COFITOK: COARSE-TO-FINE DENOISING TOKENS FOR PIXEL-SPACE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion networks predict an image-sized noise field at each evaluation. We introduce CoFiTok, which organizes this field as an ordered sequence of continuous token contributions. Each token is synthesized independently through a shallow bias-free map, giving a zero token an exact zero-component meaning. Accumulating the components yields partial clean-image estimates at a fixed noisy state and diffusion time. Prefix and component losses train these estimates along a prescribed coarse-to-fine denoising path. We relate pixel-space prefix errors to noise-space errors and use fixed-component permutations to evaluate ordering while preserving the full prediction. Across eight dataset settings, CoFiTok consistently improves path alignment. Two-seed studies with 20,000 training steps reduce path AUC from 1.0098 to 0.0378 on Tiny ImageNet and from 0.9984 to 0.0379 on ImageNet-64 HF, with endpoint-MSE increases of 3.4% and 5.0%. On ImageNet-256, the learned four-component order ranks first among all 24 permutations for both seeds. Ablations identify complementary effects of prefix and component supervision, alongside mixed endpoint and generation results. CoFiTok provides a structured interface for studying partial denoising within a diffusion prediction.

1 INTRODUCTION

Diffusion models generate images through repeated predictions of image-sized fields (Ho et al., 2020; Holl et al., 2026). Advances in network design and sampling have substantially improved visual quality (Nichol & Dhariwal, 2021; Karras et al., 2022). Within each network evaluation, the usual interface remains a single dense output, as shown in Figure 1a. Exposing useful partial contributions to this output would make the organization of a denoising prediction directly accessible: one could inspect intermediate estimates and study how their accumulation affects the final result.

Visual tokenization provides a useful starting point. TiTok compresses image content into compact sequences (Yu et al., 2024); ElasticTok and FlexTok support variable token budgets (Yan et al., 2025; Bachmann et al., 2025). Continuous autoregressive models connect sequence prediction with diffusion losses (Li et al., 2024; Gu et al., 2025; Gao & Shou, 2026). BAR learns visual bases and supervises prefix residual reconstruction (Tang et al., 2026), making the order of representation units part of the learning problem. These developments motivate applying structured sequences to the dense noise field predicted at a single diffusion time.

Two design questions arise in this setting. The first concerns supervision: an endpoint loss constrains the sum of all contributions, while opposing errors can cancel in that sum and remain large in earlier prefixes. The second concerns synthesis: contextual inputs or biases can produce image structure independently of the current token. A useful component interface therefore needs both a target for partial predictions and a precise relationship between each token and its synthesized contribution.

CoFiTok combines ordered noise factorization, restricted synthesis, and path supervision. A predictor maps the noisy image and diffusion time to continuous tokens. A separate shallow, bias-free branch maps each token to an image-sized component, ensuring $S_k(0) = 0$. Partial sums are converted to image estimates using the standard diffusion relation and trained against a prescribed coarse-to-fine denoising path. Figure 1b illustrates the resulting supervision, and Figure 2 shows the complete training architecture. The token sequence organizes contributions within one fixed-time prediction; successive sampler updates use new predictions at their respective noisy states.

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.

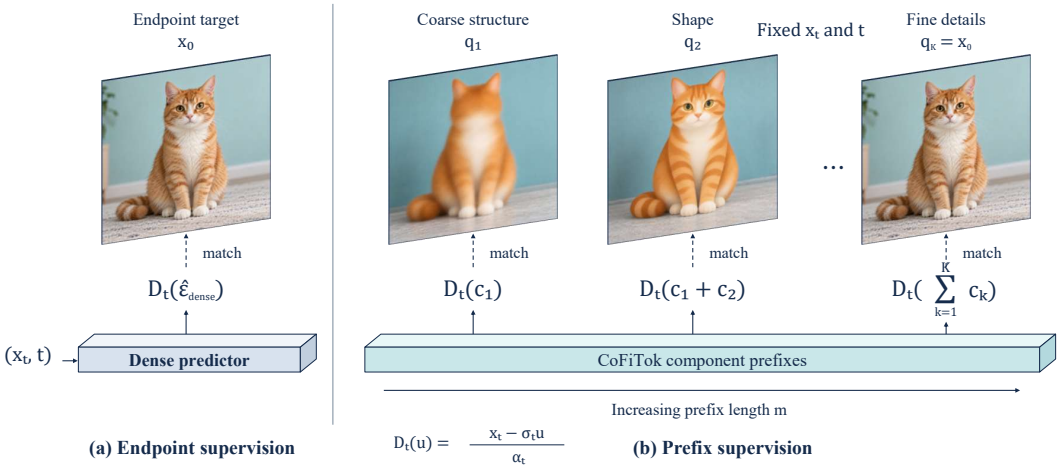

Figure 1: From endpoint supervision to ordered partial denoising. (a) A dense noise prediction produces a clean-image estimate supervised by x_0. (b) CoFiTok accumulates token contributions into prefixes $u_m = \sum_{k=1}^{m} c_k$ and supervises their converted estimates with intermediate targets q_m. The cat images illustrate these targets; the first and last show the same endpoint $q_K = x_0$. D_t is the conversion in Equation 5. Solid arrows denote computation and dashed arrows supervision. The noisy state, diffusion time, and resolution are fixed across prefixes.

Our experiments examine the resulting path–endpoint tradeoff across eight dataset settings. In the two-seed 20k studies, path AUC decreases by 96.3% and 96.2%, while endpoint MSE increases by 3.4% and 5.0%. Reordering four fixed components on ImageNet-256 preserves the endpoint and ranks the learned order first among all 24 permutations for both seeds. The contributions are:

- A fixed-time factorization of dense noise predictions into ordered continuous tokens with token-only, zero-preserving synthesis.
- Prefix and component supervision for partial denoising, together with an error identity and an endpoint-preserving order analysis.
- Multi-dataset comparisons and ablations of supervision, token feedback, and synthesis that characterize path quality, reconstruction, and generated-image distributions.

2 Background

Ordered visual representations. Discrete visual models predict codebook entries learned by a tokenizer (Van Den Oord et al., 2017); compact and adaptive representations vary the number of units used to reconstruct an image (Yu et al., 2024; Duggal et al., 2025; Yan et al., 2025; Bachmann et al., 2025). Matryoshka representations expose nested subsets (Kusupati et al., 2022; Cai et al., 2024), while sequence alignment, token folding, and spectral tokenization organize content for subsequent prediction (Wu et al., 2026; Li et al., 2025; Esteves et al., 2025). BAR learns a visual basis and encourages ordered prefix reconstruction (Tang et al., 2026). CoFiTok brings this prefix-learning perspective to the noise field of a diffusion predictor.

Diffusion and continuous prediction. MAR, DART, and D-AR connect continuous autoregressive prediction with denoising objectives (Li et al., 2024; Gu et al., 2025; Gao & Shou, 2026). Latent diffusion predicts in learned image coordinates (Rombach et al., 2022); progressive and nested diffusion models organize synthesis over multiple scales (Gu et al., 2024; Yang et al., 2025). Latent Forcing studies variable ordering along a diffusion trajectory (Baade et al., 2026), and Selftok connects diffusion with autoregressive visual tokens (Wang et al., 2025). Our setting is pixel-space noise prediction. For a clean image $x_0 \in \mathbb{R}^{C\times H\times W}$, the forward process and standard noise loss are

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \qquad \mathcal{L}_\epsilon = \mathbb{E}\left[\frac{\|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|_2^2}{CHW}\right], \tag{1}$$

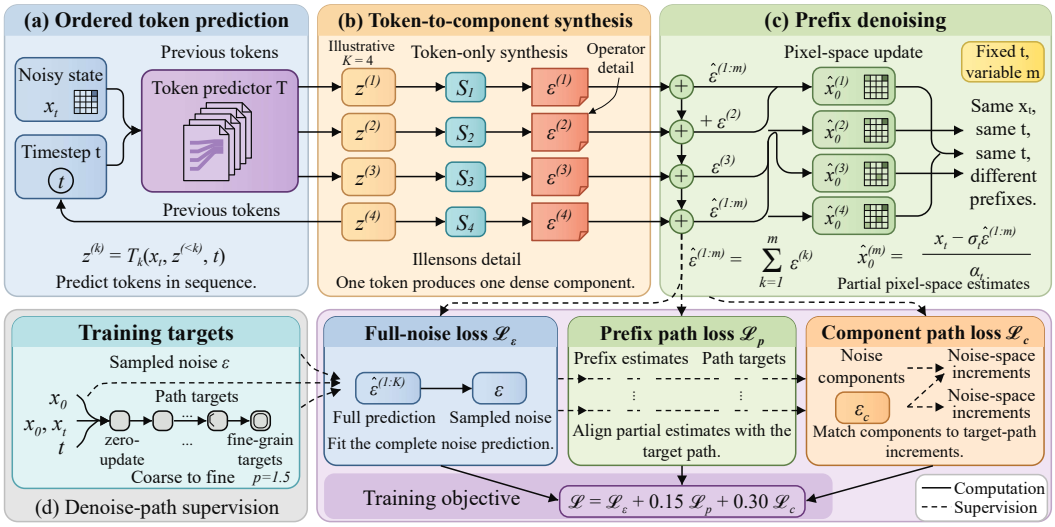

Figure 2: Overview of CoFiTok and its training objective. Ordered tokens are mapped to dense components and accumulated into prefix predictions at fixed x_t and t. Full-noise, prefix-path, and component-path losses supervise the complete and partial predictions. The figure uses $\epsilon^{(k)} = c_k$, $\hat{\epsilon}^{(1:m)} = u_m$, $\mathcal{L}_p = \mathcal{L}_{\text{prefix}}$, and $\mathcal{L}_c = \mathcal{L}_{\text{component}}$. Solid arrows denote computation; dashed arrows denote supervision. $K = 4$ is illustrative.

where α_t and σ_t are the signal and noise coefficients. The expectation is over images, diffusion times, and Gaussian noise. For $\alpha_t \neq 0$, the corresponding clean-image estimate is

$$\hat{x}_0 = \frac{x_t - \sigma_t \hat{\epsilon}_\theta(x_t, t)}{\alpha_t}. \tag{2}$$

CoFiTok applies Equation 2 to partial noise estimates at the same (x_t, t). A reverse sampler then uses the selected estimate to update the noisy state (Song et al., 2021; Karras et al., 2022).

3 Method

3.1 Ordered noise factorization

Let $\mathcal{X} = \mathbb{R}^{C\times H\times W}$ be the dense prediction space. At a fixed noisy state x_t and time t, CoFiTok predicts K continuous tokens $z^{(k)} \in \mathcal{Z}_k$. Each synthesis branch maps its token to an image-sized component:

$$c_k = S_k(z^{(k)}), \qquad S_k : \mathcal{Z}_k \longrightarrow \mathcal{X}, \qquad k = 1, \ldots, K. \tag{3}$$

All components share the same output coordinates and can contribute to the same pixel. Their cumulative sums define the partial and full noise predictions,

$$u_m = \sum_{k=1}^{m} c_k, \qquad u_0 = 0, \qquad \hat{\epsilon}_\theta(x_t, t) = u_K. \tag{4}$$

Applying Equation 2 to these sums gives a sequence of partial clean-image estimates,

$$\hat{x}_0^{(m)} = \frac{x_t - \sigma_t u_m}{\alpha_t}, \qquad m = 0, \ldots, K. \tag{5}$$

The sequence begins at $\hat{x}_0^{(0)} = x_t/\alpha_t$ and ends at the full prediction $\hat{x}_0^{(K)}$. The prefix index counts contributions while x_t, t, and the diffusion coefficients remain fixed. Figure 2 connects token prediction, synthesis, and supervision in this interface.

3.2 RESTRICTED SYNTHESIS

The predictor uses the noisy image, time, and preceding tokens through $z^{(k)} = T_k(x_t, t, z^{(<k)})$. Each synthesis branch then receives only $z^{(k)}$. We construct $S_k = C_k \circ P_k$ from a bias-free projection P_k and a bias-free local convolution C_k. Both operations are linear, so a zero token gives an exactly zero component:

$$S_k(0) = C_k(P_k(0)) = 0. \tag{6}$$

Figure 3 illustrates this construction. Contextual computation resides in the predictor, and all sample-dependent information entering a component passes through its token. The deterministic image conversion subsequently combines that component with x_t.

This organization gives token removal a direct interpretation in noise coordinates. It also separates two architectural choices that we evaluate experimentally: access to earlier tokens in the predictor and the synthesis map applied to each token. The simultaneous control uses $z^{(k)} = T_k(x_t, t)$, while the deep-synthesis control uses a biased nonlinear map with the same token-only input. Appendix S2.1 specifies the controls, and Appendix S1 analyzes the synthesis properties.

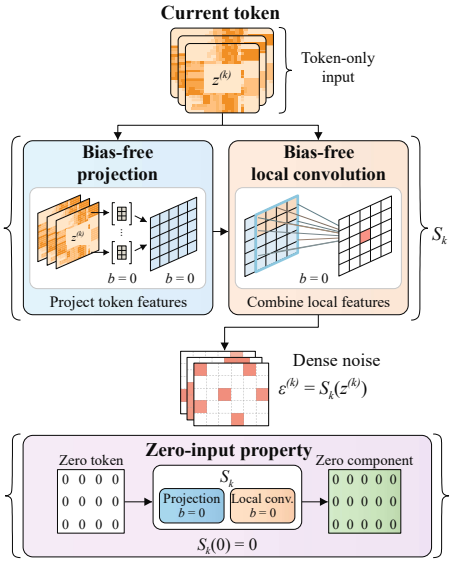

Figure 3: Single-token synthesis. A bias-free projection and local convolution produce $c_k = S_k(z^{(k)})$ and ensure $S_k(0) = 0$.

3.3 PREFIX AND COMPONENT SUPERVISION

An endpoint objective leaves the intermediate decomposition underdetermined. For example, replacing c_1 and c_2 by $c_1 + a$ and $c_2 - a$ preserves u_K while changing u_1. We supervise the partial estimates with a prescribed denoising path $q_0, \ldots, q_K$, where $q_0 = x_t/\alpha_t$ and $q_K = x_0$. The intermediate targets describe coarse-to-fine partial denoising, with default progress power $p = 1.5$. For nonzero σ_t, the equivalent cumulative noise targets and component increments are

$$v_m = \frac{x_t - \alpha_t q_m}{\sigma_t}, \qquad r_m = v_m - v_{m-1}, \qquad v_0 = 0, \quad v_K = \epsilon. \tag{7}$$

Thus $\sum_{k=1}^{m} r_k = v_m$. We jointly train the predictor and synthesis branches with

$$\mathcal{L} = \mathcal{L}_\epsilon + \lambda_p \mathcal{L}_{\text{prefix}} + \lambda_c \mathcal{L}_{\text{component}}, \qquad (\lambda_p, \lambda_c) = (0.15, 0.30). \tag{8}$$

The prefix term compares $\hat{x}_0^{(m)}$ with q_m, and the component term compares c_m with r_m. These terms constrain the cumulative estimate and its individual increments, respectively. In particular, a component can compensate for an earlier error in a prefix, while a component penalty also measures its agreement with its own target. Table 5 evaluates the two losses separately and together; Appendix S7 describes the target and loss specification accompanying the experiments.

3.4 TRAINING AND PREFIX INFERENCE

Algorithm 1 combines the training loop with a shared prefix-prediction routine. Here θ contains the parameters of the token predictors and synthesis maps. Each training example supplies a clean image, a sampled diffusion time, and Gaussian noise. The resulting noisy image is held fixed while the routine predicts all K tokens, synthesizes their contributions, and converts each cumulative sum into an image estimate. All three losses are evaluated from this same forward pass, and their gradients jointly update prediction and synthesis. This follows the noise-prediction training setup of Ho et al. (2020), with additional supervision on the internal component sequence.

The target constructor $\mathcal{Q}$ in Algorithm 1 supplies the intermediate pixel targets for the chosen progress schedule; its endpoints are assigned explicitly. Equation 7 converts these targets into noise

Algorithm 1 CoFiTok training and fixed-time prefix prediction

Require: Token predictors T_k and synthesis maps S_k with joint parameters θ;
diffusion coefficients $\alpha_t, \sigma_t \neq 0$; token count K.

function PREDICTPREFIXES(x_t, t, M) ▷ $1 \leq M \leq K$
  $u_0 \leftarrow 0$, $z^{(<1)} \leftarrow ()$
  for $k = 1, \ldots, M$ **do**
    $z^{(k)} \leftarrow T_k(x_t, t, z^{(<k)})$ ▷ Predict the next token
    $c_k \leftarrow S_k(z^{(k)})$ ▷ Synthesize from this token alone
    $u_k \leftarrow u_{k-1} + c_k$
    $\hat{x}_0^{(k)} \leftarrow (x_t - \sigma_t u_k)/\alpha_t$
  end for
  return $c_{1:M}, u_M, \hat{x}_0^{(1:M)}$
end function

Require: Data distribution p_{data}, training-time distribution $p_{\text{train}}(t)$;
target constructor $\mathcal{Q}$, progress power p, losses in Equation 8.

procedure TRAINCOFITOK
  repeat
    Sample $x_0 \sim p_{\text{data}}$, $t \sim p_{\text{train}}(t)$, $\epsilon \sim \mathcal{N}(0, I)$
    $x_t \leftarrow \alpha_t x_0 + \sigma_t \epsilon$
    $q_0 \leftarrow x_t/\alpha_t$, $q_K \leftarrow x_0$, $v_0 \leftarrow 0$
    $q_{1:K-1} \leftarrow \mathcal{Q}(x_0, x_t, t; K, p)$ ▷ Intermediate pixel targets
    $(c_{1:K}, u_K, \hat{x}_0^{(1:K)}) \leftarrow$ PREDICTPREFIXES(x_t, t, K)
    for $k = 1, \ldots, K$ **do**
      $v_k \leftarrow (x_t - \alpha_t q_k)/\sigma_t$; $r_k \leftarrow v_k - v_{k-1}$
    end for
    $\mathcal{L}_\epsilon \leftarrow \|\epsilon - u_K\|_2^2/(CHW)$
    Evaluate $\mathcal{L}_{\text{prefix}}(\hat{x}_0^{(1:K)}, q_{1:K})$ and $\mathcal{L}_{\text{component}}(c_{1:K}, r_{1:K})$
    $\mathcal{L} \leftarrow \mathcal{L}_\epsilon + \lambda_p \mathcal{L}_{\text{prefix}} + \lambda_c \mathcal{L}_{\text{component}}$
    Update θ using the gradient of $\mathcal{L}$ through prediction and synthesis
  until the training budget is exhausted
  return θ
end procedure

increments, so the prefix and component losses use two coordinate descriptions of the same prescribed path. The default uses $p = 1.5$ and $(\lambda_p, \lambda_c) = (0.15, 0.30)$. The target constructor and loss reductions are configuration inputs to the procedure, as detailed in Appendix S7.

At inference, PREDICTPREFIXES(x_t, t, M) returns the first M component contributions, their sum u_M, and the corresponding image estimates. It reads the noisy state and trained parameters directly. Choosing $M = K$ supplies the full noise prediction to the reverse sampler; choosing a shorter prefix exposes an intermediate estimate at the same noise level. After the sampler updates its state, a new call evaluates the predictor at the new (x_t, t). This separates the component loop inside CoFiTok from the time-stepping loop of a diffusion sampler (Song et al., 2021; Karras et al., 2022).

The cumulative readout reuses u_{k-1} when a new component arrives. For M synthesized components, its additions and image conversions take $O(MCHW)$ arithmetic operations. Keeping every dense prefix requires $O(MCHW)$ output storage; emitting estimates as they are formed requires only the current accumulator and image estimate, each of size CHW. The predictor separately retains the token history and internal features used by T_k. This organization makes prefix extraction a sequence of updates to a shared dense accumulator and exposes where computation is spent: token prediction, component synthesis, and cumulative readout.

Algorithm 1 also identifies the operations changed by the controls. Simultaneous prediction removes $z^{(<k)}$ from the token predictor's inputs, and deep synthesis replaces S_k at the component step. Endpoint-only training sets both path weights to zero while retaining the same forward routine. For the order experiment, we save the returned components once and construct each permuted prefix

from those tensors. This reuse makes the numerical intervention correspond directly to the fixed-component analysis below.

3.5 Error relation and order analysis

At fixed nonzero α_t and σ_t, substituting Equation 7 into Equation 5 gives

$$\|\hat{x}_0^{(m)} - q_m\|_2^2 = \frac{\sigma_t^2}{\alpha_t^2}\|u_m - v_m\|_2^2. \tag{9}$$

Pixel-space and noise-space prefix errors therefore differ by a common positive scale at the same diffusion time. This connects the image-space path metric to the errors of the accumulated components.

A second useful property follows from additivity. For any permutation π of already synthesized components, $\sum_k c_{\pi(k)} = u_K$. The full noise and image estimates remain unchanged, while partial sums can vary. We use this intervention to compare accumulation orders at an identical endpoint. Appendix S1 derives both properties and the cross terms relating prefix and component errors.

The cumulative interface also determines where each component can affect the path. Holding the other synthesized components fixed and writing I_{CHW} for the identity on vectorized images gives

$$\frac{\partial \hat{x}_0^{(m)}}{\partial c_k} = \begin{cases} -\dfrac{\sigma_t}{\alpha_t} I_{CHW}, & k \le m, \\ 0, & k > m. \end{cases} \tag{10}$$

Thus the direct contribution of an early component enters every later prefix, while a later component leaves the preceding estimates unchanged. Prefix supervision acts on these cumulative effects, and component supervision anchors each increment to its own target. In the sequential predictor, gradients additionally pass through dependencies among tokens. Equation 10 describes the output accumulation shared by both sequential and simultaneous variants, explaining how either predictor can be trained with the same ordered-prefix objective.

4 Experiments

4.1 Setup

Models and data. We evaluate small unconditional pixel-space networks on CIFAR-10, Tiny ImageNet-200, ImageNet-64 HF, strict-source ImageNet-64, FFHQ-64, AFHQv2-64, and 10% and full-source ImageNet-256. The predictor uses a U-Net-like backbone with $K = 8$ token heads at 32/64 pixels and $K = 4$ at 256 pixels. Broad comparisons use 3k training steps at 32 pixels and 5k at 64/256 pixels; repeated and order studies use 20k steps. At 64 pixels, direct dense and CoFiTok models have 80,707 and 81,808 parameters; at 256 pixels they have 15,690 and 15,776. Improved DDPM and EDM provide additional diffusion comparisons (Nichol & Dhariwal, 2021; Karras et al., 2022). Appendix S2 gives model sizes, data sources, and evaluation budgets.

Controls and metrics. Direct dense prediction evaluates endpoint reconstruction, endpoint-only factorization evaluates the effect of path supervision with the same architecture, and fixed-component permutations evaluate accumulation order. Repeated comparisons use seeds 103 and 139, with 1,024 validation images at $t = 500$. Endpoint MSE measures the final image error. To evaluate the partial predictions, let e_m be the mean squared pixel error against q_m, averaged over the N validation images:

$$e_m = \frac{1}{NCHW}\sum_{i=1}^{N}\|\hat{x}_{0,i}^{(m)} - q_{m,i}\|_2^2, \qquad A_{\text{path}} = \frac{1}{K-1}\sum_{m=1}^{K-1}\frac{e_m + e_{m+1}}{2}. \tag{11}$$

Path AUC summarizes the trajectory over prefixes $1, \ldots, K$, and endpoint MSE equals e_K because $q_K = x_0$. For $K = 8$, the final estimate has weight $1/14$ in path AUC, with the remaining weight on intermediate estimates. Reporting both quantities therefore shows how the objective distributes reconstruction error along the sequence. Tables report mean and sample SD across the two seeds. Appendix S1.4 defines path AUC, effective component count, and the zero-token energy ratio. Generation evaluation compares generated and reference distributions using low-resolution and Inception features.

4.2 PATH QUALITY AND ENDPOINT RECONSTRUCTION

Eight-dataset comparison. Table 1 shows lower path AUC for CoFiTok than for endpoint-only factorization in all eight settings, with scores from 0.0653 to 0.1416. Endpoint PSNR differences against direct dense prediction range from -0.162 to $+0.137$ dB, with improvements in three settings. The two metric groups thus show consistent path alignment and dataset-dependent endpoint effects. Appendix S3.1 gives the associated component diagnostics.

Table 1: Endpoint reconstruction and prescribed-path alignment across eight dataset settings. Direct dense prediction is the endpoint control; endpoint-only factorization is the path control. CIFAR-10 uses 3,000 training steps and all other settings use 5,000. Protocol details and component diagnostics appear in Appendices S2 and S3.1.

	Endpoint PSNR (dB)↑		Path AUC↓	
Dataset	Direct dense	CoFiTok	Endpoint only	CoFiTok
CIFAR-10	14.407	14.245	0.7906	0.0967
Tiny ImageNet-200	14.453	14.348	1.1177	0.0731
ImageNet-64 HF	15.004	14.897	1.1120	0.0672
ImageNet-64 strict	14.656	14.521	1.0502	0.0705
FFHQ-64	15.126	15.024	1.0830	0.0676
AFHQv2-64	15.154	15.167	1.0185	0.0653
ImageNet-256 10%	13.490	13.627	1.5686	0.1416
ImageNet-256	13.770	13.809	1.5003	0.1325

Effect of path supervision. Table 2 compares endpoint-only and full-objective training after 20,000 steps. Path AUC decreases from 1.0098 to 0.0378 on Tiny ImageNet and from 0.9984 to 0.0379 on ImageNet-64 HF, reductions of 96.3% and 96.2%. Endpoint MSE increases from 0.1253 to 0.1296 and from 0.1096 to 0.1151, respectively. Path supervision therefore improves intermediate alignment with endpoint costs of 3.4% and 5.0%. Figure S2 in Appendix S3.2 displays the changes and seed variability.

Table 2: Effect of path supervision after 20,000 training steps. The architecture is identical within each dataset; full CoFiTok adds prefix and component losses to endpoint-only training.

Dataset	Training objective	Endpoint MSE↓	Path AUC↓
Tiny ImageNet-200	Endpoint only	0.1253 ± 0.0001	1.0098 ± 0.1430
Tiny ImageNet-200	Full CoFiTok	0.1296 ± 0.0015	0.0378 ± 0.0007
ImageNet-64 (HF)	Endpoint only	0.1096 ± 0.0006	0.9984 ± 0.1210
ImageNet-64 (HF)	Full CoFiTok	0.1151 ± 0.0016	0.0379 ± 0.0037

Mean $\pm$ sample SD across seeds 103 and 139; 1,024 validation images per seed at $t = 500$, $K = 8$. HF denotes the Hugging Face-derived ImageNet source. Shading marks full CoFiTok. Endpoint MSE evaluates the final image; path AUC evaluates the prescribed prefixes. Both errors are lower-is-better.

Accumulation order. Table 3 evaluates all 24 permutations of four fixed components on ImageNet-256. The training order ranks first for both seeds, with path AUCs 0.0664 and 0.0678. The other 23 orders have mean AUCs 0.3157 and 0.3082, while reverse order gives 0.7192 and 0.6821. All orders share the full prediction, isolating the organization of intermediate contributions. Appendix S3.3 also compares CoFiTok with a separately trained dense predictor: endpoint MSE is $0.1133/0.1179$ for CoFiTok and $0.1234/0.1244$ for dense prediction.

Generated images. Table 4 evaluates four generators under the broad training budgets. EDM obtains the lowest low-resolution distance in seven settings and direct dense prediction in full ImageNet-256. CoFiTok obtains the lowest Inception-feature distance on CIFAR-10, one of eight settings. CoFiTok, direct dense, and Improved DDPM use 50 network evaluations; EDM uses 79, with different model capacities. The broad generation rankings favor existing diffusion baselines despite CoFiTok's lower fixed-time path error. Appendix S6 also reports 4,096-sample ImageNet-256 controls and pretrained-model comparisons.

Table 3: Effect of reordering fixed components on ImageNet-256. The 24-order evaluation is summarized by the training order $(1, 2, 3, 4)$, the mean over the other 23 orders, and the reverse order $(4, 3, 2, 1)$. Rank 1 denotes the lowest path AUC.

	Path AUC↓			
Seed	Training order	Other 23 orders (mean)	Reverse order	Training-order rank (of 24)
103	0.0664	0.3157	0.7192	1/24
139	0.0678	0.3082	0.6821	1/24

20,000 training steps; 1,024 validation images at $t = 500$. Each row is one seed. Components are computed once and then reordered, leaving the full prediction unchanged. The alternative-order mean includes the reverse order.

Table 4: Unconditional generation under short training budgets. Entries are low-resolution Fréchet-style distances between generated and reference images; smaller values indicate closer distributions under this evaluator.

	Low-resolution Fréchet-style distance↓			
Dataset	CoFiTok	Direct dense predictor	Improved DDPM	EDM
CIFAR-10	6.47	5.80	7.86	2.87
Tiny ImageNet-200	8.49	6.99	9.07	3.10
ImageNet-64 (HF)	8.09	6.89	8.88	3.23
ImageNet-64 (strict)	8.36	6.87	8.80	2.75
FFHQ-64	8.74	6.84	9.44	2.51
AFHQv2-64	7.08	5.68	8.29	1.96
ImageNet-256 (10% training)	10.44	12.59	10.40	10.39
ImageNet-256	10.54	10.24	10.40	10.40

CIFAR-10: 3,000 steps; all other datasets: 5,000. Evaluation uses 1,024 generated/4,096 reference images at 32 or 64 pixels, and 512/2,048 at 256 pixels. HF: Hugging Face-derived source; strict: downsampled ImageNet source; 10%: training subset, evaluated on the full validation source. Model sizes, batch sizes, and sampling costs are given in Appendix S2.

4.3 Ablation Study

Prefix and component losses. Table 5 evaluates architectural and objective variants on ImageNet-64 HF under the same 5k-step protocol. Endpoint-only training gives path AUC 0.9991 and endpoint MSE 0.1275. Adding only component supervision reduces path AUC to 0.1269; adding only prefix supervision reduces it to 0.1105. The full objective achieves 0.0686, with endpoint MSE 0.1300. The two path losses therefore contribute complementary constraints on partial predictions, accompanied by a small increase in endpoint error.

Feedback and synthesis. Removing token-to-token feedback changes mean path AUC from 0.0686 to 0.0709 and endpoint MSE from 0.1300 to 0.1314. The close means and two-seed variability indicate that simultaneous prediction also learns useful prefixes in this setting. Deep biased synthesis reduces path AUC to 0.0463 while increasing endpoint MSE to 0.1496. It also produces a zero-token energy ratio of 0.0485 ± 0.0050, compared with exact zero for the shallow map. This variant jointly changes depth, nonlinearity, and bias. Appendix S3.4 provides the individual comparisons and component diagnostics.

Partial denoising. Figure 4 compares endpoint-only factorization and CoFiTok on Tiny ImageNet. With path supervision, successive displayed estimates develop recognizable spatial organization while retaining noise; the final estimates are blurred. Endpoint-only training concentrates supervision on the last estimate, whereas CoFiTok directly trains the intermediate sequence. The images illustrate the change measured by path AUC and the reconstruction quality of these small models.

Component organization and sensitivity. Path supervision increases the effective component count from about 5.2–5.3 to 6.8 of eight in the repeated study, indicating more balanced component energy. Appendix S4 provides the corresponding ImageNet-64 prefix display. Appendix S5 varies loss

Table 5: Architecture and objective ablations on ImageNet-64 HF. All variants use $K = 8$, 5,000 training steps, and 1,024 validation images at $t = 500$. Values are mean $\pm$ sample SD across seeds 103 and 139. The full objective uses $(\lambda_p, \lambda_c) = (0.15, 0.30)$.

Variant	Endpoint MSE↓	Path AUC↓
Full CoFiTok	0.1300 ± 0.0027	0.0686 ± 0.0017
Endpoint loss only	0.1275 ± 0.0019	0.9991 ± 0.1607
Without prefix loss	0.1300 ± 0.0003	0.1269 ± 0.0096
Without component loss	0.1302 ± 0.0012	0.1105 ± 0.0013
Simultaneous token prediction	0.1314 ± 0.0016	0.0709 ± 0.0081
Deep biased synthesis	0.1496 ± 0.0305	0.0463 ± 0.0051

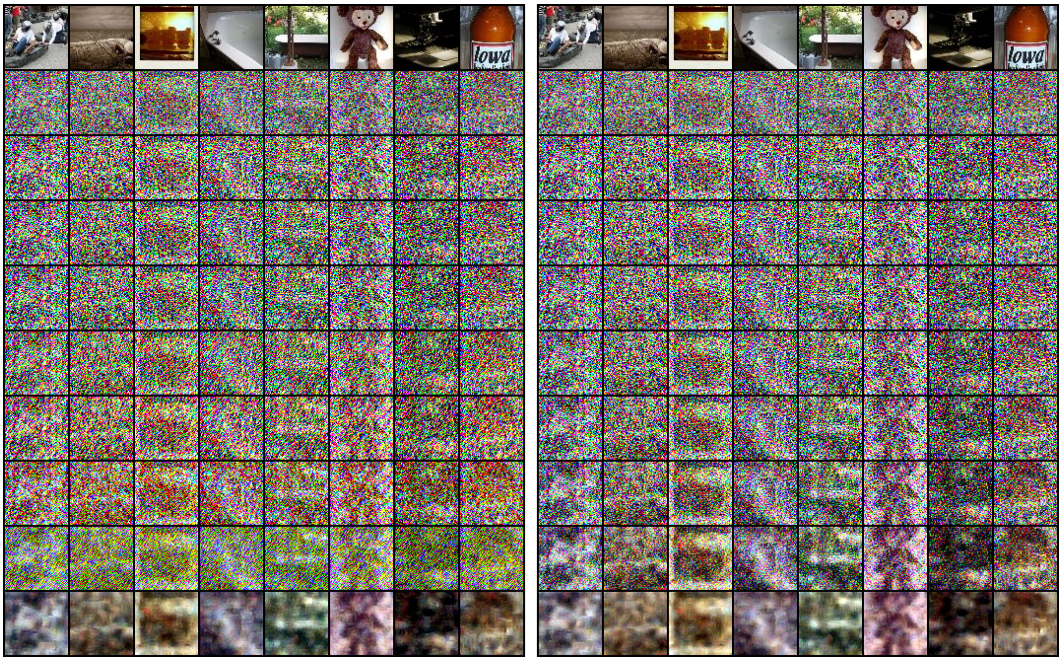

Figure 4: Tiny ImageNet prefix estimates. Left: endpoint-only factorization. Right: CoFiTok. Each half shows eight examples, with clean references above the partial and final estimates. Later CoFiTok estimates develop spatial structure while retaining residual noise.

weights, progress power, and token count: stronger path weights further reduce path error, while the four- and eight-token variants have similar mean AUCs. A decorrelation study in Appendix S3.5 reduces component cosine similarity with a small increase in path AUC.

5 Conclusion

CoFiTok organizes a dense diffusion noise prediction into ordered continuous token contributions. Token-only, bias-free synthesis gives zero tokens an exact zero-component meaning, while prefix and component supervision shape partial denoising at a fixed noisy state and time. The experiments demonstrate lower prescribed-path error across eight settings and an advantage for the learned accumulation order at a shared endpoint. The resulting path improvements coexist with reconstruction tradeoffs and generation rankings that often favor existing baselines. The present study uses small unconditional models, short training budgets, and two-seed repeated comparisons. Extending the interface to larger models, multiple noise levels, and direct measurements of sampling quality and computational cost is a natural next step.

AI USE STATEMENT

Generative AI tools were used solely to assist with and polish the writing of this manuscript. The authors reviewed and revised all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The paper specifies the token interface, synthesis restriction, and training objective. Appendices S1 and S2 provide mathematical details, metric definitions, model sizes, and evaluation protocols. The remaining appendices give extended results and ablations. Appendix S7 records the implementation details and the missing artifacts needed to reproduce training and evaluation. The repository contains the manuscript source, figures, tables, and references; it does not include the complete training implementation, checkpoints, or run-level prediction records.

REFERENCES

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, 2026.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length, 2025.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. *arXiv preprint arXiv:2405.17430*, 2024.

Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.

Carlos Esteves, Mohammed Suhail, and Ameesh Makadia. Spectral image tokenizer, 2025.

Ziteng Gao and Mike Zheng Shou. D-ar: Diffusion via autoregressive models. In *International Conference on Learning Representations*, volume 2026, pp. 68013–68032, 2026.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 30234–30262, 2024.

Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Joshua Susskind, and Shuangfei Zhai. Denoising autoregressive transformers for scalable text-to-image generation. In *International Conference on Learning Representations*, volume 2025, pp. 94564–94585, 2025.

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 *NIPS*, pp. 6626–6637, 2017.

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

Sascha Holl, Jente Vandersanden, Gurprit Singh, and Hans-Peter Seidel. Score-based generative modeling through anisotropic stochastic partial differential equations. *CoRR*, abs/2605.08976, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning, 2022.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024.

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. In *ICLR*. OpenReview.net, 2025.

Keli Liu, Julius Berner, Y HoiM, Hongkai Zheng, Jiahong Wu, Xiangxiang Chu, et al. There is no vae: End-to-end pixel-space generative modeling via self-supervised pre-training. In *International Conference on Learning Representations*, volume 2026, pp. 102818–102839, 2026.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.

Zhicong Tang, Dong Chen, Jianmin Bao, and Baining Guo. Bar: Refactor the basis of autoregressive visual generation. In *International Conference on Learning Representations*, volume 2026, pp. 26919–26947, 2026.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Bohan Wang, Zhongqi Yue, Fengda Zhang, Shuo Chen, Li'an Bi, Junzhe Zhang, Xue Song, Kennard Yanting Chan, Jiachun Pan, Weijia Wu, Mingze Zhou, Wang Lin, Kaihang Pan, Saining Zhang, Liyu Jia, Wentao Hu, Wei Zhao, and Hanwang Zhang. Selftok: Discrete visual tokens of autoregression, by diffusion, and for reasoning, 2025.

Pingyu Wu, Kai Zhu, Yu Liu, Longxiang Tang, Jian Yang, Yansong Peng, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Towards sequence modeling alignment between tokenizer and autoregressive model. In *International Conference on Learning Representations*, volume 2026, pp. 132194–132217, 2026.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. Elastictok: Adaptive tokenization for image and video. In *International Conference on Learning Representations*, volume 2025, pp. 38036–38056, 2025.

Yibo Yang, Justus Will, and Stephan Mandt. Progressive compression with universally quantized diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 37998–38019, 2025.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, Yi Wang, Limin Wang, and Yali Wang. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. In *International Conference on Learning Representations*, volume 2026, pp. 84741–84772, 2026.

S1 MATHEMATICAL DETAILS

S1.1 PIXEL-SPACE AND NOISE-SPACE ERRORS

For nonzero α_t and σ_t, the forward process and Equation 5 give the endpoint relation

$$\hat{x}_0^{(K)} - x_0 = -\frac{\sigma_t}{\alpha_t}(u_K - \epsilon), \qquad \frac{\|\hat{x}_0^{(K)} - x_0\|_2^2}{CHW} = \frac{\sigma_t^2}{\alpha_t^2}\frac{\|u_K - \epsilon\|_2^2}{CHW}. \tag{S1}$$

For a general target q_m, substitute $v_m = (x_t - \alpha_t q_m)/\sigma_t$ to obtain

$$\hat{x}_0^{(m)} - q_m = \frac{x_t - \sigma_t u_m - \alpha_t q_m}{\alpha_t} = -\frac{\sigma_t}{\alpha_t}(u_m - v_m). \tag{S2}$$

Taking squared norms proves Equation 9. The scale σ_t^2/α_t^2 is common at fixed time and varies with the noise schedule across times. The conversion becomes ill-conditioned as α_t approaches zero.

Let $\delta_k = c_k - r_k$ denote component error. Since the target increments telescope to v_m, the cumulative error satisfies

$$u_m - v_m = \sum_{k=1}^{m} \delta_k, \qquad \|u_m - v_m\|_2^2 = \sum_{k=1}^{m} \|\delta_k\|_2^2 + 2 \sum_{i<j\le m} \langle \delta_i, \delta_j \rangle. \tag{S3}$$

The cross terms allow component errors to cancel in a prefix. Component supervision acts on individual increments; prefix supervision acts on their cumulative effect. This decomposition explains the complementary losses in Equation 8.

S1.2 FIXED-COMPONENT PERMUTATION INVARIANCE

For a permutation π of already computed components,

$$u_K^{\pi} = \sum_{k=1}^{K} c_{\pi(k)} = u_K, \qquad u_m^{\pi} = \sum_{k=1}^{m} c_{\pi(k)} \quad (m < K). \tag{S4}$$

Combining Equation S4 with Equation 5 gives, for every sample,

$$\hat{x}_0^{(K),\pi} = \hat{x}_0^{(K)}, \qquad \hat{x}_0^{(m),\pi} - \hat{x}_0^{(m)} = -\frac{\sigma_t}{\alpha_t}\left(\sum_{k=1}^{m} c_{\pi(k)} - \sum_{k=1}^{m} c_k\right). \tag{S5}$$

The intervention reuses the same component tensors for each order. Re-evaluating a sequential predictor after changing its order produces new tensors and defines a separate intervention. For $K = 4$, we evaluate all $4! = 24$ accumulation orders, summarized in Table 3 and Appendix S3.3.

S1.3 ZERO PRESERVATION AND TOKEN COORDINATES

Bias-free linear projection and convolution give $C_k(P_k(0)) = 0$. This property applies to the synthesized noise component; the subsequent image conversion starts at $\hat{x}_0^{(0)} = x_t/\alpha_t$. For an invertible token-space map B_k,

$$(C_k \circ P_k \circ B_k^{-1})(B_k z^{(k)}) = (C_k \circ P_k)(z^{(k)}). \tag{S6}$$

Thus equivalent token coordinates can yield the same components and prefixes. Components may also be correlated while carrying comparable energy, which motivates reporting energy and correlation diagnostics as separate measurements.

S1.4 EVALUATION METRICS

Let N be the number of evaluation images, indexed by i. The pixel mean-squared error at prefix m is

$$e_m = \frac{1}{NCHW} \sum_{i=1}^{N} \|\hat{x}_{0,i}^{(m)} - q_{m,i}\|_2^2. \tag{S7}$$

Path AUC is an equal-spaced trapezoidal average over prefixes $1, \ldots, K$,

$$A_{\text{path}} = \frac{1}{K-1} \sum_{m=1}^{K-1} \frac{e_m + e_{m+1}}{2}, \qquad K > 1. \tag{S8}$$

The zero prefix is excluded, and endpoint MSE equals e_K because $q_K = x_0$. Path AUC measures agreement with the prescribed denoising targets. PSNR measures reconstruction quality in decibels. Generation metrics compare feature distributions through Fréchet-style distances (Heusel et al., 2017); we report low-resolution and Inception-feature results separately with their evaluation sample counts.

For normalized component-energy proportions $p_k \geq 0$ with $\sum_k p_k = 1$, the effective component count is

$$K_{\text{eff}} = \exp\left(-\sum_{k=1}^{K} p_k \log p_k \right), \tag{S9}$$

using $0 \log 0 = 0$. Equal energy gives $K_{\text{eff}} = K$, while one active component gives $K_{\text{eff}} = 1$. This measures energy balance. The zero-token ratio measures synthesis energy from $S_k(0)$ relative to observed component energy. A zero-preserving map has ratio zero whenever the normalization energy is nonzero. Repeated tables and plots use sample SD across two seeds. Relative changes in the main text are calculated from the displayed means.

S2 Experimental protocol

S2.1 Model variants

The experiments vary prediction, synthesis, and supervision within the same cumulative interface. Direct dense prediction learns a single noise field and serves as an endpoint control. Endpoint-only factorization retains all token heads and synthesis branches and trains their sum with $\mathcal{L}_\epsilon$. Full CoFiTok additionally uses prefix and component losses. The simultaneous variant removes access to earlier predicted tokens, retaining the synthesis maps and objective. Deep synthesis replaces the shallow bias-free maps with biased nonlinear token-only operators. Selecting a prefix of length m determines which components enter the image estimate; actual evaluation cost depends on the predictor implementation and selected sampler.

S2.2 Training and evaluation budgets

The broad study uses 3,000 steps for CIFAR-10 and 5,000 steps for datasets at 64 and 256 pixels. At 32 pixels, CoFiTok, direct dense, and endpoint-only factorization use batch size 64; Improved DDPM and EDM use 32. At 64 pixels all listed models use batch size 32, and at 256 pixels they use four. These budgets correspond to nominal image presentations of 192,000 versus 96,000 at 32 pixels, 160,000 at 64 pixels, and 20,000 at 256 pixels.

Broad endpoint diagnostics use 512 images at 32/64 pixels and 256 images at 256 pixels. Generation comparisons use 1,024 generated and 4,096 reference images at 32/64 pixels, and 512 generated and 2,048 reference images at 256 pixels. CoFiTok, direct dense, and Improved DDPM use 50 network evaluations; EDM uses 79. The repeated and order studies use 20,000 training steps and 1,024 validation images at $t = 500$ for each of seeds 103 and 139. Component and hyperparameter ablations use the same two-seed evaluation after 5,000 steps.

S2.3 Model capacity and direct dense control

Table S1 gives trainable parameter counts. Direct dense prediction has 1.35% fewer parameters than CoFiTok at 32/64 pixels and 0.55% fewer at 256 pixels. The 256-pixel models use a smaller capacity configuration than the 64-pixel models. The external diffusion models have roughly one to twenty-two million parameters.

Table S2 compares the nearly parameter-matched predictors on ImageNet-64 HF. CoFiTok gives endpoint MSE 0.1300 ± 0.0027, compared with 0.1292 ± 0.0006 for direct dense prediction. The factorized model additionally exposes the prefix sequence evaluated by path AUC.

Table S1: Trainable parameters by image resolution. CoFiTok and its direct dense control have closely matched sizes; the external diffusion baselines use larger architectures.

Resolution	CoFiTok	Dense	Improved DDPM	EDM
32×32	81,808	80,707	9,593,539	3,870,531
64×64	81,808	80,707	21,715,907	3,870,531
256×256	15,776	15,690	7,200,579	952,291

Table S2: Direct dense comparison on ImageNet-64 HF after 5k steps. Values are mean $\pm$ sample SD across seeds 103 and 139, using 1,024 validation images at $t = 500$. The dash indicates a single-output model with no component-prefix sequence.

Model	Endpoint MSE	Native path AUC
CoFiTok	0.1300 ± 0.0027	0.0686 ± 0.0017
Direct dense	0.1292 ± 0.0006	—

S2.4 Data sources and comparison protocols

Table S3 lists the data sources. The HF-derived and strict-source ImageNet-64 datasets are separate preparations. The ImageNet-256 10% setting uses a deterministic training subset and the full validation source. FFHQ and AFHQv2 extend evaluation to face and animal images.

We use three evaluation settings. Pixel-space models are trained for fixed optimizer-step budgets on each dataset. Pretrained FlexTok and TiTok models are evaluated for image reconstruction at 256×256 using 256 images per dataset. D-AR, MAR, and ReTok are evaluated with 50,000 ImageNet-256 samples under their respective generation protocols. Table S14 gives reconstruction results, and Table S15 gives the 50k-sample generation results.

Table S3: Dataset resolutions, splits, and sources.

Dataset label	Resolution	Split	Source
CIFAR-10	32	Train/test	Canonical CIFAR-10 dataset
Tiny-IN	64	Train/validation	Tiny ImageNet-200 dataset
ImageNet-64 HF	64	Train/validation	HF-derived ImageNet-1K 64x64 source
ImageNet-64 strict	64	Train/validation	Exact downsampled ImageNet source from Academic Torrents
FFHQ-64	64	Train/evaluation	Face-image dataset
AFHQv2-64	64	Train/evaluation	Animal-image dataset
ImageNet-256 10%	256	10% train/full validation	Deterministic subset of the prepared 256x256 dataset
ImageNet-256	256	Full train/validation	Full prepared 256x256 ImageNet dataset

S3 Extended results

S3.1 Eight-dataset diagnostics

Figure S1 displays the path and endpoint comparisons in Table 1. Path AUC improves across all eight settings; endpoint PSNR improves on AFHQv2 and the two ImageNet-256 settings. Table S4 adds effective component counts and zero-token ratios. Deep synthesis gives nonzero zero-token ratios from 0.0179 to 0.0619. At 256 pixels, CoFiTok's effective count is about 3.4 of four, compared with 3.67 for endpoint-only factorization. At lower resolutions, CoFiTok has the higher effective count, showing that energy balance varies with the dataset and model configuration.

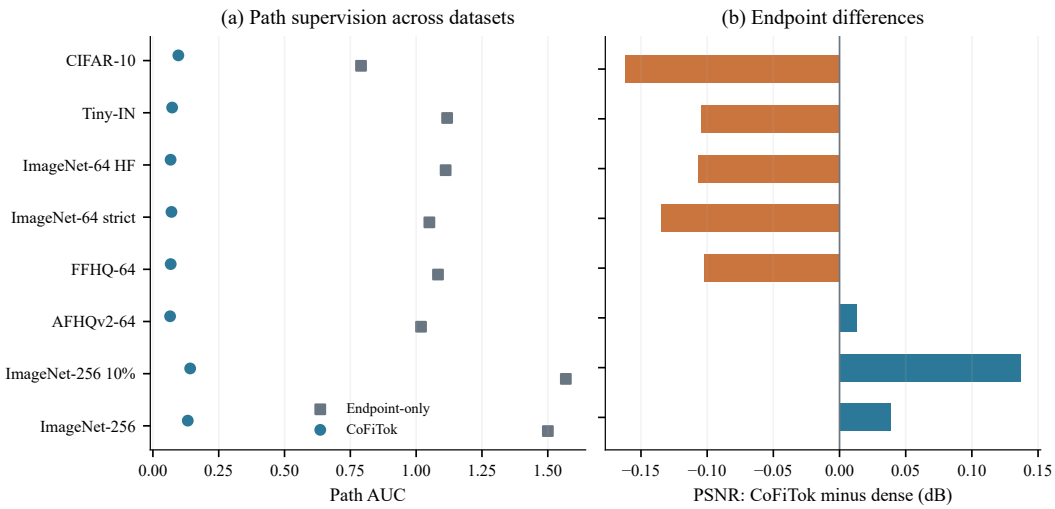

Figure S1: Eight-dataset comparison. Left: path AUC for CoFiTok and endpoint-only factorization. Right: endpoint PSNR difference between CoFiTok and direct dense prediction; positive values favor CoFiTok. Points show aggregate scores under the broad protocol.

Table S4: Component organization across eight settings. Effective counts compare endpoint-only factorization with CoFiTok; zero-token ratios compare CoFiTok with deep biased synthesis. $K = 8$ at 32/64 pixels and $K = 4$ at 256 pixels.

	Effective component count		Zero-token energy ratio	
Dataset	Endpoint only	CoFiTok	CoFiTok	Deep synthesis
CIFAR-10	5.24	6.84	0.0000	0.0619
Tiny ImageNet-200	5.57	6.80	0.0000	0.0505
ImageNet-64 (HF)	5.33	6.79	0.0000	0.0522
ImageNet-64 (strict)	5.36	6.82	0.0000	0.0535
FFHQ-64	5.41	6.82	0.0000	0.0518
AFHQv2-64	5.45	6.82	0.0000	0.0561
ImageNet-256 10%	3.67	3.36	0.0000	0.0179
ImageNet-256	3.67	3.37	0.0000	0.0185

S3.2 Repeated endpoint and prefix measurements

Figure S2 shows the lower path errors and higher endpoint errors induced by path supervision. Table S5 adds the component diagnostics: effective counts increase from 5.312 to 6.766 on Tiny ImageNet and from 5.169 to 6.752 on ImageNet-64 HF. All four zero-token ratios are zero.

Table S5: Component diagnostics after 20k steps with $K = 8$. Values are mean ± sample SD across seeds 103 and 139, with 1,024 images at $t = 500$ per seed. K_{eff} measures energy balance. Table 2 gives endpoint and path errors.

Dataset	Training objective	K_{eff}	Zero-token ratio
Tiny ImageNet-200	Endpoint only	5.312 ± 0.049	0.0000 ± 0.0000
Tiny ImageNet-200	Full CoFiTok	6.766 ± 0.015	0.0000 ± 0.0000
ImageNet-64 (HF)	Endpoint only	5.169 ± 0.108	0.0000 ± 0.0000
ImageNet-64 (HF)	Full CoFiTok	6.752 ± 0.012	0.0000 ± 0.0000

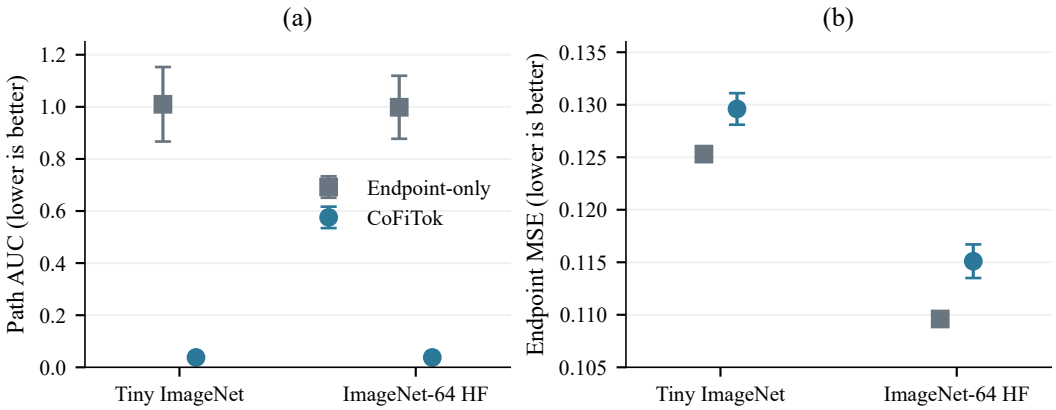

Figure S2: Path alignment and endpoint reconstruction after 20,000 steps. (a) Prescribed-path AUC. (b) Final-image MSE. Both errors are lower-is-better. Points show two-seed means and bars show sample SD. Both panels use the same 1,024-image, $t = 500$ evaluation. The endpoint panel has a restricted vertical range to make the reconstruction tradeoff visible.

S3.3 Fixed-component permutations

Figure S3 compares the training order, mean alternative order, and reverse order for each seed. The mean alternative-order penalties are 0.2493 and 0.2404. Table S6 adds the dense endpoint comparison and the paired-bootstrap lower-bound estimates of 0.2489 and 0.2400. The confidence level and resampling specification for these bounds are unavailable. Endpoint-only factorization gives path AUCs of 1.7879 and 1.4401 for seeds 103 and 139.

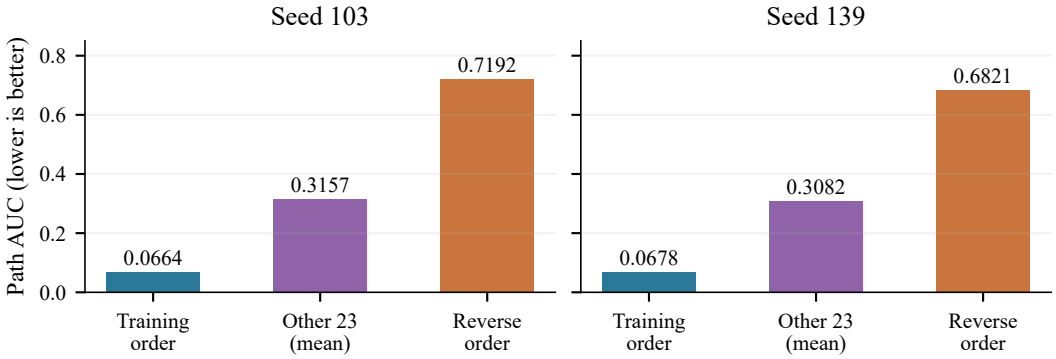

Figure S3: Reordering fixed components changes path quality. Bars display path AUC for training order, the mean of all 23 alternative orders, and reverse order. The two panels show separate seeds. All orders within each seed share the same full component sum.

Table S6: ImageNet-256 endpoint and order diagnostics with $K = 4$ after 20k steps. Each row is one seed. Endpoint columns compare separately trained dense and CoFiTok models. The last column gives the paired-bootstrap lower bound for the mean alternative-order penalty.

	Endpoint MSE↓		
Seed	Direct dense	CoFiTok	Path penalty lower bound
103	0.1234	0.1133	0.2489
139	0.1244	0.1179	0.2400

S3.4 ARCHITECTURAL AND OBJECTIVE ABLATIONS

Tables S7, S8, and S9 expand the combined comparison in Table 5. They specify token feedback, the synthesis operator, and individual loss weights. Figure S4 visualizes the path and endpoint errors.

Table S10 gives effective component counts from 5.366 to 6.829. Deep synthesis has effective count 6.774 and zero-token ratio 0.0485 ± 0.0050, combining balanced component energy with a nonzero response at zero input.

Table S7: Removing token-to-token feedback. Sequential prediction can use previously predicted tokens; simultaneous prediction cannot. The token heads, restricted synthesis and three loss terms are otherwise retained.

Token prediction	Uses earlier predicted tokens	Endpoint MSE↓	Path AUC↓
Sequential	Yes	0.1300 ± 0.0027	0.0686 ± 0.0017
Simultaneous	No	0.1314 ± 0.0016	0.0709 ± 0.0081

ImageNet-64 (HF), $K = 8$, 5,000 steps, 1,024 validation images at $t = 500$. Mean $\pm$ sample SD across seeds 103 and 139. Shading marks the default CoFiTok configuration.

Table S8: Changing the synthesis operator. Both operators receive only their own token. The shallow operator is bias-free; the deep variant adds biased nonlinear layers. The zero-token ratio measures energy produced from zero tokens relative to observed component energy; zero indicates zero-preserving synthesis.

Synthesis operator	Endpoint MSE↓	Path AUC↓	Zero-token energy ratio
Shallow, bias-free	0.1300 ± 0.0027	0.0686 ± 0.0017	0.0000 ± 0.0000
Deep, biased	0.1496 ± 0.0305	0.0463 ± 0.0051	0.0485 ± 0.0050

ImageNet-64 (HF), $K = 8$, 5,000 steps, 1,024 validation images at $t = 500$. Mean $\pm$ sample SD across seeds 103 and 139. Shading marks full CoFiTok. The deep variant jointly changes depth, nonlinearity, and bias.

Table S9: Removing the prefix and component losses. Every row retains the endpoint noise loss with weight 1. The columns λ_p and λ_c give the weights of the prefix and component losses; zero removes that term. All variants are evaluated against the same prescribed path.

Training objective	λ_p	λ_c	Endpoint MSE↓	Path AUC↓
Endpoint only	0	0	0.1275 ± 0.0019	0.9991 ± 0.1607
Without prefix loss	0	0.30	0.1300 ± 0.0003	0.1269 ± 0.0096
Without component loss	0.15	0	0.1302 ± 0.0012	0.1105 ± 0.0013
Full CoFiTok	0.15	0.30	0.1300 ± 0.0027	0.0686 ± 0.0017

ImageNet-64 (HF), $K = 8$, 5,000 steps, 1,024 validation images at $t = 500$; reference path power $p = 1.5$. Mean $\pm$ sample SD across seeds 103 and 139. Shading marks full CoFiTok.

Table S10: Component diagnostics on ImageNet-64 HF with $K = 8$, 5k steps, and 1,024 validation images at $t = 500$. Values are mean $\pm$ sample SD across seeds 103 and 139. The deep variant uses a biased nonlinear token-only operator; the other variants use bias-free synthesis.

Variant	K_{eff}	Zero-token energy ratio
Full CoFiTok	6.780 ± 0.015	0.0000 ± 0.0000
Endpoint only	5.366 ± 0.054	0.0000 ± 0.0000
Without prefix loss	6.547 ± 0.027	0.0000 ± 0.0000
Without component loss	6.752 ± 0.069	0.0000 ± 0.0000
Simultaneous prediction	6.829 ± 0.025	0.0000 ± 0.0000
Deep synthesis	6.774 ± 0.032	0.0485 ± 0.0050

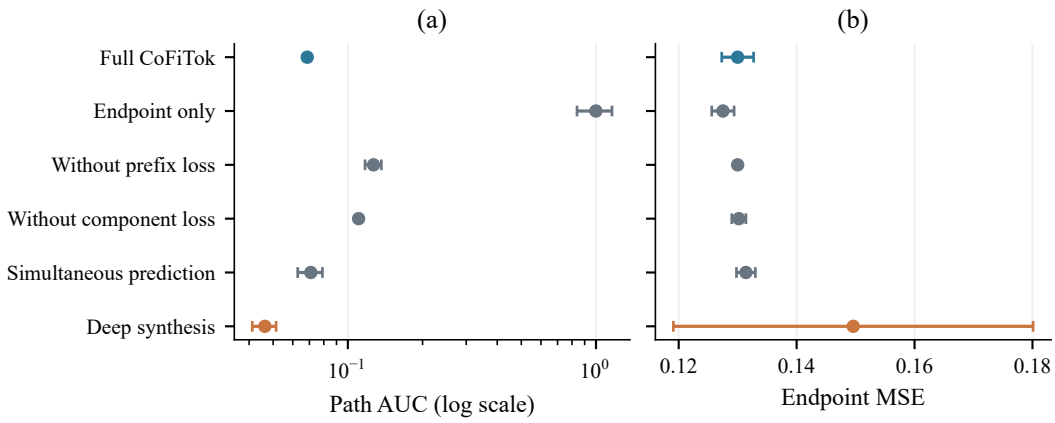

Figure S4: Component interventions on ImageNet-64 (HF), $K = 8$, 5,000 steps. (a) Path AUC on a logarithmic axis. (b) Endpoint MSE on a linear axis. Both errors are lower-is-better. Points and bars show two-seed mean and sample SD; blue denotes full CoFiTok. The orange deep-synthesis row has the lowest path AUC but a nonzero zero-input ratio and a larger endpoint mean and SD.

S3.5 Component decorrelation

We additionally study a component-decorrelation term on Tiny ImageNet-200. Its weight increases to 0.003 over 1,500 steps starting at step 3,000, reaching full weight at step 4,500 of a 5,000-step run. Relative to the default objective with zero decorrelation weight, the two-seed mean changes are $+0.0036$ in ordered path AUC, -0.0328 in mean absolute component cosine, -0.0002 in final MSE, and $+0.0336$ in effective component count. Mild separation pressure therefore reduces component correlation with a small path-AUC cost. Stronger weights further reduce cosine similarity and incur larger path penalties.

S4 Qualitative results

Figure S5 compares endpoint-only factorization and CoFiTok on ImageNet-64, complementing the Tiny ImageNet examples in Figure 4. Each panel contains eight examples. Clean reference images appear in the top row, followed by partial and final estimates. Later CoFiTok estimates show increasing spatial organization while retaining residual noise; final images remain blurred. These displays complement the quantitative path–endpoint comparisons in Table 2.

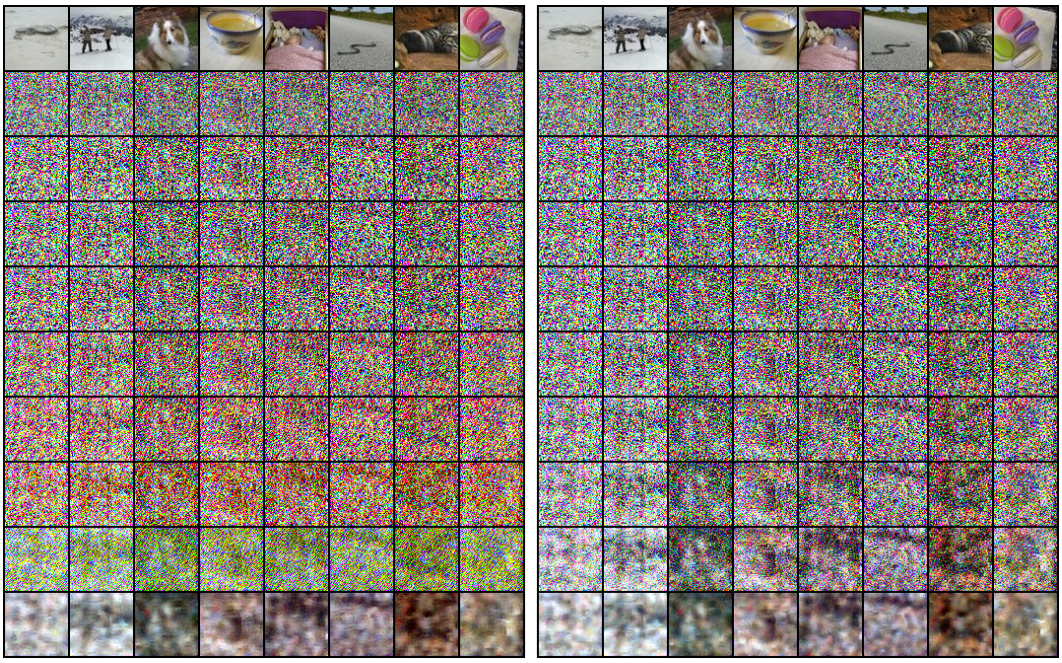

Figure S5: ImageNet-64 prefix estimates. Left: endpoint-only factorization. Right: CoFiTok. The top row contains clean references; the remaining rows show partial and final estimates for eight examples.

S5 Hyperparameter sensitivity

S5.1 Loss weights

Figure S6 and Table S11 evaluate one variable at a time from the default $(\lambda_p, \lambda_c, p, K) = (0.15, 0.30, 1.5, 8)$ on ImageNet-64 HF. Each setting uses 5k training steps, seeds 103 and 139, and 1,024 validation images at $t = 500$. Increasing λ_p from zero to 0.30 decreases mean path AUC from 0.1269 to 0.0604; increasing λ_c from zero to 0.60 decreases it from 0.1105 to 0.0581. Endpoint means vary less and non-monotonically, illustrating the path–endpoint tradeoff across loss weights.

Table S11: Hyperparameter sensitivity. Variables outside the indicated sweep retain their default values. Scores are mean ± sample SD across two seeds.

Axis	Value	MSE ↓	Path AUC ↓	K_{eff}
λ_p	0	0.1300 ± 0.0003	0.1269 ± 0.0096	6.547 ± 0.027
λ_p	0.05	0.1309 ± 0.0020	0.0894 ± 0.0009	6.694 ± 0.033
λ_p	0.15	0.1300 ± 0.0027	0.0686 ± 0.0017	6.780 ± 0.015
λ_p	0.3	0.1310 ± 0.0020	0.0604 ± 0.0015	6.819 ± 0.002
λ_c	0	0.1302 ± 0.0012	0.1105 ± 0.0013	6.752 ± 0.069
λ_c	0.1	0.1298 ± 0.0022	0.0856 ± 0.0004	6.860 ± 0.001
λ_c	0.3	0.1300 ± 0.0027	0.0686 ± 0.0017	6.780 ± 0.015
λ_c	0.6	0.1314 ± 0.0018	0.0581 ± 0.0036	6.717 ± 0.010
p	0.75	0.1344 ± 0.0025	0.4909 ± 0.0159	7.035 ± 0.102
p	1	0.1318 ± 0.0021	0.2143 ± 0.0133	7.912 ± 0.015
p	1.5	0.1300 ± 0.0027	0.0686 ± 0.0017	6.780 ± 0.015
p	2	0.1307 ± 0.0031	0.1541 ± 0.0010	5.449 ± 0.020
p	2.5	0.1308 ± 0.0005	0.3069 ± 0.0031	4.459 ± 0.007
K	4	0.1285 ± 0.0014	0.0693 ± 0.0058	3.321 ± 0.006
K	8	0.1300 ± 0.0027	0.0686 ± 0.0017	6.780 ± 0.015
K	16	0.1313 ± 0.0025	0.0711 ± 0.0018	13.676 ± 0.154

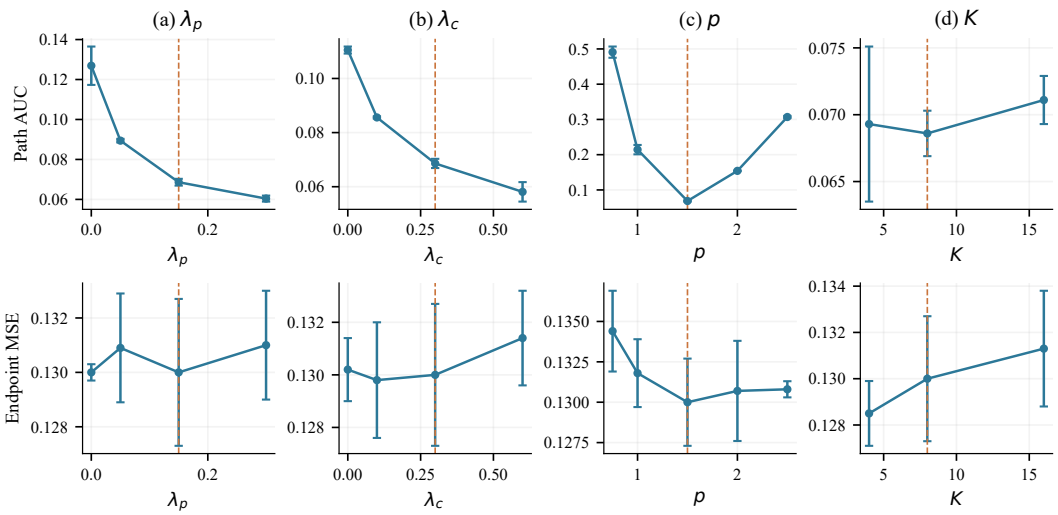

Figure S6: Sensitivity of path AUC and endpoint MSE to prefix weight, component weight, progress power, and token count. Points are two-seed means, bars are sample SD, and connecting lines are visual guides. Dashed vertical lines mark defaults. All power variants use the same power-1.5 evaluation path. Each axis has its own scale.

S5.2 Progress power and token count

With the common power-1.5 evaluation path, mean AUC is 0.0686 at $p = 1.5$, compared with 0.2143 at $p = 1$ and 0.1541 at $p = 2$. This comparison measures sensitivity to the training schedule relative to a fixed target path. For $K = 4, 8, 16$, mean path AUC is $0.0693, 0.0686, 0.0711$ and endpoint MSE is $0.1285, 0.1300, 0.1313$. Four and eight tokens yield similar path means relative to their seed variability; sixteen tokens have higher mean path and endpoint errors under this budget. Changing K also changes the prefix grid used to calculate path AUC.

S6 Generation and pretrained-model comparisons

S6.1 Short-budget generation

Tables 4 and S12 give low-resolution and Inception-feature distances across eight settings. The sample counts and training budgets follow Appendix S2. EDM has the lowest low-resolution distance in seven settings, and direct dense prediction leads on full ImageNet-256. CoFiTok has the lowest Inception-feature distance on CIFAR-10: 312.00, compared with 359.50 for dense prediction and 351.17 for EDM. On full ImageNet-256, its distance is 417.26, compared with 354.13 for dense prediction. CoFiTok's mean ranks are 3.12 in the low-resolution comparison and 2.62 in the Inception-feature comparison.

S6.2 ImageNet-256 generation controls

Table S13 evaluates 4,096 samples per method and seed. CoFiTok has the lowest Inception-feature distance for both seeds, while the low-resolution winner differs by seed. This larger evaluation complements the lower-sample broad comparison.

S6.3 Pretrained reconstruction and generation

Table S14 evaluates FlexTok and TiTok image reconstruction (Bachmann et al., 2025; Yu et al., 2024) at 256×256 on 256 images per dataset. Lower-resolution source images are evaluated at this common resolution. FlexTok PSNR ranges from 19.12 to 26.77 dB and TiTok from 15.54 to 19.35 dB. These image-content reconstruction results provide context for the ordered representations

Table S12: Inception-feature generation distances. Evaluation uses 1,024 generated and 4,096 reference images at 32/64 pixels, and 512 generated and 2,048 reference images at 256 pixels. Lower is better.

	Inception-feature distance↓			
Dataset	CoFiTok	Direct dense predictor	Improved DDPM	EDM
CIFAR-10	312.00	359.50	498.65	351.17
Tiny ImageNet-200	373.63	392.40	381.24	268.72
ImageNet-64 (HF)	390.99	417.79	378.33	258.40
ImageNet-64 (strict)	374.68	383.72	355.11	248.31
FFHQ-64	425.71	427.05	369.01	325.49
AFHQv2-64	331.06	333.42	335.18	230.40
ImageNet-256 10%	414.19	366.42	419.60	368.68
ImageNet-256	417.26	354.13	414.30	368.43

Table S13: ImageNet-256 generation controls. Evaluation uses 4,096 samples per method and seed. Lower is better.

Seed	Method	Samples	Lowres Frechet	Inception Frechet
103	Endpoint-only factorized	4096	11.1619	370.3672
103	Direct dense	4096	9.7396	449.3913
103	CoFiTok	4096	10.8232	349.4076
139	Endpoint-only factorized	4096	8.9753	404.6116
139	Direct dense	4096	10.5597	384.8126
139	CoFiTok	4096	11.0839	356.9311

discussed in Section 2. Recent pixel-space and unified tokenizer models also connect representation learning and synthesis (Liu et al., 2026; Yue et al., 2026).

Table S14: Pretrained tokenizer reconstruction at 256 pixels using 256 images per dataset. Source px gives the source resolution. PSNR is in dB; proxy is the low-resolution Fréchet-style distance.

Dataset	Source px	Flex PSNR	Flex proxy	TiTok PSNR	TiTok proxy
AFHQv2-64	64	24.04	0.036	17.29	0.217
CIFAR-10	32	26.77	0.037	19.35	0.168
ImageNet-64 strict	64	22.57	0.042	16.73	0.196
FFHQ-64	64	24.06	0.042	17.39	0.231
ImageNet-64 HF	64	23.60	0.039	18.04	0.182
ImageNet-256	256	19.12	0.044	15.54	0.196
ImageNet-256 10%	256	19.12	0.044	15.54	0.196
Tiny-IN	64	21.81	0.044	17.30	0.160

Table S15 gives pretrained ImageNet-256 generation results using 50,000 samples. The FIDs are 2.628 for D-AR, 2.338 for MAR, and 2.219 for ReTok. MAR's Inception Score of 62.90 uses an ADM-style evaluator. These models use their respective training and evaluation protocols, providing large-scale context alongside our short-budget study.

Table S15: Pretrained ImageNet-256 generation with 50,000 samples. FID and sFID are lower-is-better; IS denotes Inception Score. Each method uses its own training and evaluation protocol.

Model	Images	FID	sFID	IS	Precision	Recall
D-AR	50,000	2.628	6.720	285.89	0.803	0.588
MAR	50,000	2.338	4.700	62.90	0.822	0.572
ReTok	50,000	2.219	5.896	245.94	0.816	0.599

S7 IMPLEMENTATION SPECIFICATION

The mathematical interface specifies token-only synthesis, the additive noise estimate, and the relationship between pixel and noise targets. The experiments provide aggregate scores, seed standard deviations, and order summaries. Full replication additionally requires the exact predictor architecture and token shapes, projection and convolution configurations, optimizer and learning-rate schedule, forward noise schedule, preprocessing and dataset splits, and per-image predictions. These implementation details are absent from the accompanying materials.

The intermediate target filter, interpolation procedure, and reductions used in the prefix and component losses are also unspecified. The default progress power is $p = 1.5$. Reconstructing the target sequence requires those operations in addition to its endpoint conditions. Component-energy aggregation and zero-token normalization, the bootstrap confidence level and resampling procedure, the full permutation-score matrix, and the qualitative grids' row-to-prefix mapping and display normalization remain unspecified. These details are needed for an independently repeatable implementation of the experiments.