READ, ALLOCATE, ROUTE: ROLE-COUPLED TOKENIZATION FOR PIXEL-SPACE GENERATIVE TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel-space generative Transformers typically use one token per image patch, so capturing finer detail requires more tokens and makes global attention increasingly expensive. We observe that finer detail does not necessarily require more global communication: low-frequency structure benefits from global exchange, while high-frequency detail can be modeled locally. Based on this insight, we introduce SpecSlot, a pixel-space rectified-flow Transformer that decouples output detail from the length of the global attention sequence. SpecSlot expands each spatial anchor into multiple slots, each assigned a disjoint band of DCT coefficients; this separates anchor density from spectral subdivision. Only the lowest-frequency slot from each anchor participates in global attention, while all slots interact locally. This allows the model to increase spectral detail without enlarging the global sequence. A spatial head further adds a local correction to the spectral prediction, while the generative state, target, and loss remain in pixel space. On ImageNet-256, SpecSlot achieves a comparable final FID to a FLOPs-matched JiT control, while its FID drops faster early in training. On ImageNet-512, it improves FID by 2.84 over the control with 3.9% fewer FLOPs, and comes within 0.69 FID of JiT-B/16 at 38.2% lower cost.

Keywords: pixel-space image generation; generative Transformers; rectified flow; frequency-structured generation; efficient attention.

1 INTRODUCTION

Pixel-space generative models learn directly from noise to pixels, eliminating the separately trained autoencoder and its compression bottleneck. This simplicity, however, comes with a computational cost: the full-resolution image must be represented as Transformer tokens. Most pixel-space Transformers use one token per spatial patch, so the patch grid simultaneously determines what the model reads, what each token predicts, and how many tokens participate in global attention. As a result, finer image detail directly increases attention cost: on a 256×256 image, reducing the patch size from 16×16 to 8×8 increases the token count from 256 to 1,024, making the quadratic term of dense attention $16\times$ larger.

Existing methods address this cost from different directions, but largely retain the coupling between output detail and token communication. JiT routes every patch token through the global sequence (Li & He, 2026). Multiscale systems move detail to extra stages or detailer heads (Yu et al., 2026; Chen et al., 2026; He et al., 2026), and efficient-attention methods shorten the sequence with generic patterns—windows, selected global tokens, or learned sparsity (Liu et al., 2021; Beltagy et al., 2020; Jaegle et al., 2021; Zhou et al., 2026). Spectral Forcing brings in frequency structure, but only to filter the noisy input; its tokens remain spatial patches (Fan et al., 2026). In all of them, a token is still a spatial patch.

Our key observation is that finer detail does not necessarily require proportionally more global communication: in the frequency domain, low-frequency structure carries broad image organization and benefits from long-range exchange, while fine detail is largely local. A token carrying broad structure therefore stands to gain from exchanging with the whole image, while a token carrying fine

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.

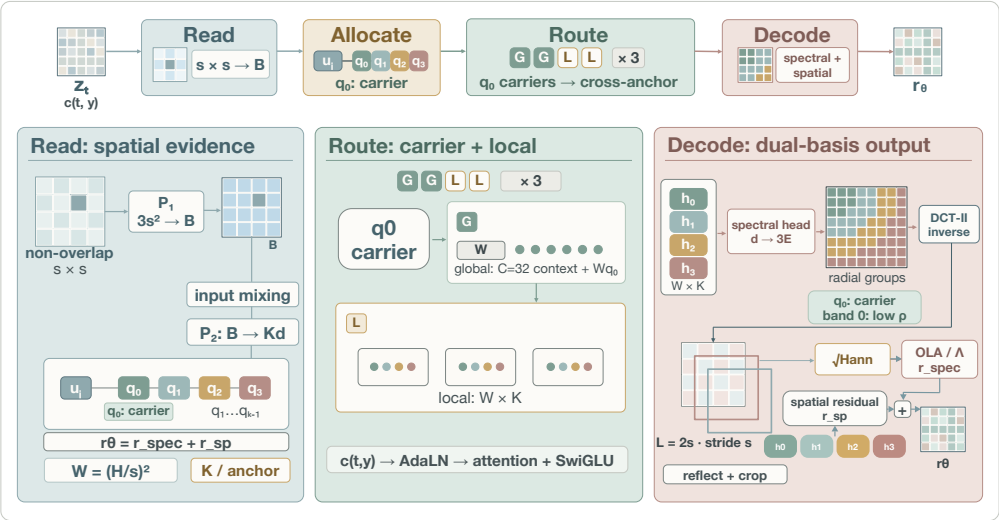

Figure 1: SpecSlot data flow. Each spatial anchor expands into K slots. Slot 0 owns the lowest-radial coefficient group and enters cross-anchor GGLL attention. The same slots decode a lapped-DCT spectral base and an anchor-aligned spatial correction; their sum is the pixel-space prediction.

detail mostly does not. Yet in existing designs, what a token communicates is fixed by where it sits, not by what it produces. Can a token's output role also determine how it should communicate?

SpecSlot builds on this idea by making output roles explicit and using them to control communication. It organizes computation around three operations: *Read*, *Allocate*, and *Route*. Read forms a grid of spatial anchors that collect input evidence. Allocate expands each anchor into K indexed slots, with each slot assigned a disjoint radial group of DCT coefficients. Route then uses these spectral roles to decide communication: only the slot assigned the lowest-frequency group from each anchor enters the global stage, while all slots interact locally within the anchor. This separates fine-grained output representation from the length of the global sequence. A spatial head further reuses the same slot states to add a local correction to the spectral prediction, without introducing additional tokens. Figure 1 illustrates this design.

This representation exposes two separate allocation variables. W controls the density of spatial anchors, while K controls the spectral subdivision at each anchor. The total number of slots grows with WK, but the global image sequence contains only W carriers. A fixed token budget can therefore be allocated either to denser spatial anchors or to finer spectral roles without forcing both to increase together. This turns tokenization from a single patch-size choice into an explicit quality–compute allocation problem.

Our contributions are:

- We introduce an indexed $W \times K$ representation in which spatial anchors collect input evidence and slots receive disjoint spectral coefficient groups. The split exposes anchor density and coefficient subdivision as separate allocation variables, and the same slot states also decode a local pixel correction.
- We define carrier-routed global–global–local–local (GGLL) attention from the slot index: one lowest-radial carrier per anchor enters global blocks, and all slots enter anchor-local blocks, so the global sequence contains W carriers rather than WK slots.
- On ImageNet-256 and ImageNet-512, SpecSlot provides two measured quality–compute operating points between JiT's low-cost and high-cost settings. At ImageNet-512, W256/K4 lowers FID by 2.84 over our FLOPs-matched JiT control (JiT-CM) with 3.9% fewer GFLOPs, and trails JiT-B/16 by 0.69 FID at 38.2% lower cost. The advantage grows with resolution: finer grids add detail without adding global communication.

2 Related Work

Latent diffusion dominates image synthesis by moving generation into an autoencoder's latent space (Rombach et al., 2022). DiT and SiT show that plain Transformers scale reliably in this space (Peebles & Xie, 2023; Ma et al., 2024). Pixel-space generation removes this autoencoder but must handle a long token sequence at full resolution. Simple Diffusion answers with a scaled, downsampling U-Net (Hoogeboom et al., 2023); PixelFlow splits generation into cascaded flows, so each stage works at a tractable resolution (Chen et al., 2025); and JiT shows that a plain large-patch Transformer already matches latent baselines (Li & He, 2026). To recover fine detail under coarse tokenization, PixelDiT, DiP, and HyperDiT attach pixel-level stages, detailer heads, or cross-scale connectors (Yu et al., 2026; Chen et al., 2026; He et al., 2026), while efficiency work merges, prunes, or reschedules tokens to shorten the sequence (Bolya & Hoffman, 2023; Chang et al., 2025; Zhao et al., 2026). These designs differ in where the compute goes, but share one premise: a token is a spatial patch, so finer detail and wider communication stay tied to the same spatial grid.

Frequency offers a complementary view of this problem. DCTdiff generates images directly in DCT space (Ning et al., 2025). VAR and NFIG generate coarse to fine, ordering the process by scale or frequency band (Tian et al., 2024; Huang et al., 2025). Spectral Forcing filters the noisy input with a frequency-dependent schedule (Fan et al., 2026), and focal frequency loss reweights spectral errors (Jiang et al., 2021). DeCo goes further and assigns low and high frequencies to a DiT and a separate decoder (Ma et al., 2026). In this line of work, frequency shapes the state, the order, or the loss—but not the tokens. In the methods discussed above, frequency does not determine which tokens communicate globally and which remain local. SpecSlot starts from this gap: it assigns tokens fixed spectral roles, and lets these roles decide both what each token predicts and how widely it communicates. We discuss each thread and its relation to SpecSlot in detail in Appendix A.

3 Methodology

SpecSlot organizes pixel-space generation around one idea: tokenization as allocation—assigning each token a fixed output role and dividing a fixed token budget between anchor density and spectral subdivision. Global attention is then spent only on the role that needs it; the rectified-flow state, target, and loss remain in pixel space.

3.1 Shared Pixel-Space Rectified-Flow Objective

Following the JiT controls and objectives (Liu et al., 2022), we mix an image $x \in [-1,1]^{3\times H\times H}$ with noise $\epsilon \sim \mathcal{N}(0, I)$ on the linear path

$$z_t = tx + (1-t)\epsilon, \tag{1}$$

where $t = \sigma(g)$ with $g \sim \mathcal{N}(-0.8, 0.8^2)$ and y is the class label. The network returns an image-shaped prediction $r_\theta(z_t, t, y)$, converted to a velocity by

$$v_\theta(z_t, t, y) = \frac{r_\theta(z_t, t, y) - z_t}{\max(1-t, \tau)}, \qquad \tau = 0.05, \tag{2}$$

where the clamp avoids the singularity at $t = 1$, and training minimizes the velocity MSE

$$\mathcal{L}_{\mathrm{RF}} = \mathbb{E}_{x,\epsilon,t,y}\left[\frac{1}{3H^2}\left\|v_\theta(z_t, t, y) - (x-\epsilon)\right\|_2^2\right]. \tag{3}$$

3.2 SpecSlot Architecture

Why is allocation needed? In a patch-grid model, adding finer detail means shrinking the patch, and the global sequence grows with it; the two cannot be adjusted separately. Prior designs keep the spatial patch as the only allocation unit (Section 2). SpecSlot instead introduces the missing variables, forming W spatial anchors and expanding each anchor into K indexed slots (Figure 2):

$$N = WK, \qquad L = 2s, \qquad W = (H/s)^2, \qquad E = \frac{L^2}{K} = \frac{4H^2}{N}. \tag{4}$$

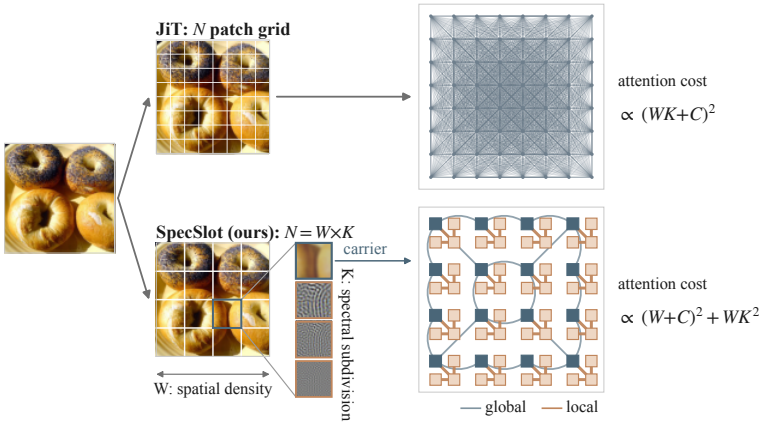

Figure 2: SpecSlot allocation. A patch grid spends all N tokens in one global sequence; splitting $N{=}WK$ into W spatial anchors and K spectral slots keeps global attention at W carriers.

The budget splits into W anchors that read the input and K slots per anchor that divide the output; each slot owns an equal share E of the coefficients per channel, with s the input cell side and L the output window side. The two variables act separately: W sets how densely the input is read, and K how finely the output is divided. At fixed N, anchor density trades directly against spectral subdivision, and output detail can grow with K while the global sequence stays at W carriers. Equal N does not imply equal cost, since communication and projection scale differently with the split.

Three operations implement this allocation, as shown in Figure 1: Read collects each anchor's input evidence (Section 3.2.1), Allocate assigns each slot a fixed output role (Section 3.2.2), and Route lets the role decide communication (Section 3.2.3).

3.2.1 Read: Spatial Anchors and Slot Expansion

The first question is where input evidence comes from; SpecSlot reads it on a grid of anchors, independent of how the output is divided. Let $\mathcal{P}_i$ extract the i-th non-overlapping $s \times s$ cell of z_t. A shared projection maps each cell to an anchor vector,

$$u_i = P_1 \,\mathrm{vec}(\mathcal{P}_i z_t) \in \mathbb{R}^B, \qquad P_1 \in \mathbb{R}^{B \times 3s^2}. \tag{5}$$

a patch embedding whose width B is free, because the output count no longer depends on it. A zero-initialized residual depthwise 3×3 convolution over the anchor grid—the input mixer—lets neighboring cells exchange evidence before the evidence is split into roles. A shared 1×1 projection expands each mixed anchor vector into K hidden states,

$$[q_{i,0}; \ldots; q_{i,K-1}] = \mathrm{reshape}(P_2 \widetilde{u}_i + b_2)\,, \qquad q_{i,k} \in \mathbb{R}^d, \tag{6}$$

where $P_2 \in \mathbb{R}^{Kd \times B}$; a two-dimensional anchor position plus a learned slot position are added. One anchor vector seeds all K slots; the roles come from the slot positions, output heads, and routing, not from the input. All WK slots are evaluated at every flow time.

3.2.2 Allocate: Radial Coefficients and Pixel Synthesis

Radial coefficient assignment. The second question is who produces what. SpecSlot assigns each slot a fixed share of the output in a DCT basis. Low radial frequencies carry broad structure and high frequencies fine detail, and natural-image spectra decay smoothly along this order (Fan et al., 2026); a radial split therefore sorts the roles from global to local without any learned assignment. For an output window of side $L = 2s$, the DCT-II indices (u, v) are ordered by $\sqrt{u^2 + v^2}$ and split into K equal groups, requiring $K \mid L^2$; slot k of each anchor owns group k, that is, $E = L^2/K$ coefficients per channel. The disjoint groups cover the grid: every coefficient is predicted exactly once. A shared affine head predicts the owned coefficients from the slot's final hidden state $h^{\mathrm{out}}_{i,k}$:

$$\widehat{c}_{i,k} = P_{\mathrm{spec}} h^{\mathrm{out}}_{i,k} + b_{\mathrm{spec}} \in \mathbb{R}^{3E}. \tag{7}$$

Sharing the head is deliberate: the slot index, not per-band parameters, defines ownership.

Lapped synthesis. Decoding each window independently would leave neighbors free to disagree on their shared edge, and the mismatch shows as block seams, most visibly early in training. SpecSlot decodes by the synthesis half of a 50%-lapped DCT frame instead: inverse DCT, a square-root periodic Hann window, and normalized overlap-add. Neighboring windows share half their pixels, so seams are removed by construction rather than learned; a matched-control ablation confirms the gain at equal compute (Table 13).

$$r_{\text{spec}} = \text{OLA}\Big(\{\text{IDCT}(\widehat{c}_i)\}_{i=1}^{W}\Big), \tag{8}$$

with padding and normalization deferred to Appendix C.

Anchor-aligned spatial correction. The same slot states can also be read out directly in pixel space. A second head adds an anchor-aligned correction alongside the spectral base: slot k writes tile k of a deterministic row-major $K_h \times K_w$ tiling ($K_hK_w{=}K$) of its anchor cell (four 8×8 quadrants for $K{=}4$, $s{=}16$). These projections are slot-specific and zero-initialized, so training starts from a pure spectral model. The correction adds output parameters but no token states or attention edges. Radial ownership applies only to the spectral base; the spatial term is a pixel readout with no frequency assignment, and no orthogonality constraint is imposed. The two terms express the same prediction in different bases, and Section 4.6.1 finds their content complementary. The prediction is

$$r_\theta(z_t, t, y) = r_{\text{spec}} + r_{\text{sp}}. \tag{9}$$

3.2.3 Route: Carrier and Local Attention

The third question is who needs to communicate globally. Sending every slot through global attention makes its quadratic term grow with K, tying detail and communication together again. The spectral roles suggest a cheaper split: the lowest radial group holds the broad structure that benefits from cross-anchor exchange, while finer groups are increasingly local; its signal also overtakes the noise earliest in the flow (Appendix I.1). SpecSlot therefore routes by role: the owner of the lowest radial group, slot 0, is the carrier and the only slot entering the global stage; all K slots then mix locally within their anchor. Since global and local blocks alternate, the carrier brings global information into its anchor and takes local detail back into the next global stage.

Let $\mathbf{Z} \in \mathbb{R}^{W\times K\times d}$ be the slot tensor and $\mathbf{Y} \in \mathbb{R}^{C\times d}$ the $C{=}32$ context states formed from the class embedding. The two routing sets are

$$\mathcal{A}_G = \text{Concat}(\mathbf{Y}, \mathbf{Z}_{:,0}), \qquad \mathcal{A}_L^{(i)} = \mathbf{Z}_{i,:}. \tag{10}$$

Two consecutive blocks process the global set and two more process each local set; this global-global-local-local (GGLL) schedule repeats three times, giving six global and six local blocks of identical dimensions. Class context appears only in the global group. Non-carrier slots traverse the six local blocks, context states the six global ones, and carriers all twelve. On ImageNet-256 with $s{=}16$ and $K{=}4$, the global sequence holds $W{+}C = 288$ tokens instead of $WK{+}C = 1{,}056$.

Ignoring projections and elementwise operations, the quadratic terms are

$$\mathcal{C}^{\text{quad}}_{\text{GGLL}} \propto 6(W + C)^2 d + 6WK^2 d, \qquad \mathcal{C}^{\text{quad}}_{\text{full}} \propto 12(WK + C)^2 d, \tag{11}$$

where d is the hidden width; the global quadratic term now grows with W, not WK. Measured GFLOPs include the full execution and decoder, not only these terms (Appendix B).

Allocation makes the trade-off measurable: with W and K independent, the quality–compute trade-off becomes a property of the tokenization, and our experiments vary exactly these knobs.

4 Experiments

4.1 Experimental Setup

Models and training. We train class-conditional models on ImageNet-1K (Deng et al., 2009) and report results at ImageNet-256 and ImageNet-512. Unless stated otherwise, ImageNet-256 models and ablations use the $W256/K4/B1536$ configuration of Section 3, with B the input-projection width. ImageNet-256 training lasts 120 epochs at effective batch 512 (the ImageNet-512 contract is in Appendix B), and the aligned models share the full training recipe.

Table 1: **Design evidence at ImageNet-256.** All rows are fully trained. **(a)** Slot allocation: the upper block fixes N=2,048, the lower fixes W=256; all use B=1536 and five seeds. **(b)** Input/output bases and **(c)** communication schedules at fixed $W256/K4/B1536$; alternatives use three seeds; $W256/K4$ rows inherit the five-seed mean of Table 2. Dense-global-CM narrows the hidden width to 560 to match the GGLL forward budget. Efficiency columns come from one profiler run.

(a) Slot allocation

Alloc.	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
Fixed slot budget				
$W64/K32$	7.28	218.52	**195.43**	**9.00**
$W256/K8$	6.28	**219.52**	220.02	6.55
$W1024/K2$	**5.66**	212.31	292.29	5.21
Fixed anchor grid				
$W256/K2$	7.39	**222.97**	**83.84**	**20.10**
$W256/K4$	6.59	215.20	129.20	9.06
$W256/K8$	**6.28**	219.52	220.02	6.55

(b) Input and output bases

Input	Output	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
Single basis					
Freq.	Freq.	7.34	**222.46**	129.30	9.35
Freq.	Spat.	7.23	215.73	129.20	**9.63**
Spat.	Freq.	7.10	221.37	129.20	9.06
Spat.	Spat.	6.99	215.56	**129.10**	9.34
Dual basis					
Spat.	Freq.+ Spat.	**6.59**	215.20	129.20	9.06

(c) Communication schedules at fixed $W256/K4/B1536$

Route	Cross-anchor image states	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
GGLL	one carrier per anchor	6.59	215.20	129.20	**9.06**
GLGL	one carrier per anchor	6.95	218.95	129.20	9.00
Dense-global-CM	all image slots	6.81	210.20	**128.96**	5.80
Full-global	all image slots	**5.84**	**237.54**	225.82	5.11

Evaluation. We sample with 50 Heun intervals on the common legacy-uniform grid and use EMA1 weights. Every row uses the best classifier-free guidance (CFG) from a per-model coarse-to-fine sweep (Appendix E); the JiT-B/8 optimum is flat over 2.9–3.25, and within this region the JiT rows report 2.9 to match SF's scale (Fan et al., 2026). FID and Inception Score (Heusel et al., 2017; Salimans et al., 2016) are means over five independently trained seeds at 50K images per seed (per-seed values in Appendix G); trained alternatives use three seeds, rows reusing the W256/K2 or W256/K4 allocations inherit those five-seed means, and compared checkpoints share class ordering and noise draws.

Compute accounting. We measure forward cost on one H800 in BF16 with whole-network TorchInductor, counting a multiply–add as two FLOPs (full profiler contract in Table 4). Throughput is the best Heun-50 value over a shared batch sweep.

Compute-matched JiT control. JiT-CM/125G keeps JiT-B/8's raw 8×8 tokenizer and routes features through a 128-dimensional input bottleneck and 32 context tokens inserted at block 4 of the 12-block trunk; width 560 with 10 heads measures 124.24 GFLOPs under the profiler above. It matches forward GFLOPs rather than trainable-parameter count and is an implementation-defined compute control, not a published JiT model size. The ImageNet-512 control JiT-CM/141G uses the corresponding 16×16 tokenizer at width 600 (141.21 GFLOPs).

4.2 Understanding the Design

4.2.1 Anchor Density and Coefficient Subdivision

Both allocation axes trade compute for quality, but they are not interchangeable: W improves the read, K refines the write (Table 1a). Increasing K at fixed W=256 lowers mean FID at higher cost, with diminishing returns (7.39, 6.59, and 6.28 for K=2, 4, 8; the last doubling buys only 0.31 FID). Shifting the split toward more anchors at fixed N=2,048 lowers FID monotonically (7.28, 6.28, and 5.66 for W=64, 256, 1,024), with no clear saturation in this range, though W also moves the carrier count, local-group size, and projection shapes at once. A teacher-forced entropy-rank probe mirrors this saturation: the effective-slot fraction falls from 0.80 at K=2 to 0.50 at K=8 (Appendix Table 16), indicating sublinear representational gains without identifying their cause. We therefore

Table 2: ImageNet quality and compute (five-seed means, 50K images per seed). SpecSlot rows report the best swept B and CFG per allocation (SF-B/32 at ImageNet-512: CFG 9.50). GFLOPs are conditional forward costs; img/s is peak Heun-50 throughput; – is the standard linear patch projection of JiT and SF.

Model	Tokens	B	CFG	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
ImageNet-256							
JiT-B/16	256	–	2.90	7.51	174.81	50.45	33.08
JiT-B/8	1,024	–	2.90	4.34	225.72	218.50	8.15
SF-B/16	256	–	2.90	7.81	172.35	50.85	32.89
JiT-CM	1,024	–	2.90	6.68	182.56	124.24	8.94
SpecSlot (W256/K2)	512	512	5.00	7.27	219.49	82.63	21.32
SpecSlot (W256/K4)	1,024	1,536	4.00	6.59	215.20	129.20	9.06
ImageNet-512							
JiT-B/32	256	–	2.90	15.88	162.12	51.50	30.61
JiT-B/16	1,024	–	2.90	8.45	196.61	219.56	7.93
SF-B/32	256	–	9.50	11.41	172.48	54.72	30.88
JiT-CM	1,024	–	2.90	11.98	178.72	141.21	8.51
SpecSlot (W256/K2)	512	3,072	10.50	10.94	183.69	94.52	18.32
SpecSlot (W256/K4)	1,024	4,096	8.25	9.14	208.24	135.64	12.69

use K2 as the lower-cost point and K4 as the intermediate point. Component controls (width, input mixer, lapping) are reported in Appendix Tables 11–13.

4.2.2 Input and Output Representations

At the same spectral output, spatial input has lower mean FID than frequency input (7.10 vs. 7.34), and every single-basis variant sits at 6.99 or above; only the dual-basis hybrid falls below 7.0, reaching 6.59—0.40 below the best single basis—without adding tokens or cross-anchor attention edges (Table 1b). This margin indicates that the two output bases are not redundant: the spatial correction supplies structure that the spectral base alone misses. Section 4.6 localizes this complementarity over the flow and across frequency bands.

4.2.3 Carrier Routing

To test routing at a fixed forward budget, we train a full-global control (Dense-global-CM) on the same $W256/K4$ configuration, replacing the carrier/local schedule with 12 full-global blocks over all 1,024 image slots; its hidden width is 560, matching the measured GGLL forward cost—a compute-matched rather than parameter-matched comparison. GGLL is both better and faster than this control at essentially identical FLOPs: 6.59 vs. 6.81 FID and 9.06 vs. 5.80 images/s (Table 1c). Full-global attention reaches a lower FID (5.84) but at 74.8% higher cost (225.82 vs. 129.20 GFLOPs) and lower throughput. The carrier design therefore targets the intermediate, attention-constrained regime, saving 42.8% of the full-global cost. GLGL keeps the same carrier set and global budget but alternates the global/local order; it reaches 6.95 at identical compute, so the order matters, not just the budget: the carrier must gather the cross-window state before local write-back can use it.

4.3 Comparison with Pixel-Space Baselines

Reading the ImageNet-256 panel of Table 2 along the compute axis, W256/K2 and W256/K4 are two measured, nondominated operating points between JiT-B/16 and JiT-B/8: W256/K2 improves on JiT-B/16 (7.27 vs. 7.51), and W256/K4 improves further (6.59) while using 40.9% fewer forward FLOPs than JiT-B/8, which remains the quality leader at 4.34.

JiT-CM gives the closer compute comparison at ImageNet-256: W256/K4 uses 4.0% more forward FLOPs, has comparable throughput, and obtains a comparable mean FID, 6.59 vs. 6.68, within the five-seed 95% interval (Appendix G). The aligned five-seed mean trajectories (Appendix Figure 6) show that SpecSlot's FID descends faster at early-to-mid checkpoints.

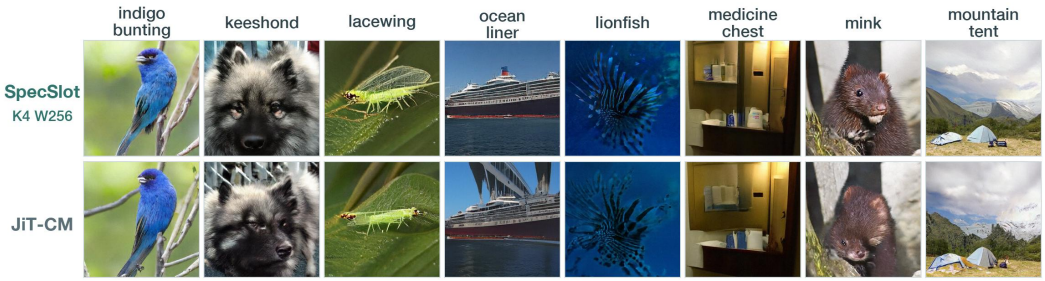

Figure 3: Qualitative comparison on ImageNet-256 with paired class and noise draws. The top row shows SpecSlot W256/K4 and the bottom row shows JiT-CM; each column uses the same class label and noise draw for both models.

4.4 ImageNet-512: Scaling to Higher Resolution

SpecSlot's advantage widens at ImageNet-512: a higher-resolution image adds mostly local detail, not global structure, so dense baselines pay quadratically for the larger grid while our global sequence stays at 288 tokens. The allocation uses wider evidence and larger cells, $B{=}4096$ for W256/K4 and $B{=}3072$ for W256/K2 (full contract in Appendix B). W256/K4 reaches 9.14 FID at 135.64 GFLOPs: 2.84 FID points lower than JiT-CM/141G (24% relative reduction) while using 3.9% fewer forward GFLOPs, and within 0.69 of JiT-B/16 (8.45) at 38.2% lower cost and $1.6\times$ higher throughput. It also attains the highest Inception Score in the ImageNet-512 panel, and both allocations outperform the input-side spectral baseline SF-B/32 (11.41). The lower-cost W256/K2 reaches 10.94 at 94.52 GFLOPs, 1.04 below JiT-CM/141G at 33.1% less compute, and saves 30.3% of W256/K4's forward cost at a 1.80 FID increase.

At nearly the same JiT compute budget (218.50 vs. 219.56 GFLOPs), the FID gap to JiT narrows from 2.25 (JiT-B/8 at ImageNet-256) to 0.69 (JiT-B/16 at ImageNet-512), while our lead over the compute-matched control grows from 0.09 to 2.84—indicating a larger measured benefit at the higher resolution.

4.5 Qualitative Analysis

Figure 3 compares paired ImageNet-256 samples with the same class and noise draw. In the shown pairs, SpecSlot has fewer visible structural collapses in the ocean liner and lacewing: the ship is more distinct, and the insect outline is less merged with the leaf. The indigo bunting and lionfish retain finer detail, while the medicine chest and mountain tent have clearer object layouts. Additional paired comparisons and selected failure cases are provided in Appendix Figures 8–9.

4.6 Mechanistic Analysis

4.6.1 Spectral Base and Spatial Correction

Figure 4(a–c) shows the spectral term has the larger self-energy, while the spatial term's share grows from 0.07 to 0.23 over the flow, and the normalized cross term stays positive—the two terms reinforce, not cancel. Projected through the matched lapped-DCT analysis frame (which does not confer radial coefficient ownership), the spatial term's share is smallest in the lowest radial group and dominates the higher groups late in the flow. Removing either term with the trunk fixed raises velocity MSE: at $t{=}0.90$ by $32.1\times$ (spatial) or $64.1\times$ (spectral), and at $t{=}0.05$ by $1.1\times$ and $2.1\times$. The spectral branch is the main scaffold throughout; the spatial branch supplies a later, higher-frequency correction.

4.6.2 Carrier Identity and Routed Communication

We vary only which slot receives cross-anchor access, holding the $W256/K4$ allocation, GGLL graph, and compute fixed (Table 3).

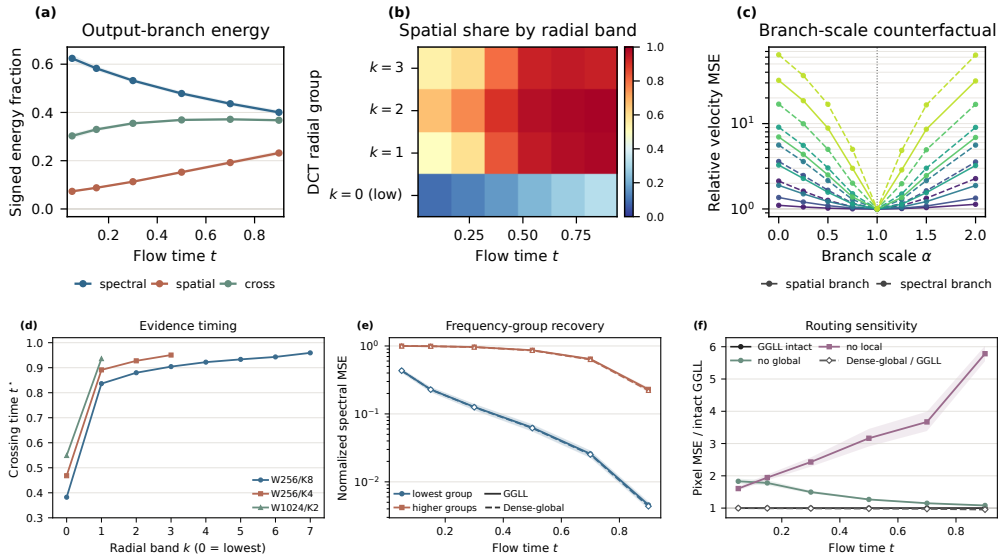

Figure 4: Mechanistic probes of the evaluated $W256/K4$ checkpoint (128 paired states, teacher-forced). **(a–c)** Output branches: (a) branch self-energies and signed cross term, normalized by total output energy; (b) the spatial/(spectral+spatial) self-energy share; (c) relative MSE under spatial (solid) or spectral (dashed) scaling, dark-to-light over $t \in \{0.05, \ldots, 0.90\}$; (a) shows 95% bootstrap intervals, (b,c) means. **(d)** Evidence timing: data-to-noise ratio (DNR) crossing time per radial band for the paired W256/K8, W256/K4, and W1024/K2 checkpoints (signal-over-noise onset; the lowest band turns reliable first). **(e,f)** Route interventions inside the shared Transformer: (e) reconstruction error for low and higher radial groups; (f) pixel-MSE ratios after removing cross-window carrier edges or local cross-band edges (carrier self edges and context reads retained). Dense-global-CM/GGLL: intact trained controls; curves are submodule diagnostics, not end-to-end FID.

Replacing the lowest-radial carrier with slot 1, 2, or 3 raises mean FID from 6.59 to 6.91, 7.21, and 6.97. The assignment matters: moving cross-anchor access to any other slot raises FID, and the lowest-radial role is also the one whose signal emerges earliest (Appendix I.2)—supporting the premise behind routing by role. Figure 4(d) shows this directly: the lowest radial band's signal becomes reliable first in the flow, which is why slot 0 is assigned the global path.

Table 3: Carrier-identity ablation at fixed $W256/K4/B1536$: only the slot with cross-anchor access changes. Alternatives use three seeds; slot 0 reuses the five-seed row of Table 2.

Global carrier	FID ↓	IS ↑
Slot 0 (lowest radial)	**6.59**	215.20
Slot 1	6.91	**220.38**
Slot 2	7.21	203.90
Slot 3	6.97	214.87

Table 1 compares separately trained communication graphs, whereas Figure 4(e,f) changes only route edges inside the shared Transformer. On paired states, the lowest radial group is recovered earlier than the higher groups; removing cross-window carrier edges costs most early, while removing local cross-band edges costs most late.

5 Conclusion

SpecSlot recasts tokenization as an allocation problem: a fixed token budget is divided between anchor density and spectral subdivision, and global attention is spent by role instead of covering every token. On ImageNet-512, this design improves FID by 2.84 over the FLOPs-matched JiT control and approaches JiT-B/16 at 38% lower cost; on ImageNet-256, it achieves a comparable final FID while descending faster early in training. The coupling between detail and communication in patch-grid models is thus a tokenization artifact rather than a necessity. Our diagnostics identify the lowest-radial role as the earliest-recovered carrier, which motivates assigning it cross-anchor access. Allocation gives pixel-space models a new axis for trading detail against communication.

AI USE STATEMENT

In this work, we used generative AI tools for polishing paragraphs and correcting grammatical mistakes during the writing process. We have not used generative AI tools for idea generation, method design, experimental design, or data analysis; the idea, method, experiments, and analyses are entirely the authors' own work. We have reviewed all AI-assisted text. We take responsibility for the final content of this work, including text and claims produced with the aid of generative AI.

ETHICS STATEMENT

This work studies class-conditional image generation on ImageNet-1K. It does not involve human participants, private user data, or user profiling. We use ImageNet under its existing access and use terms and do not redistribute the dataset. The generated-image examples and failure cases in the appendix are used only to characterize model behavior; they are not claims about real people or real-world deployments.

REPRODUCIBILITY STATEMENT

The main text and Appendix B specify the model and training contracts for both resolutions, including the tokenization, widths, routing, optimizer, schedule, batch sizes, and hardware measurement setup. Appendix D records the shared evaluation protocol, while Appendix E describes CFG selection and Appendix F describes the latency measurements. Per-seed results and across-seed statistics are given in Appendix G; the checklist in Appendix M records the checkpoint, seed, EMA, sampler, CFG, image-count, statistics, and profiler contracts used to accept each reported row.

REFERENCES

Sotiris Anagnostidis, Gregor Bachmann, Yeongmin Kim, Jonas Kohler, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Albert Pumarola, Ali Thabet, and Edgar Schönfeld. Flexidit: Your diffusion transformer can easily generate high-quality samples with less compute, 2025. URL https://arxiv.org/abs/2502.20126.

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LFUaHUCO1a.

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, 2023. URL https://arxiv.org/abs/2209.12152.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4599–4603, 2023. doi: 10.1109/CVPRW59228.2023.00484.

Shuning Chang, Pichao WANG, Jiasheng Tang, Fan Wang, and Yi Yang. Sparsedit: Token sparsification for efficient diffusion transformer. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=jTBxyQempF.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow, 2025. URL https://arxiv.org/abs/2504.07963.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space, 2026. URL https://arxiv.org/abs/2511.18822.

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 *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Peter Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Lucic, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VpGFHmI7e5.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Weitao Du. Frequency-forcing: From scaling-as-time to soft frequency guidance. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026. URL https://openreview.net/forum?id=dvTYdwKbLj.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–12878, Los Alamitos, CA, USA, June 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01268. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01268.

Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models, 2025. URL https://arxiv.org/abs/2505.11278.

Weichen Fan, Haiwen Diao, Penghao Wu, and Ziwei Liu. Show the signal, hide the noise: Spectral forcing for pixel-space diffusion, 2026. URL https://arxiv.org/abs/2606.15236.

Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet score-based generative modeling. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 478–491. Curran Associates, Inc., 2022. doi: 10.52202/068431-0035. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/03474669b759f6d38cdca6fb4eb905f4-Paper-Conference.pdf.

Yu He, Lichen Ma, Zipeng Guo, Xinyuan Shan, Jingling Fu, Dong Chen, Junshi Huang, and Yan Li. Hyperdit: Hyper-connected transformers for high-fidelity pixel-space diffusion, 2026. URL https://arxiv.org/abs/2605.15741.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images, 2023. URL https://arxiv.org/abs/2301.11093.

Zhihao Huang, Xi Qiu, Yukuo Ma, Yifu Zhou, Junjie Chen, Hongyuan Zhang, Chi Zhang, and Xuelong Li. Nfig: Multi-scale autoregressive image generation via frequency ordering, 2025. URL https://arxiv.org/abs/2503.07076.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation, 2023. URL https://arxiv.org/abs/2212.11972.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention, 2021. URL https://arxiv.org/abs/2103.03206.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13899–13909, 2021. doi: 10.1109/ICCV48922.2021.01366.

Dong Hoon Lee and Seunghoon Hong. Variable-length tokenization via learnable global merging for diffusion transformers, 2026. URL https://arxiv.org/abs/2606.20076.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026. URL https://arxiv.org/abs/2511.13720.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00986. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00986.

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, 2024. URL https://arxiv.org/abs/2401.08740.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation, 2026. URL https://arxiv.org/abs/2511.19365.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations, 2021. URL https://arxiv.org/abs/2103.03841.

Mang Ning, Mingxiao Li, Jianlin Su, Jia Haozhe, Lanmiao Liu, Martin Benes, Wenshuo Chen, Albert Ali Salah, and Itir Onal Ertugrul. DCTdiff: Intriguing properties of image generative modeling in the DCT space. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=vDoAA8xKXL.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00387.

Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10199–10208, 2023. doi: 10.1109/CVPR52729.2023.00983.

Hao Phung, Quan Dao, Trung Dao, Hoang Phan, Dimitris Metaxas, and Anh Tran. Dimsum: Diffusion mamba – a scalable and unified spatial-frequency method for image generation, 2025. URL https://arxiv.org/abs/2411.04168.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022. doi: 10.1109/CVPR52688.2022.01042.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Keyu Tian, Yi Jiang, Zehuan Yuan, BINGYUE PENG, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=gojL67CfS8.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion, 2025. URL https://arxiv.org/abs/2507.23268.

Weiyi You, Jinhua Zhang, Xingyu Zhou, Wei Long, Junyu Lou, and Shuhang Gu. Advanced pixel diffusion model with guided sparse global refinement, 2026. URL https://arxiv.org/abs/2609.00798.

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

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Hao Luo, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dydit++: Diffusion transformers with timestep and spatial dynamics for efficient visual generation, 2026. URL https://arxiv.org/abs/2504.06803.

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

Yifan Zhou, Zeqi Xiao, Tianyi Wei, Shuai Yang, and Xingang Pan. Trainable log-linear sparse attention for efficient diffusion transformers, 2026. URL https://arxiv.org/abs/2512.16615.

Appendix Contents

A. Extended Related Work 14
B. Implementation Contract 15
C. Lapped Spectral-Base Synthesis 17
D. Evaluation Protocol 18
E. Classifier-free Guidance Sweep 18
F. Sampling-step Latency Scaling 18
G. Five-seed ImageNet-256 Results 20
H. Additional Ablation Tables 21
I. Additional Mechanistic Diagnostics 21
J. Paired Qualitative Comparisons 23
K. Failure Cases 25
L. Limitations and Future Work 26
M. Reproducibility Checklist 26

A Extended Related Work

A.1 Pixel-space generation

Diffusion and score-based models learn to reverse a gradual noising process and provide the iterative foundation of modern image synthesis (Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021); rectified flow turns the same idea into a straight flow from noise to data (Liu et al., 2022). Latent diffusion moved this process into the latent space of a separately trained autoencoder (Esser et al., 2021; Rombach et al., 2022), and better autoencoders keep improving this line (Zheng et al., 2025). DiT, U-ViT, and SiT showed that plain Transformers scale reliably in this space (Peebles & Xie, 2023; Bao et al., 2023; Ma et al., 2024). Pixel-space generation removes the autoencoder entirely, but the model must then process the full-resolution state. Simple Diffusion pays this cost with a scaled, downsampling U-Net (Hoogeboom et al., 2023), and Hourglass DiT downsamples tokens inside a single network (Crowson et al., 2024). Cascaded diffusion and PixelFlow instead split generation across resolutions, so a coarse model handles global structure and later stages add detail (Ho et al., 2022; Chen et al., 2025). JiT showed that a large-patch Transformer with a plain recipe matches latent baselines (Li & He, 2026), and PixNerd decodes patch tokens into pixels with a neural field (Wang et al., 2025). A parallel line assigns detail to extra components: PixelDiT adds a pixel-level DiT on top of a patch-level one (Yu et al., 2026), DiP adds a patch-detailer head (Chen et al., 2026), and HyperDiT connects coarse and fine flows with cross-attention connectors (He et al., 2026). SpecSlot builds on JiT's recipe, but instead of separating the global and local roles into extra stages, heads, or flows, it assigns both roles to indexed slots inside a single backbone.

A.2 Frequency structure and transform-domain generation

Frequency structure has entered generative modeling at four points. First, the state or forward process: inverse heat dissipation replaces noise with a DCT-based dissipation operator (Rissanen et al., 2023), wavelet models generate directly in wavelet space (Guth et al., 2022; Phung et al., 2023), sparse transform codes compress the generated representation (Nash et al., 2021), and DCTdiff makes the DCT coefficients themselves the generation state (Ning et al., 2025). Second, the generation order or conditioning: VAR generates scale by scale (Tian et al., 2024), NFIG orders generation by frequency band (Huang et al., 2025), Spectral Forcing filters the noisy input by a frequency schedule (Fan et al., 2026), Latent Forcing reorders the trajectory with a semantic auxiliary stream (Baade et al., 2026), and Frequency-Forcing guides generation with a learnable wavelet stream (Du, 2026). Third, the loss: focal frequency loss reweights spectral errors (Jiang et al., 2021). Fourth, analysis: Fourier studies characterize the spectral behavior of diffusion without changing the model (Falck et al., 2025). Two recent denoisers move frequency roles into the architecture itself: DiMSUM alternates spatial and wavelet-subband processing in one model (Phung et al., 2025), and DeCo assigns low frequencies to a DiT and high frequencies to a separate pixel decoder (Ma et al., 2026). SpecSlot differs in kind: the state, target, and loss remain in pixel space, and frequency enters only as fixed, disjoint DCT groups assigned to output slots—not a transformed state, an input filter, an auxiliary stream, or a second component.

A.3 Visual tokenization and allocation

Vision Transformers tokenize an image into a grid of spatial patches (Dosovitskiy et al., 2021), and NaViT extends this grid to variable resolutions (Dehghani et al., 2023). Efficiency work treats these spatial tokens as the unit to be saved: ToMe merges similar tokens (Bolya & Hoffman, 2023), SparseDiT prunes them (Chang et al., 2025), DyDiT++ reschedules computation across denoising time and space (Zhao et al., 2026), variable-length tokenization learns a global merge (Lee & Hong, 2026), and FlexiDiT spends fewer tokens at easier steps (Anagnostidis et al., 2025). In all of these methods, however, the spatial patch remains the only allocation unit. SpecSlot adds a second, spectral axis: each of W anchors owns K slots with disjoint coefficient groups, so a fixed $N{=}WK$ budget splits explicitly between anchor density and spectral subdivision, and both are directly measurable.

A.4 Global-local and sparse communication

Long token sequences are handled by restricting attention: Swin uses local windows with shifted boundaries (Liu et al., 2021), Longformer combines local windows with a few global tokens (Beltagy et al., 2020), and Perceiver routes everything through a small latent bottleneck (Jaegle et al., 2021). In pixel-space generation, RINs exchange information through learned interface states (Jabri et al., 2023), trainable sparse attention learns its pattern from data (Zhou et al., 2026), and PixSGR chooses content-dependent interactions from coarse-scale attention (You et al., 2026). These topologies are implicit or learned, whereas SpecSlot's graph is fixed by construction: one low-frequency slot per anchor attends globally while all slots interact locally, with no learned router.

B Implementation Contract

Table 4 records the implementation contracts used by the final SpecSlot models.

The input mixer applies a residual depthwise 3×3 convolution to the anchor grid, $\widetilde{u} = u + W_{\mathrm{DW}} * u$ with W_{DW} zero-initialized, and the spatial correction head is likewise zero-initialized. Their constructors preserve the baseline CPU random-number state, so enabling these branches does not shift the initialization of shared parameters.

Each Transformer block uses pre-RMSNorm with $\epsilon = 10^{-6}$ and RMS-normalized queries and keys. Both contracts use a SwiGLU width of 3,072 (ratio 4.0), as listed in Table 4. Timestep conditioning uses a sinusoidal 256-dimensional embedding followed by Linear–SiLU–Linear layers; class conditioning uses a 1,000-class embedding plus a null class and 32 context states. In each block, the summed condition is mapped by SiLU–Linear$(d, 6d)$ to shift, scale, and AdaLN branch

Table 4: Implementation contract for the final SpecSlot models. The ImageNet-256 resource rows are measured on one H800 with BF16 whole-network TorchInductor at profiler batch 128. The ImageNet-512 GFLOPs and throughput are the corresponding main-table measurements; its peak-memory row is the logged training peak per GPU at batch 128.

ImageNet-256	
Quantity	Value
Image side H	256
Input cell / stride s	16 / 16
Spatial anchors W	256
Slots per anchor K	4
Image slots $N = WK$	1,024
Hidden / FFN width / heads / blocks	768 / 3,072 / 12 / 12
Evidence width B	1,536
Input mixer	zero-initialized residual depthwise 3×3
Spectral-base window / stride	32 / 16
Spatial correction tiles / anchor stride	2×2 / 16
Routing	$(G, G, L, L) \times 3$
Class-context slots	32
Input DCT / frequency gate	none / none
Prediction / loss	image-shaped prediction / pixel velocity MSE
Trainable parameters	138.025728M
Profiler device / precision	H800 / BF16
Conditional forward GFLOPs	129.20
Conditional peak memory (batch 128)	5.56 GB
Heun-50 peak memory (batch 128)	5.84 GB
Heun-50 throughput (batch 128)	9.06 img/s
ImageNet-512	
Quantity	Value
Image side H	512
Input cell / stride s	32 / 32
Spatial anchors W	256
Slots per anchor K	4
Image slots $N = WK$	1,024
Hidden / FFN width / heads / blocks	768 / 3,072 / 12 / 12
Evidence width B	4,096
Input mixer	zero-initialized residual depthwise 3×3
Spectral-base window / stride	64 / 32
Spatial correction tiles / anchor stride	2×2 / 32
Routing	$(G, G, L, L) \times 3$
Class-context slots	32
Input DCT / frequency gate	none / none
Prediction / loss	image-shaped prediction / pixel velocity MSE
Training epochs / GPUs	120 / 8
Batch per GPU / accumulation	128 / 1
Reported training seeds	$\{0, 1, 2, 3, 4\}$
Optimizer / schedule	AdamW / constant after warmup
Actual learning rate	2×10^{-4}
Trainable parameters	160.859904M
Profiler device / precision	H800 / BF16
Conditional forward GFLOPs	135.64
Training peak memory (batch 128 / GPU)	17.39 GB / GPU
Heun-50 throughput (batch 128)	12.69 img/s

gates for the attention and feed-forward branches. The final normalization is RMSNorm with a SiLU–Linear($d, 2d$) shift/scale projection.

For registered allocations, the spatial head uses deterministic row-major near-square tilings of the anchor cell (K2: 1×2, K4: 2×2, K8: 2×4, and K32: 4×8); the main model uses the K4 quadrants.

Formally, $r_{\mathrm{sp}} = \sum_{i,k} \mathrm{Scatter}_{i,k}\Big(P_{\mathrm{sp},k} h^{\mathrm{out}}_{i,k} + b_{\mathrm{sp},k}\Big)$, where $\mathrm{Scatter}_{i,k}$ writes the output of slot k into tile k of anchor i.

All aligned ImageNet-256 runs train for 120 epochs on eight GPUs with 64 images per GPU (effective batch 512). We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, zero weight decay, and no gradient clipping. The base learning rate is 5×10^{-5} and scales to 10^{-4} at batch 512 under the code's reference batch 256; the schedule is constant after five warmup epochs. EMA decays are $(0.9999, 0.9996)$, and images are scaled to $[-1, 1]$. For the shared rectified-flow objective, $g \sim \mathcal{N}(-0.8, 0.8^2)$, $t = \sigma(g)$, the noise scale is one, $t_\epsilon = 0.05$, the prediction parameterization is x, and class-label dropout is 0.1.

The ImageNet-512 contract uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, zero weight decay, and no gradient clipping; EMA decays are $(0.9999, 0.9996)$, and the learning-rate schedule is constant after warmup. It trains for 120 epochs on eight GPUs with 128 images per GPU, no gradient accumulation (effective batch 1,024), BF16 autocast, activation checkpointing, pinned memory, spawn-context data loading, and 12 data-loader workers. With the command-line base rate 5×10^{-5}, the batch scaling rule gives an actual learning rate of 2×10^{-4}; warmup lasts five epochs. The final W256/K4/B4096 row uses independently trained seeds $\{0, 1, 2, 3, 4\}$ for the reported mean, EMA1 for evaluation, the legacy-uniform 50-step Heun sampler with generation batch 512, and the selected CFG 8.25; the seed-0 reference run uses generation seed 0. The reported model's spatial evidence projection uses 32-pixel cells; the lapped spectral base uses 64-pixel windows at stride 32, and the hybrid output uses the frequency IDCT/overlap-add branch plus the zero-initialized anchor-aligned spatial correction over the 32-pixel anchor. The profiler values are tied to the stated batch and compiled kernel shapes, so throughput is an empirical hardware measure rather than a direct pixel-count scaling metric.

For one image, timestep, and class label, the implementation executes:

1. Form $z_t = tx + (1 - t)\epsilon$ in pixel space.
2. Project non-overlapping cells to the anchor grid, apply the residual depthwise mixer, and expand each anchor into its K slots.
3. Add separate anchor and slot positions and apply the GGLL schedule.
4. Project every slot to its radial DCT group, assemble lapped windows, and compute the spectral base by inverse DCT and normalized overlap-add.
5. Project the slots at each anchor to anchor-aligned spatial quadrants and add the resulting residual to the spectral base.
6. Convert the sum to v_θ with Equation 2 and optimize Equation 3.

C Lapped Spectral-Base Synthesis

The decoder uses the synthesis half of a matched lapped DCT frame. We describe the corresponding analysis only to state its reconstruction property.

Let P reflection-pad an image by $s/2$ pixels, R_i extract an $L \times L$ window with stride $s = L/2$, g be the separable square-root periodic Hann window, and $\mathcal{D}$ be the orthonormal two-dimensional DCT-II. The matched analysis coefficients are

$$c_i(x) = \mathcal{D}\left(g \odot R_i P x\right). \tag{12}$$

For arbitrary coefficient windows c_i, define

$$S(\{c_i\}) = Q\left[\Lambda^\dagger \odot \sum_i R_i^\top \left(g \odot \mathcal{D}^\top c_i\right)\right], \qquad \Lambda = \sum_i R_i^\top (g^2), \tag{13}$$

where Q crops the central image and $\Lambda^\dagger$ is reciprocal on nonzero folded energy. With the stated padding, $QP = I$. Substituting Equation 12 into Equation 13 yields $S(\{c_i(x)\}) = x$ up to floating-

Table 5: Coarse CFG sweep for JiT-B/8 on ImageNet-256. All points use the same epoch-120 EMA1 checkpoint and evaluation protocol.

CFG	FID-50K ↓	IS ↑
1.00	29.4893	53.4115
2.00	7.3136	154.0057
2.90	**4.3579**	225.1920
4.00	5.0462	281.4794
5.00	6.3617	312.9500
6.00	7.5583	334.1862
7.00	8.5149	349.8076
8.00	9.2679	359.5481

Table 6: Fine CFG sweep for JiT-B/8 on ImageNet-256 around the coarse-grid optimum. The coarse point 2.90 is included because it attains the best FID overall; other coarse points are not duplicated.

CFG	FID-50K ↓	IS ↑
2.25	5.7930	176.5661
2.50	4.9153	197.2529
2.75	4.4765	215.3064
2.90	**4.3579**	225.1920
3.25	4.3611	246.4003
3.50	4.5283	259.3960
3.75	4.7617	270.9640

point error. Predicted coefficient windows need not form a unique analysis representation because the lapped frame is redundant; the training loss constrains their synthesized pixel field.

Radial order is defined by $\sqrt{u^2 + v^2}$ on integer DCT indices, with ties resolved by $u + v$ and then row-major index. We require $K \mid L^2$, so each spectral slot owns $E = L^2/K$ coefficients per channel. Concatenating slots at an anchor recovers radial order before scattering into the two-dimensional window.

D Evaluation Protocol

We use the common Heun grid and classifier-free guidance interval specified in Section 4.1. The same class ordering and noise draw are used for compared checkpoints. CFG selection, image count, and profiler settings follow the shared protocol in that section; every quality row pairs FID and IS at one CFG.

E Classifier-free Guidance Sweep

We use JiT-B/8 on ImageNet-256 as a concrete example of the guidance-scale search. Every point evaluates the same epoch-120 EMA1 checkpoint on 50K images with a per-rank generation batch size of 512, the legacy-uniform Heun sampler with 50 steps, generation seed 0, and guidance interval $[0.1, 1.0]$; only CFG changes. We first run a coarse grid at CFG $\in \{1.0, 2.0, 2.9, 4.0, 5.0, 6.0, 7.0, 8.0\}$ (Table 5). We then refine the low-FID region with a local ± 1 window and increments of 0.25. Table 6 lists the fine-grid evaluations retained in the archive; the coarse optimum 2.90 is also the fine optimum. We use CFG 2.9 for JiT in the main comparisons, rather than 3.0, to match the scale used by Spectral Forcing (Fan et al., 2026).

F Sampling-step Latency Scaling

The main tables use 50 Heun intervals for quality evaluation. We additionally measured sampling latency at 20, 30, and 50 intervals for the trained $W256/K4$ model and two routing controls. The measurement uses one H800, BF16, a fixed sampling batch of 512, one warm-up repeat, and three

Table 7: Fixed-batch sampling latency on one H800. The three routes use the same legacy-uniform Heun sampler, BF16 precision, sampling batch 512, one warm-up repeat, and three timed repeats. "Evals/image" is the actual network-call count under classifier-free guidance; "TFLOPs/image" is the analytic model forward count. GGLL is the SpecSlot route. Dense-global-CM has matched per-evaluation FLOPs, while Full-global is a wider control. CFG is 4.0, 4.25, and 3.5 for GGLL, Dense-global-CM, and Full-global, respectively.

Route	Heun	Evals/image	GFLOPs/eval	TFLOPs/image	ms/image	img/s
GGLL	20	73	128.59	9.39	47.56	21.02
Dense-global-CM	20	73	128.96	9.41	69.89	14.31
Full-global	20	73	225.22	16.44	80.99	12.35
GGLL	30	113	128.59	14.53	73.09	13.68
Dense-global-CM	30	113	128.96	14.57	109.20	9.16
Full-global	30	113	225.22	25.45	124.83	8.01
GGLL	50	187	128.59	24.05	121.88	8.20
Dense-global-CM	50	187	128.96	24.11	179.95	5.56
Full-global	50	187	225.22	42.12	208.60	4.79

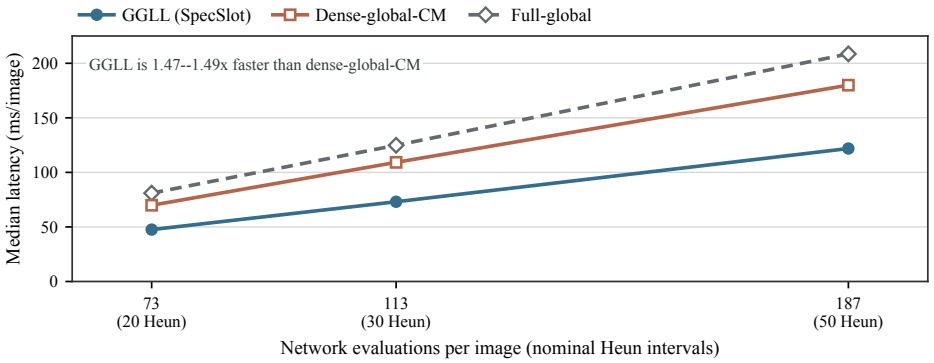

Figure 5: Sampling latency as the evaluation budget increases. The measurements use one H800, BF16, a fixed batch of 512, and three timed repeats. The x-axis gives the profiler-observed network evaluations per image; the nominal Heun intervals appear in parentheses. GGLL remains below the compute-matched dense-global control at all three budgets; the exact medians are in Table 7.

timed repeats. The sampler is the same legacy-uniform Heun implementation used for the quality results. Under the profiler's classifier-free-guidance accounting, the corresponding network-evaluation counts are 73, 113, and 187 per image. We report both quantities to avoid confusing an integration budget with a network-evaluation count. The route-specific CFG values are 4.0 for GGLL, 4.25 for Dense-global-CM, and 3.5 for Full-global.

Table 7 reports the median model-sampling latency. The compute-matched dense-global control has nearly the same per-evaluation FLOPs as GGLL, so it is the primary routing comparison; the full-global row uses a wider network and is included as a cost reference. At every tested budget, GGLL is faster than both controls. Relative to Dense-global-CM, its throughput is 1.47–1.49$\times$ higher across the three budgets; relative to full-global, it is 1.70–1.71$\times$ higher. Figure 5 shows the same scaling in latency.

These measurements isolate fixed-batch model-sampling latency. The run uses per-network compilation, but not whole-network compilation, fused local attention, or batched CFG; its throughput is therefore not the peak-throughput contract used in the main tables. FID and IS are not evaluated at reduced step counts, and these are not end-to-end 50K wall-clock timings: startup, compilation, image writing, and metric calculation are outside this profiler measurement. The quality numbers in the main tables therefore remain the 50-step evaluations defined in Section 4.1.

Table 8: Per-training-seed ImageNet-256 evaluation on 50K generated images per checkpoint. FID and IS are reported for training seeds 12345–12349.

Allocation	Training seed	FID-50K ↓	IS ↑
$W256/K2$	12345	7.2191	214.2811
	12346	7.0054	237.4801
	12347	7.6025	206.7721
	12348	7.3332	226.7448
	12349	7.1983	212.1924
$W256/K4$	12345	6.4266	232.6772
	12346	6.6197	212.0868
	12347	6.4503	224.0468
	12348	6.8389	201.4354
	12349	6.6250	205.7543

Table 9: Across-training-seed FID statistics for the same ImageNet-256 runs. SD is the sample standard deviation, SE is $\mathrm{SD}/\sqrt{n}$, and the interval is the two-sided 95% Student-t interval with $n-1$ degrees of freedom. The interval quantifies variation across independently trained checkpoints; it is not a confidence interval over individual generated images.

Allocation	n	Mean FID	SD	SE	95% t interval	Median	Min–Max
$W256/K2$	5	7.2717	0.2192	0.0981	[6.9995, 7.5439]	7.2191	[7.0054, 7.6025]
$W256/K4$	5	6.5921	0.1660	0.0742	[6.3860, 6.7982]	6.6197	[6.4266, 6.8389]

G FIVE-SEED IMAGENET-256 RESULTS

The main tables report means over independently trained seeds. This section exposes the individual runs and the across-seed uncertainty for the two $W256$ allocations used in the ImageNet-256 comparison. Table 8 lists each training seed, while Tables 9 and 10 report FID and Inception Score statistics. The training seed identifies the independently trained checkpoint; the generation seed is fixed to zero for every row. Evaluations use epoch-120 EMA1 weights, 50K generated images, the legacy-uniform 50-step Heun sampler, and the same class ordering and noise draws. The selected CFG is 5.0 for K2 and 4.0 for K4, as in Table 2; the K2 runs use evidence width $B = 512$.

All summary statistics are computed from the four-decimal per-seed values in Table 8 and rounded to four decimals only after computation.

The aligned five-seed mean trajectories in Figure 6 complement the endpoint statistics above.

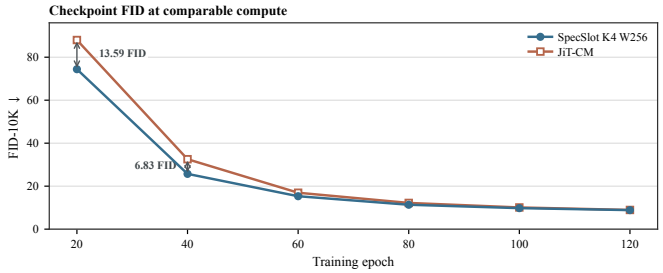

Figure 6: Five-seed mean FID-10K at epochs 20–120 under the aligned protocol (fixed CFGs of 4.0/2.9 for SpecSlot/JiT-CM); a convergence diagnostic, not the 50K endpoint evaluation.

Table 10: Across-training-seed Inception Score statistics for the same ImageNet-256 runs. SD, SE, and the 95% Student-t interval are defined as in Table 9.

Allocation	n	Mean IS	SD	SE	95% t interval	Median	Min–Max
$W256/K2$	5	219.4941	12.4339	5.5606	[204.0554, 234.9328]	214.2811	[206.7721, 237.4801]
$W256/K4$	5	215.2001	12.9624	5.7970	[199.1051, 231.2951]	212.0868	[201.4354, 232.6772]

H Additional Ablation Tables

The remaining component controls vary one part of the final contract at a time. Table 11 compares evidence-projection widths and finds the lowest FID at $B = 1536$. Table 12 shows the reduction from the residual depthwise input mixer, while Table 13 shows the corresponding benefit of 50% overlapping synthesis windows.

Table 11: Evidence-width ablation at the fixed anchor–slot contract.

Evidence width B	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
$B = 1024$	6.77	221.58	**128.19**	**9.18**
$B = 1536$	**6.59**	215.20	129.20	9.06
$B = 2048$	6.64	**224.80**	129.87	9.04

Table 12: Input-mixing ablation at the fixed anchor–slot contract.

Input mixer	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
Identity	6.83	**220.38**	**129.20**	**9.06**
Residual depthwise mixer	**6.59**	215.20	**129.20**	**9.06**

I Additional Mechanistic Diagnostics

This appendix collects measurements that probe how SpecSlot uses its explicit roles. They are complementary to the trained comparisons in Section 4.2: unless stated otherwise, a probe loads a checkpoint, intervenes on its states, and does not update any parameter. We therefore use these results to characterize representation and communication, not to claim an additional end-to-end quality gain. The sample count and checkpoint contract are stated with each probe because some diagnostics use a smaller allocation or a reduced ImageNet split. For the route probe, the intervention is made inside the shared Transformer after slot formation and before the output head, so its interpretation concerns the spectral-slot/GGLL submodule rather than a particular output basis. The other probes state their representation and checkpoint contract explicitly; the branch and gradient probes are specific to the dual-basis decoder and are labeled as such. The compact route curves in Figure 4(e,f) are reported as a mechanism diagnostic and are not used as end-to-end quality rows.

I.1 Evidence timing and route interventions

We first measure an empirical data-to-noise ratio (DNR) for each radial coefficient group in the lapped frame. For group k, $\mathrm{DNR}_k(t) = t^2\mathbb{E}\|A_k x\|^2/[(1-t)^2\mathbb{E}\|A_k\epsilon\|^2]$, where A_k denotes the matched lapped-frame analysis restricted to that group. The crossing time $t^\star$ is the first flow time at which the group reaches DNR ≥ 1; it is a descriptive measure of when the noisy input contains more signal than noise, not a learned gate. Table 14 reports the crossings for the paired W256/K8, W256/K4, and W1024/K2 checkpoints.

The frontiers move when W or K changes: the low group becomes usable earlier, while progressively finer groups become usable later. This is consistent with assigning the early cross-anchor path to the lowest radial role. In the matched teacher-forced route intervention, removing the global carrier path primarily increases the low-group and early error; removing local mixing primarily increases detail-group error later in the trajectory. For example, the mid-time log-MSE penalties for

Table 13: Lapped-window ablation at the fixed anchor–slot contract.

Output window	FID ↓	IS ↑	GFLOPs ↓	img/s ↑
Non-overlapping	6.78	211.23	**129.19**	**9.13**
50% overlapping	**6.59**	**215.20**	129.20	9.06

Table 14: Empirical DNR frontiers from the lapped-frame probe. Entries list the crossing times $t^\star$ in increasing radial-group order.

Allocation	$t^\star$ for successive radial groups
$W256/K8$	$\{0.382, 0.836, 0.880, 0.904, 0.922, 0.933, 0.943, 0.959\}$
$W256/K4$	$\{0.468, 0.891, 0.928, 0.951\}$
$W1024/K2$	$\{0.549, 0.936\}$

the W256/K8 and W256/K4 probes are approximately $0.14/1.66$ and $0.20/2.14$ (global/local). The on-policy intervention has the same ordering: global removal changes the early coarse field (low-frequency change about 0.98, 0.99, and 1.05 for the three allocations), while local removal continues to change mid/high frequency content in the middle and late segments. These measurements are consistent with the role assignment used by GGLL.

I.2 Which frequency can serve as the carrier?

Table 15 summarizes the carrier-identity experiment, which keeps the GGLL graph fixed and changes only the physical radial group exposed to the global path. It uses the W256/K8 contract on the 100-class, 100-images-per-class subset for 5,000 updates and the same raw-CFG/Heun diagnostic protocol. The lowest group (b0) is visible much earlier and has the lowest final regression loss.

The result supports an evidence-aligned carrier: the useful criterion is early observability and stable recovery. No FID or IS conclusion is drawn from this reduced training probe.

A trained counterpart of this probe, which moves only the slot with cross-anchor access in the full ImageNet-256 model at fixed $W256/K4/B1536$, is reported in Section 4.6.2 (Table 3).

I.3 Allocation and effective use of additional slots

To separate single-step fitting from the use of extra role slots, a paired fixed-$W = 256$, fixed-$B = 1536$ analysis compares $K \in \{2, 4, 8\}$. The models share the input, lapped output, and GGLL interfaces used by the registered K-rate diagnostic. For reproducibility, the effective-slot value is the entropy effective rank of each anchor's slot Gram matrix, divided by K; the effective-local-key value is $\exp(H_q)/K$ averaged over local queries, with $H_q = -\sum_j p_{qj} \log p_{qj}$; and detail-to-carrier is the mean local attention probability from detail queries to the carrier key. Table 16 shows that although the teacher-forced velocity error changes little, the fraction of effective slots and the fraction of local keys decline as the within-anchor sequence grows.

Thus adding slots can slightly improve teacher-forced fitting without providing the same number of independent, strongly coupled directions. This is consistent with an allocation sweet spot: K controls how finely the coefficient ownership is divided, but it is not a promise that every added slot contributes an equal amount of information. The analysis does not by itself identify a unique cause of free-running FID differences.

I.4 The two output bases are both used

The main text reports branch energy, frequency-band attribution, and branch scaling for the hybrid decoder. We additionally measured gradients on the same W256/K4 checkpoint using 16 paired validation images at each of the six teacher-forced times. Figure 7 shows that the frequency head receives the larger direct output gradient, while the spatial correction still has a finite gradient. In shared Transformer layers, the branch-gradient cosine takes both signs and remains close to zero on

Table 15: Carrier-identity probe. Visibility is the archived probe's mean coefficient-level input-gate response at $t = 0.25$; this reduced diagnostic is separate from the final input-gate-free contract. The last column is the cycle-3 global-update RMS divided by the carrier seed RMS.

Carrier	final RF loss	visibility	recon. MSE ($t = 0.90$)	update/seed
b0 (lowest)	.0835	.891	.00443	1.31
b3 (middle)	.0894	.00087	.00519	1.14
b7 (highest)	.0891	.00014	.00450	2.45

Table 16: K-rate diagnostics at fixed W and evidence width. Values are means over paired teacher states; they describe utilization, not a new quality row.

K	velocity MSE	effective slots/K	effective local keys	detail $\rightarrow$ carrier
2	.10670	.803	.729	.330
4	.10549	.620	.682	.312
8	.10497	.495	.621	.275

average; the probe therefore does not show a persistent large conflict between the two paths. The spatial shared-layer gradient becomes smaller toward the clean end, so the figure is not evidence that both branches receive equal optimization signal. It is a diagnostic supporting the more limited statement established in the main text: the spatial correction does not replace the spectral base, and the two terms are functionally complementary in the evaluated checkpoint.

J Paired Qualitative Comparisons

Figure 8 compares selected ImageNet-256 generations from JiT-CM and SpecSlot W256/K4 under the same class and noise draw. In these pairs, SpecSlot renders class-specific fine structure more coherently: feather boundaries are clearer for brambling, chickadee, and rooster; the cornet has more regular tubing and bell geometry; and the boathouse preserves sharper roofline and reflection structure. It also reduces visible local distortions in the anemone fish, banister, and barrel examples, where contours or repeated parts are less coherent in the JiT-CM samples. The hay example shows a more regular field-scale texture. These selected examples illustrate qualitative differences and are not used to estimate their frequency over the full sample distribution.

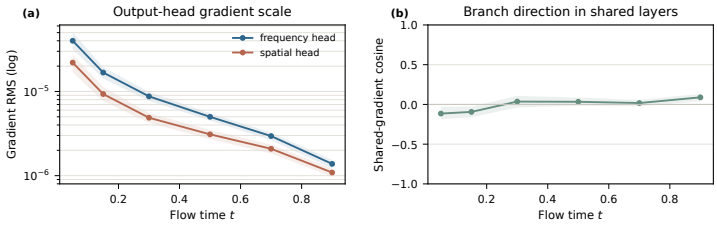

Figure 7: Shared-gradient diagnostic for the hybrid output. (a) RMS gradients of the two output heads; (b) cosine between the frequency and spatial branch gradients in shared Transformer layers. Each point averages the 16-image probe at a teacher-forced flow time. This fixed-checkpoint measurement is descriptive and does not replace a retrained ablation.

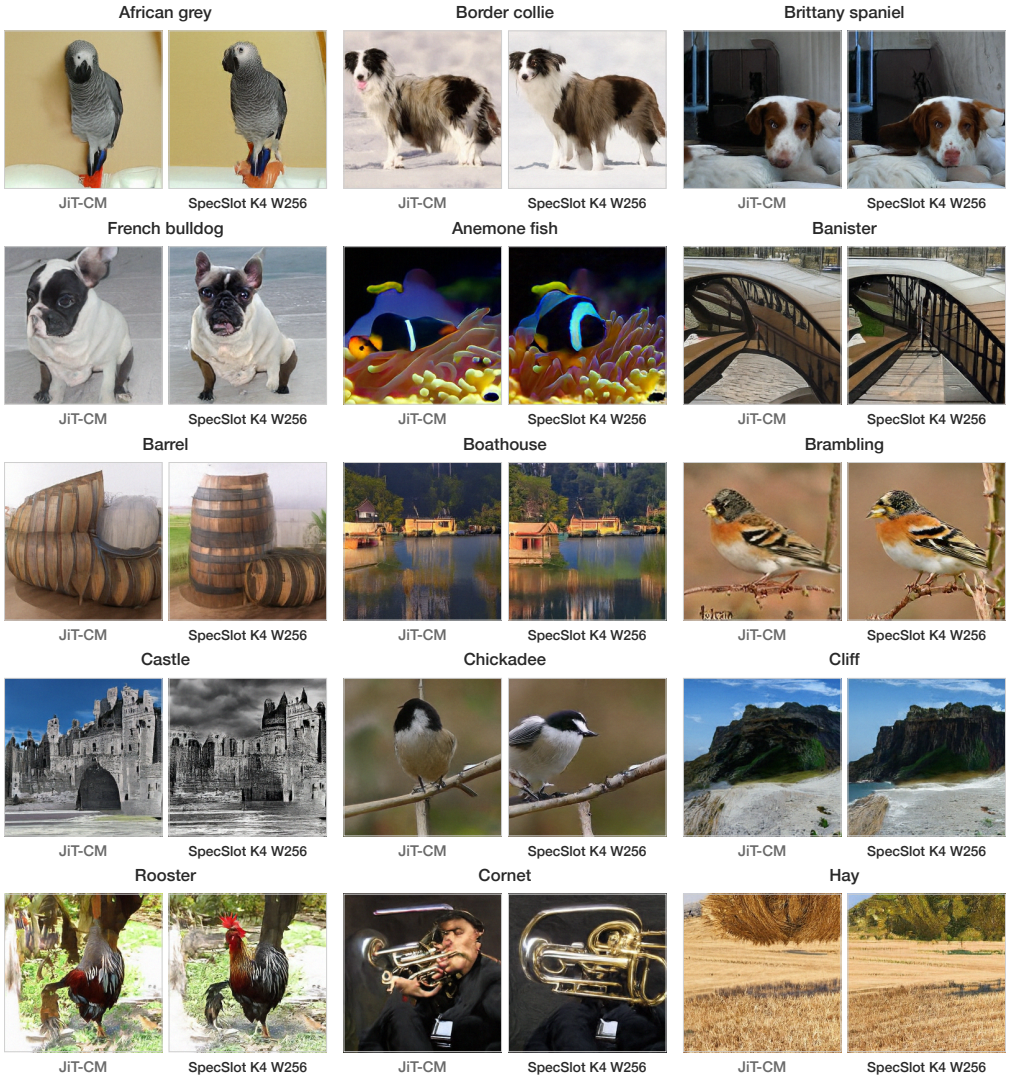

Figure 8: Selected paired generations on ImageNet-256. Within each pair, JiT-CM is shown on the left and SpecSlot W256/K4 on the right; both use the same class and noise draw. The examples highlight differences in fine structure, object geometry, and local distortion.

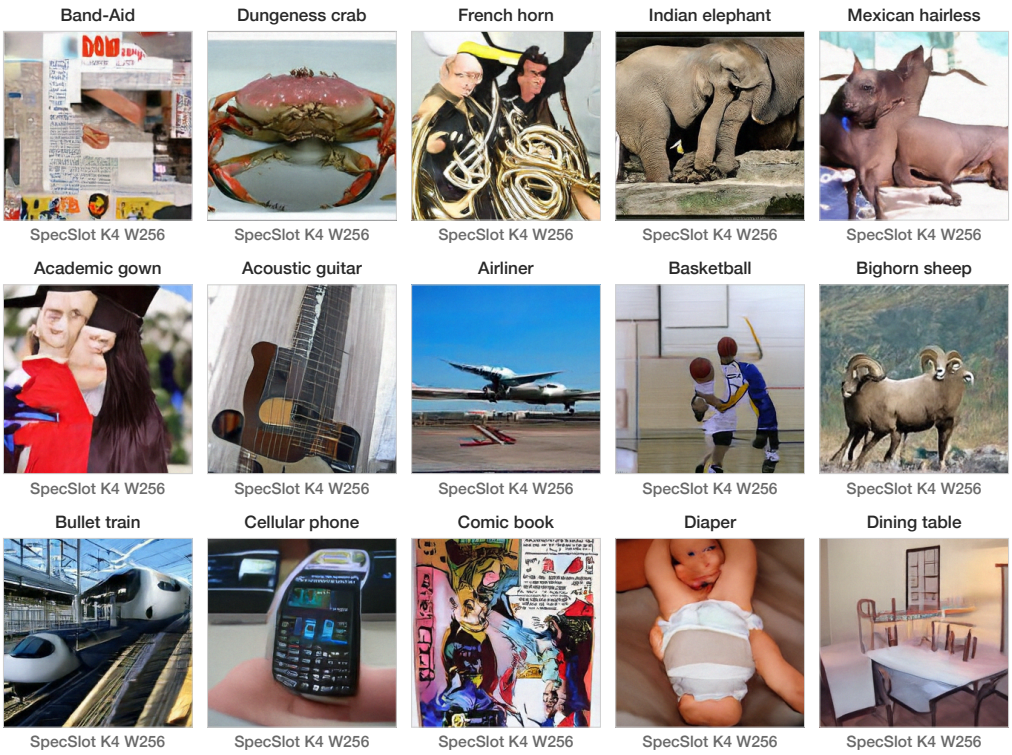

Figure 9: Selected failure cases from SpecSlot W256/K4 on ImageNet-256. The corresponding matched JiT-CM outputs also contain visible structural errors and are omitted here for space. The examples cover malformed human features, merged animal instances, distorted instruments and vehicles, and incoherent articulated or text-bearing objects.

K Failure Cases

Figure 9 collects selected SpecSlot samples for which the matched JiT-CM generation, using the same class and noise draw, also contains substantial structural errors. SpecSlot does not consistently resolve these shared failures. Human-centered classes such as academic gown and diaper contain malformed faces or fused body parts. The French horn and acoustic guitar examples lose the geometry of intertwined tubing, strings, and instrument boundaries. Multi-instance animals, including Indian elephant, Mexican hairless, and bighorn sheep, exhibit merged bodies or duplicated heads and limbs; the Dungeness crab also has fused limb structure. Airliner and bullet train show inconsistent part attachment and perspective, while the cellular phone has an irregular screen and keypad. The Band-Aid and comic-book examples retain illegible text; basketball and dining table show incoherent human–object or articulated geometry. These selected examples characterize visible failure modes rather than their frequency over the full sample distribution.

L Limitations and Future Work

Our evaluation is scoped to class-conditional ImageNet at two resolutions under a fixed recipe per resolution. The most informative extensions are text-to-image generation and pixel-space world models, such as those in VLA systems: removing the autoencoder avoids a learned-tokenizer bottleneck, but full-resolution attention dominates the cost—precisely the regime in which an allocation axis between detail and communication should matter most.

The JiT-CM control is a fair same-recipe JiT implementation at comparable compute, but it is not necessarily the best JiT configuration at that budget: matching FLOPs required narrowing its width. A rebalanced configuration at the same compute could close part of the reported gaps; we keep JiT-CM because depth and protocol parity isolate the tokenization change.

The two allocation knobs entangle several physical factors: changing W also changes the input cell size, the DCT window size, the per-slot coefficient share, the local attention group size, and the projection shapes. This bounds single-knob attribution, but the joint (W, K) reallocation we report is how the design is meant to be used—a fixed budget redistributed between the two axes. The K-rate probe in Appendix I.3 and the component controls in Appendix H isolate individual pieces of this space; a fully factorized sweep remains future work.

Finally, our targets are operating points on the quality–compute frontier rather than the absolute FID frontier, and we do not claim state-of-the-art quality: the gains concentrate in the mid-compute regime, where the budget is tight enough for allocation to matter. Whether the advantage persists at larger models and longer training, and whether the coefficient assignment should be learned rather than fixed for data whose spectra differ from natural images, remain open.

M Reproducibility Checklist

For every reported row we archive the checkpoint contract, model arguments, training seed, EMA choice, sampler grid, CFG interval and scale, generation seed, image count, reference-statistics path, and profiler backend. A row enters a cross-model table only after its log confirms the intended W, K, B, input/output representation, input mixer, lapping, routing, and prediction parameterization.