STAIRDROP: HIERARCHICAL TOKEN DROPPING FOR DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent diffusion transformers have demonstrated that substantial training compute can be saved by allowing only a subset of tokens to traverse intermediate blocks while preserving competitive sample quality. Yet existing token-sparse designs usually treat the resulting sparsity pattern as a fixed computational shortcut, leaving open how information should be organized across depth. Inspired by U-Net's progressive contraction and expansion, we introduce StairDrop, a hierarchical token-dropping schedule that progressively reduces and restores participation, with skip fusion at each restoration and depth allocated across the resulting stages. The hierarchy is used during training; at inference, all tokens traverse the full backbone with the fusion layers retained. Controlled experiments on SiT-B/2 and SiT-XL/2 show that intermediate participation improves generation and that depth allocation matters. On ImageNet 256×256 with guided sampling, at half of the training compute of the strongest token-dropping system, StairDrop matches its classifier-free-guidance FID-50K of 1.96, and at equal compute it reaches 1.47 against its best reported 1.55. These results support hierarchical token participation as an effective design for efficient diffusion transformer training.

1 INTRODUCTION

Diffusion transformers produce high-quality images at substantial training cost (Peebles & Xie, 2023; Ma et al., 2024). Token dropping reduces this cost by allowing positions to skip intermediate blocks (Krause et al., 2025; Park et al., 2026). Masking, routing and placeholder methods (Zheng et al., 2024; Gao et al., 2023; Sehwag et al., 2025; Krause et al., 2025; Park et al., 2026) share one shape: a span of blocks runs on a fraction of the positions between dense ends, and the network is deployed dense. The strongest of them runs 24 of the 28 blocks of SiT-XL/2 on a quarter of the positions, at about a third of the cost of a dense step, and is better than the dense model at equal steps. The participation schedule determines both how much computation a block performs and which features it processes. We study how to organise this participation hierarchically. Figure 1 previews the resulting trade-off: at the training compute of the strongest token-dropping system, StairDrop reaches FID-50K 1.47 on ImageNet 256×256 against its 1.55, and at half of that compute it matches its classifier-free-guidance result of 1.96.

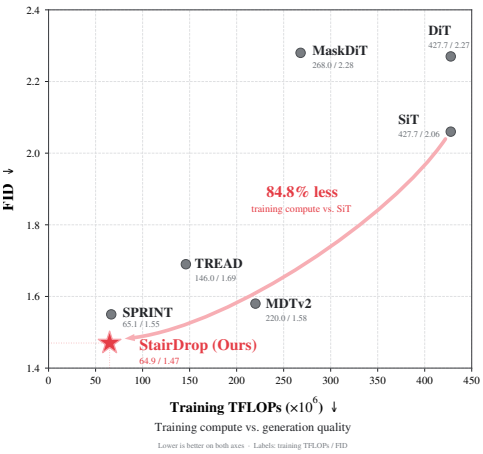

Figure 1: **Training compute–quality trade-off.** Selected systems from Table 3 at their reported guided sampling settings; labels give training TFLOPs ($\times10^6$) and FID-50K. Arrow: comparison with the published SiT-XL/2 system.

A common schedule reduces participation once, processes the retained tokens through a sparse span, and restores full participation afterwards. This saves compute, but leaves a structural question: *should the transition between full participation and*

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.

the sparse core happen in one step? An intermediate participation level would let features undergo further processing before the next reduction and after the first restoration. It also creates a depth-allocation problem: for a fixed backbone, how many blocks should operate at each level? These choices trade the cost of each training step against the quality learned over training. The question is distinct from the one asked by U-shaped transformers for diffusion (Tian et al., 2024; Crowson et al., 2024), which pay for their hierarchy at every sampling step; here it exists only while the network is trained.

We draw inspiration from the progressive contraction and expansion of U-Net (Ronneberger et al., 2015). In a one-level token-dropping model, selection and restoration surround a reduced-participation span with a skip connection: withholding positions plays the part of downsampling, restoring them the part of upsampling, the fusion with the saved entry state the part of the skip, and the sparse span sits where the bottleneck would be. Spatial-interaction probes support this reading of the span (Section 3.2): it depends on distant interactions more than the corresponding dense blocks do, and its output is read locally after restoration. What a U-Net has and the one-level schedule lacks is the intermediate scales, with processing at each of them on the way down and on the way up. This motivates a hierarchy of participation levels, with processing between successive reductions and restorations. The correspondence with U-Net is in this staged organisation; token coordinates remain fixed, and the hierarchy is used during training. Section 4 tests whether this design improves generation.

StairDrop instantiates this idea with nested token selection and restoration. Participation decreases from full to half to quarter, then returns through half to full, with a skip fusion at each restoration. The existing Transformer blocks are divided among these five stages. The intermediate stages provide processing at a density between the full sequence and the sparse core, without adding Transformer blocks. At inference, every position traverses every block, with the two fusion layers retained. The design therefore changes how the backbone is trained while preserving full token participation during generation.

We evaluate the hierarchy from controlled design choices to the overall training-compute trade-off. On SiT-B/2, introducing intermediate participation lowers FID-10K from 34.9 to 29.5 at matched training steps against a one-level control with the same depth, fusion positions and parameters. Allocation studies vary one stage at a time and yield three observations that carry to the deeper backbone, where the staircase is 1.2 FID below one-level dropping at 400K steps. Table 3 compares complete systems: the 1.47-FID model uses 64.9×10^6 training TFLOPs. Combining StairDrop with representation alignment (Yu et al., 2025) further yields FID-50K 1.58 at 19.2×10^6 training TFLOPs, the lowest training compute in the table.

Our contributions are, first, a hierarchical training schedule based on nested token participation, which reads the one-level schedule as a single-scale U-Net and completes it with an intermediate scale on each side; second, controlled studies of the intermediate level and of depth allocation on SiT-B/2 and their transfer to SiT-XL/2; and third, an efficient training system that reaches the lowest FID and the lowest training compute among published token-dropping systems, and whose gains combine with representation alignment.

2 Related Work

Efficient training through token participation. Masked autoencoders (He et al., 2022) reduce the cost of visual representation learning by applying the encoder only to visible patches. In diffusion models, MaskDiT (Zheng et al., 2024) combines masked training with an asymmetric encoder–decoder and an auxiliary reconstruction objective. MDT (Gao et al., 2023) uses masked latent modelling and a side interpolater to recover missing positions, while MicroDiT (Sehwag et al., 2025) mixes patch features before masking. Routing offers another way to retain information from tokens that skip computation: TREAD (Krause et al., 2025) carries their features across selected layers, and SPRINT (Park et al., 2026) fuses sparse outputs with dense features through a residual path. These methods establish token participation as a useful control over training cost. StairDrop studies its organisation along depth: nested participation levels define a hierarchy, and the backbone's blocks are allocated among its stages. We use the placeholder restoration and linear fusion of SPRINT within each level, and also evaluate the hierarchy with identity restoration.

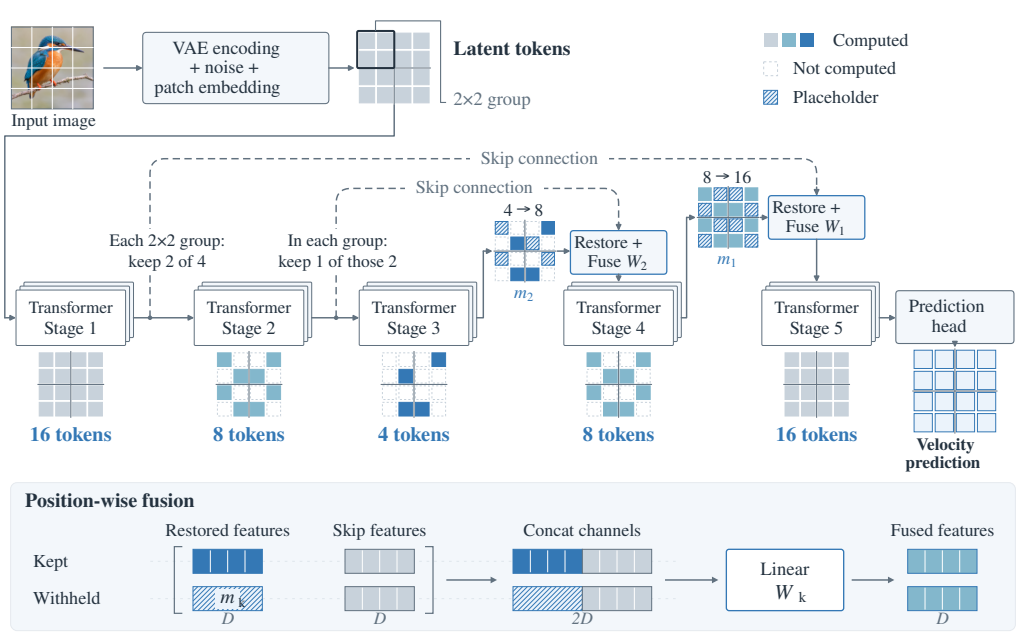

Figure 2: **StairDrop during training.** A 16-position illustration of the 256-token model. Dashed skips carry features from before each selection. **Bottom:** fusion concatenates each position's computed feature or fixed placeholder with its corresponding skip feature along channels, then applies a linear layer W_k, shared across positions within the same level.

Hierarchical and U-shaped architectures. U-Net (Ronneberger et al., 2015) combines progressive spatial contraction and expansion with skip connections between corresponding levels. Diffusion transformers adopt different aspects of this design. U-ViT (Bao et al., 2023) connects shallow and deep blocks through long skips without spatial downsampling. HDiT (Crowson et al., 2024) builds a spatial hierarchy by merging neighbouring tokens and splitting them during expansion; U-DiT (Tian et al., 2024) combines a U-shaped backbone with downsampled self-attention. UDT (Yun et al., 2026) instead uses feature similarity for progressive token merging and unmerging. Outside diffusion, fractional max-pooling (Graham, 2014) explores gradual spatial reduction, and DynamicViT (Rao et al., 2021) progressively prunes tokens for efficient classification. Our design draws on progressive contraction and expansion, applying them to participation density during training. Tokens retain their original coordinates and feature width as nested subsets are selected and restored. At inference, all positions pass through every Transformer block, with the fusion layers retained.

Complementary training and sampling methods. Representation alignment provides a separate route to more efficient diffusion training. REPA (Yu et al., 2025) aligns projected hidden states with clean-image features from pretrained visual encoders, supplying an auxiliary learning signal. StairDrop changes which tokens participate in each stage and can be combined with this objective, as evaluated in Section 4.4. Sampling also affects the quality achievable by a trained model: classifier-free guidance (Ho & Salimans, 2022) and guidance intervals (Kynkäänniemi et al., 2024) modify the denoising trajectory, while Sparse Guidance (Krause et al., 2026) uses token sparsity to construct a guidance signal. Our experiments assess the participation schedule with unguided sampling and report guided system-level results separately.

3 STAIRDROP

3.1 PRELIMINARIES

Flow matching. We work with SiT (Ma et al., 2024), trained as a flow-matching model on the linear interpolant between a latent x_0 and Gaussian noise ε,

$$x_t = (1-t)\,x_0 + t\,\varepsilon, \qquad t \in [0,1], \tag{1}$$

with $t = 1$ at the noise end. The network $v_\theta(x_t, t, c)$, a stack of L identical transformer blocks conditioned on the class label c, predicts the velocity of the interpolant,

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,\,\varepsilon,\,t}\,\big\|\, v_\theta(x_t,t,c) - (\varepsilon - x_0)\,\big\|^2, \tag{2}$$

and samples are drawn by integrating the learned velocity from $t = 1$ to 0 with an ODE or an SDE sampler. Classifier-free guidance (Ho & Salimans, 2022) sharpens the conditional velocity with the unconditional one,

$$\tilde{v}_\theta(x_t,t,c) = v_\theta(x_t,t,\varnothing) + w\,\big(v_\theta(x_t,t,c) - v_\theta(x_t,t,\varnothing)\big), \qquad w > 1. \tag{3}$$

One-level token dropping. A one-level schedule runs one span of Transformer blocks on a subset S of the N image-token positions, with full participation before and after the span. The subset is redrawn for every sample and training step. Writing D for the hidden width, $h_{\text{in}} \in \mathbb{R}^{N\times D}$ for the span's entry state and h_S for its output on the retained positions, restoration takes the form

$$h \;\leftarrow\; F\big(\,\mathrm{restore}(h_S;\,p)\,,\; h_{\text{in}}\big), \tag{4}$$

where $\mathrm{restore}$ places h_S at its original positions and fills the withheld positions with p; F combines this sequence with the saved entry state. TREAD (Krause et al., 2025) samples half of the positions uniformly, restores withheld features from h_{in}, and uses no fusion. SPRINT (Park et al., 2026) keeps one position per 2×2 grid cell, restores the others with a fixed placeholder $\mathbf{m} \in \mathbb{R}^D$, and fuses by channel concatenation followed by a shared linear projection, $F(u, h_{\text{in}}) = W[u; h_{\text{in}}] + b$, with $W \in \mathbb{R}^{D\times 2D}$. We use cell-based selection and this placeholder-and-fusion operation for each span of StairDrop.

For both one-level training and StairDrop, the loss (2) includes every position. At inference all positions pass through every block; placeholder filling is inactive, while the fusion layers operate on the full sequence.

3.2 THE STAIRCASE

StairDrop replaces a single reduction in token participation with a hierarchy. To motivate this design, we first examine spatial interactions in a one-level model, using 100K-step SiT-B/2 checkpoints at noise level $t = 0.5$. Within the sparse span, we restrict attention to active positions within radius d and measure the fraction of the unrestricted attention benefit retained over self-only attention. After restoration, a second probe restricts direct attention to retained outputs, leaving attention to restored positions unrestricted, and measures the output's loss benefit relative to its removal.

At a radius of two patches, the one-level span retains 44% of its unrestricted attention benefit, compared with 87% in the corresponding dense blocks (Figure 3a). The sparse span is therefore more sensitive to removing distant interactions under this intervention. After restoration, a one-patch reading radius preserves 96% of the retained output's measured loss benefit (Figure 3b). Together, these observations suggest a bottleneck-like role: the reduced set of positions relies on broad spatial interactions, while its outputs remain useful under nearby direct access after restoration.

Together with the selection, restoration and skip around the sparse span, this interpretation suggests a U-Net-like organisation of token participation (Ronneberger et al., 2015). U-Net reaches its bottleneck through progressive contraction and expands through successive levels.

StairDrop adds one intermediate level of participation on each side of the core, visited on the way down and on the way up, and keeps the core density unchanged. Participation decreases in nested steps, with processing between reductions, and is restored in reverse order through corresponding skips. The hierarchy is in participation density; the original token coordinates remain unchanged.

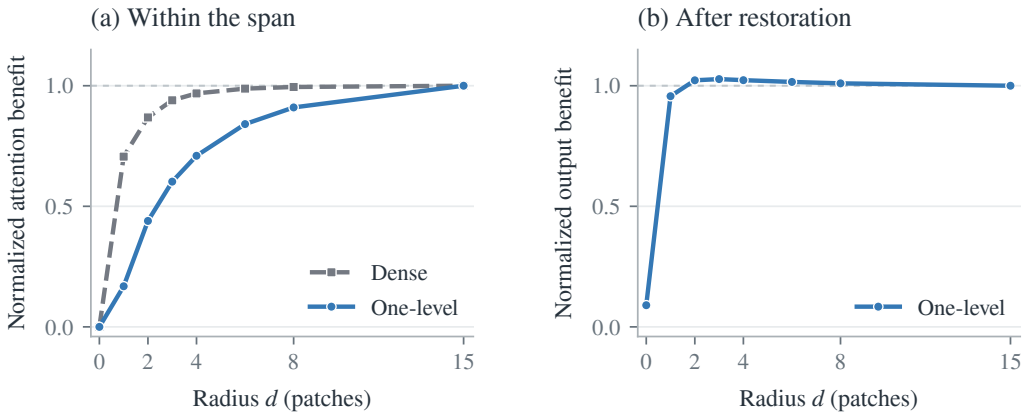

Figure 3: **Spatial interactions in a one-level model.** (a) Restricting attention within the span, with dense blocks as a reference. (b) Restricting direct access to its retained outputs after restoration. Radius uses original-grid patch units; $d = 15$ is unrestricted. Means $\pm$ one SD over three probe sampling seeds; SiT-B/2 at 100K steps, training graph, $t = 0.5$. Normalisations: Appendix E.

Nested selection. The schedule is built from the operations of Section 3.1 applied twice (Figure 2). After d_0 dense blocks, the outer selection keeps two uniformly drawn positions of every 2×2 cell of the grid, $N/2$ tokens, and the next d_1 blocks run on them. The inner selection then keeps one of the two survivors of every cell, $N/4$ tokens, and the d_c core blocks run on those. The inner set is a subset of the outer one, so the inner span is nested in the outer span, and a position withheld at the first cut stays withheld until the outer restore. A retained token keeps its position embedding, so attention within a span is ordinary attention among fewer positions of the same grid.

Nested restoration. Each level records the state that entered it and owns a placeholder $\mathbf{m}_k$ and a fusion (W_k, b_k); the levels are restored from the inside out. When the core ends, the $N/4$ positions withheld at the second cut are filled with $\mathbf{m}_2$, and the resulting $N/2$-token sequence is fused, position by position, with the state that entered the inner span, Eq. (4) with $c = \mathbf{m}_2$ and $F = W_2[\cdot\,;\cdot] + b_2$; the next d_2 blocks run on these $N/2$ tokens. When they end, the positions withheld at the first cut are filled with $\mathbf{m}_1$, the full sequence is fused with the dense state that entered the outer span through (W_1, b_1), and the last d_3 blocks run on all N tokens. A retained position thus leaves each level with its own deep computation, a withheld position with the placeholder, and both with their state from before the level through the skip. The loss is taken on every position, withheld or not.

Cost and deployment. Over one-level dropping with one fusion, the staircase adds a placeholder and a fusion, $2D^2 + D$ parameters (2.7M on SiT-XL/2), and no Transformer blocks; the fusions start from a Xavier initialisation with zero bias and the placeholders are fixed. At inference nothing is withheld: every position passes through every block, the placeholders are never used, and the two fusions read the full sequence, so the deployed model is the dense backbone with two linear layers.

Depth allocation and cost. With L blocks the staircase is described by five depths, d_0 dense blocks before the first drop, d_1 blocks at $N/2$ tokens, d_c blocks at $N/4$, d_2 blocks at $N/2$ after the inner restore and d_3 dense blocks after the outer restore, with $d_0 + d_1 + d_c + d_2 + d_3 = L$ and at least one block per stage. The cost of a training step relative to a dense step is the token-weighted block count,

$$\frac{(d_0 + d_3) + \frac{1}{2}(d_1 + d_2) + \frac{1}{4}\,d_c}{L}. \tag{5}$$

We use $2/2/4/2/2$ on SiT-B/2 and $4/6/8/6/4$ on SiT-XL/2, with relative step costs of 0.58 and 0.57 by Eq. (5). Measured on two H200s, the SiT-B/2 staircase trains at 4.5K samples per second against 4.4K for one level at a quarter of the tokens and 2.6K dense, the shorter attention sequences of the one-level span buying less than their FLOP count suggests.

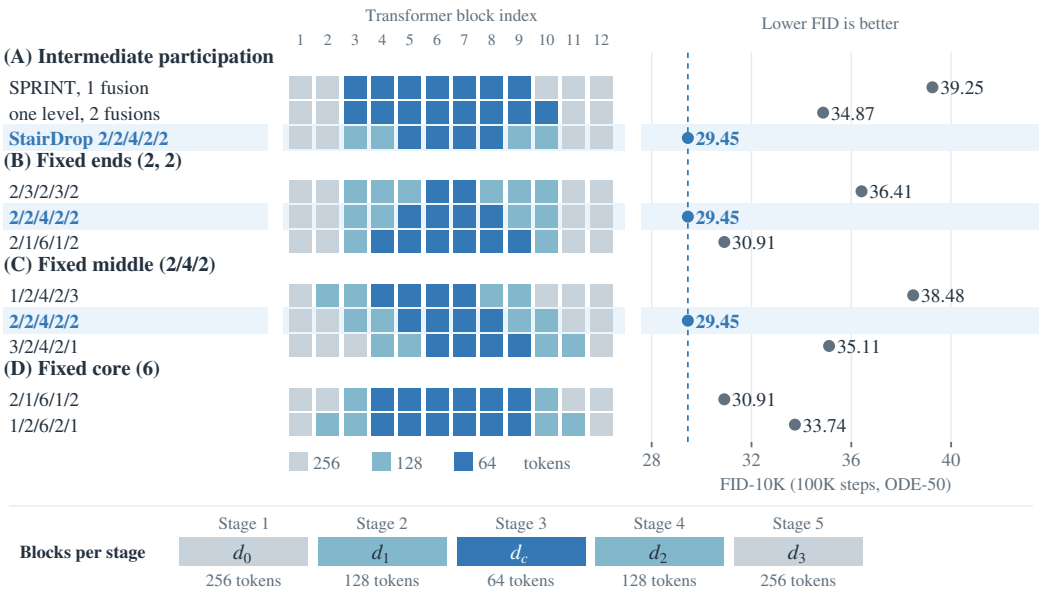

Figure 4: **Design study on SiT-B/2.** FID-10K at 100K steps (ODE-50, no guidance). Each square is a block, coloured by the tokens it sees in training; a layout lists the five stage depths, so $2/2/4/2/2$ assigns 2, 2, 4, 2, 2 blocks to the 256/128/64/128/256-token stages. (A) Intermediate participation against a one-level control with the staircase's span, fusion positions and parameters. (B)–(D) Depth allocation, one thing varied at a time: dense ends fixed (B), middle fixed (C), core fixed (D). The highlighted row is $2/2/4/2/2$ in every panel.

4 Experiments

Setup. All experiments are class-conditional ImageNet 256×256 (Deng et al., 2009) in the latent space of the Stable Diffusion VAE (Rombach et al., 2022), scored with the ADM evaluator (Dhariwal & Nichol, 2021) against its ImageNet reference batch: FID (Heusel et al., 2017), Inception score (Salimans et al., 2016), and precision and recall (Kynkäänniemi et al., 2019). Two backbones and two protocols are used. The design study and the scaling study (Sections 4.1 and 4.2) use the original SiT backbone and recipe (Ma et al., 2024), SiT-B/2 for ablations and probes and SiT-XL/2 for scale, every arm trained from scratch on the same latents in the same order and read out on the sparse weights without guidance, with the 50-step Euler ODE sampler of SPRINT's protocol (ODE-50) (Park et al., 2026). The system-level comparison (Section 4.3) adopts SPRINT's published setting in full, its improved backbone, dense stage and guided sampling protocol, described there. TREAD and SPRINT are our re-implementations of their released code inside one code base, so that all methods share the backbone, the recipe, the data order and the evaluator. Recipes, fine-tuning, sampling and compute accounting are given in Appendices A and B.

4.1 Design study on SiT-B/2

The design study runs on SiT-B/2, twelve blocks, at 100K steps under the ODE-50 protocol (Figure 4; every arm with its other metrics is in Appendix F). It asks two questions: whether an intermediate level of participation improves generation when the parameters and the depth are held fixed, and, once it does, where a fixed budget of blocks should go.

Introducing an intermediate level. One-level dropping at SPRINT's published layout is the reference (Figure 4A): two dense blocks, seven blocks on one token per cell, three dense blocks, one fusion; it reaches 39.3. The control is a one-level layout with the staircase's span, fusion positions and parameters: it keeps 64 tokens through blocks 3–10 and adds a second placeholder and fusion at the depths where the staircase restores, and it reaches 34.9. The staircase $2/2/4/2/2$ differs from this control in four blocks only, blocks 3–4 and 9–10, which see 128 tokens instead of 64, and it

Table 1: **SiT-B/2 along training.** $2/2/4/2/2$ against the baselines; ODE-50, no guidance.

Model	100K FID-10K	400K FID-50K
Dense	70.72	38.58
TREAD	54.71	20.58
SPRINT	39.25	11.48
StairDrop	**29.45**	**7.66**

Table 2: **Scaling to SiT-XL/2.** Original backbone, sparse weights, ODE-50; the one-level row is SPRINT's published layout. The last column is FID-10K with classifier-free guidance on $t \leq 0.7$ at the scale tuned per layout, which selects the XL layout (Section 4.3).

	FID-10K, unguided			at 400K	
Layout	200K	300K	400K	FID-50K	guided FID-10K
one-level $2/-/24/-/2$	11.52	9.65	8.75	6.04	–
StairDrop $2/2/20/2/2$	9.74	8.47	8.08	5.28	–
StairDrop $3/3/16/3/3$	11.70	8.46	**7.52**	**4.76**	6.89
StairDrop $4/6/8/6/4$	11.20	8.76	7.68	4.89	**6.77**

reaches 29.5. The gain over the control cannot come from the added fusion parameters, since the control has them; it comes from the intermediate level of participation.

Allocating depth across stages. With the intermediate level in place, the twelve blocks can be divided among the five stages in many ways; one-level layouts have a single free variable, the width of the dense ends, and the staircase has four. We vary one thing at a time around $2/2/4/2/2$ (Figure 4B–D). With the dense ends fixed, moving depth from the intermediate stages into the core helps until the intermediate stages are down to a single block, and then hurts (B). With the middle fixed, the balanced split of the dense blocks is best by a wide margin, and the encoder end is the more sensitive one (C). With the core fixed, the remaining depth is better spent at the dense ends than at the intermediate stages (D).

Among the tested layouts $2/2/4/2/2$ is the best, and it is the default from here on. What we carry to the deeper backbone is not the layout but three observations: (1) a core at least as deep as either intermediate stage, (2) intermediate stages and dense ends of at least two blocks each, and (3) ends of equal depth. The candidates of Section 4.2 are built to satisfy them, and whether they hold at that scale and under longer training is tested there.

Longer training. Table 1 follows the four arms to 400K steps on 50K samples: the ordering holds, and the staircase ends 3.8 FID below one-level dropping.

4.2 Scaling to SiT-XL/2

Table 2 repeats the controlled comparison on SiT-XL/2, a backbone twice as deep, with three staircase layouts built from the SiT-B/2 observations. The one-level baseline is trained at SPRINT's published layout under the same recipe and protocol, since its own tables report XL numbers only after a dense fine-tune on the improved backbone. The design pays at scale: at 400K steps the best staircase is 1.2 FID below the one-level layout on 10K samples at the same recall, and 0.4 to 0.6 below it at matched training FLOPs, since a staircase step costs more than a one-level step. The gain is the intermediate level: $2/2/20/2/2$ is the one-level layout with the staircase inserted, and it is ahead of it at every step.

The thin-ended $2/2/20/2/2$ leads early and the thick-ended layouts pass it by 300K to 400K steps. The SiT-B/2 comparisons admit all three candidates and do not rank them; the larger model and the longer training do. The two thick-ended layouts end 0.13 FID apart. Since the system-level comparison is made under guidance, we choose between them under guidance, on the sparse weights before the dense stage (last column of Table 2): $4/6/8/6/4$ reverses the unguided order, consistent with its higher recall, which guidance converts into quality. We take $4/6/8/6/4$ as the XL layout, continue it on the sparse schedule to 800K steps (Appendix F), and every system-level row below is this layout.

4.3 System-level comparison

Protocol. To compare with published numbers we adopt SPRINT's setting in full. The backbone is the improved SiT-XL/2 of its paper, with query-key normalisation (Dehghani et al., 2023; Esser et al., 2024), two-dimensional rotary position embeddings (Lu et al., 2024) and log-normal

Table 3: **System-level comparison on ImageNet 256×256, classifier-free guidance, 50K samples.** Published rows carry the numbers of their papers at their own sampling protocols; SPRINT's rows are quoted as published (Appendix C); our rows use the SDE sampler with 250 steps. Epochs are images seen divided by 1.28M, every training stage counted at its batch size: SPRINT's rows include the 100K-step dense stage at batch 1024 that its own table omits, MaskDiT's count the masked stage only. Training compute is SPRINT's measurement, forward FLOPs of the training graph at batch 256 times the iterations, scaled to each paper's training length (Appendix B). The StairDrop rows are $4/6/8/6/4$ after 800K, 1.72M and (with REPA) 400K sparse steps, each with the same 20K-step dense stage at batch 1024. †: improved SiT backbone from SPRINT; ‡: interval or scheduled guidance; PDG: path-drop guidance; –: not available.

Method	Epochs	Params	Training TFLOPs ($\times10^6$)	FID-50K	IS	Prec.	Rec.
ADM-G,U (Dhariwal & Nichol, 2021)	–	673M	–	3.94	215.8	0.83	0.53
LDM-4 (Rombach et al., 2022)	167	400M	–	3.60	247.7	0.87	0.48
U-ViT-H/2 (Bao et al., 2023)	400	501M	–	2.29	263.9	0.82	0.57
DiT-XL/2 (Peebles & Xie, 2023)	1400	675M	427.7	2.27	278.2	0.83	0.57
FiTv2-XL/2 (Wang et al., 2024)	400	671M	–	2.26	261.0	0.81	0.59
SiT-XL/2 (Ma et al., 2024)	1400	675M	427.7	2.06	277.5	0.83	0.59
MDTv2-XL/2‡ (Gao et al., 2023)	920	742M	220.0	1.58	314.7	0.79	0.65
MaskDiT (Zheng et al., 2024)	1600	730M	268.0	2.28	276.6	0.80	0.61
TREAD‡ (Krause et al., 2025)	740	675M	146.0	1.69	292.7	0.81	0.63
SiT-XL/2†	400	675M	122.2	2.04	–	0.82	0.60
+ SPRINT, CFG (Park et al., 2026)	460	677M	65.1	1.96	–	0.80	0.61
+ SPRINT, PDG‡	460	677M	65.1	1.55	–	0.80	0.64
SiT-XL/2† + StairDrop	176	680M	32.8	**1.96**	248.9	0.81	0.61
SiT-XL/2† + StairDrop‡	360	680M	64.9	**1.47**	307.2	0.79	0.65
SiT-XL/2† + StairDrop + REPA‡	96	680M	19.2	**1.58**	292.1	0.80	0.64

timestep sampling (Esser et al., 2024); the staircase sits on top of them: at 400K sparse steps, unguided, $4/6/8/6/4$ reads 6.76 on this backbone against 7.68 on the original one. Training is the sparse schedule followed by the dense stage that MaskDiT, MicroDiT, TREAD and SPRINT all use (Zheng et al., 2024; Sehwag et al., 2025; Krause et al., 2025; Park et al., 2026), 20K steps at batch 1024 with every position kept (SPRINT: 100K steps). Our guided results use SiT's SDE sampler (Ma et al., 2024) at 250 steps and are scored on 50K samples. The 800K row uses classifier-free guidance (Ho & Salimans, 2022) throughout the trajectory; rows marked ‡ restrict guidance to the lower 70% of the noise range (Kynkäänniemi et al., 2024). Compute is counted as SPRINT counts it, forward FLOPs of the training graph at batch 256 times the number of iterations (Appendix B).

Results. Table 3 compares selected diffusion backbones and token-sparse training systems, each at its reported configuration. After 800K sparse steps and the dense stage, StairDrop matches SPRINT's CFG result of 1.96 with 32.8×10^6 training TFLOPs against 65.1×10^6, approximately half the compute. Extending sparse training to 1.72M steps and keeping the same dense stage yields FID 1.47 at 64.9×10^6 TFLOPs, improving on SPRINT's best reported result of 1.55 with PDG and scheduled guidance at a comparable training budget. With representation alignment added (Section 4.4), StairDrop reaches 1.58 at 96 epochs. Among the listed systems, the staircase achieves the lowest FID, and its REPA variant uses the lowest reported training compute.

Figure 5 shows samples from the evaluation set of the 1.47 model; more are in Appendix G.

4.4 Compatibility with representation alignment

The staircase changes only which positions a block sees during training, so it can be combined with methods that change the training objective. We test this with REPA (Yu et al., 2025), the training-side lever with the largest published gain on SiT-XL/2: it aligns the state after block 8 with DINOv2



Figure 5: **Selected ImageNet samples from StairDrop.** One image from each of 18 categories, generated by the SiT-XL/2 model achieving FID-50K 1.47 in Table 3 (SDE-250, interval CFG $w = 1.6$). All images are generated at 256×256.

features (Oquab et al., 2024), and one-level dropping is reported to compose with it (Park et al., 2026). We add REPA at the same block, which on $4/6/8/6/4$ is the fourth block of the intermediate level, and align the 128 tokens present there at their own positions, with coefficient 0.5. At 100K steps the unguided FID-10K of the staircase falls from 13.11 to 8.36 (IS 107 to 159, precision 0.767 to 0.802, recall 0.599 to 0.618) at the same time per step. At 400K the gain narrows, as it does on dense SiT: 6.76 to 5.98 (IS 190 to 243, precision 0.793 to 0.821, recall 0.666 to 0.640). With guidance, after the same dense stage with the alignment kept, it reaches FID-50K 1.58 at 96 epochs and 19.2×10^6 TFLOPs (last row of Table 3); at 96 epochs the aligned staircase sits between the plain rows at 176 and 360 epochs, so the two levers add.

5 Conclusion

Inspired by U-Net's progressive contraction and expansion, StairDrop organises training-time token participation into nested levels, restores each level through a skip fusion, and allocates the backbone's blocks across the resulting stages. The whole addition is one placeholder and one linear fusion, and the deployed network is the dense backbone. The controlled comparisons on SiT-B/2 show that the intermediate level, not the added parameters, improves generation, and that how the depth is divided among the stages matters as much as the levels themselves: a core at least as deep as the intermediate stages, no stage thinner than two blocks, and ends of equal depth. These observations carried to SiT-XL/2, where the layout chosen under guidance halves the training compute of the best token-dropping method at equal FID and lowers its FID from 1.96 to 1.47 at equal compute, and where representation alignment adds to the staircase rather than replacing it, reaching 1.58 at less than a third of that compute. Two directions follow directly: a third level for larger token grids, where a 2×2 cell is a small fraction of the image, and the allocation of depth in backbones whose blocks are not identical. More broadly, the results argue for treating the participation schedule as part of how a backbone is trained rather than as a cost-saving device: which positions a block sees shapes what it learns to do.

AI USE STATEMENT

Generative AI tools assisted with reviewing the literature, developing and cleaning up code, carrying out experiments, improving the presentation of the work, providing manuscript feedback and editing the text. They were not used to generate data, to formulate or prove mathematical claims, to translate, or to clean datasets. The authors independently reviewed and verified all AI-assisted results, made the final scientific and methodological decisions, and take full responsibility for the contents of this paper.

REPRODUCIBILITY STATEMENT

The code base behind every experiment, with the configuration of every arm, the scripts that build the latent and feature caches, run the probes, sample and score, is submitted as supplementary material. Training recipes, the dense stage and the sampling protocols are in Appendix A, the compute and epoch accounting in Appendix B, every design-study arm with all of its metrics in Appendix F, the evaluation of SPRINT's released checkpoint in Appendix C and the probe definitions in Appendix E. All metrics are computed with the public ADM evaluator against its reference batch.

REFERENCES

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

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

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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*, 2017.

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

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. TREAD: Token routing for efficient architecture-agnostic diffusion training. In *IEEE/CVF International Conference on Computer Vision*, 2025.

Felix Krause, Stefan Andreas Baumann, Johannes Schusterbauer, Olga Grebenkova, Ming Gui, Vincent Tao Hu, and Björn Ommer. Guiding token-sparse diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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*, 2019.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, 2024.

Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. FiT: Flexible vision transformer for diffusion model. In *International Conference on Machine Learning*, 2024.

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 *European Conference on Computer Vision*, 2024.

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

Dogyun Park, Moayed Haji-Ali, Yanyu Li, Willi Menapace, Sergey Tulyakov, Hyunwoo J. Kim, Aliaksandr Siarohin, and Anil Kag. SPRINT: Sparse-dense residual fusion for efficient diffusion transformers. In *International Conference on Learning Representations*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 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*, 2016.

Vikash Sehwag, Xianghao Kong, Jingtao Li, Michael Spranger, and Lingjuan Lyu. Stretching each dollar: Diffusion training from scratch on a micro-budget. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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*, 2024.

Zidong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, and Lei Bai. FiTv2: Scalable and improved flexible vision transformer for diffusion model. *arXiv preprint arXiv:2410.13925*, 2024.

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*, 2025.

Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. UDT: Reconciling U-Nets and diffusion transformers with data-adaptive token reduction. *arXiv preprint arXiv:2608.01298*, 2026.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

A TRAINING AND EVALUATION PROTOCOL

Backbones. SiT-B/2 (12 blocks, width 768, 12 heads) and SiT-XL/2 (28 blocks, width 1152, 16 heads) with patch size 2 on the $32\times32\times4$ latents of `sd-vae-ft-ema`, i.e. 256 tokens. The original backbone uses absolute sinusoidal position embeddings and uniform timestep sampling. The improved backbone used for Table 3 adds RMS normalisation of the per-head queries and keys (Dehghani et al., 2023; Esser et al., 2024), two-dimensional rotary position embeddings (Lu et al., 2024) that follow the kept token subset at every level, and log-normal timestep sampling (Esser et al., 2024), which is the configuration SPRINT reports all of its numbers on.

Pre-training. Global batch 256, AdamW with learning rate 10^{-4} and no weight decay, bf16 autocast, EMA decay 0.9999, linear interpolant with velocity target, no gradient clipping. The token masks are drawn per sample and per step from a dedicated generator that is checkpointed with the model, so that a resumed run is bitwise identical to an uninterrupted one. Withheld positions carry the same loss weight as kept ones. Every SiT-B/2 arm is trained from scratch on the same latents with the same seed, so that differences of 0.1 FID between arms are real: two independent runs of one configuration agree to 10^{-7}. Training compute follows SPRINT's convention, forward FLOPs of the training graph at batch 256 times the number of iterations, measured with a FLOP counter (Appendix B). TREAD is run at its published setting for this depth (route from block 3 to block 9, 50% of the tokens through an identity path); SPRINT at its published setting (blocks 3 to 9 on one token per 2×2 cell, placeholder restore, one fusion). On SiT-XL/2 the staircase $4/6/8/6/4$ runs blocks 5 to 24 on 128 tokens and blocks 11 to 18 on 64.

Dense stage. Before guided sampling the sparse weights are trained for 20K steps with every position kept, at global batch 1024 with a 2.5K-step warm-up, learning rate 2×10^{-4} for the 800K-step row (SPRINT's fine-tuning setting) and 10^{-4} for the 1.72M-step row, everything else unchanged. Unguided rows use the sparse weights directly.

Sampling and scoring. SDE-250: the Euler–Maruyama sampler of SiT with 250 steps. ODE-50: 50 Euler steps. Guidance uses the classifier-free rule $\tilde{v} = v(x,\varnothing) + w\,(v(x,c) - v(x,\varnothing))$ and is applied only for interpolation times $t \le 0.7$ (the 70% of the trajectory closest to the data); the guidance scale is chosen on 10K samples and the reported 50K number is a single run at that scale ($w = 1.2$ for the 800K row without the interval, $w = 1.6$ for the 1.72M row with it). All samples are scored with the ADM evaluator (the unmodified `evaluator.py` of guided-diffusion) against the ImageNet 256×256 reference batch and its stored statistics, with 10K or 50K samples as stated. Every reported number is EMA weights, seed 0.

B COMPUTE ACCOUNTING

Training compute follows SPRINT's convention: the forward FLOPs of the training graph at batch 256, multiplied by the number of iterations. Published results use the source papers' reported configurations; where available, their training compute is taken from SPRINT's measurements and scaled to those training lengths. Unavailable compute is left unreported.

Epochs in Table 3 follow the source papers or their tabulation in SPRINT, with the following training-stage accounting. For our models and the SPRINT rows, epochs are images seen divided by 1.28M, with every stage counted at its batch size. Both SPRINT rows use 1.9M sparse iterations at batch 256 plus 100K dense iterations at batch 1024, giving 460 epochs including the dense stage omitted from SPRINT's own epoch count. MaskDiT's 1600 epochs cover masked training only, because the batch size of its 75K-step unmasking stage is unspecified. For MDTv2 the paper's own training length, 4.6M iterations at batch 256, replaces the 1080 epochs of SPRINT's table, and its compute is SPRINT's figure scaled to that length ($258.3 \times 4.6/5.4 = 220.0$).

The additional backbone comparisons use the metrics paired with their reported FID in the original papers: ADM-G,U (Dhariwal & Nichol, 2021), LDM-4 (Rombach et al., 2022), U-ViT-H/2 (Bao et al., 2023) and FiTv2-XL/2 (Wang et al., 2024). ADM's 673M parameters include its base model, upsampler and guidance classifier; we do not assign a single epoch count to this multi-stage system. U-ViT-H/2 uses 500K steps at batch 1024, corresponding to 400 epochs; FiTv2 uses

2M steps at batch 256, also 400 epochs. These baselines retain their original samplers rather than our SDE-250 protocol.

A dense SiT-XL/2 forward is 240 GFLOPs per sample, 61.4 TFLOPs per batch of 256, measured with `torch.utils.flop_counter` and consistent with the 61 TFLOPs per iteration implied by SPRINT's Table 1. The $4/6/8/6/4$ training graph measures 136.5 GFLOPs per sample, 34.9 TFLOPs per step, a ratio of 0.569 to dense (the token-weighted block count gives 0.571). The layouts $2/2/20/2/2$ and $3/3/16/3/3$ are counted by their token-weighted block counts, 0.393 and 0.464.

The dense stage costs $20\text{K} \times 4 \times 61.4 = 4.9 \times 10^6$ TFLOPs. The StairDrop rows of Table 3 are therefore $800\text{K} \times 34.9 + 4.9 \times 10^6 = 32.8 \times 10^6$ and $1.72\text{M} \times 34.9 + 4.9 \times 10^6 = 64.9 \times 10^6$ TFLOPs. The REPA row counts its alignment head, 2.1 GFLOPs per sample on the 128 tokens of the aligned block and 4.2 on 256 in the dense stage: $400\text{K} \times 35.5 + 5.0 \times 10^6 = 19.2 \times 10^6$. SPRINT's quality metrics and training compute are quoted as published; decoded from their Tables 1 and 9, their 400-epoch row is 1.9M sparse steps at 21.3 TFLOPs plus 100K dense steps at batch 1024, so 63% of its 65.1 million TFLOPs are the dense stage.

C SPRINT'S PUBLISHED NUMBERS AND ITS PUBLIC ARTIFACTS

We ran the released `sprint_XL_imagenet256` checkpoint (SHA-256 verified against the Hugging Face upload, the file behind the 400-epoch rows of their Table 3) through their released `sample_ddp.py` and evaluation configurations, byte-identical to the GitHub repository, and scored the samples with the ADM evaluator on 10K samples. With classifier-free guidance at their stated scale of 1.35 the checkpoint gives FID-10K 5.03. With path-drop guidance at the same scale it gives 5.57, and their released REPA checkpoint gives 6.04 under the same pipeline. On every 10K/50K pair we have measured, unguided and guided, ODE and SDE, B/2 and XL, the FID-10K exceeds the FID-50K by 2.75 to 3.02, and the $1/N$ slope of FID against sample count is the same on their sample sets as on ours (40K to 41K). The classifier-free number therefore corresponds to the printed 1.96. The path-drop numbers correspond to about 2.5 and 3.0, not to the printed 1.62 and 1.59; under their SDE sampler the path-drop pass is 0.54 worse than the classifier-free pass on their own weights, and the whole of the difference is a shift of the Inception feature mean. Table 3 retains the classifier-free result for the matched-FID comparison and the best published scheduled path-drop result for the comparable-compute comparison.

D CHANGING THE RESTORE

Table 4: Restoration variants on SiT-B/2 $2/2/3/2/3$ at 100K steps (SDE-250, FID-10K).

Restore	FID-10K
StairDrop: placeholder at each withheld position, then fusion with the skip	**31.95**
Span output appended as context tokens for the decoder; no restore, no fusion	55.57
Placeholder replaced by a learned upsampler of the span's output	49.15
Skip half of the fusion dropped with probability 0.1 during training	47.50

Table 4 compares alternatives to the inherited placeholder-and-skip interface. Appending the span output as context tokens changes both restoration and fusion; the other variants change the placeholder or regularise the skip. All three perform worse in this setting. These comparisons support retaining the original interface, but do not isolate why each alternative fails or establish that every component is necessary under other training choices.

E MOTIVATION PROBE PROTOCOL

Figure 3 uses the dense and one-level (SPRINT) 100K-step SiT-B/2 EMA checkpoints from Table 6. We evaluate the training graph on 2048 ImageNet training latents per probe seed, with class labels, no guidance and fixed noise level $t = 0.5$. Three probe sampling seeds vary the images, latent and noise draws, and token masks; they are not independent training runs. Each draw is replayed

Table 5: Motivation probes across noise levels. Mean normalised loss benefits from Eqs. (6) and (7) over three probe sampling seeds; radii are in original-grid patch units.

Probe / model	t	$d=0$	1	2	3	4	6	8	15
$C(d)$: Dense	0.1	0.00	0.80	0.91	0.95	0.97	0.98	0.99	1.00
	0.5	0.00	0.71	0.87	0.94	0.97	0.99	0.99	1.00
	0.9	0.00	0.84	0.94	0.98	0.99	1.00	1.00	1.00
$C(d)$: One level	0.1	0.00	0.25	0.56	0.71	0.80	0.88	0.93	1.00
	0.5	0.00	0.17	0.44	0.60	0.71	0.84	0.91	1.00
	0.9	0.00	0.06	0.21	0.37	0.51	0.71	0.83	1.00
$R(d)$: One level	0.1	0.14	0.83	0.93	0.95	0.97	0.98	0.99	1.00
	0.5	0.09	0.96	1.02	1.03	1.02	1.02	1.01	1.00
	0.9	0.09	0.81	0.90	0.94	0.97	0.99	1.00	1.00

across restrictions. Radii are Chebyshev distances on the original 16×16 token grid, so $d = 15$ is unrestricted. The implementation is `scripts/probe_roles.py`.

Within the span. In blocks 3–9, each query attends only to active positions within radius d, including itself. For the resulting velocity loss ℓ_d, we report the normalised attention benefit

$$C(d) = \frac{\ell_0 - \ell_d}{\ell_0 - \ell_{15}}. \tag{6}$$

The reference ℓ_0 uses self-only attention; residual paths and MLPs remain active. Thus $C(d)$ measures the benefit of cross-position attention relative to this reference, not the contribution of the entire span. The dense reference tests the same block indices, with all positions active.

After restoration. In blocks 10–12 of the one-level model, queries may attend directly to positions retained through the span only within radius d; attention to restored positions remains unrestricted. Let ℓ_d^{out} be the loss with the span output present and ℓ_d^{ref} the loss when it is replaced by the fixed placeholder before fusion, keeping the skip and the same attention restriction. The normalised output benefit is

$$R(d) = \frac{\ell_d^{\text{ref}} - \ell_d^{\text{out}}}{\ell_{15}^{\text{ref}} - \ell_{15}^{\text{out}}}. \tag{7}$$

Restored positions can relay information across later blocks, so this intervention does not bound all downstream spatial dependencies. $C(d)$ and $R(d)$ have different reference losses; both are normalised contrasts, can exceed one, and are computed within each seed before averaging. Figure 3 shows the means and sample standard deviations. Table 5 retains the full radius sweep at all three measured noise levels for these same probes.

F Full metrics

Table 6: Complete SiT-B/2 design study (ODE-50, no guidance): 10K samples at 100K steps; 50K at 400K. The one-level rows retain 64 tokens through blocks 3–9 (one fusion) or 3–10 (two fusions).

	100K steps		400K steps
Model	FID-10K	IS	FID-50K
Dense	70.72	20.0	38.58
TREAD (route 50%, blocks 3–9)	54.71	28.1	20.58
SPRINT (one level, one fusion)	39.25	41.7	11.48
One level, two fusions	34.87	48.6	
StairDrop $2/2/4/2/2$	**29.45**	**53.0**	**7.66**
StairDrop $2/1/6/1/2$	30.91	50.4	
StairDrop $1/2/6/2/1$	33.74	44.5	
StairDrop $3/2/4/2/1$	35.11	43.0	
StairDrop $2/3/2/3/2$	36.41	40.4	
StairDrop $1/2/4/2/3$	38.48	40.6	
StairDrop $2/2/3/2/3$, first level at 128 tokens	33.52	45.4	
StairDrop $2/2/3/2/3$, first level at 192 tokens	36.93	41.4	

Table 7: SiT-B/2 $2/2/4/2/2$ along training, SDE-250, FID on 50K samples, no guidance.

Steps	FID	sFID	IS	Prec.	Rec.
100K	24.89	9.08	54.4	0.653	0.459
200K	12.25	7.44	107.0	0.761	0.452
300K	8.72	7.02	137.8	0.791	0.473
400K	7.18	6.91	160.5	0.807	0.463

Table 8: SiT-XL/2 $4/6/8/6/4$ on the original backbone along training, ODE-50, 50K samples, no guidance, sparse weights.

Steps	FID	sFID	IS	Prec.	Rec.
400K	4.89	4.77	173.8	0.816	0.520
500K	4.40	4.68	185.3	0.820	0.515
600K	4.02	4.66	193.0	0.823	0.530
700K	3.78	4.64	200.1	0.823	0.528
800K	3.61	4.57	205.5	0.827	0.543

Table 9: The three XL layouts of Table 2 along training (ODE-50, FID-10K, no guidance).

Layout	100K	200K	300K	400K
$2/2/20/2/2$	17.03	9.74	8.47	8.08
$3/3/16/3/3$	25.07	11.70	8.46	7.52
$4/6/8/6/4$	22.41	11.20	8.76	7.68

Table 10: Guidance-scale sweep for the StairDrop rows of Table 3 (SDE-250, FID-10K). Interval guidance applies only at $t \leq 0.7$.

Model	w=1.2	1.3	1.4	1.5	1.6	1.7	1.8
800K + 20K, whole trajectory	**4.98**	5.24	–	–	–	–	–
800K + 20K, interval	–	–	4.74	**4.73**	–	–	–
1.72M + 20K, interval	–	–	4.77	4.58	**4.53**	4.61	4.74

G GENERATED SAMPLES

Additional selected samples from the same evaluation set as Figure 5. Each category has 27 distinct 256×256 images, arranged in rows of 3, 6, 9 and 9 at different display sizes.

Samoyed



Red panda



Figure 6: Selected StairDrop samples: Samoyed and red panda.

Flamingo



Coral reef



Figure 7: Selected StairDrop samples: flamingo and coral reef.

Pizza



Volcano



Figure 8: Selected StairDrop samples: pizza and volcano.