SPEND-WHERE-IT-MATTERS (SWIM): CONTENT-AWARE EFFICIENT DIFFUSION MODELS FOR IMAGE AND VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models achieve state-of-the-art performance in image synthesis and editing, but remain costly due to long denoising trajectories and large backbone models. Existing accelerations reduce the number of sampling steps or reuse cached features on a static schedule, so every input receives the same computation, even though which parts of the backbone matter varies across inputs and timesteps. We introduce SWIM, a model-agnostic, content-aware diffusion framework that decides, at each denoising step, which backbone blocks to execute under a user-set compute budget, while leaving the backbone weights frozen. A lightweight recurrent gating module reads the current sample's intermediate features, the change in the predicted score across diffusion steps, and bypasses the blocks that contribute least to the current update, so compute is spent where the sample needs it rather than on a fixed schedule. Because per-step compute is no longer uniform, we report cumulative multiply–accumulate operations (CMACs) as the measure of compute cost, over the full trajectory. For SD3.5 T2I generation, SWIM uses under 50% of DeepCache's CMACs yet outperforms it across all T2I-CompBench++ benchmarks with a 38% relative mean score improvement. On FLUX 2, SWIM cuts CMACs by 23% while staying within 0.01 of the T2I-CompBench++ mean. On ImageNet-256, SWIM + PixelDiT uses 18% fewer CMACs and 19% fewer parameters than PixelDiT at competitive FID, and improves FID over DiT-XL with 79% fewer CMACs. SWIM also extends to video generation, reducing CogVideoX-5B inference time by 13%.

1 INTRODUCTION

Diffusion models (Dhariwal & Nichol, 2021; Ho et al., 2020; Song & Ermon, 2019), built on frameworks like UNet (Ronneberger et al., 2015) and DiT (Peebles & Xie, 2023), have become the leading method in image generation. These models have been applied in a wide range of tasks, including image editing (Avrahami et al., 2022; Kawar et al., 2023; Meng et al., 2022), image enhancement (Gao et al., 2023; Li et al., 2022a; Saharia et al., 2022b), image-to-image translation (Choi et al., 2021; Saharia et al., 2022a; Tumanyan et al., 2023), text-to-image generation (Nichol et al., 2022; Saharia



Figure 1: **Fidelity vs. efficiency.** SWIM preserves semantic integrity (errors in red) while generating in 10 s per image vs. 14 s for DeepCache (1.4× faster), both at 1024×1024 on a single A100.

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

et al., 2022b; Zhang et al., 2023a) and text-to-3D creation (Lin et al., 2023; Luo & Hu, 2021; Poole et al., 2021). Despite their success, slow inference and intricate architectures, due to many denoising steps and model parameters, pose hurdles for practical deployment. The performance of diffusion models relies on executing a lengthy Markov chain of hundreds or thousands of denoising steps. Because this iterative process requires evaluating a heavy backbone network—such as a deep U-Net, a Vision Transformer (ViT), or UViT (Bao et al., 2023)—at every single step, it results in an immense computational burden during inference.

Current research accelerates sampling by skipping timesteps using specialized ODE/SDE solvers (Song et al., 2021a; Lu et al., 2022), or knowledge distillation (Salimans & Ho, 2022). However, these approaches overlook intrinsic computational redundancy within each diffusion timestep, where many network blocks contribute marginally when the denoising update is relatively unchanged across consecutive timesteps.

In diffusion sampling, each iteration has two controls: the *solver* chooses the step size, which determines how far the current sample state moves along the noise-to-image trajectory, while the *model* provides the update direction by predicting the score or the velocity, and thus determines how many network blocks are executed to estimate that move. Advanced solver-based methods, such as those introduced by EDM (Song et al., 2021b; Karras et al., 2022; Lu et al., 2022; Yu et al., 2026), focus on making fewer, more efficient informed steps along the trajectory, but they overlook intra-step redundancy by applying the full model depth uniformly. Orthogonally, quantization and cache-based methods (Liu et al., 2025c; Agarwal et al., 2024; Lu et al., 2024; Li & Gu, 2023) aim to reduce computation by respectively reducing feature map size or reusing feature maps from previous samples. But they often suffer from performance degradation such as losing critical context (as we demonstrate in our experiment with Deepcache (Ma et al. (2024)) in Figure 1).

Because all paradigms rely on a static, one-size-fits-all allocation of compute, they fail to adapt to inputs that vary in complexity. Consequently, the computational burden remains massive; for example, for U-Net as the denoising step, a standard diffusion model trained on ImageNet requires 1,000 denoising steps, translating to 1,000 full forward passes through a U-Net with over 500M parameters, or 387.3 T multiply–accumulate operations (MACs) per image. With improved solvers or architectures like Pixel DiT (Yu et al., 2026), inference still demands at least 15.5 T MACs, rendering high-throughput, latency-sensitive deployment challenging even on powerful server-grade accelerators like the NVIDIA A100.

In this paper, we propose SWIM: **S**pend-**W**here-**I**t-**M**atters, a content-aware adaptive diffusion framework that dynamically allocates computation at every timestep based on the observed behavior of the earlier timesteps. SWIM infers local trajectory curvature from predicted score (gradient of the log probability density) changes at the previous timestep while tracking intermediate generations. This drives a content-aware recurrent gating network that adaptively allocates computation by activating only the most informative backbone blocks at each denoising timestep. Unlike LoRA-based methods (Hu et al., 2022) that apply static weight adaptations for specific subtasks, our approach operates dynamically at inference time, tailoring computation to each input instance while keeping the backbone completely frozen.

We evaluate SWIM on T2I-CompBench++, DrawBench, FID, video CLIP score, and a 34-participant human preference study. Across modalities, SWIM cuts compute at a modest quality cost: on text-to-image, 16.1% (SD3.5) and 23.0% (FLUX 2) fewer CMACs (Cumulative MACs), within 1.35% of FLUX 2's T2I-CompBench++ mean (0.73 vs. 0.74); on ImageNet-256, 17.7% fewer CMACs and 19.4% fewer parameters than PixelDiT (FID 2.78 $\rightarrow$ 3.69), and 78.6% fewer CMACs than DiT-XL with better FID (4.23 $\rightarrow$ 3.69); and on video, 12.9% lower CogVideoX-5B inference time (3m 30s $\rightarrow$ 3m 03s) at an 8.1% CLIP-score drop.

The core contributions of our work are as follows:

- **Content-aware block selection under a budget**: SWIM adds a lightweight recurrent gate that, at each denoising step and for each sample, selects which backbone blocks to execute under a user-set compute budget, conditioned on the trajectory of the score function. The backbone stays frozen, and the same gate transfers across single-stream Multi Modal Diffusion Transformer (MMDiT) (SD3.5), dual-stream (FLUX 2), pixel-space DiT (PixelDiT), and video DiT (CogVideoX) backbones.

- **Analysis of block redundancy**: We show that block sensitivity depends on prompt content and denoising stage rather than prompt complexity alone, that skippable capacity differs between double- and single-stream blocks in FLUX 2 and across depth in SD3.5, and that content-aware selection beats random and fixed-position block dropping.
- **Cumulative compute as the efficiency measure**: NFE, the standard efficiency measure, assumes every denoising step costs the same and therefore cannot capture savings from approximations within each step. We instead report cumulative MACs (CMACs), total multiply-accumulate operations summed over the full trajectory, which provides one consistent metric to evaluate step-reduction, caching, and block-gating methods.

2 Related Work

Diffusion models have gradually become the leading method in image generation, surpassing GAN (Arjovsky et al., 2017; Goodfellow et al., 2014) and VAE (Higgins et al., 2017; Kingma & Welling, 2013).

Temporal modification methods or solvers: One area of focus is improving sampling efficiency by shortening the overall sampling process. Some techniques modify the variance schedule (Nichol & Dhariwal, 2021) or change the denoising steps (Song et al., 2021a; Zhang et al., 2023b; Wang et al., 2025a) to eliminate unnecessary stages. Parallel sampling methods (Li et al., 2024; Shih et al., 2023; Zheng et al., 2023a) are also being explored to boost inference times. Another line of research involves cache-based strategies (Chen et al., 2024d; Ma et al., 2024; Wimbauer et al., 2024; Lu et al., 2024; Agarwal et al., 2024), which reduce inference time by storing and reusing the outputs from earlier stages in the model. Other studies aim to speed up sampling through faster solvers for Stochastic Differential Equations (SDE) (Dockhorn et al., 2022; Liu et al., 2022a) or Ordinary Differential Equations (ODE) (Lu et al., 2022; 2025). The pursuit of efficient diffusion models has driven significant research into accelerating the sampling process by modeling the trajectory as an ODE. Leading efforts in this area, notably EDM (Karras et al., 2022) and the DPM-Solver (Zheng et al., 2023b; Lu et al., 2022), achieve substantial gains by leveraging higher-order numerical integration schemes and derivative information to take significantly larger steps, effectively skipping numerous timesteps while maintaining high fidelity. Prior approaches like DCT-Diff (Ning et al., 2025) convert images into Discrete Cosine Transform (DCT) domain and then measure the difference between their frequency components, to save computations. While these methods minimize the external number of required network evaluations, a critical gap remains in their focus: they do not explicitly consider removing redundancy within the individual time-steps. Although ADSF (Mukherjee et al., 2026) investigates the score function across various timesteps, it fails to provide a scalability justification or account for how distinct input signals affect architectural block selection.

Architectural modification methods or accelerating models: Structural-level acceleration techniques aim to enhance efficiency by modifying the network design. Some researchers focus on creating lighter architectures (Li et al., 2023b) or embedding frequency-based priors into the model structure (Yang et al., 2023). OMS-DPM (Liu et al., 2023), for instance, builds a model zoo that dynamically selects different diffusion models at various stages of the process. Other approaches (Meng et al., 2023; Salimans & Ho, 2022; Wang et al., 2023a) leverage distillation to simplify the model while maintaining performance. Methods like Ominicontrol (Tan et al., 2025) rely on LoRA (Hu et al., 2022) to reduce the number of trainable parameters. While LoRA reduces the number of trainable parameters during the fine-tuning phase, the resulting model still utilizes its full architectural depth (and the additional overhead of the LoRA rank matrices) for every single step of the denoising trajectory. Meanwhile, pruning-based methods (Castells et al., 2024; Wang et al., 2023b; Zhang et al., 2024) cut down the number of parameters to reduce computational load, and quantization-based techniques (Li et al., 2023a; Li & Gu, 2023; Li et al., 2022b; Liu et al., 2025c) compress models by representing weights and activations with lower bit precision. Existing gating-based approaches in diffusion models, such as (Zhu et al., 2025), employ gates to selectively choose tokens for attention computation (Yang et al., 2024), thereby reducing the computational load for only a subset of instances. Methods like DeepCache (Ma et al., 2024) and TeaCache (Liu et al., 2025a), choose to compute blocks based on the cache from an earlier timestep. This results in the model forgetting the semantic relation in the prompt, as demonstrated in Figure 1 and Figure 16 in Appendix. TeaCache's reliance on computing feature differences between consecutive steps introduces a hidden bottleneck, where the memory bandwidth required to evaluate high-dimensional intermediate tensor distances can eclipse the compute savings gained from skipping backbone layers. This

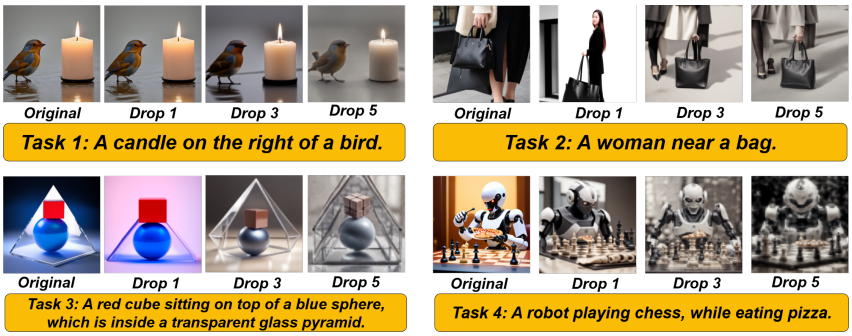


Figure 2: **Less is (Almost) More: Observing Graceful Degradation in SD3:** For simpler text prompts (1,2), image quality remains stable despite block drops (though the specific trajectory of intermediate steps causes Task 2 to lose proportion and degrade in visual quality earlier). However, complex prompts (3,4) lack this structural redundancy; a single drop leads to significant compositional failure, suggesting that complex layouts require the full depth of the model to maintain coherence.

overhead adds to the baseline computation. In contrast, SWIM addresses this limitation by reducing computational cost across all timesteps in a content-aware manner.

3 METHODS

3.1 MOTIVATION

While structural pruning is a key mechanism for accelerating generative models, existing optimization schemes largely depend on static heuristics that assume uniform resilience across similar tasks. We empirically find that computational requirements often cannot be captured by a fixed schedule, as they are fundamentally driven by the evolving interaction between prompt content and intermediate states. Consequently, effective dynamic computation demands a shift toward content-aware paradigms that continuously condition on real-time generative features rather than pre-computed assumptions. This dependency is clearly illustrated when examining how different types of generations react to structural pruning across four distinct example tasks (as shown in Figure 2). To examine this behavior, we evaluated four distinct scenarios representing different combinations of prompt complexity and generation difficulty:

Variable Stability in Simple Prompts: Even prompts sharing a lower level of complexity exhibit drastically different structural robustness. For instance, in Task 1, the bird's silhouette and relative positioning remain stable throughout early pruning steps. In contrast, Task 2, shown alongside it, suffers an immediate collapse in spatial proportions upon dropping a single block. This divergence proves that nominal prompt complexity alone is insufficient to predict computational slack.

Fragility in Complex Prompts: Complex prompts, such as Task 3 and Task 4,fail to maintain semantic coherence upon the very first block removal. In Task 3, structural bounds fail as elements overshoot, while Task 4 immediately loses the spatial grounding required to correctly compose the scene.

In summary, these empirical disparities highlight a fundamental limitation of static allocation: because identical prompt classes can exhibit erratic stability depending on the state of generation, optimal block selection at every timestep must continually condition on the real-time evolution of intermediate generative features rather than relying on prompt structure alone.

3.2 ARCHITECTURAL OVERVIEW

SWIM attaches a small recurrent gate to a frozen diffusion backbone. Given a user-set budget, a fraction α of the B backbone blocks, the gate reads the current state of the generation at every denoising step and selects which blocks to execute; skipped blocks are replaced by the identity. The gate is a Gated Recurrent Unit (GRU) (Cho et al., 2014b;a) with about 1M parameters, whose hidden state carries information about the trajectory from earlier steps to the current one (Section 3.2.2 explains why a recurrent gate suits this task). The pipeline is shown in Figure 3.

3.2.1 INPUTS TO THE GATE

Redundancy is content-dependent (Figure 2): prompts of similar complexity tolerate very different numbers of dropped blocks, so neither prompt length nor a fixed schedule can decide what to skip (Appendix Section A.5 compares against random and fixed-position dropping). The gate therefore conditions on the evolving state of the generation through three signals.

Preprocessed content inputs: Image features and Prompt. All visual input lives in the latent space of the backbone's image encoder. This image encoder is dependent on the backbone. For instance, we reuse the VAE for SD3.5, to obtain the corresponding features. During training, the encoded image and mixed with Gaussian noise according to a sampled timestep t, which yields the noisy latent $\mathbf{z}_t \in \mathbb{R}^{C\times H\times W}$ that represents the current denoising state; at inference, $\mathbf{z}_t$ is simply the sampler's current state. The prompt, if available, is processed by the pipeline's text encoders into token-level embeddings, which the backbone consumes through attention, and a pooled embedding $\mathbf{e}_{\text{text}}$. The gate uses only the pooled embedding, as a compact semantic summary of the caption. Class-conditional image models have no prompt and use the image descriptor alone.

Score change (Δs_t). Following ADSF (Mukherjee et al., 2026), we track the local curvature of the denoising trajectory through a scalar $\tilde{\kappa}_t$ (defined in Appendix A.1), which reflects how sensitive the backbone's output is to perturbations at the current step. High curvature marks a winding region where the model makes sharp adjustments and is sensitive to skipped blocks; low curvature marks a flat region where the update direction is stable and some blocks become redundant.

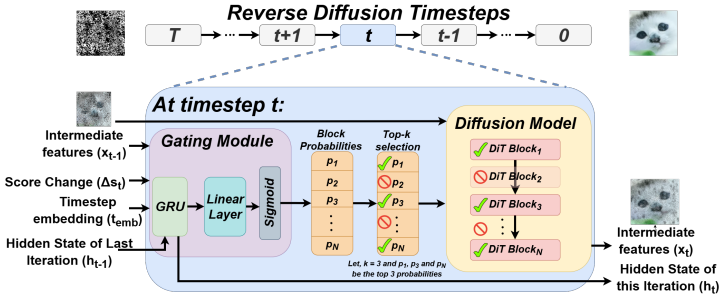


Figure 3: **SWIM pipeline:** Our approach leverages a GRU-based network to evaluate significance of each block (in every timestep), allowing the system to selectively process only the most relevant blocks while skipping less important ones thus achieving an efficient diffusion denoising process.

Timestep (τ) Sensitivity to pruning varies across the denoising process (Appendix Section A.9), so the gate also receives the normalized timestep $\tau = t/T$, which locates the current step within the schedule, through a sinusoidal embedding t_{emb}. Together, these signals let the gate move from a full-capacity backbone in volatile phases to a lean, refinement-only mode when the trajectory is stable.

3.2.2 RECURRENT GATING NETWORK

Controller. At step t, the gate concatenates the three signals,

$$\mathbf{u}_t = \big[\,\text{CNN}(\mathbf{z}_t);\; W\mathbf{e}_{\text{text}};\; t_{\text{emb}};\; \tilde{\kappa}_t\,\big], \tag{1}$$

and updates its recurrent state, $(\mathbf{o}_t, \mathbf{s}_t) = \text{GRU}(\mathbf{u}_t, \mathbf{s}_{t-1})$. A small MLP maps the output to one logit per backbone block, $\boldsymbol{\ell}_t = \text{MLP}(\mathbf{o}_t) \in \mathbb{R}^B$, and these logits determine which blocks run at step t.

Why a recurrent gate. We use a GRU for four reasons, each tied to a property of the task. *(i) The right decision depends on history.* Whether a block can be skipped at step t depends on how the trajectory arrived there, for example how quickly structure has been emerging, which a stateless per-step predictor (such as an MLP on the current features) cannot distinguish; the hidden state accumulates this context at every step. *(ii) The gate must be almost free.* The state has a fixed size, so the cost per step is constant and the whole module has about 1M parameters (roughly 0.06% of the active backbone parameters of SWIM + SD3.5), whereas attention over past steps would grow with the trajectory and require caching earlier features. *(iii) Decisions are causal and online.* A block must be selected before it would execute, using only the present and the past, which is exactly what a recurrence provides, and it works for any number of sampling steps or schedule because

no fixed-length input is needed. *(iv) A GRU is sufficient at this scale.* Denoising trajectories have tens of steps, so the extra cell state of an LSTM is not needed, and the GRU's two gates give it fewer parameters for comparable short-range modeling. The ablation in Table 3 is consistent with (i): making the selection once before denoising (SWIM (static)) is 18.4% worse in FID than re-deciding at every step at the same budget.

Budgeted block selection. The budget is enforced by hard top-k selection with $k = \lfloor \alpha B \rfloor$, using a straight-through estimator (STE) so the gate remains trainable:

$$\mathbf{m}_t = \mathrm{TopK}\big(\sigma(\boldsymbol{\ell}_t), k\big) \in \{0,1\}^B, \qquad \mathbf{g}_t = \mathbf{m}_t + \sigma(\boldsymbol{\ell}_t) - \mathrm{sg}\big[\sigma(\boldsymbol{\ell}_t)\big], \tag{2}$$

where $\mathrm{sg}[\cdot]$ is the stop-gradient operator, so the forward pass is exactly discrete while gradients flow through the sigmoid.

Blocks judged important from the current state and the trajectory history are executed and the rest are skipped, so the compute per step is fixed by α while *which* blocks run varies with the sample and the step. We use $\alpha = 0.8$, chosen empirically (Figure 7a and Appendix Section A.3).

Gated execution. Each backbone block F_l is wrapped so that

$$\mathbf{h}_{l+1} = g_{t,l}\, F_l(\mathbf{h}_l) + (1 - g_{t,l})\, \mathbf{h}_l, \tag{3}$$

which is exactly a skip when $g_{t,l} = 0$. Skipped blocks are not evaluated, so the saving is realized whenever a block is skipped for all samples in the batch.

For single stream architectures, like SD 3.5, the initial blocks establish global structure and are highly sensitive to pruning in multimodal models (Figure 2; consistent with (Xu et al., 2025; Chen et al., 2024c)), so the first n_0 blocks are always executed and top-k selects among the remaining blocks so that $\lfloor \alpha B \rfloor$ blocks run in total. For SD3.5 we set the first 4 blocks as fixed. The intuition has been described Appendix, Section A.2. However, this anchor is not required for FLUX or other architectures. Only the gate is trained; the backbone weights stay frozen. The gate is optimized with the backbone's flow-matching (velocity-prediction) loss, $\mathcal{L} = \mathcal{L}_{\text{flow}}$, computed with the gated forward pass.

Gate implementation. The gate has four input branches and one output head. A two-layer CNN with 3×3 kernels and strided downsampling reduces the 16-channel latent to a 64-dimensional spatial descriptor; a linear adapter compresses the pooled prompt embedding to a 256-dimensional semantic vector; the timestep is encoded as a 256-dimensional sinusoidal embedding; and the curvature $\tilde{\kappa}_t$ enters as a scalar. The concatenated vector $\mathbf{u}_t$ feeds a GRU with a 256-dimensional hidden state, and an MLP maps the GRU output to B block logits ($B = 24$ for SD3.5). Because the top-k mask is discrete in the forward pass, the executed path is a genuine sub-network, not a soft mixture, while the STE keeps it differentiable for training. Together with the anchored early blocks, this preserves the structural integrity of the image.

Other backbones. The same gate applies to FLUX 2, PixelDiT and CogVideoX with B set to the number of gated blocks of each backbone. For dual stream architectures as FLUX, we implement two separate gates fo rthe different streams. The ablation for checking how the individual components affect the overall performance is provided in Figure 7b.

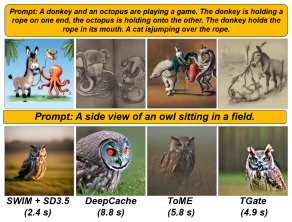


Figure 4: High resolution (1024 × 1024) and complex prompts from DrawBench. We report the latency numbers for SWIM and other approximation methods, averaged across the entire dataset.

4 Evaluation

We evaluate SWIM on three families of backbones: latent text-to-image transformers (SD3.5, FLUX 2), a pixel-space DiT (PixelDiT), and a video DiT (CogVideoX). In every case the backbone is frozen from its released checkpoint and only the gate (∼1M parameters) is trained, using COCO (Lin et al., 2014) for SD3.5 and FLUX, ImageNet (Deng et al., 2009) for PixelDiT, and the Disney Video Generation Dataset (Wild-Heart, 2024) for CogVideoX. Training the PixelDiT gate takes 12 hours, and the FLUX and CogVideoX gates about 20 hours,

on a single NVIDIA A100 (80GB) GPUs with PyTorch 2.4 and CUDA 12.1. Models are checkpointed every epoch (storing EMA and gating weights) so that diffusion trajectories can be replayed deterministically.

Metrics. We measure compositional quality with T2I-CompBench++ (Huang et al., 2025b) and DrawBench (Saharia et al., 2022b), text alignment with CLIP score (Hessel et al., 2021), and class-conditional quality with FID. To summarize quality per unit of compute we also report quality-normalized compute (QNC), the sum of the CLIP score and the mean T2I-CompBench++ score divided by CMACs. QNC is a compact summary combining quality and compute dimensions.

Measuring efficiency beyond NFE: The Number of Function Evaluations (NFE) counts forward denoising passes and implicitly assumes every pass costs the same. Because SWIM skips blocks inside a step, two methods with the same NFE can differ substantially in compute cost, so we compare methods by cumulative multiply–accumulate operations (CMACs; summed across all steps) counted with python `thop` command. Since SWIM executes a fixed fraction of blocks at every step, a reduction in per-step MACs carries over unchanged to the whole trajectory.

Table 1: Evaluation of computational cost (CMACs) and active parameters. The reduced parameter usage in DeepCache (via block-caching) and OminiControl (via LoRA adapters) is compared against SWIM and the full SD-3.5 model, demonstrating varying trade-offs between architectural complexity and generation efficiency.

Method	Active Params (M)	CMACs
DeepCache (Ma et al., 2024)	1028.7	18.18 T
Ominicontrol (Tan et al., 2025)	1076	158.97 T
SD-3.5 (Esser et al., 2024)	2028.33	10.47 T
TGate (Liu et al., 2025b)	2304.67	55.97 T
ToMe (Bolya & Hoffman, 2023)	1327.8	24.65 T
SWIM + SD3.5	1654.33	8.78 T

4.1 Text to Image Generation



Figure 5: Visual comparison of SWIM with contemporary acceleration methods ToMe (Bolya & Hoffman, 2023), TGate (Liu et al., 2025b), and DeepCache (Ma et al., 2024). Captions have been used from the COCO dataset. Compared to prior methods, SWIM better preserves fine prompt details and avoids spurious artifacts.

Resource Usage: Table 1 compares the compute of SWIM with the vanilla backbone and with representative acceleration and control frameworks under the same profiler. SWIM + SD3.5 has the lowest cost of all methods in the table at 8.78 T, a 16.1% reduction relative to vanilla SD-3.5 (10.47 T) and less than half the cost of DeepCache (18.18 T), while being trained only through a small gate on a frozen backbone. TGate and ToMe, which are also designed to accelerate inference, are respectively $6.4\times$ and $2.8\times$ more expensive under the same measurement. Our gate is lightweight: its $\sim$1M parameters are about 0.06% of the 1653M active parameters of SD3.5, and the reported reduction is a net figure that includes the gate. Unlike caching, which reuses stale features on a manually chosen refresh schedule, SWIM decides at each step which blocks to run, using the current latent, the prompt embedding, the timestep, and the recent score change.

Image Quality: Efficiency is only useful if quality is retained, so we evaluate on T2I-CompBench++ (Table 2) (Huang et al., 2025b). Two comparisons matter. Firstly, *against the unpruned backbones,* SWIM keeps nearly all of the quality while removing compute. The unpruned models are the natural upper bound for quality, and SWIM remains close to them. SWIM + FLUX 2 reduces MACs by 23.0% (23.8 T to 18.32 T) and stays within 1% of the FLUX 2 mean (0.73 vs. 0.74), with matching 3D-spatial and numeracy scores (0.78) and a CLIP score of 24.88 vs. 24.98. SWIM + SD3.5 reduces MACs by 16.1% and stays within 1.5% of the SD3.5 mean (0.65 vs. 0.66). As a result, QNC improves by 17.4% on SD3.5 (2.3 to 2.7) and by 29.9% on FLUX 2 (1.07 to 1.39). Secondly, *against other acceleration methods,* SWIM is clearly stronger. SWIM + SD3.5 outperforms DeepCache in all eight T2I-CompBench++ categories, a 38% relative improvement in the mean (0.65 vs. 0.47), while using less than half of its compute. The gains are largest on the compositional categories that reflect semantic fidelity: 2D spatial (0.59 vs. 0.50), 3D spatial (0.63 vs. 0.43) and numeracy (0.70 vs. 0.54), at under half of its compute. Taking mean quality and compute together, SWIM + SD3.5 and SWIM + FLUX 2 are two of the four Pareto-optimal points in Table 2 (the other two being the unpruned SD3.5 and FLUX 2 they are derived from): every other listed method is matched or beaten by one of these four on both axes. Figure 5 shows the qualitative picture on COCO validation captions: where the baselines lose the

"playing" motion in the tennis samples, the drawn face on the hydrant, or the proportions of the man with a suitcase, SWIM keeps these details, and it is the only method to render both the orange and the red flowers in the last column. Figure 10 in the Appendix extends this comparison. Figure 4 compares average high-resolution generation latency (measured over the entire dataset at batch size 1) alongside qualitative visual results for two representative prompts from DrawBench. SWIM uses roughly 50% of the time than contemporary acceleration methods.

Table 2: Quantitative Comparison on T2I-CompBench++. Rankings are denoted as: **First**, **Second**, and **Third**. We report the quality-normalized compute (QNC) (mean+CLIP score/ CMACs) to denote the performance per compute of our model as compared to other state of the art methods.

Model	Color	Shape	Texture	2D Spatial	3D Spatial	Numeracy	Non-Spatial	Complex	Mean	CLIP Score	CMACs (T)	QNC ↑
Other competitive methods												
FLUX.1 (Labs et al., 2025)	0.74	0.57	0.69	0.65	0.67	0.62	0.31	**0.92**	0.64	23.89	98.40	0.25
Stable XL (Podell et al., 2024)	0.59	0.47	0.53	0.21	0.36	0.50	0.31	0.85	0.48	24.30	74.52	0.33
Composable + SD v2 (Liu et al., 2022b)	0.41	0.33	0.36	0.08	0.28	0.43	0.30	0.50	0.34	22.38	32.50	0.70
Structured + SD v2 (Feng et al., 2023)	0.50	0.42	0.49	0.14	0.32	0.46	0.31	0.81	0.43	23.01	32.50	0.72
Stable v1-4 (Rombach et al., 2022)	0.38	0.36	0.42	0.12	0.30	0.45	0.31	0.77	0.39	22.53	15.82	1.42
Deepcache (Ma et al., 2024)	0.39	0.36	0.43	0.50	0.43	0.54	0.58	0.56	0.47	22.12	18.18	1.24
Stable v2 (Rombach et al., 2022)	0.51	0.42	0.49	0.13	0.32	0.46	0.31	0.82	0.43	23.12	16.25	1.44
Attn-Exct + SD v2 (Chefer et al., 2023)	0.64	0.45	0.60	0.15	0.32	0.48	0.31	0.82	0.47	23.08	16.25	1.44
GORS-unbiased (Huang et al., 2025a)	0.64	0.45	0.60	0.17	0.33	0.48	0.32	**0.86**	0.48	21.52	16.25	1.35
GORS + SD v2 (Huang et al., 2025a)	0.66	0.48	0.63	0.18	0.36	0.48	0.32	0.84	0.49	21.52	16.25	1.35
Pixart-α-ft (Chen et al., 2024b)	0.67	0.49	0.65	0.21	0.39	0.50	0.32	**0.86**	0.51	**24.55**	**12.10**	**2.07**
SD 3.5 based backbones												
Stable v3.5 (Esser et al., 2024)	**0.81**	**0.59**	**0.73**	**0.67**	**0.72**	**0.75**	**0.69**	0.71	**0.7**	23.55	**10.47**	**2.3**
TeaCache (Liu et al., 2025a)+ SD3.5	0.63	0.45	0.48	0.48	0.52	0.57	0.48	0.67	0.53	18.71	10.20	1.88
SWIM + SD3.5 (Ours)	0.77	0.55	0.59	0.59	**0.63**	**0.70**	0.59	**0.82**	0.65	22.81	**8.78**	**2.7**
FLUX 2 based backbones												
FLUX 2 (Labs, 2026)	**0.85**	**0.61**	**0.77**	**0.72**	**0.78**	**0.78**	**0.73**	**0.75**	**0.74**	**24.98**	23.8	1.07
TeaCache+FLUX 2	0.75	0.49	0.72	0.63	0.72	0.64	0.58	0.60	0.64	20.40	20.02	1.05
SWIM + FLUX 2 (Ours)	**0.84**	**0.6**	**0.75**	**0.7**	**0.78**	**0.78**	**0.71**	0.73	**0.73**	**24.88**	18.32	1.39

Generalization beyond COCO. The gates are trained on COCO captions, so we also test on prompts that are not COCO-style. On DrawBench (Saharia et al., 2022b), which contains complex, high-resolution (1024 × 1024) prompts, SWIM attains a clearly positive overall score of 0.582 (Figure 6), whereas DeepCache, TGate and ToMe have negative or near-zero averages. The advantage is most visible on positional reasoning, text rendering and conflicting prompts, categories that depend on fine-grained prompt–image alignment, which is what naïve caching and token reduction tend to erode. Figure 4 shows that this quality is obtained together with lower latency than the other approximation methods on a single A100. Finally, in a human preference study across 34 participants (Appendix Section A.7), SWIM was preferred 52.0% cases, against 30.1% for the next best method (TGate).

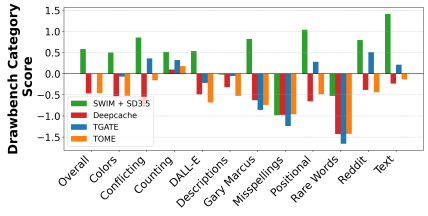
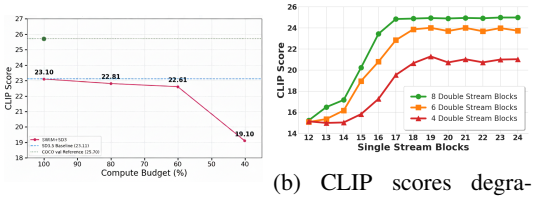


Figure 6: SWIM's performance across DrawBench categories compared to SOTA methods.

Choice of top-k and where redundancy lives. Empirically setting k to maintain an 80% block activation budget achieves graceful quality degradation (Figure 7a, Appendix Figure 9, and Section A.3). Furthermore, redundancy analysis on FLUX 2 (Figure 7b) reveals that double-stream blocks preserve critical cross-modal interactions, whereas single-stream blocks tolerate pruning down to 18 with minimal penalty. Consistent with these findings, later blocks emerge as the preferred candidates for skipping (Appendix A.6).



(a) Semantic fidelity (CLIP Score) degradation in SD3.5.

(b) CLIP scores degradation across multiple double- and single-stream blocks for FLUX.

Figure 7: Analysis of compute budgets its effect on performance using CLIP score. We sequentially drop compute and observe the effect on performance for both SWIM +SD3.5 and SWIM +FLUX 2.

4.2 Class-conditional Image Generation

On class-conditional ImageNet-256, SWIM + PixelDiT reduces CMACs by 17.7% (15.5 T to 12.75 T) and parameters by 19.4% (797.5 M to 642.5 M) relative to PixelDiT, at a modest FID in-

crease from 2.78 to 3.69. It also uses 78.6% fewer CMACs than DiT-XL and 4.8% fewer parameters while achieving a better FID (3.69 vs. 4.23), and it is better than every other non-PixelDiT baseline in Table 3 on FID (REPA, LightningDiT, FARMER, FractalMAR-H and DDPM) at a fraction of their compute. Overall, SWIM gives the lowest compute and parameter count in the table with the second-best FID, i.e. a favorable efficiency–quality trade-off.

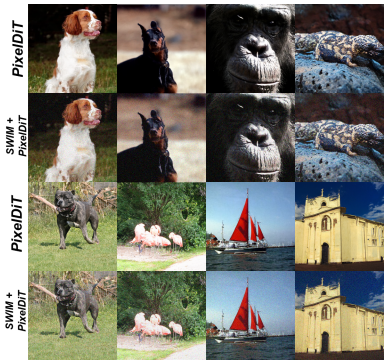


Figure 8: Image generations – Top: PixelDiT, Bottom:SWIM + PixelDiT generations on ImageNet.

Table 3: Quantitative comparison of efficiency and quality on ImageNet-256. SWIM achieves competitive FID with significantly fewer CMACs and parameters. **Best**, **second**, and **third** results are highlighted. The bottom part provides an ablation study showing which features in SWIM are most important in deciding on which blocks to skip.

Model	CMACs	Params	FID
DDPM (Ho et al., 2020)	382.5T	1.2 B	4.59
DiT-XL (Peebles & Xie, 2023)	59.5 T	675 M	4.23
REPA (Yu et al., 2025)	59.5 T	675 M	5.90
LightningDiT (Yao et al., 2025)	59.5 T	675 M	4.54
FARMER (Zheng et al., 2026)	115.2 T	1.1 B	5.40
FractalMAR-H Li et al. (2025)	102.44 T	844 M	6.15
PixelDiT (Yu et al., 2026)	15.5 T	797.5 M	2.78
SWIM + PixelDiT (Ours)	12.75 T	642.5 M	3.69
SWIM (only image) + PixelDiT	12.75 T	642.5 M	3.87
SWIM (only score) + PixelDiT	12.75 T	642.5 M	6.78
SWIM (only timestep) + PixelDiT	12.75 T	642.5 M	9.31
SWIM (static) + PixelDiT	12.74 T	642.5	4.52
SWIM (random anchored) + PixelDiT	12.74 T	642.5	55.25

Ablations: every input to the gate matters. All ablation variants in Table 3 run at the same block budget, so differences in FID reflect *which* blocks are executed and not *how many*. Combining the (partially denoised, intermediate) image, score and timestep signals gives the best FID (3.69). The image signal is the strongest on its own (3.87), and removing it degrades FID to 6.78 (score only) and 9.31 (timestep only); the complementary signals therefore refine, rather than replace, the content signal. This is intuitive: image features directly encode the current visual state (object structure, layout, semantics) and are the most informative about which features can be safely skipped, whereas the timestep and score signals describe the denoising stage and prediction uncertainty. Making the block selection once before denoising (SWIM (static)) raises FID from 3.69 to 4.52, an 18.4% worse result at identical compute, so re-deciding at every step contributes measurably. Randomly skipping blocks at the same budget, with the same anchored early blocks, collapses generation (FID 55.25), so the quality of SWIM comes from the learned selection and not from the budget or from anchoring alone. Appendix A.5 gives the corresponding random-drop and drop-last comparison on SD3.5, where SWIM achieves a mean CLIP score of 29.67 against 21.24 for random dropping and 28.75 for dropping the last three blocks.

4.3 Video Generation

Our gate design transfers to video without architectural changes. With the CogVideoX backbone frozen and only the gates trained on the Disney Video Generation Dataset, SWIM reduces inference time by 12.9% (3m 30s to 3m 03s) relative to the unaccelerated model (Table 4), making it the fastest method in the table. This comes at a CLIP score of 30.81 against 33.53 for the full model, an 8.1% drop that we regard as a reasonable tradeoff. A key obervation is that SWIM continues to maintain semantic relevance over cache based methods. SWIM + CogVideoX remains 27.4% and 35.8% faster than VideoCrafter2 and LaVie, respectively, while achieving higher text alignment than both (30.81 vs. 29.80 and 28.50). A visual representation can be found in Appendix Figure 16.

Table 4: Quantitative comparison of text-to-video alignment (CLIP Score) and inference efficiency on video generation task.

Method	CLIP ↑	Time ↓
SWIM + CogVideoX (5B) **(Ours)**	30.81	3m 03s
CogVideoX (5B) (Yang et al., 2025)	33.53	3m 30s
TeaCache (Liu et al., 2025a) + CogVideoX (5B)	18.27	5 m 1 s
VideoCrafter2 (Chen et al., 2024a)	29.80	4m 12s
LaVie (Wang et al., 2025b)	28.50	4m 45s

5 Conclusion

SWIM speeds up diffusion's denoising process that generalizes across tasks. Its technical contribution is a content-aware gating framework that decides, at each denoising step, which backbone blocks to execute under a fixed compute budget, with a frozen backbone. While SWIM currently selects which blocks to run under a fixed budget, extending the framework to support dynamic, per-sample adaptive budgets represents a promising direction for future research.

AI USE STATEMENT

We used generative AI tools to assist with drafting, copyediting, and improving the writing clarity of the manuscript. Additionally, we used AI to help identify potential competing baselines and relevant datasets, as well as to assist with the formatting and visual organization of our plots.

We have not used generative AI tools for data collection, experimental execution, or primary scientific ideation, and such tasks are not applicable to this work. We have reviewed all AI-assisted work; specifically, all AI-suggested text and structural edits were thoroughly checked for factual accuracy and alignment with our original notes, and stylistic adjustments were selectively curated to preserve our original arguments and tone. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have incorporated detailed architectural descriptions directly into the main text of the paper. Furthermore, our implementation code will be made publicly available upon acceptance.

REFERENCES

Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving Text-to-Image diffusion models. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 1173–1189, Santa Clara, CA, April 2024. USENIX Association. ISBN 978-1-939133-39-7. URL https://www.usenix.org/conference/nsdi24/presentation/agarwal-shubham.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18208–18218, 2022.

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*, pp. 22669–22679, 2023.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4599–4603, 2023.

Thibault Castells, Hyoung-Kyu Song, Bo-Kyeong Kim, and Shinkook Choi. Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 821–830, 2024.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7310–7320. IEEE, 2024a.

Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=eAKmQPe3m1.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024c.

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

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In Dekai Wu, Marine Carpuat, Xavier Carreras, and Eva Maria Vecchi (eds.), *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012/.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14367–14376, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems*, 35:30150–30166, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PUIqjT4rzq7.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10021–10030, 2023.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025a.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. https://karine-h.github.io/T2I-CompBench-new/, 2025b.

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.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6007–6017, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://huggingface.co/black-forest-labs/FLUX.2-klein-9B, 2026.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022a.

Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7183–7193, 2024.

Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Qk9kn6lOlW.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023a.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36:20662–20678, 2023b.

Zhikai Li and Qingyi Gu. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17065–17075, 2023.

Zhikai Li, Liping Ma, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Patch similarity aware data-free quantization for vision transformers. In *European conference on computer vision*, pp. 154–170. Springer, 2022b.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 300–309, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 21915–21936. PMLR, 2023.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. *Transactions on Machine Learning Research*, 2025b. URL https://openreview.net/forum?id=xXs2GKXPnH.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022a.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022b.

Xuewen Liu, Zhikai Li, and Qingyi Gu. Cachequant: Comprehensively accelerated diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23269–23280, 2025c.

Chen-Yi Lu, Shubham Agarwal, Md Mehrab Tanjim, Kanak Mahadik, Anup Rao, Subrata Mitra, Shiv Kumar Saini, Saurabh Bagchi, and Somali Chaterji. Recon: Training-free acceleration for text-to-image synthesis with retrieval of concept prompt trajectories. In *European Conference on Computer Vision*, pp. 288–306. Springer, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, pp. 1–22, 2025.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2837–2845, 2021.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15762–15772, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

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*, pp. 14297–14306, 2023.

Preeti Mukherjee, Shayok Bagchi, Jagannath Prabhakaran, Siddhant Prakash, and Chaoyue Liu. ADSF: Approximating diffusion process through tracking the score function. In *ECCV Workshop on Representation Learning with Very Limited Resources: When Data, Modalities, Labels, and Computing Resources are Scarce*, pp. 1–14, 2026. URL https://openreview.net/forum?id=bfzUOI3KjW.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pp. 16784–16804. PMLR, 2022.

Mang Ning, Mingxiao Li, Jianlin Su, Haozhe Jia, Lanmiao Liu, Martin Beneš, Albert Ali Salah, and Itir Onal Ertugrul. Dctdiff: Intriguing properties of image generative modeling in the dct space. In *The Forty-Second International Conference on Machine Learning (ICML 2025)*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2021.

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*, pp. 10684–10695, 2022.

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

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022a.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022b.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1921–1930, 2023.

Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12934–12944, 2025a.

Luozhou Wang, Shuai Yang, Shu Liu, and Ying-cong Chen. Not all steps are created equal: Selective diffusion distillation for image manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7472–7481, 2023a.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078, 2025b.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al. Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in neural information processing systems*, 36:72137–72154, 2023b.

Wild-Heart. Disney-videogeneration-dataset. `https://huggingface.co/datasets/Wild-Heart/Disney-VideoGeneration-Dataset`, 2024.

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6211–6220, 2024.

Zhuoyan Xu, Khoi Duc Nguyen, Preeti Mukherjee, Saurabh Bagchi, Somali Chaterji, Yingyu Liang, and Yin Li. Learning to inference adaptively for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3552–3563, 2025.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024.

Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 22552–22562, 2023.

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. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 87400–87442, 2025. URL `https://proceedings.iclr.cc/paper_files/paper/2025/file/d9e42b4d7163931f3689d6d6fbaa11d0-Paper-Conference.pdf`.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

Dingkun Zhang, Sijia Li, Chen Chen, Qingsong Xie, and Haonan Lu. Laptop-diff: Layer pruning and normalized distillation for compressing diffusion models. *CoRR*, abs/2404.11098, 2024. URL https://doi.org/10.48550/arXiv.2404.11098.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023a.

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: generalized denoising diffusion implicit models. In *International Conference on Learning Representations*, 2023b.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25730–25741, 2026.

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, pp. 42390–42402. PMLR, 2023a.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. DPM-Solver-v3: Improved Diffusion ODE Solver with Empirical Model Statistics. *Advances in Neural Information Processing Systems*, 36: 55502–55542, 2023b.

Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7664–7674, 2025.