (CFG), autoguidance (Karras et al., 2024), and the proposed cheap classifier-free guidance (CCFG) (see Figure 13). We notice that AG produces results with most variation, including wider ranges of camera poses, compositions with multiple subjects and objects occlusion. By comparing results across different weights, we notice that AG remains most closely aligned with low guidance weight results, avoiding the mode collapse effect visible for CFG and CCFG that pushes samples towards more object-centric representations for the given class. We attribute this observation to the lower Inception Scores obtained by AG in Figure 5. Both AG and CCFG produce improved results which are particularly noticeable in complex concepts such as humans. CCFG combines the object-centric behavior of CFG, while reaping improved generation of complex objects from AG.

Additional Qwen-Image Results. We provide in Figure 15 additional qualitative comparison for ELIT-Qwen-Image against the original model. Thanks to CCFG, our model performs sampling with 69% of the FLOPs with respect to Qwen-Image and is able to produce a smooth tradeoff between sample quality and model FLOPs by varying the amount of tokens in the latent interface. In the cheapest shown configuration, ELIT-Qwen-Image uses

Figure	Prompt
Figure 1	*"The image portrays a woman with dark skin wearing a gold headpiece adorned with a blue jewel. Her gaze is directed towards something off-camera, giving her a focused expression. The background appears to be blurred, drawing attention to her face and headpiece."*
Figure 1	*"The image features actor Liev Schreiber in a snowy scene from a movie or TV show. He is dressed in black tactical gear, including a vest with "ARCTIC OCEAN" written on it, and a helmet with goggles. The setting appears to be a bustling city street filled with people and vehicles, all covered in snow."*
Figure 1	*"The image features a woman walking down a city street at night. She is wearing a black leather jacket, a white crop top, and a short black skirt. The street is illuminated by neon signs and streetlights, creating a vibrant atmosphere. There are other people visible in the background, but they are not the main focus of the image."*
Figure 15	*"The image portrays a man with long black hair and red eyes, wearing a black hooded cloak. He has a red gem on his forehead and holds a red orb-like object in his hand. The background features a circular pattern with red and black colors."*
Figure 15	*"The image features a large, white robot-like creature with wings standing on a desert landscape. The creature has sharp claws and appears to be looking down at something. Its body structure resembles a fusion of humanoid and bird-like characteristics. The background consists of a clear blue sky and rocky terrain."*
Figure 15	*"The image features a man wearing a blue knit cap, looking upwards with a serious expression. The background is dark blue, creating a contrast with the man's face and hat."*
Figure 15	*"The image showcases a vibrant sneaker with a red upper and blue accents. The shoe features a gold star design on the side and has red laces. The background appears to be a dark gray or black surface, providing a stark contrast to the colorful sneaker."*
Figure 15	*"The image captures a lively scene in a city square where people are walking around a fountain that is spraying water into the air. The square is surrounded by colorful buildings, creating a vibrant atmosphere. People are dressed in various styles of clothing, including dresses and suits, indicating a diverse crowd. Some individuals are carrying handbags, suggesting they might be tourists or shoppers. The sky above is blue"*
Figure 15	*"The image portrays a woman dressed in full armor, holding a small picture frame with a portrait of another woman inside. The background features dramatic clouds and fire, adding intensity to the scene."*
Figure 15	*"The image portrays a woman inside a large, ornate heart with wings. The heart is surrounded by red roses and intricate designs, creating a fantastical and romantic atmosphere."*
Figure 15	*"The image portrays a woman with purple hair and tattoos on her arm. She has striking blue eyes and is wearing a black tank top and jeans. The background is a solid color, possibly pink or magenta."*
Figure 15	*"The image depicts a futuristic cityscape with tall buildings and domed structures illuminated by orange lights. The city is surrounded by mountains and is situated near a body of water. The sky above the city appears cloudy."*
Figure 15	*"The image features a woman with intricate blue tattoos on her face and neck. She has a serious expression and is adorned with gold jewelry, including earrings and a necklace. Her hair is styled in braids, and she wears a flower crown. The background is dark, which contrasts with her colorful appearance."*
Figure 15	*"The image features a luxurious black leather armchair with gold accents. The chair has a high backrest adorned with buttons and a footrest. It is positioned against a dark background, creating a dramatic effect."*

Table 7: Prompts used to produce the showcased qualitative results for Qwen-Image Wu et al. (2025).

only 35% of the FLOPs with respect to the original model. As the number of latent tokens is decreased, the model preserves structural details, prioritizing changes in the least prominent image details.

Additional ImageNet-1k 512px Results. We provide in Figure 16, Figure 17 and Figure 18 additional qualitative comparison on ImageNet-1k 512px where we compare baseline DiT method with ELIT-DiT using CFG and CCFG. Class Ids and samples were randomly selected.

F FAILED EXPERIMENTS

Spatial token masking for flexible inference computation. We explored ideas from masked diffusion transformers Zheng et al. (2024); Gao et al. (2023); Krause et al. (2025) as a way to obtain variable inference budget by dropping tokens in the spatial domains. We found token dropping in the spatial domain not to produce satisfactory results when applied at inference time and attribute its lower performance to the unrecoverable information loss in the spatial regions corresponding to dropped tokens.

Per group latent tokens count. We experiment with automatic per-group budget assignment, i.e. making $\tilde{J}$ different for each group rather than uniform across groups, with the aim of assigning more tokens to groups with more complex content, further improving compute reallocation. To achieve this, we use the loss map to supervise an additional DiT block positioned at the beginning of the DiT which predicts importance score for every group according to the loss map. Given a desired total number of tokens, we automatically distribute latent tokens to different groups, assigning more tokens to groups with higher importance score. We find this variant to increase model and implementation complexity while matching the performance of ELIT. We hypothesize that our read operation is already tailored to read more from spatial tokens with higher loss as shown in Figure 2.

G LLM USAGE

We used LLMs in the preparation of this work to aid in writing and expand the coverage of the related works. For writing aid, we prompt LLMs to discover grammar errors, ambiguous formulations, clarity issues, opportunities to improve conciseness, and inconsistent usage of terms. For related work search, we compile a first version of the related works including all relevant references and prompt the model to read all provided references and expand the search to include additional works linked to the initial set. The returned results are reviewed and considered for inclusion in the related work section.

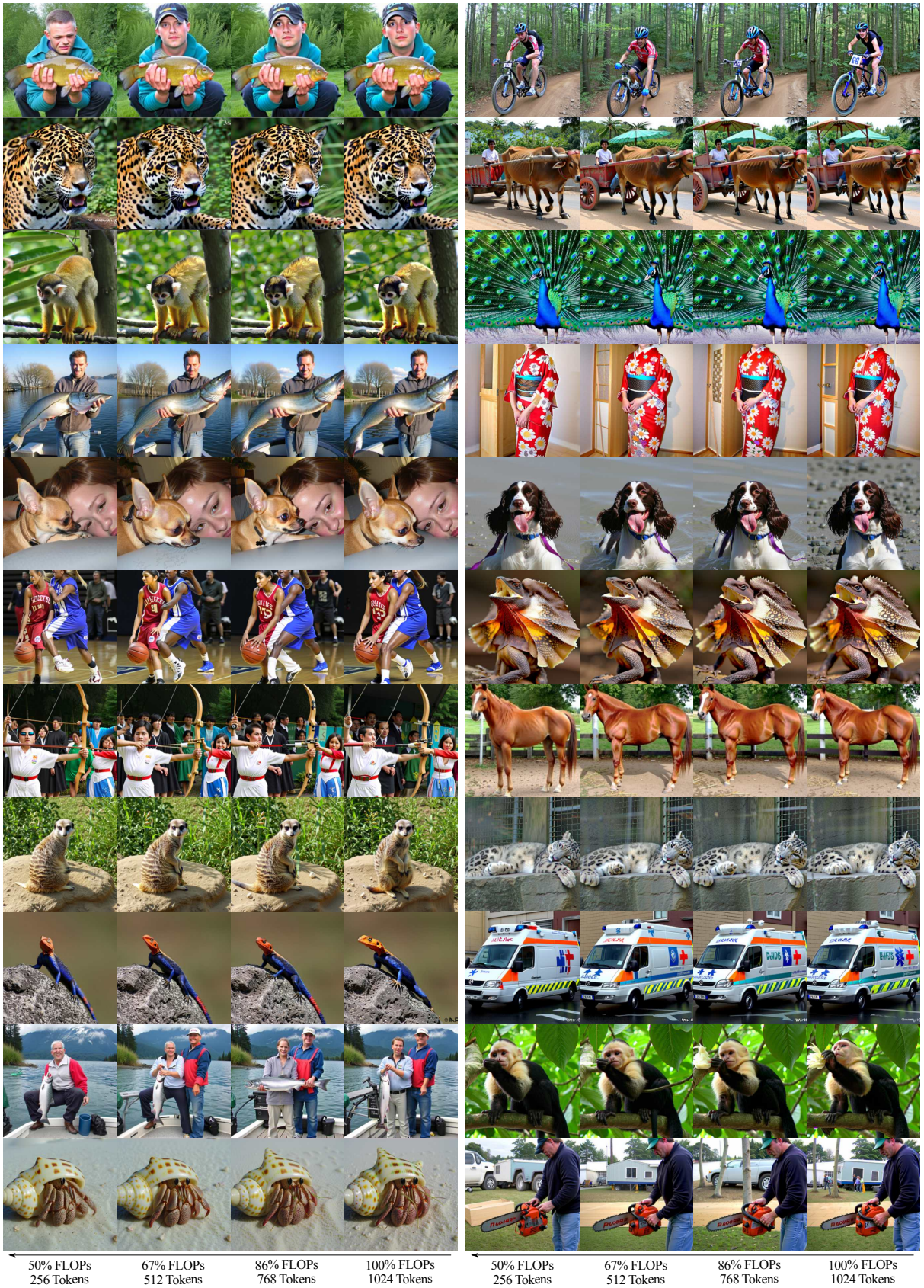

Figure 12: Qualitative results produced by ELIT-DiT on ImageNet-1K 512px with CCFG 4.0 for varying number of tokens in the latent interface. As the tokens and model FLOPs are reduced, the model preserves structure, while varying image details, producing gradual image changes. FLOPs are expressed relative to the model variant where no latent tokens are dropped.

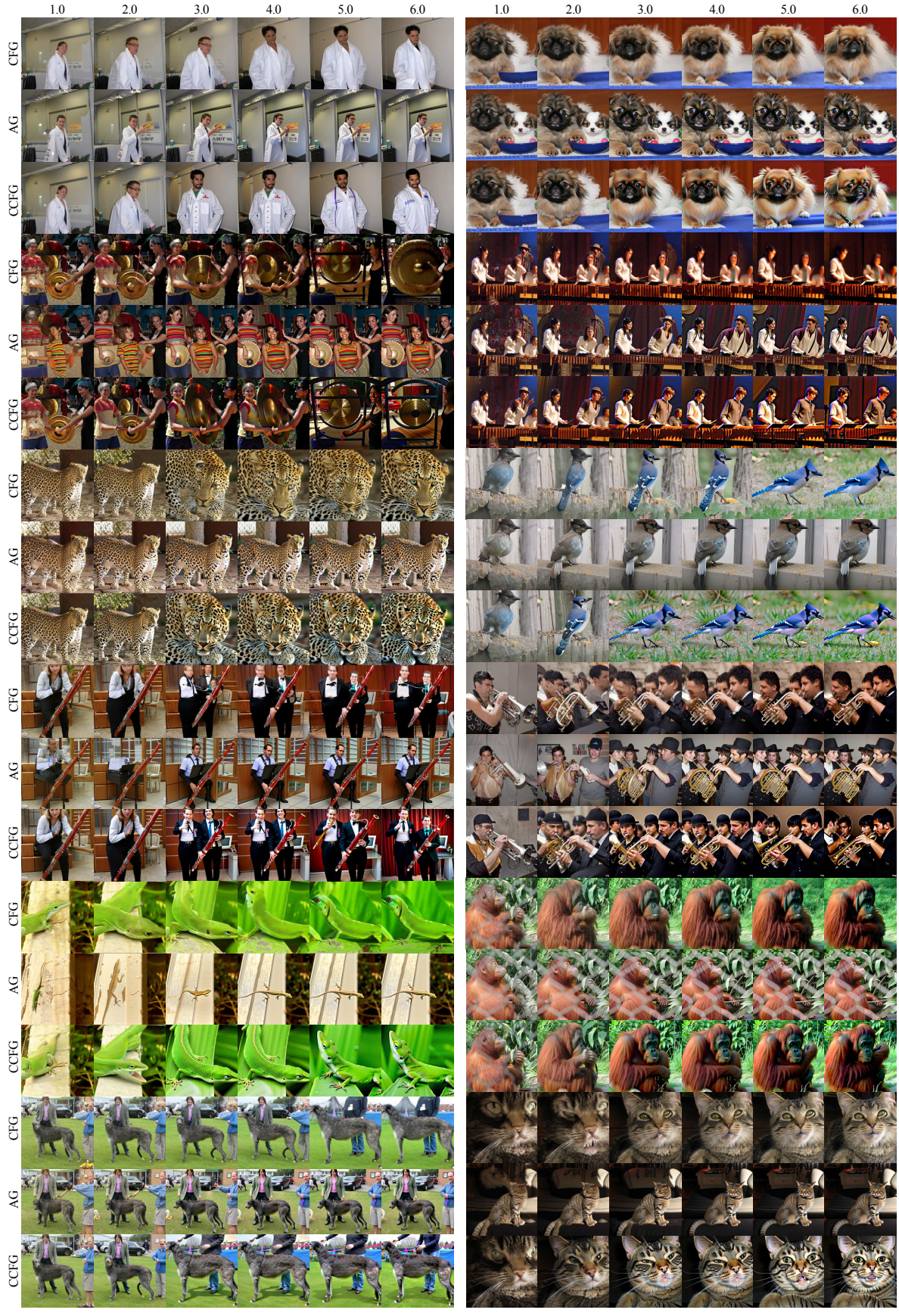

Figure 13: Qualitative comparison of classifier-free guidance (CFG), autoguidance (Karras et al., 2024) (AG), cheap classifier-free guidance (CCFG) with different weights, when applied to ELIT-DiT trained on the ImageNet-1K 512px dataset. AG produces the most varied samples, generating results with similar structure across guidance weights, as opposed to CFG and CCFG which favor object-centric generations. Both AG and CCFG produce better generations of complex concepts such as human faces.

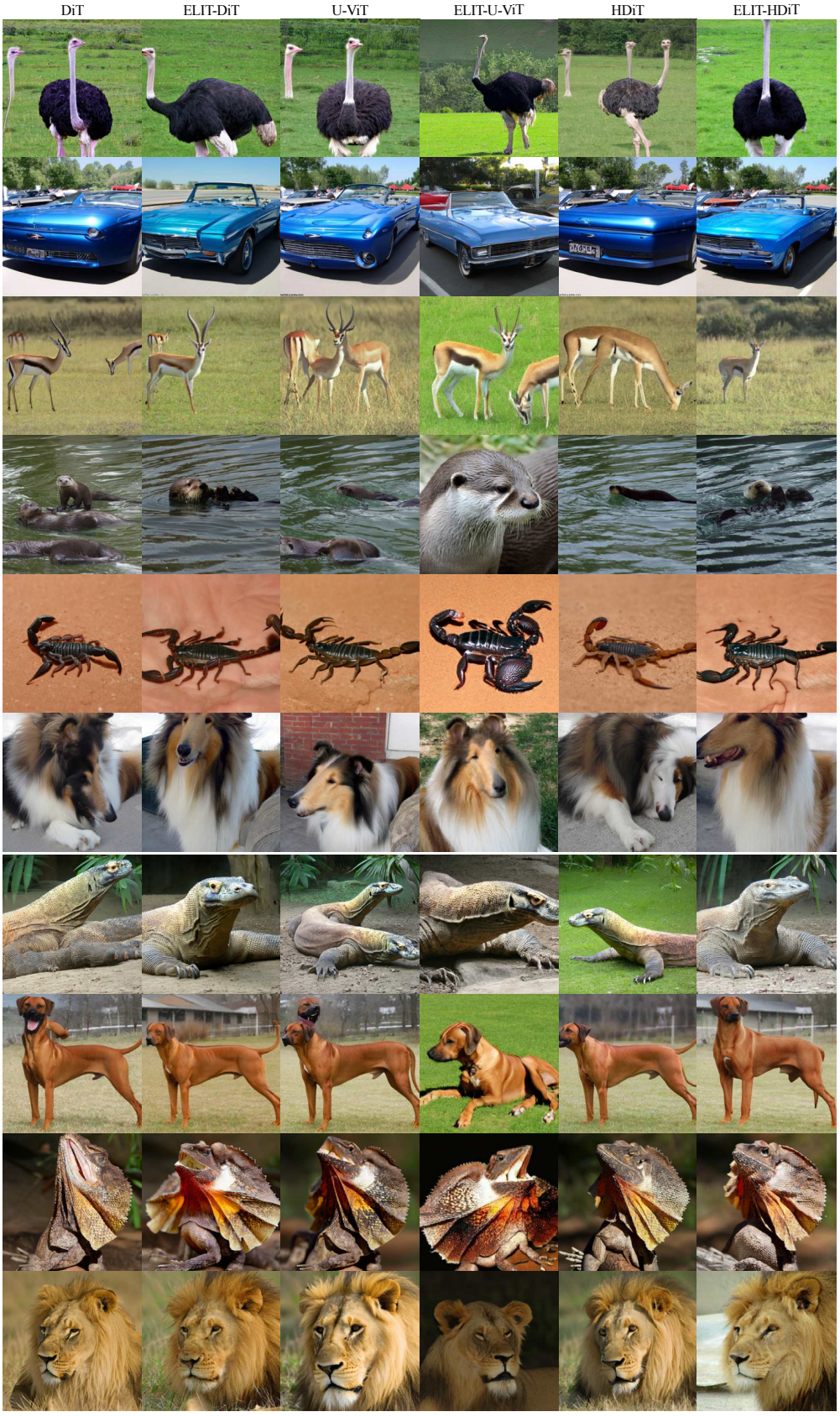

Figure 14: Qualitative comparison of ELIT against baselines on the ImageNet-1K 512px dataset. Results are produced using CFG with weight 4.0 for all methods.

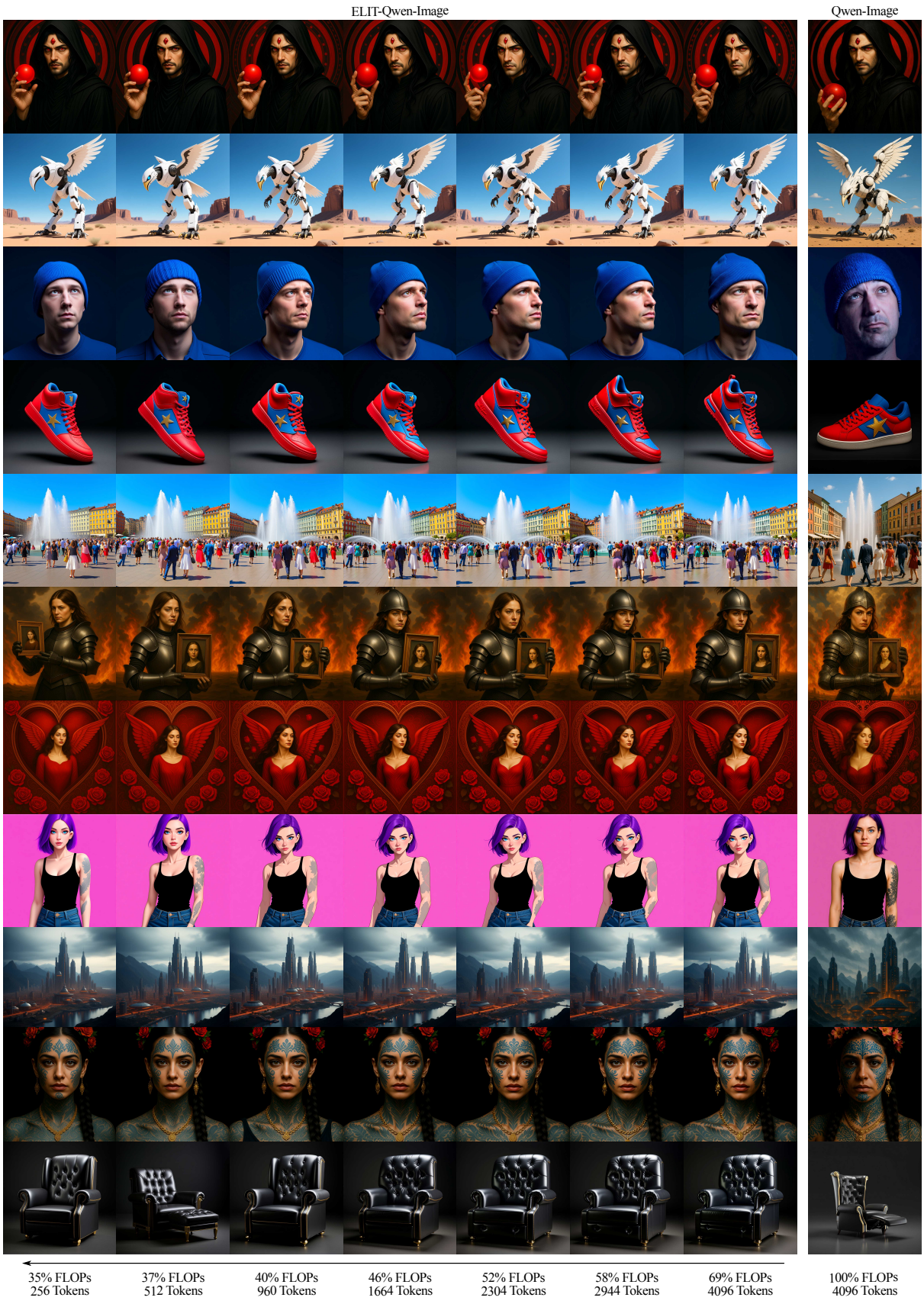

Figure 15: Qualitative results produced by ELIT-Qwen-Image for varying number of tokens in the latent interface. As the number of tokens is decreased and model FLOPs are reduced, our method can preserve structural details, while prioritizing changes in image details, preserving perceptual quality. Reported FLOPs are expressed relative to the original Qwen-Image and account for both the sampling FLOPs reductions brought by CCFG and the reduction in the number of tokens in the latent interface.


Class 653
Class 759
Class 30
Class 127
Class 166
DiT (w/ CFG)
ELIT-DiT (w/ CFG)
ELIT-DiT (w/ CCFG)


Class 429
Class 432
Class 558
Class 603
Class 367
DiT (w/ CFG)
ELIT-DiT (w/ CFG)
ELIT-DiT (w/ CCFG)


Class 581
Class 206
Class 742
Class 674
Class 727
DiT (w/ CFG)
ELIT-DiT (w/ CFG)
ELIT-DiT (w/ CCFG)