ATTENTION RESIDUALS FOR PIXEL-SPACE FLOW TRANSFORMERS: AN EMPIRICAL STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention Residuals (AttnRes) replace fixed residual accumulation with content-dependent retrieval over earlier residual increments. We study the published Full AttnRes operator in class-conditional, pixel-space JiT flow Transformers under matched backbone and training settings. On ImageNet 256×256, Full AttnRes reduces six-encoder FDr^6 from 15.80 to 14.72 at B/16 (6.8%) and from 11.92 to 10.81 at L/16 (9.3%), with lower ratios in all six representation spaces at both scales. Learned parameters increase by approximately 0.01% and profiled inference FLOPs by below 1%, although full-history storage and mixing remain important systems costs. We also insert zero-query Full AttnRes into a shared 100-epoch standard-residual checkpoint. Equal-budget continuation raises IS from 224.6 for unchanged continuation to 230.6 after conversion. We explain approximate function preservation under explicit pre-normalization assumptions, localize most of an ordered ablation's endpoint gap to condition-token history, and observe flow-time-dependent depth routing. These results support studying Full AttnRes both from scratch and as a continued-training feature, while highlighting representation-dependent effect sizes and the need for repeated-run and systems evaluation.

1 INTRODUCTION

Generative Transformers combine token mixing, conditional modulation, and pre-normalized residual streams (Peebles & Xie, 2023; Lipman et al., 2023; Ma et al., 2024; Vaswani et al., 2017; He et al., 2016). In a standard residual stream, a later branch reads a fixed sum of earlier increments. Attention Residuals (AttnRes) (Kimi Team, 2026) instead let each token retrieve a content-dependent mixture from this depth history. The operator was introduced for language-model pre-training; its behavior in conditioned image generation remains a separate empirical question.

Pixel-space flow models provide a useful setting for this question. Their inputs evolve from noise-dominated to data-dominated states, and condition-bearing prefix tokens evolve alongside image tokens across depth. Prior analyses of denoising (Choi et al., 2022) and architectures with long feature paths (Rombach et al., 2022; Crowson et al., 2024; Chen et al., 2025; Wang et al., 2026) motivate examining whether the useful history changes across the generation trajectory. We study this possibility without introducing a new routing operator or a timestep-conditioned query network.

We instantiate the published Full AttnRes operator in a JiT-style pixel-space flow Transformer and call the resulting experimental model AttnRes-JiT. The backbone, conditioning interface, objective, optimizer, and sampling recipe are matched to the standard-residual control. This isolates the effect of replacing fixed residual aggregation with learned history retrieval. Because the history already depends on time and class conditioning, a static learned query can still produce sample- and time-dependent weights. Figure 1 illustrates the cross-layer readout and how its weights can change with flow time.

Our study addresses three questions:

1. **From-scratch quality and cost.** Does Full AttnRes improve matched B/16 and L/16 models, and how do parameter/FLOP overheads differ from full-history storage and mixing costs?

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

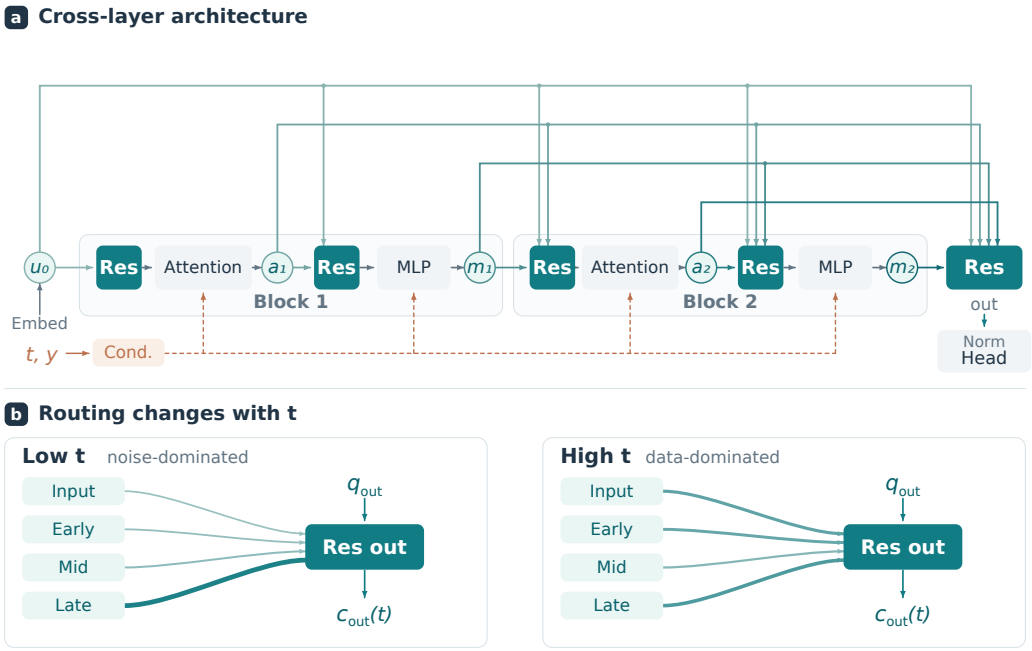

Figure 1: **Full AttnRes with explicit cross-layer reads.** **(a)** Two blocks are unrolled: each Res module reads all preceding increments, including the initial embedding $\mathbf{u}_0$ and gated branch outputs $\mathbf{a}_l, \mathbf{m}_l$. Normalization, modulation, and gates are omitted inside branch boxes. This connectivity extends through the full backbone; image and condition-token histories remain separate. **(b)** Output routing with the same query at different t. Line widths illustrate source-depth group weights, not measured values or a grouped implementation; time changes the conditioned history.

2. **Post-hoc conversion.** Can zero-query Full AttnRes be inserted into a standard-residual checkpoint and improve on equal-budget continuation? We explain why its initial mean carrier is approximately function-preserving under explicit normalization assumptions.
3. **Location and measurement of the effect.** How much of an ordered ablation's endpoint gap is associated with condition-token history, how does routing change with flow time, and how does the measured difference depend on the evaluation representation?

The contribution is a controlled empirical study of an existing operator: whether conditioned histories support useful depth retrieval without a new routing parameterization, and whether that readout can be introduced during continued training. The metric ensemble characterizes the effect rather than establishing a universally superior evaluator.

2 Related Work

From residual addition to depth retrieval. ResNets and DenseNets establish additive and concatenative feature reuse (He et al., 2016; Huang et al., 2017). DLCL and DenseFormer learn depth-combination coefficients (Wang et al., 2019; Pagliardini et al., 2024); DeepCrossAttention makes cross-layer mixing input-dependent (Heddes et al., 2025). Hyper-Connections and mHC instead mix multiple residual streams (Zhu et al., 2025; Xie et al., 2026). AttnRes (Kimi Team, 2026) retrieves earlier increments using softmax depth attention. We retain its Full variant; neither learned depth aggregation nor the operator itself is our contribution.

Routing in diffusion Transformers. U-ViT, Skip-DiT, and DDT revisit long paths or encoder–decoder structure (Bao et al., 2023; Chen et al., 2025; Wang et al., 2026). DAR (Xu et al., 2026) directly studies diffusion depth routing, including static queries, explicit timestep injection, dynamic queries, and chunked aggregation. Our distinction is the controlled study of Full AttnRes in pixel-

space JiT, with separate prefix histories and shared-checkpoint conversion. Static queries alone do not distinguish us from DAR; matched routing comparisons remain open.

Pixel-space generation and representation learning. Pixel modeling has progressed from simple diffusion and hourglass Transformers (Hoogeboom et al., 2023; 2025; Crowson et al., 2024) to JiT and PixelDiT (Li & He, 2025; Yu et al., 2026a). Recent PixelU uses U-shaped feature paths (Guo et al., 2026). PixelREPA adapts representation alignment to JiT (Shin et al., 2026), while PixelDiT2 injects frozen-encoder features during denoising (Yu et al., 2026b). These change topology or external representation guidance; our matched study changes retrieval of internally computed history. Compatibility is a hypothesis, not an evaluated result.

Conditioning tokens and registers. FiLM and adaLN modulate features (Perez et al., 2018; Peebles & Xie, 2023), whereas registers provide additional computational tokens (Darcet et al., 2024). Starodubcev et al. (2026) show register-like behavior in JiT in-context tokens and introduce Register Guidance. Our ablation retains the prefix and changes its history readout; it does not isolate class information from register-like computation.

Conversion and evaluation. Function-preserving transfer has a precedent in Net2Net (Chen et al., 2016); our conversion depends specifically on zero-query means and pre-normalization. Evaluation studies identify encoder dependence, finite-sample bias, and preprocessing sensitivity (Stein et al., 2023; Chong & Forsyth, 2020; Parmar et al., 2022). These motivate reporting FID alongside FDr^6 (Yang et al., 2026), without treating their agreement as independent evidence of perceptual superiority. Appendix G expands these distinctions and attributes the evaluation encoders.

3 Full AttnRes in Pixel-Space JiT

Our class-conditional backbone uses the interpolation

$$\mathbf{z}_t = t\mathbf{x} + (1-t)\boldsymbol{\epsilon}, \qquad t \in [0,1], \tag{1}$$

where $\mathbf{x}$ is a clean image and $\boldsymbol{\epsilon}$ is Gaussian noise. Small t is noise-dominated and large t is data-dominated. Both residual variants retain the same patch embedding, attention/MLP branches, prefix-token conditioning, existing adaLN-style modulation, and prediction head. We replace only the carrier read before each branch.

3.1 The Residual Carrier as a Design Axis

A standard pre-norm Transformer sublayer updates its accumulated state as

$$\mathbf{h}_l = \mathbf{h}_{l-1} + \mathcal{F}_l(\mathrm{Norm}(\mathbf{h}_{l-1})), \tag{2}$$

Writing this state in terms of the initial embedding $\mathbf{u}_0$ and subsequent branch increments $\mathbf{u}_i$ makes the residual carrier explicit:

$$\mathbf{s}_n = \sum_{i=0}^{n-1} \mathbf{u}_i, \qquad \mathbf{u}_n = \mathcal{F}_n\big(\mathrm{Norm}(\mathbf{s}_n)\big). \tag{3}$$

The next branch therefore reads a fixed unit-weight sum of all preceding increments. Full AttnRes replaces this sum with a learned read from the same history $\mathcal{H}_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$. This separates two interactions: self-attention mixes tokens within a sequence, while residual attention selects which earlier sublayer outputs supply the next branch. Our experimental instantiation retains the JiT branch and conditioning computations and changes only this depth-wise carrier.

3.2 The Full AttnRes Operator

Operator. For a single token group, let $\mathbf{u}_0$ be its initial embedding and later entries $\mathbf{u}_i \in \mathbb{R}^{B\times S\times D}$ be residual increments, where B is the batch size, S is the sequence length, and D is the hidden dimension. Given the residual-increment history $\mathcal{H}_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$, Full AttnRes first stacks it along a depth axis,

$$\mathbf{U} = \mathrm{stack}(\mathcal{H}_n, \mathrm{dim} = 2) \in \mathbb{R}^{B\times S\times n\times D}. \tag{4}$$

Each Full AttnRes module owns a learned query vector $\mathbf{q}_n \in \mathbb{R}^D$. Scores are computed independently for every batch element and token:

$$e_{n,i,b,s} = \mathbf{q}_n^\top \mathrm{RMSNorm}\big(\mathbf{U}_{b,s,i,:}\big), \tag{5}$$

$$\alpha_{n,i,b,s} = \mathrm{softmax}_{i=0}^{n-1}(e_{n,i,b,s}), \tag{6}$$

where RMSNorm (Zhang & Sennrich, 2019) is parameter-free here and used only for score computation. The residual carrier is the weighted sum of the original, unnormalized increments:

$$\mathrm{AttnRes}_n(\mathcal{H}_n)_{b,s,:} = \sum_{i=0}^{n-1} \alpha_{n,i,b,s} \mathbf{U}_{b,s,i,:}. \tag{7}$$

There are no query, key, or value projection matrices in the residual-attention module, and no value projection W_v. The only learned parameter of one Full AttnRes module is its query vector $\mathbf{q}_n$.

History and conditioning. Each attention or MLP branch appends its conditioned, gated output as a new increment, rather than adding it to the retrieved carrier. A depth-L model therefore has $2L+1$ image-history entries. A separate `output_residual` mixer reads this history before the normalized prediction head. Token groups inserted at different depths maintain separate, shape-aligned histories, which are mixed before concatenation for attention. The query itself is not condition-derived: time and class influence routing through the conditioned increments. Figure 1 shows the cross-layer connectivity; Appendix A gives the recurrence, and Appendix D gives the conditioned block pseudo-code.

3.3 Zero-Query Initialization and Checkpoint Conversion

All queries start at zero, so history weights are uniform. For the standard residual sum $\mathbf{s}_n = \sum_{i=0}^{n-1} \mathbf{u}_i$, the initial carrier satisfies

$$\mathrm{AttnRes}_n(\mathcal{H}_n) = \frac{\mathbf{s}_n}{n}, \qquad \mathrm{Norm}\Big(\frac{\mathbf{s}_n}{n}\Big) = \mathrm{Norm}(\mathbf{s}_n). \tag{8}$$

The second equality assumes scale-invariant normalization with zero numerical ϵ. With identical increments, pre-normalized branches, and a normalized final readout, it extends through the network. Nonzero ϵ makes this an approximate argument, not an unconditional identity for arbitrary architectures; the detailed derivation is in Appendix A.2.

For conversion, we copy the shared embedding, branch, conditioning, normalization, and output-head weights from the standard-residual checkpoint and initialize the added queries to zero. The comparison is continued training of this converted model versus equal-budget continuation of the unchanged checkpoint. Including the final readout, Full AttnRes adds $(2L+1)D$ query parameters. Its full-history storage and repeated depth mixing remain distinct costs from this small parameter increase.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation. We evaluate class-conditional generation on ImageNet (Deng et al., 2009; Russakovsky et al., 2015) at 256×256 resolution under our JiT-style flow-matching recipe. Following standard practice, we report Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) using 50K generated samples. To characterize generation quality beyond a single Inception feature space, we additionally report FDr^6 (Yang et al., 2026). For a representation encoder ϕ, the normalized ratio is $\mathrm{FDr}_\phi = \mathrm{FD}_\phi(\mathcal{G},\mathcal{T})/\mathrm{FD}_\phi(\mathcal{V},\mathcal{T})$, where $\mathcal{G}$, $\mathcal{T}$, and $\mathcal{V}$ denote the generated set, ImageNet training set, and ImageNet validation set, respectively. FDr^6 is the arithmetic mean of these ratios across Inception-v3, ConvNeXt-v2, DINOv2, MAE, SigLIP2, and CLIP; lower is better. We evaluate the B/16 models with classifier-free guidance (CFG) (Ho & Salimans, 2022) scale 2.8 and the L/16 and H/16 models with CFG scale 2.4. For comparisons with published systems, we report representation space and denoiser parameter counts; these external rows contextualize quality rather than rank inference cost.

Baselines. Our primary comparisons use matched standard-residual JiT models. Appendix B places these results alongside published latent- and pixel-space generators with different compute and sampling budgets; those external rows are contextual references, not controlled comparisons.

Training. Unless otherwise specified, AttnRes-JiT is trained with the same flow-matching objective, optimizer, EMA, training length, sampling protocol, prefix-token interface, and adaLN-style conditioning path as the matched standard-residual baseline. AttnRes-JiT uses the full residual-increment history, two Full Attention Residual modules per block, and zero-initialized query vectors. Detailed hyperparameters are reported in Appendix E.

4.2 Main Results: FDr^6 and FID

Table 1: **Matched ImageNet** 256×256 **evaluation.** Each pair uses the same JiT backbone and training recipe; only the residual carrier changes. B/16 uses CFG 2.8 and L/16 uses CFG 2.4. Lower FID and FDr^6 are better; higher IS is better.

Scale	Residual carrier	Params	GFLOPs	FID↓	IS↑	FDr^6↓
B/16	Standard residual	131M	50.45	4.08	264.9	15.80
	Full AttnRes	131M	50.60	**3.95**	**266.9**	**14.72**
L/16	Standard residual	459M	175.89	2.85	278.8	11.92
	Full AttnRes	459M	176.59	**2.84**	**283.0**	**10.81**

Full AttnRes reduces FDr^6 by approximately 6.8% at B/16 and 9.3% at L/16 in the matched comparisons. Table 1 reports the aggregate results: FDr^6 changes from 15.80 to 14.72 and from 11.92 to 10.81, respectively. The corresponding FID changes are 4.08 to 3.95 and 2.85 to 2.84. Table 2 presents the complete encoder-wise results in the main text: all six representation-space ratios decrease at both matched scales. Its H/16 column is a standalone result, not a matched comparison. Appendix B provides the broader published-model comparison.

Table 2: **FDr^6 and complete encoder-wise results.** FDr^6 decreases by 6.8% at B/16 and 9.3% at L/16, with lower ratios in all six feature spaces. The six encoder rows report normalized FDr ratios (lower is better); IS is higher-is-better. H/16 has no matched standard-residual control.

Metric	B/16		L/16		H/16
	Standard	Full AttnRes	Standard	Full AttnRes	Full AttnRes
CFG	2.8	2.8	2.4	2.4	2.4
IS↑	264.9	266.9	278.8	283.0	299.8
FDr^6↓	15.80	**14.72**	11.92	**10.81**	9.09
Inception-v3	2.31	2.24	1.63	1.61	1.47
ConvNeXt-v2	5.01	4.92	4.08	4.03	3.26
DINOv2	10.13	9.10	6.32	5.11	4.26
MAE	14.29	13.57	11.38	10.75	8.70
SigLIP2	26.97	24.06	18.16	15.12	11.70
CLIP	36.11	34.45	29.93	28.25	25.16

FDr^6 reveals representation-dependent effect sizes. Figure 2 makes the L/16 contrast explicit: Inception FID changes little, while the representation ensemble shows a larger relative difference. All six constituent ratios decrease at both matched scales. These are point estimates; the small FID differences alone do not establish statistical significance.

These results do not establish universal accuracy of FDr^6 or perceptual superiority. The encoders capture complementary but statistically dependent aspects of the generated distribution. Establishing which metric best agrees with perceived quality would require human judgments or an external downstream criterion, neither of which is evaluated here.

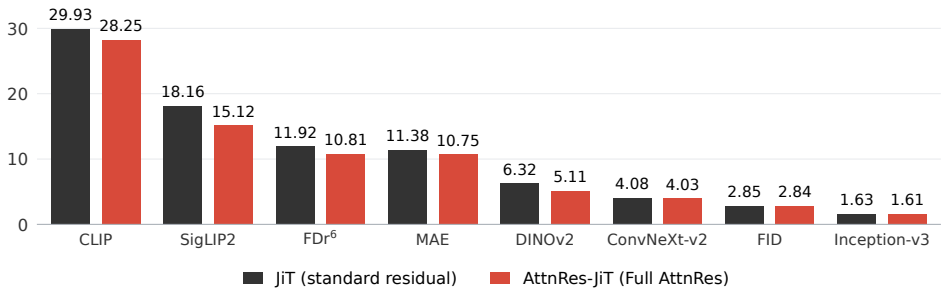

Figure 2: **Metric dependence in the matched L/16 comparison (CFG 2.4).** JiT uses standard residuals; AttnRes-JiT uses Full AttnRes. FID changes from 2.85 to 2.84, while FDr^6 decreases from 11.92 to 10.81 (9.3%). Encoder-named categories show normalized FDr ratios. All metrics are lower-is-better; numerical scales differ across metrics.

Qualitative comparison. Figure 3 complements the distributional metrics with three matched L/16 sample pairs at CFG 2.4. The examples illustrate the appearance of generated objects and backgrounds under the same sampling configuration; selected images do not establish a general perceptual ranking. Appendix F provides a broader sample grid.



Figure 3: **Three matched L/16 sample pairs at CFG 2.4.** Standard residuals are shown on the left of each pair and Full AttnRes on the right. These examples illustrate sample appearance; aggregate comparisons are reported in Tables 1 and 2.

Nearest-neighbor diagnostic. We generate 1,000 images from the Full-AttnRes JiT-L/16 model at CFG 2.4 and perform an exact global top-1 search over all 1,281,167 ImageNet training images using normalized InceptionV3 pool-2048 features. Retrieval is not restricted by class. For a generation g_i, we use cosine distance

$$d_i = 1 - \max_{x \in \mathcal{D}_{\text{train}}} \hat{f}(g_i)^\top \hat{f}(x). \tag{9}$$

None of the 1,000 queries has a zero-distance match. The minimum, median, and mean distances are 0.0144, 0.1331, and 0.1404, respectively; 81.3% of the retrieved neighbors share the query class. This is a limited Inception-feature diagnostic rather than a general test of memorization.

Appendix C gives the full distance summary and retrieved image pairs.

4.3 Attention Residuals as a Continued-Training Feature

We test whether Full AttnRes can be introduced during continued training of an existing standard-residual model. Both protocols start from the same Standard-B/16 checkpoint after 100 epochs and train for 100 additional epochs. The control keeps standard residuals; the conversion variant inserts the zero-query modules described in Section 3.3. Both reinitialize the Muon optimizer state and use the same continuation recipe, sampler, sampling-step budget, and classifier-free guidance sweep. The comparison therefore holds prior optimization and the additional training budget fixed.

Table 3 separates conversion from unchanged continuation of the shared checkpoint. Conversion improves FID from 5.22 to 5.06 and IS from 224.6 to 230.6 relative to the equal-budget control. In this experiment, Full AttnRes is useful without being present throughout from-scratch training: it preserves the original backbone weights at insertion and adapts their history readout during continued optimization.

Table 3: **Post-hoc Full-AttnRes conversion on B/16.** Both continuation rows start from the same 100-epoch standard-residual checkpoint and receive 100 additional epochs under the same recipe.

Continuation protocol	Residual path	Extra epochs	FID↓	IS↑
Start checkpoint	Standard residual	0	6.16	221.5
Standard-residual continuation	Standard residual	100	5.22	224.6
Full AttnRes insertion + continuation	Learned full-history Full AttnRes	100	**5.06**	**230.6**

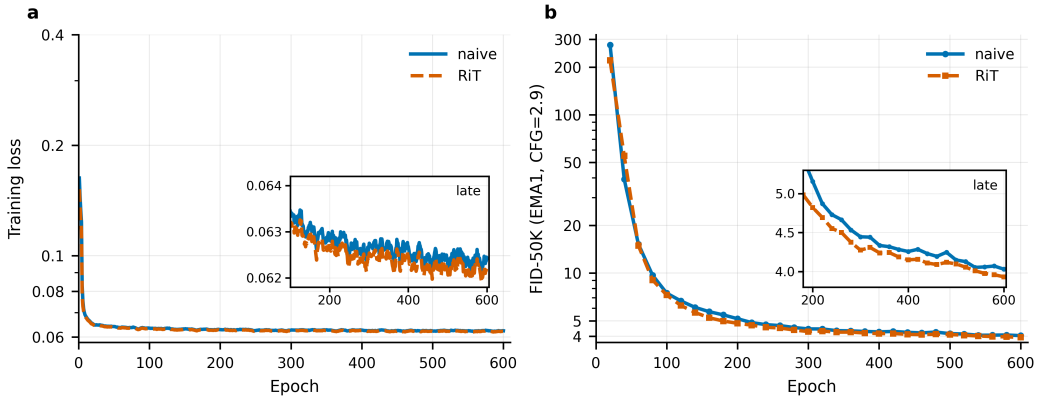

Figure 4: **Training dynamics of matched B/16 models.** Full AttnRes and the standard-residual baseline are compared over 600 epochs in training loss (a) and FID-50K from EMA weights at CFG 2.9 (b); lower is better. This trajectory evaluation uses a different CFG from the B/16 endpoint comparison in Table 1 (2.8).

4.4 Computational Efficiency and Training Dynamics

Parameter and profiled-FLOP overhead. Each mixer adds one D-dimensional query, with no residual QKV/value projections or extra backbone layers. Parameters increase by approximately 0.01%, and profiled per-forward inference FLOPs by below 1%. These figures omit important systems costs: storing and repeatedly mixing the depth history incurs activation storage and memory traffic, so they do not establish wall-clock efficiency.

Optimization behavior. We also compare training loss and FID trajectories of the matched B/16 runs over 600 epochs to assess how full-history retrieval affects learning.

Figure 4 shows similar optimization trajectories, with a consistently lower late-stage loss for Full AttnRes. After the early transient, Full AttnRes also maintains lower FID through most of training and finishes below the matched baseline. The difference extends beyond a single endpoint in these runs, but does not establish variability across independent training seeds.

5 Where the Effect Appears: Condition History and Routing

AttnRes-JiT maintains depth histories for both image tokens and auxiliary in-context tokens. We first isolate how condition-bearing token histories contribute to the gain, and then examine how flow time changes the source depth selected from those conditioned histories.

5.1 Contribution of In-Context Token History

At their configured start block in the JiT-style backbone, in-context tokens are inserted as an auxiliary prefix before the image-token sequence. They carry conditioning information and subsequently traverse the Transformer alongside the image tokens. To isolate whether their depth-wise

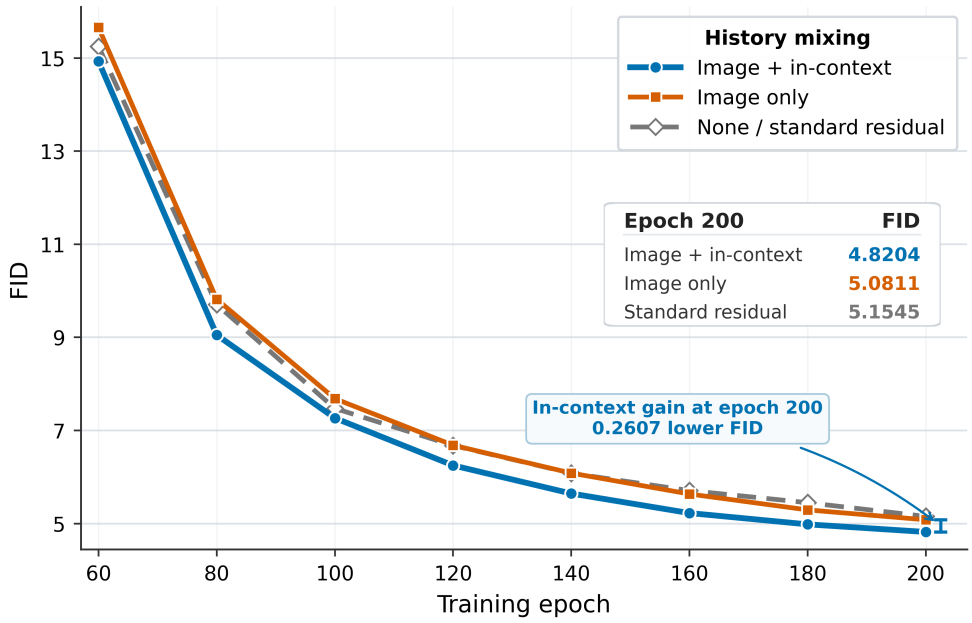

Figure 5: **Condition-token history ablation.** Full history gives the lowest FID at the evaluated checkpoints. At epoch 200, restoring condition-token history contributes 78% of the endpoint gap along the ordered standard → image-only → full-history path.

history matters, we compare full AttnRes-JiT, which mixes the residual histories of both token types, against an image-only variant that applies history mixing only to image tokens, and against a standard-residual baseline.

Figure 5 shows that Full AttnRes with both token histories achieves the lowest FID at every evaluated checkpoint. At epoch 200, Full AttnRes reaches 4.8204 FID, compared with 5.0811 for image-only history mixing and 5.1545 for standard residuals. Image-only mixing reduces FID by 0.0734 relative to the baseline, whereas restoring in-context-token history yields a further 0.2607 reduction. The latter difference is 78% of the total 0.3341 endpoint gap along this ordered ablation path. This path-dependent decomposition does not by itself separate main effects from interactions between image-token and condition-token history, but it shows that retaining the depth-wise evolution of the condition-bearing prefix is associated with most of the observed endpoint difference and motivates examining whether history readout changes with flow time.

5.2 Output-Residual Routing Across Flow Time

Because prefix tokens may also act as registers (Starodubcev et al., 2026), the ablation does not attribute the gap exclusively to class conditioning. We next inspect image-token routing in the final output mixer across flow time; this is a separate diagnostic, not a direct measurement of prefix-history routing.

Figure 6 visualizes the learned routing distribution of the final `output_residual` mixer at checkpoint 200. We evaluate flow times $t \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$ under the interpolation in Eq. (1). For each t, residual-increment weights are averaged over image tokens only, using 8 ImageNet classes, 16 random noise seeds, and all image-token positions. We group history entries by source depth: *input* denotes the initial patch embedding $\mathbf{u}_0$ (`input_embed`), *early* denotes blocks 0–3, *mid* denotes blocks 4–7, and *late* denotes blocks 8–11. The plotted value is the sum of residual-increment softmax mass within each depth group, so the four group masses sum to one for each t.

The measured routing weights vary systematically with flow time. When t is small and $\mathbf{z}_t$ is noise-dominated, the final mixer assigns almost all softmax mass to late history entries. As t increases

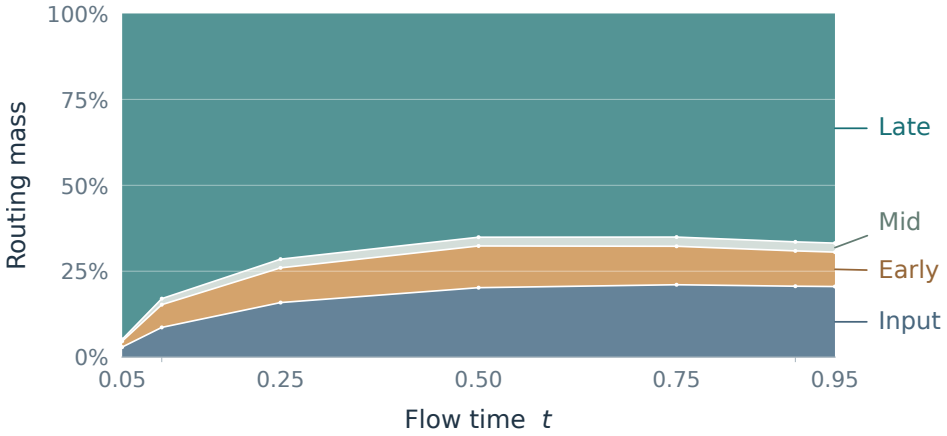

Figure 6: **Trajectory-dependent routing of the final output mixer.** Stacked bands show each depth group's share of routing mass, averaged over 8 classes, 16 noise seeds, and image tokens. Low-t inputs concentrate on late histories; higher t reallocates mass toward input and early histories. Values are digitized from the original raster figure, with estimated reading precision of about 0.2 percentage points.

and $\mathbf{z}_t$ becomes more data-dominated, it reallocates mass toward the initial input and early blocks, while late entries remain dominant. Because the values are unnormalized residual increments, these weights describe routing mass rather than a complete causal decomposition of output contribution.

Taken together, the analyses show an association between temporal conditioning and cross-layer routing. Flow time does not generate the Full AttnRes query directly; instead, time and class signals shape image-token and in-context-token residual histories through the backbone's standard prefix-token and adaLN pathways. The static query consequently produces different routing weights as these histories change, without an explicit timestep-conditioned routing network.

6 Discussion and Conclusion

Full AttnRes can be studied as both a from-scratch residual carrier and a continued-training feature for pixel-space JiT. Matched experiments show modest FID differences and larger changes across the evaluated representation ensemble; shared-checkpoint continuation improves the reported FID and IS relative to unchanged continuation. Condition-history ablations and trajectory-wise routing localize where the observed differences arise. These findings concern the behavior of an existing operator in a specific generative setting.

Limitations. The study covers one dataset, resolution, conditioning regime, and backbone family. The reported results are point estimates without uncertainty over independent training runs; the L/16 FID difference is not a statistically established gain. Representation encoders are not independent, and no human judgments or downstream tests establish perceptual superiority. The condition-history decomposition depends on the ablation order and does not isolate interactions. Full AttnRes requires $O(L)$ history storage and $O(L^2)$ depth mixing; profiled FLOPs do not establish wall-clock or memory efficiency. The conversion argument assumes pre-normalized branches, normalized readout, and idealized zero-ϵ normalization. Repeated training seeds, direct conversion-error measurements, system profiling, a matched H/16 control, and comparisons with alternative depth-routing mechanisms remain necessary for broader conclusions.

AI USE STATEMENT

Generative AI tools assisted translation and language editing of the manuscript. During revision, they also assisted LaTeX formatting, cross-reference and bilingual consistency checks, literature retrieval, bibliography verification, and discussion of related-work positioning and interpretive limitations. These activities did not generate experimental data or new experimental results. The research did not use generative AI to develop theoretical models, formulate or prove mathematical claims, propose research hypotheses, design experiments, implement the research methods, or clean datasets. Interpretation-related revision assistance was limited to discussing the scope and limitations of claims against the cited literature. The authors retain responsibility for the final text, technical claims, and reported results.

ETHICS STATEMENT

This study evaluates class-conditional image generation on ImageNet. Such models can reproduce biases or sensitive content present in their training data and can be misused to create misleading imagery. The nearest-neighbor diagnostic in Section 4.2 and Appendix C is limited to a particular feature space and sample set; it does not certify absence of memorization or privacy risk. No human-subject evaluation is reported. Use of the dataset, pretrained feature encoders, and any generated or retrieved images should respect their applicable access and licensing conditions. Distributional quality metrics alone do not establish safety or fairness.

REPRODUCIBILITY STATEMENT

Section 3 specifies the Full AttnRes operator and conversion assumptions. Section 3.1 gives the residual-carrier formulation, and Appendix A provides the full recurrence and initialization derivation; Appendix D provides PyTorch-style block pseudo-code; and Appendix E lists the reported architecture, training, sampling, and evaluation settings. Section 4.3 defines the shared-checkpoint continuation control. Table 2 reports the complete encoder-wise metrics, Section 4.2 specifies the nearest-neighbor diagnostic, and Section 6 states the limits of the available experimental evidence. The pseudo-code describes the core mechanism rather than a complete training and evaluation pipeline.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2209.12152.

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 *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Towards_Stabilized_and_Efficient_Diffusion_Transformers_through_Long-Skip-Connections_with_Spectral_ICCV_2025_paper.html.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.00426.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2Net: Accelerating Learning via Knowledge Transfer. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.05641.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2204.00227.

Min Jin Chong and David Forsyth. Effectively unbiased FID and inception score and where to find them. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Chong_Effectively_Unbiased_FID_and_Inception_Score_and_Where_to_Find_CVPR_2020_paper.html.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Proc. of the International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2401.11605.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.16588.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2105.05233.

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. *arXiv preprint arXiv:2403.03206*, 2024. URL https://arxiv.org/abs/2403.03206.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024. URL https://arxiv.org/abs/2407.11633.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. PixelU: A U-Shaped Transformer for Efficient End-to-End Pixel Diffusion. *arXiv preprint arXiv:2606.27760*, 2026. URL https://arxiv.org/abs/2606.27760.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://arxiv.org/abs/1512.03385.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2111.06377.

Mike Heddes, Adel Javanmard, Kyriakos Axiotis, Gang Fu, MohammadHossein Bateni, and Vahab Mirrokni. DeepCrossAttention: Supercharging Transformer Residual Connections. In *Proc. of the International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.06785.

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 (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2006.11239.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *Proc. of the International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2301.11093.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 FID on ImageNet512 with pixel-space diffusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2410.19324.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://arxiv.org/abs/1608.06993.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024.

Kimi Team. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026. URL https://arxiv.org/abs/2603.15031.

Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2303.00848.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

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 *Proc. of the European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2401.08740.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

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 (NeurIPS)*, 2024. URL https://arxiv.org/abs/2402.02622.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2104.11222.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2212.09748.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. URL https://arxiv.org/abs/1709.07871.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of the International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2103.00020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.10752.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. URL https://arxiv.org/abs/1409.0575.

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 (NeurIPS)*, 2016. URL https://arxiv.org/abs/1606.03498.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1701.06538.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation Alignment for Just Image Transformers is not Easier than You Think. *arXiv preprint arXiv:2603.14366*, 2026. URL https://arxiv.org/abs/2603.14366.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.02502.

Nikita Starodubcev, Ilia Sudakov, Ilya Drobyshevskiy, Artem Babenko, and Dmitry Baranchuk. Registers Matter for Pixel-Space Diffusion Transformers. *arXiv preprint arXiv:2605.16147*, 2026. URL https://arxiv.org/abs/2605.16147.

George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.04675.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://arxiv.org/abs/1512.00567.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, et al. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.03762.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://arxiv.org/abs/1906.01787.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2504.05741.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2301.00808.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, et al. mHC: Manifold-Constrained Hyper-Connections. *arXiv preprint arXiv:2512.24880v2*, 2026. URL https://arxiv.org/abs/2512.24880v2.

Chao Xu, Maohua Li, Qirui Li, Yixuan Xu, Yanke Zhou, Yunhe Li, Cuifeng Shen, Hanlin Tang, Kan Liu, Tao Lan, Lin Qu, and Shao-Qun Zhang. Rethinking cross-layer information routing in diffusion transformers. *arXiv preprint arXiv:2605.20708*, 2026. URL https://arxiv.org/abs/2605.20708.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fréchet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026. URL https://arxiv.org/abs/2604.28190.

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 *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.06940.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel diffusion transformers for image generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. URL https://arxiv.org/abs/2511.20645.

Yongsheng Yu, Wei Xiong, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT2: Representation-Grounded Pixel Diffusion Transformers. *arXiv preprint arXiv:2609.24919*, 2026b. URL https://arxiv.org/abs/2609.24919.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1910.07467.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. URL https://arxiv.org/abs/2510.11690.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-Connections. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.19606.

A Residual-Carrier Details

Section 3.1 develops the residual-carrier view. Here we give the full attention/MLP recurrence and the detailed zero-query initialization argument supporting Section 3.3.

A.1 Full-History Residual Routing

We use the full sublayer history rather than grouping layers into coarse stages. Let $\mathbf{u}_0$ be the initial token embedding. Each attention or MLP sublayer appends one residual increment to the history, so a Transformer block contributes two history entries.

The schematic recurrence below suppresses conditional modulation and gates. The stored increments in the conditioned block include these operations, as expanded in Appendix D.

Given a history $\mathcal{H}_n$, the attention sublayer retrieves its carrier and appends its output:

$$\mathbf{x}_l^{\mathrm{attn}} = \mathrm{AttnRes}_l^{\mathrm{attn}}(\mathcal{H}_n), \tag{10}$$

$$\mathbf{a}_l = \mathrm{MSA}_l\big(\mathrm{LN}_1(\mathbf{x}_l^{\mathrm{attn}})\big), \tag{11}$$

$$\mathcal{H}_{n+1} = [\mathcal{H}_n, \mathbf{a}_l]. \tag{12}$$

The MLP sublayer then retrieves from the updated history:

$$\mathbf{x}_l^{\mathrm{mlp}} = \mathrm{AttnRes}_l^{\mathrm{mlp}}(\mathcal{H}_{n+1}), \tag{13}$$

$$\mathbf{m}_l = \mathrm{MLP}_l\big(\mathrm{LN}_2(\mathbf{x}_l^{\mathrm{mlp}})\big), \tag{14}$$

$$\mathcal{H}_{n+2} = [\mathcal{H}_{n+1}, \mathbf{m}_l]. \tag{15}$$

There is no ordinary residual update of the form $x + \mathrm{MSA}(x)$ or $x + \mathrm{MLP}(x)$ inside this residual mode. The residual path is the Full AttnRes carrier itself.

For a model with L Transformer blocks, the full history contains the initial token embedding plus two residual increments per block:

$$|\mathcal{H}| = 2L + 1. \tag{16}$$

After all Transformer blocks, we use a separate final mixer, denoted `output_residual`, to perform a final readout over the full image-token residual-increment history and construct the carrier for the prediction head:

$$\mathbf{x}^{\mathrm{out}} = \mathrm{AttnRes}^{\mathrm{out}}(\mathcal{H}_{2L+1}^{x}), \tag{17}$$

$$\widehat{\mathbf{y}} = \mathrm{Head}\big(\mathrm{Norm}(\mathbf{x}^{\mathrm{out}})\big), \tag{18}$$

Here, $\mathcal{H}_{2L+1}^{x}$ denotes the full image-token residual-increment history. This is the final output residual mixer visualized in Section 5.

A.2 Zero-Query Mean Initialization and Post-Hoc Insertion

Following the original Full AttnRes design (Kimi Team, 2026), every pseudo-query is initialized to zero:

$$\mathbf{q}_n = \mathbf{0}. \tag{19}$$

Then every score in Eq. (5) is zero and the softmax over history is uniform:

$$\mathrm{AttnRes}_n(\mathcal{H}_n) = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{u}_i. \tag{20}$$

This starts the residual carrier as deterministic mean pooling over the available history rather than random routing.

This mean carrier is structurally aligned with the standard residual stream. Let $\mathbf{u}_i$ denote the residual increment produced by sublayer i, with $\mathbf{u}_0$ denoting the initial token embedding. A standard pre-norm Transformer presents the next sublayer with

$$\mathbf{s}_n = \sum_{i=0}^{n-1} \mathbf{u}_i, \qquad \mathbf{u}_n = \mathcal{F}_n\big(\mathrm{Norm}(\mathbf{s}_n)\big). \tag{21}$$

If Full AttnRes stores the same increments in its history $\mathcal{H}_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$, zero-query initialization gives

$$\bar{\mathbf{s}}_n = \mathrm{AttnRes}_n(\mathcal{H}_n) = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{u}_i = \frac{1}{n}\mathbf{s}_n. \tag{22}$$

In the idealized zero-ϵ formulation, RMSNorm and LayerNorm (Zhang & Sennrich, 2019; Ba et al., 2016) are invariant to positive scalar rescaling:

$$\mathrm{Norm}(\bar{\mathbf{s}}_n) = \mathrm{Norm}\left(\frac{1}{n}\mathbf{s}_n\right) = \mathrm{Norm}(\mathbf{s}_n). \tag{23}$$

Therefore the next branch computation is initialized to the same normalized input as the standard-residual baseline:

$$\mathcal{F}_n\big(\mathrm{Norm}(\mathrm{AttnRes}_n(\mathcal{H}_n))\big) = \mathcal{F}_n\big(\mathrm{Norm}(\mathbf{s}_n)\big). \tag{24}$$

Under these assumptions, the model starts from a mean-pooling carrier that is functionally equivalent after pre-normalization to the standard residual sum, rather than from a random routing pattern. With the nonzero numerical ϵ used in implementation, the equivalence is approximate. During training, the query vector can learn to prefer different historical increments for different tokens.

The same initialization motivates post-hoc insertion into a trained standard-residual backbone. Given a trained checkpoint, we copy the shared Transformer weights—token embeddings, normalization layers, attention modules, MLPs, conditioning modules, and output head—and replace each local residual carrier with a Full AttnRes module over the accumulated increments. The relevant baseline is *standard-residual continuation*: continue training the unchanged checkpoint for the same extra budget, with the same reinitialized Muon optimizer, EMA, sampler, and data. The resulting comparison measures the effect of conversion under this continuation protocol.

Full AttnRes is parameter-light: including the two per-block residual mixers and the final output mixer, a depth-L, width-D backbone adds only $(2L+1)D$ learned query parameters, corresponding to approximately 0.01% parameter overhead in our evaluated settings. Its practical cost therefore comes from storing and mixing the full residual-increment history rather than from learned weights, which we discuss in the limitations.

B Extended Generation Results

Table 4: **Broader ImageNet** 256×256 **positioning.** Published diffusion/flow systems are external reference points rather than matched controls. Parameters count denoisers only, excluding autoencoders, external encoders, and guidance networks. Sampling and FLOP conventions vary across sources, so inference costs are not compared here. H/16 is a standalone Full-AttnRes run because a matched standard-residual H/16 run is unavailable.

Model	**Space**	**Params**	**FID↓**	**IS↑**
Multi-step latent-space diffusion/flow				
DiT-XL/2 (Peebles & Xie, 2023)	latent	675M	2.27	278.2
SiT-XL/2 (Ma et al., 2024)	latent	675M	2.06	277.5
SiT-XL/2 + REPA (Yu et al., 2025)	latent	675M	1.42	305.7
RAE + DiT^{DH}-XL/1 (Zheng et al., 2025)	latent	839M	1.13	262.6
Multi-step pixel-space diffusion/flow				
ADM-G (Dhariwal & Nichol, 2021)	pixel	554M	4.59	186.7
SiD, U-ViT (Hoogeboom et al., 2023)	pixel	2B	2.44	256.3
VDM++ (Kingma & Gao, 2023)	pixel	2B	2.12	267.7
SiD2, Flop Heavy (Hoogeboom et al., 2025)	pixel	–	1.38	–
JiT-G/16 (Li & He, 2025)	pixel	2B	1.82	292.6
PixelDiT-XL/16 (Yu et al., 2026a)	pixel	797M	1.61	292.7
Standard-B/16 (matched)	pixel	131M	4.08	264.9
AttnRes-JiT-B/16	pixel	131M	3.95	266.9
Standard-L/16 (matched)	pixel	459M	2.85	278.8
AttnRes-JiT-L/16	pixel	459M	2.84	283.0
AttnRes-JiT-H/16	pixel	953M	2.65	299.8

The REPA FID of 1.42 uses a guidance interval; the RAE FID of 1.13 uses autoguidance; and the PixelDiT FID of 1.61 is the 320-epoch configuration. The SiD row refers to Hoogeboom et al.'s U-ViT, distinct from Bao et al.'s similarly named architecture.

C Nearest-Neighbor Memorization Diagnostic

Section 4.2 reports the global retrieval protocol, Eq. (9), and the principal statistics. Table 5 supplements these with the distance distribution, including its tails: 3.1% of queries have cosine distance at most 0.05. Figure 7 shows examples sampled from the 16 closest matches and pairs near the mean distance. Retrieval uses all training classes; visual similarity and nonzero feature distance do not establish absence of memorization.

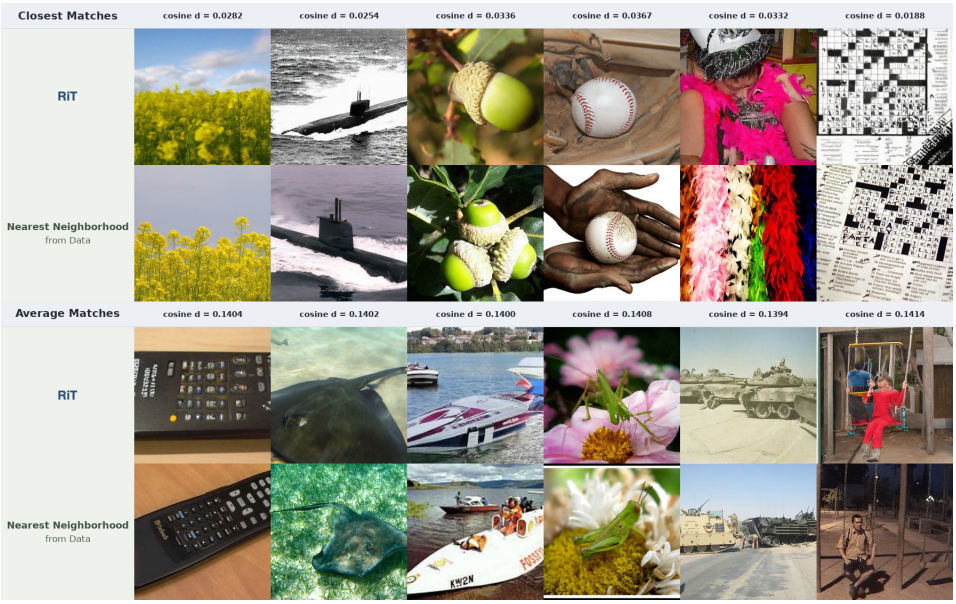

Figure 7: **Global ImageNet training-set nearest neighbors.** The upper panel samples from the 16 closest matches; the lower panel shows pairs near the mean distance. Within each panel, generations appear above their retrieved training images.

Table 5: **Nearest-neighbor summary for 1,000 Full-AttnRes JiT-L/16 generations.**

Metric	Value
Training candidates	1,281,167
Exact zero-distance matches	0 / 1,000
Minimum cosine distance	0.0144
5th percentile	0.0584
Median	0.1331
Mean $\pm$ std.	0.1404 ± 0.0572
95th percentile	0.2388
Maximum cosine distance	0.3233
Distance ≤ 0.05	3.1%
Same-class nearest neighbor	81.3%

D PyTorch Pseudo-code

We provide PyTorch-style pseudo-code for the core Full AttnRes mechanism to facilitate reproducibility.

```
import torch
import torch.nn as nn

def modulate(x, shift, scale):
    return x * (1 + scale[:, None]) + shift[:, None]


class FullAttentionResidual(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.query = nn.Parameter(torch.zeros(dim))

    def forward(self, history):
```

```
        # history contains one shape-aligned token group.
        states = torch.stack(history, dim=2)
        normalized = states * torch.rsqrt(
            states.square().mean(dim=-1, keepdim=True) + 1e-6
        )
        weights = (normalized @ self.query).softmax(dim=2)
        return (weights.unsqueeze(-1) * states).sum(dim=2)


class AttnResJiTBlock(nn.Module):
    def __init__(self, dim, cond_dim, attn, mlp):
        super().__init__()
        self.pre_attn_residual = FullAttentionResidual(dim)
        self.pre_mlp_residual = FullAttentionResidual(dim)
        self.norm1 = nn.LayerNorm(dim, elementwise_affine=False)
        self.norm2 = nn.LayerNorm(dim, elementwise_affine=False)
        self.attn, self.mlp = attn, mlp
        self.adaLN = nn.Sequential(
            nn.SiLU(), nn.Linear(cond_dim, 6 * dim)
        )
        nn.init.zeros_(self.adaLN[-1].weight)
        nn.init.zeros_(self.adaLN[-1].bias)

    def forward(self, history, condition):
        # Shift, scale, and gate for attention and MLP.
        sa, ca, ga, sm, cm, gm = self.adaLN(condition).chunk(6, dim
            =-1)

        x = self.pre_attn_residual(history)
        x = modulate(self.norm1(x), sa, ca)
        history.append(ga[:, None] * self.attn(x))

        x = self.pre_mlp_residual(history)
        x = modulate(self.norm2(x), sm, cm)
        history.append(gm[:, None] * self.mlp(x))
        return history
```

This is the block path used with adaptive layer normalization. Each block owns one residual-attention mixer per branch, while the condition produces shift, scale, and gate parameters for both attention and MLP. The Full AttnRes queries and final adaLN projection are zero-initialized. When token groups enter at different depths, the implementation keeps a separate shape-aligned history for each group and applies the same Full AttnRes operator independently before concatenating active groups for token mixing; this is implementation bookkeeping rather than a separate model component. Shape annotations and mixed-precision implementation details are omitted.

E Detailed Hyperparameters

Unless otherwise stated, the standard-residual baseline and AttnRes-JiT use the same data, optimizer, EMA, sampling, and evaluation settings; they differ only in the residual carrier. The architecture and training recipe follow our JiT-style pixel-space flow-matching setup.

Table 6: **Architecture settings.** Model-specific dimensions are shown separately; shared backbone choices span all three scales.

Setting	AttnRes-JiT-B/16	AttnRes-JiT-L/16	AttnRes-JiT-H/16
Depth	12	24	32
Parameters	131M	459M	953M
Hidden dimension	768	1024	1280
Attention heads	12	16	16
Image size		256×256	
Patch size		$\text{image_size}/16$	
Bottleneck	128	128	256
Dropout	0	0	0.2
In-context class tokens		32, if used	
In-context start block	4	8	10
Conditioning interface	Prefix-token conditioning; existing adaLN-style path retained		
Baseline residual carrier	Standard residual stream		
AttnRes-JiT residual carrier	Full AttnRes before attention and before MLP		
Full AttnRes query initialization	Zero		

Table 7: **Training, sampling, and evaluation settings.** These settings are shared across model scales except for the reported CFG value.

Setting	Value
Training	
Dataset	ImageNet, class-conditional
Representation	Pixel space
Objective	Flow matching with $\mathbf{z}_t = t\mathbf{x} + (1-t)\boldsymbol{\epsilon}$
Epochs	200 for ablations; 600 for main runs
Warmup epochs	5
Optimizer	Muon (Jordan et al., 2024)
Batch size	1024
Learning rate	2×10^{-4}
Learning-rate schedule	Constant
Weight decay	0
EMA decay	$\{0.9996, 0.9998, 0.9999\}$
Time sampler	$\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = -0.8$, $\sigma = 0.8$
Noise scale	$1.0 \times \text{image_size}/256$
Clip of $(1-t)$ in division	0.05
Class-token drop for CFG	0.1
Sampling and evaluation	
ODE solver	Heun
ODE steps	50
Time steps	Linear in $[0, 1]$
CFG scale sweep	$[1.0, 4.0]$
Reported CFG scale	B/16: 2.8; L/16: 2.4; H/16: 2.4
CFG interval	$[0.1, 1]$, if used
Evaluation samples	50K generated images
Metrics	FID, IS, and FDr^6
FDr^6 feature encoders	Inception-v3, ConvNeXt-v2, DINOv2, MAE, SigLIP2, and CLIP

F QUALITATIVE COMPARISONS

All comparisons use the same L/16 sampling configuration at CFG 2.4. Figure 8 uses vertically aligned groups: Full AttnRes above and the matched standard-residual model below. Figure 9 extends the adjacent-pair comparison in the main-text Figure 3, with standard residuals on the left and Full AttnRes on the right.



Figure 8: **Matched qualitative comparison on ImageNet** 256×256**.** The upper group (labeled RIT) shows Full-AttnRes samples; the lower group (labeled DiT) shows matched standard-residual JiT samples. Corresponding images are aligned vertically.

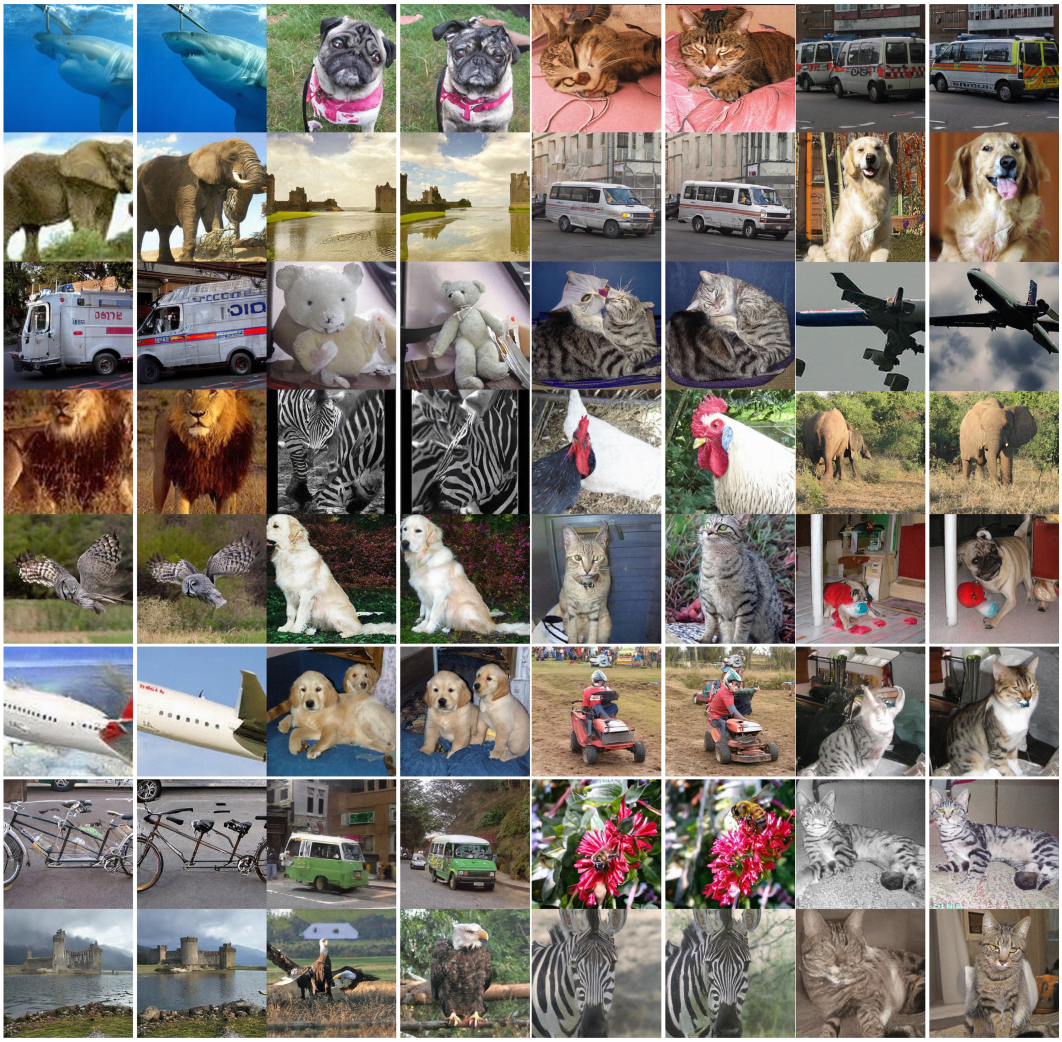

Figure 9: **Additional L/16 sampling results.** The grid covers a broader set of ImageNet classes under the same paired protocol.

G Extended Literature Positioning

What is being routed? Feature reuse, residual-stream mixing, token interaction, and expert selection operate on different objects. Table 8 summarizes the residual mechanisms most relevant to this study. It is a conceptual comparison, not an experimental ranking.

Table 8: **Mechanistic context for depth retrieval.** The final row states what this study evaluates; the remaining rows describe prior mechanisms.

Research line	Object of reuse and relation to this study
ResNet / DenseNet (He et al., 2016; Huang et al., 2017)	Additive shortcuts / concatenated features establish cross-layer reuse; they do not implement the Full AttnRes readout.
DLCL / DenseFormer (Wang et al., 2019; Pagliardini et al., 2024)	Learned depth coefficients combine earlier representations. Learned coefficients need not be token-dependent retrieval weights.
DeepCrossAttention (Heddes et al., 2025)	Input-dependent layer mixing constructs attention inputs from different depths; token attention and the depth-mixing interface differ from Full AttnRes.
HC / mHC (Zhu et al., 2025; Xie et al., 2026)	Multiple concurrent residual streams are mixed; mHC constrains residual mixing with doubly stochastic matrices. This differs from storing every prior increment.
Full / Block AttnRes (Kimi Team, 2026)	Softmax retrieves earlier increments or block summaries. The block variant addresses history costs; our experiments use the Full variant.
DAR (Xu et al., 2026)	Diffusion-specific depth aggregation studies query conditioning and chunked sources, including a static-query variant.
AttnRes-JiT (this study)	Applies Full AttnRes to matched pixel-space JiT models; studies prefix history, flow-time routing, and zero-query checkpoint conversion.

Generative modeling and representation interfaces. Denoising diffusion, DDIM, and flow matching establish the generative setting (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023). Latent diffusion, DiT/SiT, PixArt-α, and rectified-flow Transformers advance representation spaces or conditioned backbones (Rombach et al., 2022; Peebles & Xie, 2023; Ma et al., 2024; Chen et al., 2024; Esser et al., 2024). REPA supplies an external representation target, whereas RAE changes the autoencoder representation (Yu et al., 2025; Zheng et al., 2025). PixelREPA reports that directly transferring REPA to JiT can fail and adapts the alignment interface (Shin et al., 2026); PixelDiT2 retains frozen-encoder conditioning during inference (Yu et al., 2026b). Our evaluation encoders do not provide a training loss or denoiser conditioning. Combining history retrieval with these approaches requires a new controlled comparison.

Conditioning and computation allocation. AdaLN/FiLM modulate features, and mixture-of-experts methods select branch computation (Perez et al., 2018; Shazeer et al., 2017; Fei et al., 2024). Prefix registers and Register Guidance instead affect auxiliary-token computation or sampling (Darcet et al., 2024; Starodubcev et al., 2026). These distinctions prevent equating any use of the word "routing" with retrieval over residual history.

Scope of checkpoint conversion. Net2Net studies function-preserving transformations (Chen et al., 2016), and Skip-DiT also discusses conversion through continued training (Chen et al., 2025). Thus post-training architectural conversion is not itself new. Here the specific question is whether zero-query means, under the normalization assumptions in Appendix A.2, allow Full AttnRes conversion to improve on unchanged continuation of the same checkpoint. We do not establish superiority over other conversion methods.

Evaluation provenance and limits. The six encoders are Inception-v3 (Szegedy et al., 2016), ConvNeXt V2 (Woo et al., 2023), DINOv2 (Oquab et al., 2023), MAE (He et al., 2022), SigLIP 2 (Tschannen et al., 2025), and CLIP (Radford et al., 2021). FDr^6 follows the normalized-ratio construction of Yang et al. (2026); we use the evaluation metric, not their FD-loss training procedure. Encoder-dependent rankings (Stein et al., 2023), finite-sample bias (Chong & Forsyth, 2020), and resizing/compression effects (Parmar et al., 2022) remain relevant. These citations identify the encoder families and methodological context; they do not substitute for the exact checkpoints, feature layers, and preprocessing needed for reproduction.