Rethinking Cross-Layer Information Routing in Diffusion Transformers

Chao Xu[*,3], **Maohua Li**[*,1,2,3,♯], **Qirui Li**[3,4,♯], **Yixuan Xu**[3], **Yanke Zhou**[1,3,♯], **Yunhe Li**[3,5,♯]
Cuifeng Shen[3], **Hanlin Tang**[†,3], **Kan Liu**[3], **Tao Lan**[3], **Lin Qu**[3], **Shao-Qun Zhang**[1,2,§]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Intelligent Science and Technology, Nanjing University, China
[3]RTP, Alibaba, China
[4]College of Computer Science and Technology, Zhejiang University, China
[5]Department of Computer Science, City University of Hong Kong, Hong Kong

{limaohua, zhangsq}@lamda.nju.edu.cn
{shuaichao.xc, wanghaisheng.whs}@alibaba-inc.com

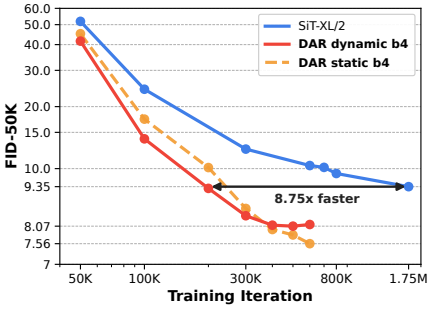
Abstract

Diffusion Transformers (DiTs) have become a de facto backbone of modern visual generation, and nearly every major axis of their design — tokenization, attention, conditioning, objectives, and latent autoencoders — has been extensively revisited. The residual stream that governs how information accumulates across layers, however, has been directly inherited from the original Transformer. In this paper, we present a systematic empirical analysis of cross-layer information flow in DiTs, jointly along depth and denoising timestep, and identify three concrete symptoms of traditional residual addition, namely monotonic forward magnitude inflation, sharp backward gradient decay, and pronounced block-wise redundancy. Motivated by this diagnosis, we propose Diffusion-Adaptive Routing (DAR), a drop-in residual replacement that performs *learnable, timestep-adaptive, and non-incremental* aggregation over the history of sublayer outputs. Moreover, the proposed DAR is compatible with many modern Transformer enhancement methods, such as REPA. On ImageNet 256×256, DAR improves SiT-XL/2 by 2.11 FID (7.56 vs. 9.67) and matches the baseline's converged quality with $8.75\times$ fewer training iterations. Stacked on top of REPA, it yields a $2\times$ training acceleration in the early stage, suggesting cross-layer information routing as an underexplored design axis in diffusion modeling, one that operates orthogonally to existing representation-alignment objectives. Beyond pretraining, DAR can also be applied during the fine-tuning stage of large-scale T2I models and preserves high-frequency details during Distribution Matching Distillation. Code is available here.



Baseline DAR

(a) DMD visual comparison



(b) Training FID curves

Figure 1: Overview of the main empirical results. (a) Our method preserves high-frequency details during DMD. (b) Training FID curves on ImageNet 256×256.

* Equal contribution. Chao Xu initiated the project and built the codebase and infra; Maohua Li refined the idea, led the paper writing and REPA-related work. †Project lead ♯Work done during internship at Alibaba §Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Advances in the design and optimization of Diffusion Transformers (DiTs) that replace convolutional U-Nets with token-based Transformer denoisers [Peebles and Xie, 2023] have led to significant breakthroughs in modern visual generation tasks [Wu et al., 2025, Kong et al., 2024, Labs, 2024, HaCohen et al., 2026, Cai et al., 2025, Team Seedream et al., 2025]. A central challenge for modern visual generation with DiTs is to capture the time-varying dynamics of the denoising process by developing architectural innovations. Recent years have seen extensive efforts devoted to key components of DiTs, including macro structure design [Bao et al., 2023, Peebles and Xie, 2023, Esser et al., 2024, Li et al., 2024], attention mechanisms [Xie et al., 2024, Chen et al., 2023, Peebles and Xie, 2023], conditioning mechanisms [Tan et al., 2025, Zhang et al., 2025b], learning objectives [Yu et al., 2025, Leng et al., 2025], latent autoencoders [Yao et al., 2025, Chen et al., 2024c, Zheng et al., 2025], and causal and autoregressive DiTs [Deng et al., 2024, Huang et al., 2025, Cheng et al., 2025]. However, the pre-normalized residual stream in DiTs and its variants — a fundamental design inherited from standard NLP practice — has remained largely unchanged, leaving its role in cross-layer information accumulation during time-varying denoising unresolved.

This work starts with an in-depth investigation of cross-layer information routing in DiTs, jointly along depth and denoising timestep. On the one hand, our analysis suggests that this seemingly innocuous default residual addition in DiTs gives rise to three symptoms that emerge in lockstep with depth: hidden-state magnitudes inflate monotonically, backward gradients decay sharply, and adjacent transformer blocks become increasingly redundant, as shown in Fig. 2. Strikingly, these symptoms collectively echo the *PreNorm dilution* phenomenon [Xiong et al., 2020] recently characterized in Large Language Models (LLMs) [Kimi Team et al., 2026, Li et al., 2026]. On the other hand, cross-layer information flow within DiTs is inherently time-varying: as denoising progresses across a continuum of noise levels, the intermediate representations that matter most should shift from coarse-structure features in high-noise regimes to fine-detail features in low-noise regimes [Ho et al., 2020, Sclocchi et al., 2025]. Thus, the *fixed, time-agnostic, and uniform-weighted* aggregation, as in conventional LLMs, is poorly suited to DiTs.

Several works have revisited the depth-wise structure of DiTs. A representative line of research [Bao et al., 2023, Tian et al., 2024, Chen et al., 2025, Li et al., 2024] grafts U-Net-style long skip connections onto DiTs to bridge shallow and deep layers, with the goal of restoring the fixed hierarchical inductive bias of U-Nets. Other studies explore layer- and timestep-dependent feature reuse to accelerate DiT inference [Ma et al., 2024b, Chen et al., 2024d]. Our key insight is that the denoising timestep — the very dimension that distinguishes DiTs from a standard Transformer — should play a vital role in adaptive routing. This motivates depth-wise aggregation mechanisms in DiTs to be *learnable, timestep-adaptive, and non-incremental*, to capture time-varying dynamics.

Building on the above insights, this work elevates cross-layer information routing in DiTs from an inherited convention to an explicit design axis, with contributions on two complementary fronts.

On the diagnostic side, we systematically analyze cross-layer information flow in DiTs across depth and denoising timesteps. We reveal that the three symptoms identified above and illustrated in Fig. 2 persist throughout training and vary systematically with the noise level, thereby suggesting that the role of the pre-normalized residual stream extends beyond stabilizing deep training, and exposing a spatiotemporal structure of PreNorm dilution that is invisible to LLM-side analyses.

On the methodological side, we propose Diffusion-Adaptive Routing (DAR), a drop-in residual replacement that performs *learnable, timestep-adaptive, and non-incremental* aggregation. Inspired by Kimi Team et al. [2026], we replace the running residual at each sublayer with a softmax attention over preceding sublayer outputs, where the query is computed from the current adaLN-modulated hidden state, allowing the routing mechanism to inherit both content and timestep dependence from DiT's existing conditioning pathway. This preserves the isotropic and homogeneous Transformer stack without introducing manually specified layer pairing, and remains compatible with modern Transformer enhancement methods, such as REPA [Yu et al., 2025].

Empirically, on ImageNet 256×256, DAR consistently outperforms vanilla SiT in our experiments, achieving 7.56 FID with SiT-XL/2 ($2.11 \downarrow$ over the baseline) while matching the baseline's converged quality in roughly $8.75\times$ fewer training iterations. Critically, the gains of DAR are orthogonal to representation-alignment objectives: Combining DAR with REPA [Yu et al., 2025] yields a $2\times$ training acceleration in the early stage over REPA alone. This suggests that cross-layer information

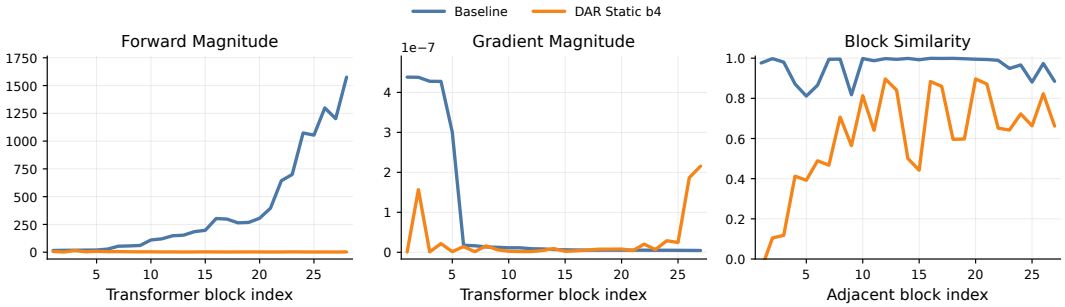


Figure 2: Three diagnostic symptoms of standard residual routing in DiTs across depth: Forward magnitude inflation, backward gradient decay, and block-wise redundancy. Measured at t=1.0.

routing is a promising and underexplored direction for improving diffusion models, complementary to existing learning objectives. Quantitatively, the three dilution symptoms identified by our diagnosis tighten in lockstep with these FID gains, linking our diagnosis to the observed gains.

Overall, the main contributions of this paper are summarized as follows

- We analyze cross-layer information flow in DiTs across depth and denoising timesteps and identify three concrete symptoms of standard residual routing: *forward magnitude inflation*, *backward gradient decay*, and *block-wise redundancy*.
- We propose DAR, a drop-in residual replacement for DiTs that performs *learnable, timestep-adaptive, and non-incremental* aggregation. The design operates purely along the depth dimension, preserving the isotropic and homogeneous Transformer stack, and remains compatible with many modern Transformer enhancement methods, such as REPA.
- Our method improves both convergence speed and final quality of diffusion transformers: on SiT, we achieve $8.75\times$ faster training and a 2.11 FID improvement over the baseline. Stacked on top of REPA [Yu et al., 2025], it yields a $2\times$ training acceleration in the early stage over REPA alone, demonstrating that depth-wise routing operates synergistically with existing representation-alignment objectives.

The rest of this paper is organized as follows. Section 2 reviews previous studies related to this work. Section 3 presents an in-depth investigation of cross-layer information flow in DiTs. Section 4 introduces DAR. Section 5 conducts experiments to demonstrate the effectiveness of our proposed DAR. Section 6 concludes this work.

2 Related Work

This section reviews seminal studies on cross-layer information routing and DiT architectures. An extended discussion is provided in Appendix A.

Evolution of Cross-Layer Information Routing. Cross-layer information routing in deep networks begins with standard residual connections, where layers communicate through fixed additive recursion [He et al., 2016, Srivastava et al., 2015]. Subsequent work mainly improves this residual pathway for optimization stability, including gated or scaled variants such as ReZero [Bachlechner et al., 2021], LayerScale [Touvron et al., 2021], and DeepNorm [Wang et al., 2024], which adjust residual strength without fundamentally changing the routing topology. Beyond single-stream propagation, Hyper-Connections [Zhu et al., 2024] introduces multi-stream recurrence with learned mixing, which mHC [Xie et al., 2025] subsequently refines by imposing doubly stochastic constraints on the mixing for more stable signal propagation at scale. In parallel, another line of work grants layers more direct access to earlier representations, from dense connectivity in DenseNet [Huang et al., 2017] to learned depth aggregation in DenseFormer [Pagliardini et al., 2024] and explicit depth-wise softmax attention in Attention Residuals [Kimi Team et al., 2026]. Overall, prior studies show a clear transition from fixed residual recursion toward learned, selective, and increasingly dynamic routing across depth. Despite the rapid architectural evolution of generative Transformers, the depth-wise routing dimension remains far less explored than these architectural developments.

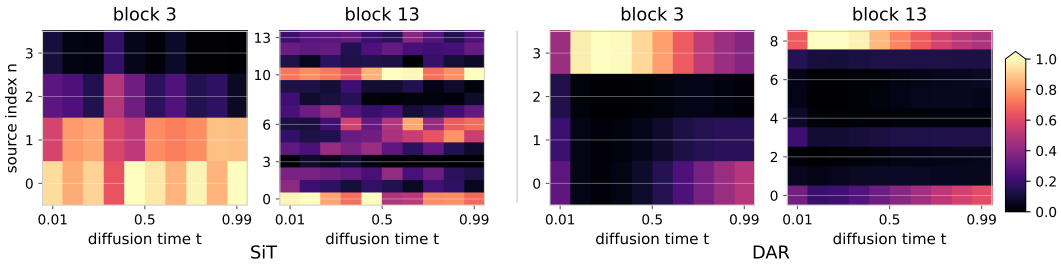


Figure 3: Source-mixing patterns across denoising timesteps. For the SiT model with standard residual routing, we add measurement-only scalar gates initialized to 1 on historical residual sources, keep the forward pass unchanged, and plot loss gradients with respect to the gates as counterfactual source importance; for DAR, we plot the learned softmax routing weights.

Evolution of Diffusion Transformers. DiTs have evolved from ViT-style U-Net replacements to specialized architectures for scalable generation. U-ViT shows that noisy image patches, timesteps, and conditions can be treated as tokens in a Transformer denoiser while retaining long skip connections [Bao et al., 2023]. DiTs further simplify this design into a pure latent-space Transformer and establish clear scaling behavior [Peebles and Xie, 2023]. Subsequent work has mainly progressed along two directions. One improves multimodal fusion and conditioning. For example, PixArt [Chen et al., 2024a, 2023, 2024b] retains conventional cross-attention, whereas MM-DiT [Esser et al., 2024] shifts to a unified self-attention framework. This trend also accompanies the adoption of stronger language models as condition encoders: Lumina-T2X [Gao et al., 2024], Playground v3 [Liu et al., 2024], and Sana [Xie et al., 2024] use decoder-only LLMs as text encoders, while Qwen-Image [Wu et al., 2025] further extends this design with a vision-language encoder. The other direction advances generative formulations and training objectives. SiT [Ma et al., 2024a] unifies diffusion- and flow-based objectives, while Stable Diffusion 3 [Esser et al., 2024] stresses rectified-flow training at scale. Notably, REPA [Yu et al., 2025] accelerates DiT training by introducing a representation-alignment objective that aligns hidden states of DiTs with pretrained visual representations. Overall, the recent evolution of DiTs has focused heavily on backbone scaling, conditioning pathways, and training objectives, whereas the residual pathway itself has remained largely unchanged.

3 Diagnosing Cross-Layer Information Flow in DiTs

In this section, we provide an empirical investigation of cross-layer information routing in DiTs, jointly along depth and denoising timestep. We analyze two models, that is, a vanilla SiT-XL/2 baseline and a static variant of DAR with chunk size $S = 4$. Both models are checkpointed after 600K training iterations, and diagnostics are computed on 4096 ImageNet samples. For each transformer block $k \in \{1, \ldots, 28\}$, we record three statistics of its output hidden state z_k. The first is the forward magnitude $\mathrm{RMS}(z_k)$ (root-mean-square of the feature values, averaged over batch and tokens). The second is the backward gradient magnitude $\mathrm{RMS}(\partial \mathcal{L}/\partial z_k)$, where $\mathcal{L}$ is the velocity-prediction MSE used for SiT training. The third is the block similarity $\cos(z_k, z_{k+1})$, defined as the per-token cosine similarity between consecutive block outputs averaged over batch and tokens. For DAR, we use z_k to denote the aggregated state passed to block $k + 1$; when $k = 28$, z_k denotes the final aggregated state fed to the prediction head.

Fig. 2 plots the forward hidden-state magnitude, backward gradient magnitude, and block-wise similarity as functions of the Transformer block index. The blue curves, corresponding to the standard residual baseline, reveal three diagnostic symptoms that all intensify with depth. The forward hidden-state magnitude grows monotonically from ~ 15.5 at block 1 to ~ 1576 at block 28, corresponding to roughly $100\times$ inflation. Combined with the unit-RMS normalization applied at each block input, this growth forces deeper blocks to produce ever-larger raw outputs in order to retain influence over the residual stream, echoing the *PreNorm dilution* phenomenon characterized in LLMs [Xiong et al., 2020, Kimi Team et al., 2026, Li et al., 2026]. The backward gradient magnitude drops sharply after the first five blocks. Early blocks receive substantial signal ($\sim 5 \times 10^{-7}$), whereas later blocks are lower by more than an order of magnitude and remain close to zero throughout the deep stack. This pattern suggests that the standard residual pathway provides limited control over gradient flow, leaving deeper layers with substantially weaker optimization signals. The per-token cosine

similarity between consecutive block outputs stays above 0.9 throughout the deep stack, indicating that neighboring deep blocks produce highly similar representations. This high similarity suggests substantial representational redundancy under the standard residual routing.

We next probe the timestep dimension, the key axis that distinguishes DiTs from standard Transformers. For DAR, the softmax weights $\alpha_{i\to l}(t)$ are directly observable; for the SiT baseline, which exposes no router by construction, we attach a scalar gate initialized to 1 on each historical residual source and read out the gradient of the denoising loss with respect to that gate as a counterfactual importance of how a baseline-equivalent router would reweight each source if one existed, while keeping the forward pass numerically identical to the unmodified baseline. Fig. 3 visualizes both quantities at a shallow and a deep location, and two observations stand out. Although the baseline never sees a router during training, its counterfactual importance map already varies systematically along t at both depths, with the preferred sources at high noise differing visibly from those at low noise — the standard residual stack exhibits timestep-dependent source preferences, suggesting the value of timestep-conditioned aggregation. DAR's learned weights provide the missing degree of freedom suggested by this diagnostic: the softmax concentrates sharply on a small subset of historical sources, and this selection itself shifts smoothly with t at both shallow and deep blocks, confirming that timestep-adaptive cross-layer routing is not an externally imposed inductive bias but a latent need of the DiT residual pathway that DAR directly meets.

Taken together, these findings point to an inherent rigidity in standard residual routing, which is associated with three issues: PreNorm dilution driven by residual-stream magnitude growth [Nguyen and Salazar, 2019, Xiong et al., 2020, Li et al., 2026, Kimi Team et al., 2026], imbalanced gradient propagation across depth [Kimi Team et al., 2026, Xie et al., 2025, Zhu et al., 2024], and high feature similarity and redundancy [Jiang et al., 2024, Song et al., 2024, Men et al., 2025, Chen et al., 2026]. These observations suggest that standard residuals provide cross-layer propagation, but lack adaptive control over which previous representations should be emphasized or suppressed.

4 Exploring Cross-Layer Interaction Spaces in DiTs

4.1 Cross-layer Routing in DiTs

Motivated by the diagnostic results, we revisit how existing DiT architectures route information. Rather than viewing cross-layer information routing as a post-hoc architectural add-on, we treat it as a fundamental design dimension already implicit in DiTs.

Standard residual routing in DiTs. Standard DiTs inherit the residual routing of the original Transformer. For clarity, we treat each self-attention or MLP sublayer as an individual transformation, indexed by $l = 1, \ldots, L$

$$h_{l+1} = h_l + f_l(h_l; t)\,, \tag{1}$$

where $h_l \in \mathbb{R}^{T\times d}$ denotes the hidden token sequence entering sublayer l, t is the diffusion or flow timestep, and f_l is the corresponding attention or MLP transformation. Let $v_0 = h_1$ be the input embedding and $v_l = f_l(h_l; t)$. Omitting the conditioning signal for simplicity, we obtain

$$h_l = v_0 + \sum_{i=1}^{l-1} f_i(h_i; t) = \sum_{i=0}^{l-1} v_i\,. \tag{2}$$

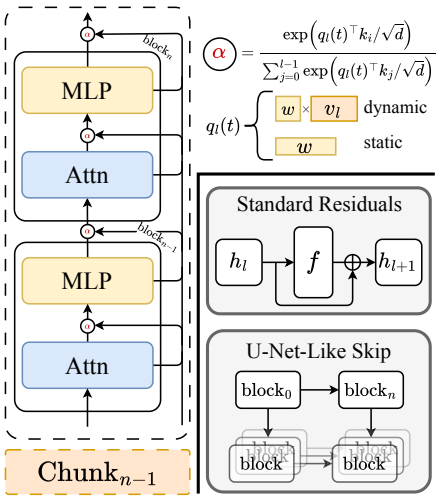


Figure 4: Diffusion-Adaptive Routing (DAR) compared with previous methods.

Standard DiTs already perform a form of cross-layer information routing. However, this routing pattern is fixed, since all previous outputs enter the residual stream with unit coefficients. Thus, standard DiTs cannot explicitly decide which earlier representations should be retrieved or suppressed at a given depth or denoising stage.

U-Net-like skip routing. Previous works [Bao et al., 2023, Tian et al., 2024, Chen et al., 2025, Li et al., 2024] introduce a U-Net-like routing pattern for diffusion models. Abstractly, for a deep layer l, U-Net-like long skip routing augments its input with a paired shallow representation

$$\tilde{h}_l = \psi_l \left(h_l, h_{\pi(l)} \right) \ , \tag{3}$$

where $\pi(l) < l$ indexes the corresponding shallow layer and ψ_l denotes the skip-fusion operation. The layer update can then be written as

$$h_{l+1} = \tilde{h}_l + f_l(\tilde{h}_l; t) \ . \tag{4}$$

From a routing perspective, U-Net-like skip routing shows that diffusion Transformers can benefit from multi-level feature fusion. Nevertheless, U-Net-like skip routing uses a manually specified topology that weakens the homogeneity underlying Transformer scalability.

4.2 Diffusion-Adaptive Routing

Drawing on the recently proposed Attention Residuals (AttnRes) framework [Kimi Team et al., 2026], which replaces fixed residual accumulation with softmax attention over depth, we instantiate Diffusion-Adaptive Routing (DAR) for DiTs with several design choices tailored to the diffusion setting. Let v_0 denote the input embedding and $v_i = f_i(h_i; t)$ the output of sublayer i, for $i = 1, \ldots, L$. Standard residual routing accumulates these sources into a running stream $h_l = \sum_{i=0}^{l-1} v_i$ with unit weights; DAR instead uses softmax-weighted aggregation

$$h_l = \sum_{i=0}^{l-1} \alpha_{i \to l}(t)\, v_i \quad \text{with} \quad \alpha_{i \to l}(t) = \frac{\exp\left(q_l(t)^\top k_i / \sqrt{d}\right)}{\sum_{j=0}^{l-1} \exp\left(q_l(t)^\top k_j / \sqrt{d}\right)} \ , \tag{5}$$

where $k_i = \mathrm{RMSNorm}(v_i)$ is the key associated with source v_i, and the softmax is computed over the source set $\mathcal{S}_l = \{v_0, v_1, \ldots, v_{l-1}\}$. The aggregated h_l then enters the sublayer transformation following $v_l = f_l(h_l; t)$.

Query parameterization. The per-layer query $q_l(t)$ admits two natural choices

$$q_l(t) = \begin{cases} w_l \ , & \text{(static)} \\ W_q^{(l)}\, v_{l-1} \ , & \text{(dynamic)} \end{cases} \tag{6}$$

where $w_l \in \mathbb{R}^d$ is a layer-specific learnable vector and $W_q^{(l)} \in \mathbb{R}^{d \times d}$ is a layer-specific projection. Notably, this is a sharp departure from the LLM-side observation in AttnRes, where the dynamic variant improves only marginally over the static one. We attribute this departure to the diffusion timestep dimension unique to DiTs, which is a structural feature absent in the LLM setting and fundamentally reshapes how the per-layer query should be conditioned. We discuss this below.

Timestep injection. Concretely, the main difference between static and dynamic query parameterization lies in how t enters the per-layer query. The former keeps w_l time-independent by construction, whereas the latter injects t implicitly since the network input x_t is itself a noised latent and further amplified at each sublayer through DiT's adaLN-Zero conditioning pathway. Additionally, we consider an explicit injection variant that augments w_l with the timestep embedding $e(t)$ reused from DiT's existing t-embedder, i.e., $q_l(t) = w_l + e(t)$ at no additional parameter cost. The final layer of the t-embedder is zero-initialized, so that $e(t) = 0$ at initialization and the model exactly recovers the pure static variant at the start of training. Overall, this yields three query variants of timestep injection: pure static, explicit timestep injection, and dynamic. A more detailed comparison that disentangles timestep awareness from input dependence is provided in Section 5.3.

Chunked aggregation. Retaining all sublayer outputs increases the activation footprint linearly with depth. To reduce this cost, we support a chunked variant that partitions the L sublayers into N chunks of size $S = L/N$. Chunk $n \in \{1, \ldots, N\}$ contains sublayers $(n-1)S + 1, \ldots, nS$ and is summarized by $c_n := v_{nS}$, the output of its last sublayer. For any sublayer l in chunk n, the source set is replaced by

$$\mathcal{S}_l \ = \ \underbrace{\{c_0, c_1, \ldots, c_{n-1}\}}_{\text{embedding and prior summaries}} \ \cup \ \underbrace{\{v_{(n-1)S+1}, \ldots, v_{l-1}\}}_{\text{current intra-chunk sources}} , \tag{7}$$

Table 1: System-level comparison on ImageNet 256×256 generation, with and without classifier-free guidance [Ho and Salimans, 2022], with the best results marked in **bold**. We use CFG with $w = 1.5$. Here, c4 denotes a chunk size of 4.

Method	Iters.	Params	w/o guidance					w/ guidance				
			FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Standard Residuals												
DiT_{ODE}	1.75M	675M	10.58	5.64	114.2	0.65	0.67	2.25	4.56	239.2	0.80	0.59
DiT_{SDE}			9.62	–	121.5	0.67	0.67	2.27	–	278.2	0.83	0.57
SiT_{ODE}	1.75M	675M	9.67	6.40	124.1	0.66	0.68	2.15	4.60	258.1	0.81	0.60
SiT_{SDE}			8.61	6.32	131.7	0.68	0.67	2.06	4.49	270.3	0.82	0.59
$\text{SiT-Plus}_{\text{ODE}}$	1M	752M	10.85	5.57	115.2	0.66	0.67	2.36	4.40	244.6	0.80	0.58
$\text{SiT-Plus}_{\text{SDE}}$			10.02	5.18	119.9	0.68	0.66	2.34	4.56	254.1	0.82	0.57
U-Net-Like Routing												
$\text{U-ViT-H/2}_{\text{ODE}}$	500K	585M	–	–	–	–	–	2.29	5.68	263.9	0.82	0.57
$\text{U-DiT-L}_{\text{SDE}}$	250K	810M	7.54	5.27	135.5	0.70	0.66	3.00	4.40	286.6	**0.86**	0.52
Our Method												
Static c4_{ODE}	600K	675M	7.56	5.18	131.1	0.69	0.68	2.08	4.42	272.9	0.83	**0.61**
Static c4_{SDE}			**6.92**	5.27	**138.8**	0.70	0.67	2.23	4.49	**287.0**	0.84	0.57
Dynamic c4_{ODE}	500K	751M	8.07	**5.07**	129.0	0.68	**0.69**	**2.05**	**4.39**	270.1	0.82	0.60
Dynamic c4_{SDE}			7.39	5.20	134.7	**0.71**	0.67	2.17	4.49	284.8	0.83	0.57

which consists of the input embedding, summaries from previous chunks, and all preceding sublayer outputs within the current chunk. The softmax aggregation therefore runs over $|\mathcal{S}_l| \leq S + N$ sources, reducing source memory from $O(Ld)$ to $O((S + N)d)$. Our chunked design differs from the LLM-side instantiation of AttnRes in both the choice of chunk summary and the chunk size. We analyze this gap and its dependence on depth in Appendix C.3 and Section 5.5.

5 Experiments

5.1 Setup

Implementation details. Unless otherwise specified, we follow the setup of SiT [Ma et al., 2024a]. We train on ImageNet-1K [Russakovsky et al., 2015] at 256×256 resolution, with the same data preprocessing as SiT. To ensure a fair comparison, we retrain SiT-XL/2 from scratch under the identical training recipe. For experiments combining DAR with REPA [Yu et al., 2025], we follow the original REPA configuration. We do not use any additional training tricks beyond those in the original SiT recipe. Full hyperparameters and experimental details are provided in Appendix C.

Evaluation. We report Fréchet Inception Distance (FID; [Heusel et al., 2017]), Inception Score (IS; [Salimans et al., 2016]), spatial Fréchet Inception Distance (sFID; [Nash et al., 2021]), precision and recall (Pre. and Rec. [Kynkäänniemi et al., 2019]), computed using 50,000 samples. We use both ODE and SDE samplers as in SiT, with 250 function evaluations by default; unless otherwise specified, we report ODE results.

5.2 Better Quality and Faster Convergence

We first show in Tab. 1 that DAR improves both final quality and convergence speed. Compared to the vanilla SiT-XL/2 baseline trained for 1.75M iterations, the static variant of DAR achieves a substantially better FID of 6.92 (SDE) without CFG, while training for only 600K iterations. The dynamic variant attains the best ODE FID with CFG (2.05), again outperforming the SiT baseline with far fewer training iterations. To rule out the possibility that these gains arise simply from increased model size, we further train SiT-Plus, a widened SiT-XL/2 matched to the 752M parameter count of dynamic c4. Despite using $2\times$ the training budget, SiT-Plus still trails DAR by a wide margin, confirming that the gains of DAR cannot be explained simply by parameter scaling.

Table 2: Ablation on timestep awareness in DAR. We report FID↓ at different training iterations.

Method	100K	200K	400K
Static w/o t-injection	22.36	15.47	11.51
Dynamic	13.95	9.29	8.10
Static w/ t-injection	17.39	10.12	7.97

Table 3: Compatibility with REPA. We report FID↓ at different training iterations.

Method	100K	200K	300K
SiT + REPA	9.89	6.89	6.29
DAR + REPA	7.09	5.92	5.68

A natural follow-up question is whether the gains of DAR merely reproduce what can already be achieved by equipping DiTs with U-Net-like skip pathways. We therefore compare against two representative models from this family in Tab. 1: U-ViT-H/2 [Bao et al., 2023] and U-DiT-L [Tian et al., 2024]. Under SDE with CFG, DAR static c4 outperforms U-DiT-L by 0.77 FID while using only $0.83\times$ as many parameters. Under ODE, DAR dynamic c4 further improves over U-ViT-H/2 by 0.24 FID. The gap is informative: hand-designed skip topologies wire shallow and deep layers together at predetermined depths and with fixed, time-invariant fusion weights, whereas DAR replaces these manual choices with learned and timestep-adaptive aggregation. Most importantly, DAR preserves the isotropic, homogeneous Transformer stack that underlies modern scaling.

5.3 Timestep Awareness Is Crucial for Routing in DiTs

Cross-layer information routing in DiTs differs fundamentally from its LLM counterpart in one key respect: the denoising timestep t is a first-class control signal that modulates every layer's computation, and the very motivation for adaptive routing, that different noise levels demand different mixtures of shallow and deep features, presupposes that the router itself is aware of t. The three query parameterizations of §4.2 differ precisely in how this awareness is introduced. The pure static variant, $q_l = w_l$, leaves the router blind to the timestep. The dynamic variant, $q_l(t) = W_q^{(l)} v_{l-1}$, derives its query from the most recent sublayer output, so any timestep information already encoded in v_{l-1} is naturally inherited by the routing weights. We refer to this as implicit timestep injection. The explicit-injection variant, $q_l(t) = w_l + e(t)$, instead reuses DiT's existing timestep embedding to add a direct timestep signal to an otherwise time-invariant query, at no additional parameter cost. As shown in Tab. 2, both timestep-aware variants substantially outperform the timestep-blind baseline at matched compute. This supports the view that timestep awareness — regardless of how it is injected — is the key ingredient.

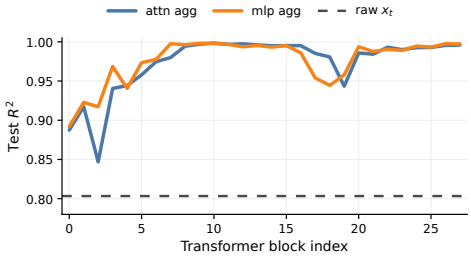


Figure 5: Linear-probe test R^2 for decoding the scalar denoising timestep t from the aggregated hidden states that feed the router at each depth in DAR-Dynamic.

The effectiveness of the dynamic variant rests on a non-trivial premise: the input to the dynamic query, v_{l-1}, actually retains sufficient timestep information for $W_q^{(l)}$ to extract. We test this premise directly with a linear-probe diagnostic. For a frozen DAR-Dynamic checkpoint, we sweep t over a uniform grid while holding the underlying (x_0, x_1) image-noise pairs fixed, collect the aggregated hidden state h_l that feeds each sublayer's router, and fit a ridge regressor that maps the pooled feature back to the scalar t on disjoint pair-level train/test splits. Fig. 5 reports the resulting test R^2 across all 28 blocks. Both the attention and MLP aggregator inputs sit well above the raw input latents x_t baseline ($R^2 \approx 0.80$) at every depth, exceed 0.95 within the first five blocks, and remain close to 1.0 throughout the deep stack. Thus the timestep is not merely present but linearly decodable from the very tensors the dynamic query consumes, confirming that $q_l(t) = W_q^{(l)} v_{l-1}$ enjoys direct access to a strong t-signal without any explicit conditioning, and is consistent with the mechanism underlying the early-training advantage of the dynamic variant.

5.4 DAR Is Orthogonal to REPA

REPA [Yu et al., 2025] accelerates DiT training by aligning intermediate hidden states with a pretrained visual encoder, an objective-level intervention that leaves the residual pathway untouched. DAR, in contrast, restructures how those hidden states are aggregated across depth, a purely architec-

tural change that is agnostic to the training loss. The two interventions therefore act along orthogonal axes, and a natural question is whether their gains compound or merely overlap. Tab. 3 answers this directly: stacking DAR on top of REPA improves FID from 9.89 to 7.09 at 100K iterations and from 6.89 to 5.92 at 200K iterations. Notably, DAR+REPA at 100K already matches REPA alone at 200K, a roughly 2× early-stage speedup, while DAR+REPA at 200K surpasses even the 300K FID of REPA alone, indicating that the routing-level and representation-level accelerations compound rather than offset each other.

5.5 Chunk Size Choices

The chunked aggregation in §4.2 exposes a single knob, the chunk size S, that interpolates between two extremes: $S = 1$ degenerates to the dense, all-source variant where every sublayer output enters the routing softmax, while large S collapses each chunk to a single summary c_n and aggressively compresses the source set. Tab. 4 reports a sweep of $S \in \{1, 4, 8\}$ on DAR under matched compute (300K iterations) and reveals a clear U-shaped pattern with $S = 4$ at the bottom. We now show that this U-shaped pattern is not coincidental: the cost can be decomposed into two competing terms whose unique minimum lies precisely in the $S \approx 4$ regime predicted for DAR.

Table 4: Effects of chunk size S in chunked aggregation on SiT-XL/2, ImageNet 256×256, no classifier-free guidance.

Chunk size	1	4	8
FID↓	10.41	8.39	11.14
IS↑	107.2	121.7	103.51

Under a mild rate-distortion model, the cost per aggregator decomposes as

$$\mathcal{L}(S) \;=\; \log\big(L/S + S\big) \;+\; \alpha \log S \quad \text{with} \quad \alpha \in (0, 1)\,, \tag{8}$$

where L is the total number of sublayers and $S \in (0, L]$ denotes the chunk size. The first term is the maximum-entropy lower bound on a softmax over $|\mathcal{S}|$ slots and caps the routing precision attainable by a bounded-norm query; the second is the rate-distortion cost of compressing S sublayer outputs into a single d-dimensional summary, with α discounting the loss to reflect partial recoverability through vertical attention over earlier summaries.

Proposition 1 (U-shaped cost of chunked aggregation)**.** *Let $L > 0$ and $\alpha \in (0, 1)$. Then $\mathcal{L}(S)$ in Eq.* (8) *is strictly decreasing on $(0, S^\star)$ and strictly increasing on $(S^\star, \infty)$, where*

$$S^\star = \sqrt{L \cdot \frac{1-\alpha}{1+\alpha}}. \tag{9}$$

Consequently, $\mathcal{L}(S)$ is U-shaped and has a unique global minimizer at $S^\star$.

A detailed proof and further analysis are provided in Appendix B.

5.6 Large-Scale T2I Model Post-Training

Beyond ImageNet, we apply Distribution Matching Distillation [Yin et al., 2024b,a] to Qwen-Image [Wu et al., 2025] equipped with DAR. We find that adaptive cross-layer information routing helps the distilled model preserve high-frequency details, including sharp edges and fine textures, which are easily attenuated during aggressive few-step distillation. Full setup and samples are deferred to Appendix E.

6 Conclusion

In this paper, we conducted a systematic investigation of cross-layer information routing in DiTs, jointly along depth and denoising timestep, identified three symptoms of the pre-normalized residual stream inherited from the original Transformer — forward magnitude inflation, backward gradient decay, and block-wise redundancy — and accordingly proposed DAR, a drop-in residual replacement that performs learnable, timestep-adaptive, and non-incremental aggregation. On ImageNet 256×256, DAR attains a best FID of 6.92 on SiT-XL/2, matches the baseline with 8.75× fewer iterations, and further delivers a 2× early-stage speedup over REPA. These results identify cross-layer routing as an underexplored design axis that complements prevailing advances in diffusion modeling. Further discussion of limitations and future work is provided in Appendix G.

Acknowledgments and Disclosure of Funding

This research was supported by the National Natural Science Foundation of China (62406138, 62632005). Maohua Li also received support from the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. 26CXJH0431).

This research was performed during the academic cooperation between LAMDA and RTP-LLM (Alibaba). The proposed method was evaluated and deployed in Alibaba's internal systems and on the Wuli platform.

References

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In: Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence, pp. 1352–1361. 2021.

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.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699. 2025.

Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Xiaoye Qu, Tianlong Chen, and Yu Cheng. Towards stabilized and efficient diffusion transformers through long-skip-connections with spectral constraints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17708–17718. 2025.

Hanqi Chen, Xu Zhang, Xiaoliu Guan, Lielin Jiang, Guanzhong Wang, Zeyu Chen, and Yi Liu. SortBlock: Similarity-aware feature reuse for diffusion model. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2859–2867. 2026.

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

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-Σ: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. In: Proceedings of the 18th European Conference on Computer Vision, pp. 74–91. 2024a.

Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. PixArt-δ: Fast and controllable image generation with latent consistency models. arXiv preprint arXiv:2401.05252. 2024b.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. arXiv preprint arXiv:2410.10733. 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. arXiv preprint arXiv:2406.01125. 2024d.

Xinle Cheng, Tianyu He, Jiayi Xu, Junliang Guo, Di He, and Jiang Bian. Playing with transformer at 30+ FPS via next-frame diffusion. arXiv preprint arXiv:2506.01380. 2025.

En Ci, Shanyan Guan, Yanhao Ge, Yilin Zhang, Wei Li, Zhenyu Zhang, Jian Yang, and Ying Tai. Describe, don't dictate: Semantic image editing with natural language intent. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19185–19194. 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Shi Guang, and Haoqi Fan. Causal diffusion transformers for generative modeling. arXiv preprint arXiv:2412.12095. 2024.

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: Proceedings of the 41st International Conference on Machine Learning, pp. 12606–12633. 2024.

Peng Gao, Le Zhuo, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, Chen Lin, Rongjie Huang, Shijie Geng, et al. Lumina-T2X: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. arXiv preprint arXiv:2405.05945. 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient joint audio-visual foundation model. arXiv preprint arXiv:2601.03233. 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. 2016.

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: Advances in Neural Information Processing Systems 30, pp. 6627–6638. 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598. 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems 33, pp. 6840–6851. 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708. 2017.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009. 2025.

Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. Tracing representation progression: Analyzing and enhancing layer-wise similarity. arXiv preprint arXiv:2406.14479. 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In: Advances in Neural Information Processing Systems 35, pp. 26565–26577. 2022.

Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, et al. Attention residuals. arXiv preprint arXiv:2603.15031. 2026.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In: Advances in Neural Information Processing Systems 34, pp. 21696–21707. 2021.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603. 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In: Advances in Neural Information Processing Systems 32, pp. 3904–3913. 2019.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`. 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742. 2025.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers. arXiv preprint arXiv:2504.10483. 2025.

Tianyu Li, Dongchen Han, Zixuan Cao, Haofeng Huang, Mengyu Zhou, Ming Chen, Erchao Zhao, Xiaoxi Jiang, Guanjun Jiang, and Gao Huang. SiameseNorm: Breaking the barrier to reconciling Pre/Post-Norm. arXiv preprint arXiv:2602.08064. 2026.

Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-DiT: A powerful multi-resolution diffusion transformer with fine-grained Chinese understanding. arXiv preprint arXiv:2405.08748. 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747. 2022.

Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, João Pedro Gandarela de Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. arXiv preprint arXiv:2409.10695. 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003. 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: Proceedings of the 18th European Conference on Computer Vision, pp. 23–40. 2024a.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-Cache: Accelerating diffusion transformer via layer caching. In: Advances in Neural Information Processing Systems 37, pp. 133282–133304. 2024b.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In: Findings of the Association for Computational Linguistics: ACL 2025, pp. 20192–20204. 2025.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In: Proceedings of the 38th International Conference on Machine Learning, pp. 7958–7968. 2021.

Toan Q Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In: Proceedings of the 16th International Conference on Spoken Language Translation. 2019.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In: Proceedings of the 38th International Conference on Machine Learning, pp. 8162–8171. 2021.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. DenseFormer: Enhancing information flow in transformers via depth weighted averaging. In: Advances in Neural Information Processing Systems 37, pp. 136479–136508. 2024.

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.

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.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252. 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In: Advances in Neural Information Processing Systems 29, pp. 2234–2242. 2016.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. Proceedings of the National Academy of Sciences, 122(1): e2408799121. 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 2256–2265. 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502. 2020a.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. In: Proceedings of the 41st International Conference on Machine Learning, pp. 46136–46155. 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456. 2020b.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387. 2015.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and universal control for diffusion transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14940–14950. 2025.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, Xiaowen Jian, Huafeng Kuang, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yanzuo Lu, Zhengxiong Luo, Tongtong Ou, Guang Shi, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Wenxu Wu, Yonghui Wu, Xin Xia, Xuefeng Xiao, Shuang Xu, Xin Yan, Ceyuan Yang, Jianchao Yang, Zhonghua Zhai, Chenlin Zhang, Heng Zhang, Qi Zhang, Xinyu Zhang, Yuwei Zhang, Shijia Zhao, Wenliang Zhao, and Wenjia Zhu. Seedream 4.0: Toward next-generation multimodal image generation. arXiv preprint arXiv:2509.20427. 2025.

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. arXiv preprint arXiv:2503.20314. 2025.

Yuchuan Tian, Zhijun Tu, Hanting Chen, Jie Hu, Chao Xu, and Yunhe Wang. U-DiTs: Downsample tokens in U-shaped diffusion transformers. In: Advances in Neural Information Processing Systems 37, pp. 51994–52013. 2024.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 32–42. 2021.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(10):6761–6774. 2024.

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 preprint arXiv:2508.02324. 2025.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. SANA: Efficient high-resolution image synthesis with linear diffusion transformers. arXiv preprint arXiv:2410.10629. 2024.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections. arXiv preprint arXiv:2512.24880. 2025.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In: Proceedings of the 37th International Conference on Machine Learning, pp. 10524–10533. 2020.

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 preprint arXiv:2408.06072. 2024.

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, pp. 15703–15712. 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In: Advances in Neural Information Processing Systems 37, pp. 47455–47487. 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6613–6623. 2024b.

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: Proceedings of the 13th International Conference on Learning Representations, pp. 87400–87442. 2025.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. arXiv preprint arXiv:2505.23884. 2025a.

Yuxuan Zhang, Yirui Yuan, Yiren Song, Haofan Wang, and Jiaming Liu. EasyControl: Adding efficient and flexible control for diffusion transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19513–19524. 2025b.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690. 2025.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. arXiv preprint arXiv:2409.19606. 2024.

In the Appendix, we provide supplementary materials for our work "Rethinking Cross-Layer Information Routing in Diffusion Transformers", organized to follow the main paper.

A More Discussion on Related Work

A.1 Diffusion Models

Diffusion models were originally formulated as a finite-step Markov chain that progressively corrupts data with Gaussian noise and learns to reverse the process via a variational bound [Sohl-Dickstein et al., 2015, Ho et al., 2020], and Song et al. [2020b] unified this view with score matching by recasting the forward and reverse processes as a continuous-time SDE with an equivalent probability-flow ODE. Subsequent work refined the noise schedule and parameterization [Nichol and Dhariwal, 2021, Kingma et al., 2021, Karras et al., 2022, Song et al., 2020a]. To avoid pixel-space cost, latent diffusion performs denoising in the compressed latent space of a pretrained autoencoder [Rombach et al., 2022]. A parallel line reformulated generation as learning a deterministic velocity field that transports noise to data, with Flow Matching [Lipman et al., 2022] regressing onto the conditional velocity along a prescribed probability path and Rectified Flow [Liu et al., 2022] favoring straight transport for few-step inference; SiT [Ma et al., 2024a] casts diffusion- and flow-based objectives under a single interpolant framework, and Stable Diffusion 3 [Esser et al., 2024] scales rectified-flow training to large text-to-image models. Diffusion models have since become a dominant generative paradigm, achieving remarkable success in image generation [Wu et al., 2025, Xie et al., 2024, Labs, 2024], image editing [Labs et al., 2025, Ci et al., 2025, Zhang et al., 2025a], video generation [Yang et al., 2024, Team Wan et al., 2025], and related visual synthesis tasks.

B Proof of Proposition and Empirical Verification

For convenience, we restate the proposition.

Proposition 1 (U-shaped cost of chunked aggregation)**.** *Let $L > 0$ and $\alpha \in (0, 1)$. Then $\mathcal{L}(S)$ in Eq.* (8) *is strictly decreasing on $(0, S^\star)$ and strictly increasing on $(S^\star, \infty)$, where*

$$S^\star = \sqrt{L \cdot \frac{1-\alpha}{1+\alpha}}. \tag{10}$$

Consequently, $\mathcal{L}(S)$ is U-shaped and has a unique global minimizer at $S^\star$.

Proof. Differentiating Eq. (8) yields

$$\mathcal{L}'(S) = \frac{S^2 - L}{S(S^2 + L)} + \frac{\alpha}{S} = \frac{(1+\alpha)S^2 - (1-\alpha)L}{S(S^2 + L)} .$$

Since $S(S^2 + L) > 0$ for all $S > 0$, the sign of $\mathcal{L}'(S)$ is determined by

$$(1+\alpha)S^2 - (1-\alpha)L .$$

Thus,

$$\mathcal{L}'(S) < 0 \quad \text{for} \quad S < S^\star , \qquad \mathcal{L}'(S) = 0 \quad \text{for} \quad S = S^\star , \qquad \mathcal{L}'(S) > 0 \quad \text{for} \quad S > S^\star ,$$

where

$$S^\star = \sqrt{L \cdot \frac{1-\alpha}{1+\alpha}} .$$

Therefore, $\mathcal{L}(S)$ is strictly decreasing on $(0, S^\star)$ and strictly increasing on $(S^\star, \infty)$, which proves that $S^\star$ is the unique global minimizer. □

Empirical agreement. For SiT-XL/2 (depth 28, two sublayers per chunk, hence $L = 56$), Eq. (9) predicts $S^\star \in [3.7, 4.9]$ over the realistic range $\alpha \in [0.4, 0.6]$, identifying $S = 4$ as the model-predicted optimum. This agrees quantitatively with Tab. 4: $S = 4$ improves on both the under-compressed extreme $S = 1$ (no chunking, $|\mathcal{S}| = L$, dominated by the routing-entropy term) and the over-compressed $S = 8$ (small $|\mathcal{S}|$ but large per-summary distortion). We therefore use $S = 4$ throughout the main experiments. Since Eq. (9) predicts $S^\star$ to scale as $\sqrt{L}$, we further conjecture that scaling DiTs to substantially deeper backbones will require a proportionally larger chunk size.

C Additional Implementation Details

C.1 Experimental Configuration

Unless otherwise specified, all ImageNet experiments follow the training recipe of SiT [Ma et al., 2024a]. We train models with a global batch size of 1024, a learning rate of 1×10^{-4}, and bfloat16 mixed precision. All compared models are trained under the same optimization and data-processing settings unless explicitly stated otherwise. For experiments involving REPA [Yu et al., 2025], we adopt the original REPA configuration. Specifically, we use DINOv2-B as the pretrained visual encoder, set the representation-alignment coefficient to 0.5, and apply the alignment loss to the hidden representation at the eighth layer. No additional modifications are introduced beyond replacing the residual routing mechanism with the proposed DAR. To ensure a fair comparison, we rerun the SiT and REPA baselines under our experimental setting rather than using previously reported results.

C.2 Compute Resources

Experiments were conducted on NVIDIA H20 GPUs with 192 CPU cores and 1024 GB of system memory. Training throughput and memory measurements with the fused implementation are reported in Appendix F.

C.3 Additional Implementation Details for DAR

Dedicated final aggregator. In our design, the final aggregator that produces the input to the prediction layer has access to the input embedding, the prior chunk summaries, and all raw sublayer outputs of the last chunk

$$\mathcal{S}_{\text{final}} = \underbrace{\{c_0, c_1, \ldots, c_{N-1}\}}_{\text{embedding and prior summaries}} \cup \underbrace{\{v_{(N-1)S+1}, v_{(N-1)S+2}, \ldots, v_L\}}_{\text{raw last-chunk sublayer outputs}}. \tag{11}$$

This is in contrast to AttnRes, whose final layer aggregates strictly over the N chunk summaries. The intuition is that the most recent sublayer outputs carry the most task-specific signal; thus, exposing them in raw form, rather than compressed into a single summary, lets the final layer recover fine-grained information that the chunk-level summary would otherwise discard. This yields about a 2-point FID gain after 200K training iterations.

REPA-specific implementation. For experiments that combine DAR with REPA, we depart from the dedicated final aggregator described above and instead let the last chunk reuse its own MLP aggregator parameters — the static/dynamic query (w_l or $W_q^{(l)}$) and the per-source $\mathrm{RMSNorm}$ — to perform the final aggregation. This is an empirically motivated design that improves performance.

D Direct Comparison with Attention Residuals

We compare a direct application of Attention Residuals [Kimi Team et al., 2026] to SiT-XL/2 with DAR under the same training and evaluation protocol on ImageNet 256×256. Table 5 reports FID at the available checkpoints.

Table 5: Direct comparison with Attention Residuals on SiT-XL/2, ImageNet 256×256. We report FID↓ using ODE sampling without classifier-free guidance. DAR uses the static query with explicit timestep injection and chunk size $S = 4$. Dashes indicate unreported checkpoints.

Method	**100K**	**200K**	**300K**	**400K**	**500K**	**600K**	**700K**	**800K**
Attention Residuals	21.78	14.28	11.52	10.28	9.51	8.99	8.71	8.73
DAR (Static c4)	17.39	10.12	8.62	7.97	7.81	7.56	–	–

Attention Residuals reaches its best reported FID of 8.71 at 700K iterations and yields 8.73 at 800K. DAR reaches 7.56 at 600K, improving on the best reported Attention Residuals result by 1.15 FID with fewer training iterations. This comparison supports the additional benefit of DAR's diffusion-specific routing design over directly applying Attention Residuals to SiT.

E Details of Large-Scale T2I Model Post-Training

For large-scale T2I post-training, we apply DMD to Qwen-Image [Wu et al., 2025] with DAR inserted into the MM-DiT backbone. We use LoRA fine-tuning with rank 64, and train the student branch with a learning rate of 5×10^{-6} together with a fake branch learning rate of 2×10^{-6}. Distillation is performed with 4 denoising steps and guidance scale 4.0. We train at 1024^2 resolution using bfloat16 mixed precision and a per-GPU batch size of 1. More visualizations are provided in the supplementary materials.

Quantitative evaluation. We compare standard routing and DAR under matched Qwen-Image DMD settings using the same GenEval2 prompts. Both metrics in Table 6 are computed relative to the teacher outputs: LPIPS measures perceptual discrepancy, while the high-frequency-ratio (HFR) deviation derived from RAPSD compares frequency content above the threshold 0.2. Lower values indicate closer agreement with the teacher.

Table 6: Quantitative comparison on Qwen-Image DMD using GenEval2 prompts. Both metrics are measured against the teacher outputs; lower is better.

Method	**LPIPS**$\downarrow$	**RAPSD HFR**(> 0.2) **deviation**$\downarrow$
Standard routing + DMD	0.538	$+0.340$
DAR + DMD	$\mathbf{0.512}$	$\mathbf{+0.215}$

DAR reduces teacher-relative LPIPS from 0.538 to 0.512 and the high-frequency-ratio deviation from $+0.340$ to $+0.215$. These results show that DAR stays closer to the teacher in both perceptual similarity and frequency content, supporting the qualitative detail-preservation observations.

F Infrastructure

A naive implementation of DAR's vertical aggregation $h_l = \sum_{i<l} \alpha_{i\to l}(t)\, v_i$ decomposes into separate kernels for per-source RMSNorm, query–key dot product, softmax, and weighted sum, each launching its own CUDA kernel, materializing $[N, B, T, D]$-shaped intermediates in HBM, and reading the source tensor four times per forward pass; since N scales with depth, this baseline is both memory- and bandwidth-bound. We collapse the entire forward path into a single Triton kernel that uses an online-softmax recurrence to fuse the normalization constant with the weighted accumulator in one streaming loop over the N sources, so that $\{v_i\}_{i<l}$ is read from HBM exactly once and all per-source intermediates — $\mathrm{RMS}(v_i)$, k_i, $q^\top k_i$, $\exp(\cdot)$ — live entirely in registers. The backward kernel applies the same idea in two passes: it first streams the sources to recover the softmax statistics (m, Z, s), then streams them again to recompute the RMSNorm intermediates on the fly and emit $\partial\mathcal{L}/\partial v_i$, $\partial\mathcal{L}/\partial q$, and $\partial\mathcal{L}/\partial w_{\text{NORM}}$ in two HBM reads instead of four to five; we further fuse the downstream LayerNorm and adaLN `modulate` into the same kernel. Microbenchmarks at the SiT-XL/2 working point ($N = 57$) in Fig. 6 show the fused kernel reducing forward latency from 22.5 ms to 1.96 ms ($11.5\times$) and backward from 115.8 ms to 13.6 ms ($8.5\times$) for the dynamic variant, with peak activation memory dropping by 78.7% in the forward and 74.6% in the backward pass (and up to 82.1% for the static variant); the savings grow monotonically with N, keeping the chunked aggregator viable as DiTs scale to deeper backbones. The fused kernel is numerically equivalent to the reference PyTorch path up to floating-point reordering and serves as a drop-in replacement used throughout the main paper.

Training-time efficiency. Table 7 compares SiT-XL/2 and DAR static c4 on the same NVIDIA H20 setup with a global batch size of 1024. Both models use the latest fused implementation; fused operators applicable to SiT are also enabled for the baseline. The measured training rates are 1.83 and 1.73 steps/s, respectively, corresponding to a 5.8% per-step overhead for DAR.

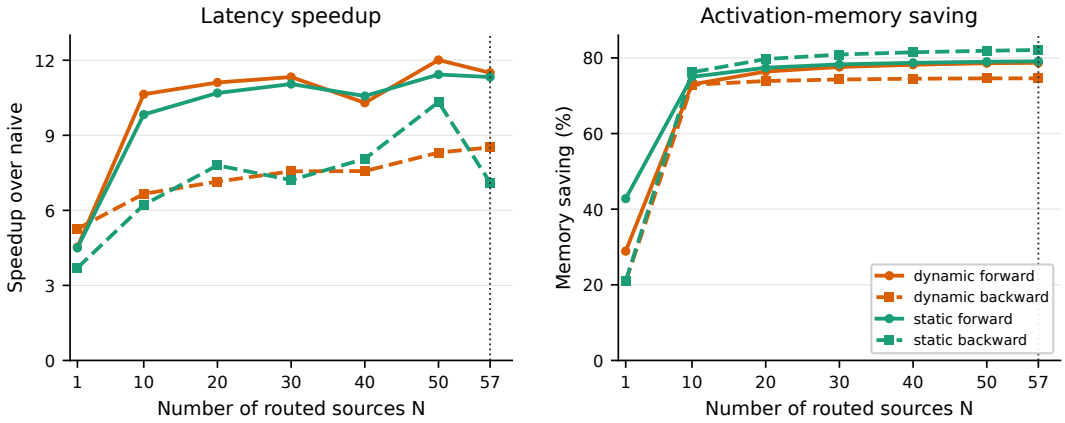


Figure 6: Infrastructure benchmark of the fused Triton implementation for DAR. Left: latency speedup over a naive implementation as a function of the number of routed sources N. Right: activation-memory saving, shown for the dynamic/static variants and the forward/backward passes.

Table 7: Training efficiency with fused operators on SiT-XL/2, ImageNet 256×256. Step times are rounded to two decimal places.

Model	**Time (s/step)**	**Steps/s**	**Images/s**	**Peak memory (GB)**
SiT-XL/2	0.55	1.83	1874	54.56
DAR (Static c4)	0.58	1.73	1771	69.97

Using the reported $8.75\times$ iteration reduction to the baseline-quality target (FID 9.67), the measured training rates imply an estimated wall-clock speedup of

$$\frac{T_{\mathrm{SiT}}}{T_{\mathrm{DAR}}} = \frac{N_{\mathrm{SiT}}}{N_{\mathrm{DAR}}} \frac{r_{\mathrm{DAR}}}{r_{\mathrm{SiT}}} \approx 8.75 \times \frac{1.73}{1.83} \approx 8.27\times,$$

where N is the number of training iterations and r is the measured rate in steps/s. Thus, after accounting for the per-step overhead, the estimated training time is approximately 12.1% of the baseline's at this target FID.

G Limitations and Future Work

We view the most compelling next step as pushing DAR along the two scale axes that dominate modern generative Transformers: large-scale pretraining and large-scale post-training. On the pretraining side, state-of-the-art T2I and T2V backbones such as MM-DiT [Esser et al., 2024], Qwen-Image [Wu et al., 2025], FLUX [Labs, 2024], and HunyuanVideo [Kong et al., 2024] routinely scale to several billion parameters with substantially deeper Transformer stacks, where the PreNorm-dilution symptoms diagnosed in Section 3 should manifest more severely than in SiT-XL/2; Eq. (9) further predicts that the optimal chunk size grows as $\sqrt{L}$, hinting that the headroom for DAR may widen rather than saturate with depth, which makes a systematic scaling study on multi-billion-parameter MM-DiT and video-DiT pretraining the most natural and informative follow-up direction we envision. On the post-training side, our preliminary DMD [Yin et al., 2024b,a] experiment on Qwen-Image equipped with DAR (Appendix E) suggests that the better-conditioned gradient flow afforded by adaptive routing has a detail-preserving effect on otherwise brittle distillation procedures where the vanilla counterpart loses fine details, and we plan to extend this observation to a broader family of post-training objectives — including supervised fine-tuning, RL-style preference optimization, and few-step distillation — across multiple large-scale T2I and T2V backbones, to assess whether DAR can serve as a general-purpose technique across diverse diffusion post-training settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: This work aims at expositing and developing the conventional pre-normalized residual streams in DiTs for achieving learnable, timestep-adaptive, and softmax-weights aggregations, which have been claimed clearly in Abstract and Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We have provided an in-depth discussion on the limitations and future studies in Appendix G. Besides, there are detailed discussions about our investigations and related works in Section 1 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: This paper provides a theoretical analysis of the U-shaped cost of chunked aggregation in Proposition 1, which is proved in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: Section 5 details the experimental configurations and results for demonstrating the effectiveness of our findings and proposed DAR method.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The data we used is of open license. We plan to release the code publicly.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 5 details the experimental configurations, including but not limited to the datasets, model versions, devices, hyper-parameters, and some training skills.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following the standard evaluation protocol of DiT/SiT, all metrics are computed over 50K generated samples per run; repeating full SiT-XL/2 training across multiple seeds is computationally prohibitive, and our gains are consistent across iteration checkpoints, ODE/SDE samplers, and with/without classifier-free guidance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendices C and F report the compute resources, training throughput, and peak memory used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We promise to obey the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: This paper presents work whose goal is to advance the fields of Computer Vision and Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The assets we used is of open license. We have cited the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.