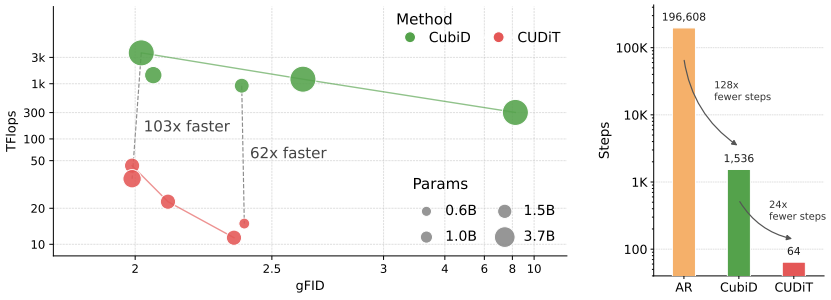
CUDiT: Cubic Uniform Diffusion for Efficient Generation of High-Dimensional Discrete Visual Representations

Anonymous authors
Paper under double-blind review

Abstract

Discrete visual generation casts image synthesis as token prediction, providing a modeling interface shared with language and a natural path toward unified multimodal architectures. Recent visualtokenizers pursue larger effective vocabularies to reduce quantization error and higher-dimensional representations to retain the semantic information required for visual understanding. Conventional quantization methods face substantial challenges when scaling to high-dimensional representations and large codebooks. Dimension-wise quantization provides a simple and effective route to near-lossless quantization, but induces an enormous and highly redundant product space. The enormous code space makes direct joint categorical modeling intractable, so practical generation must rely on parallel factorization. The existing absorbing-diffusion approach uses large sampling budgets in this space and reports substantially lower quality at fewer steps. Our analysis shows that the resulting mismatch between the product of coordinate-wise marginals and the joint distribution becomes increasingly severe as fewer sampling steps require larger parallel updates. To address this problem, we propose CUDiT (**C**ubic **U**niform **Di**ffusion **T**ransformer), a full-support discrete diffusion framework for efficient parallel generation of high-dimensional representation cubes. By denoising through continuous-time Markov chain dynamics over noisy token states, CUDiT avoids the severe degradation that arises when a model trained only on clean tokens encounters the intrinsic noise of parallel generation. On ImageNet-256, CUDiT-XL achieves gFID 1.99 in 64 steps, delivering comparable generation quality with up to two orders of magnitude fewer estimated backbone FLOPs than CubiD.



(a) gFID vs. TFlops on ImageNet-256. Circle size denotes model size.

(b) Steps to generate $768\times16\times16$ tokens.

Figure 1: **Efficiency of CUDiT.** Left: ImageNet-256 quality–compute trade-off of CubiD vs. CUDiT. Marker size scales with model parameter count, and solid lines connect evaluations of the same model scale at different inference-step budgets. Right: steps required by different methods to generate a $768\times16\times16$ dimension-wise token cube.

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.

1 INTRODUCTION

Discrete visual generation represents images as token sequences, giving vision and language a shared prediction interface for unified multimodal models (Team, 2024; Chang et al., 2022; Sun et al., 2024). Within this interface, the visual tokenizer determines both the fidelity of the representation and the information available to downstream models. Recent tokenizers have therefore advanced along two complementary axes: enlarging the effective vocabulary to reduce quantization error (Yu et al., 2024a; Zhuang et al., 2026b;a; Ai et al., 2026), and increasing the representation dimension to retain the semantics of foundation vision features (Oquab et al., 2023; Tschannen et al., 2025; Zheng et al., 2026).

Despite these advances, scaling discrete visual representations along either axis remains difficult for conventional vector quantization. High-dimensional semantic features make samples sparse in the ambient space and render codebook clustering increasingly unreliable, while larger vocabularies require rapidly growing numbers of codewords to maintain adequate coverage (Van Den Oord et al., 2017; Esser et al., 2021). Dimension-wise quantization sidesteps both difficulties by discretizing each scalar independently with a small, training-free codebook rather than clustering full feature vectors (Wang et al., 2025; 2026). This simple construction preserves native high-dimensional RAE features with nearly lossless quantization and realizes a large effective vocabulary through the Cartesian product of scalar codebooks (Wang et al., 2025; 2026). However, this advantage comes with a structural cost: the Cartesian product produces an enormous and highly redundant implicit code space. A $768\times16\times16$ representation contains $196{,}608$ categorical coordinates, while valid configurations occupy only a sparse subset of the product space.

This trade-off leaves a fundamental question unanswered: how can such a large discrete cube be generated efficiently without sacrificing quality? Cubic Discrete Diffusion (CubiD) (Wang et al., 2026) applies absorbing mask diffusion to dimension-wise quantized RAE features, reducing the sequential budget from $\mathcal{O}(10^5)$ autoregressive steps ($196{,}608$ for a $768\times16\times16$ cube) to 1536 denoising steps while achieving competitive ImageNet-256 quality (Figure 1b). However, this improvement does not extend to aggressive few-step inference. For CubiD-XXL, reducing the budget from 1536 to 128 steps worsens gFID from 2.02 to 8.25 and lowers IS from 214.8 to 155.8 (Table 1). The intermediate results at 512 and 256 steps show the same monotonic deterioration. As the budget decreases, each update predicts more correlated coordinates in parallel, and generation quality declines sharply. We therefore seek an efficient generator for high-dimensional, dimension-wise quantized representations that remains effective at small step budgets.

Table 1: **CubiD generation quality degrades sharply with fewer inference steps.**

Steps	gFID↓	IS↑
128	8.25	155.8
256	3.96	185.9
512	2.63	201.5
1536	2.02	214.8

This limitation points to a deeper modeling issue: the full joint distribution over the discrete cube is too large to parameterize directly. Practical generators replace an intractable joint categorical distribution with coordinate-wise heads and sample many variables in parallel. For correlated coordinates, however, the resulting product of marginals need not reproduce the joint distribution, even when every marginal is exact. We characterize this discrepancy as *parallel factorization error* through conditional total correlation (Section 3). As the step budget shrinks, each parallel block grows and incorporates more cross-coordinate dependencies, causing this error to become increasingly severe. The resulting noisy content states are then fed back to a denoiser trained only on clean tokens and masks, creating a train–inference gap precisely in the few-step regime.

Motivated by this observation, we propose CUDiT (**C**ubic **U**niform **Di**ffusion **T**ransformer), a full-support discrete diffusion framework for efficient generation of high-dimensional representation cubes. CUDiT couples a factorized uniform source with the data distribution through a prescribed probability path and learns the corresponding continuous-time Markov chain transition rates (Gat et al., 2024). Unlike clean-plus-mask training, this formulation places noisy content states directly on the training path while retaining parallel prediction across the representation cube. On ImageNet-256, CUDiT-XL achieves gFID 1.99 in 64 steps, compared with gFID 2.06 in 1536 steps for CubiD-XL, requiring $24\times$ fewer inference steps. Across comparisons with CubiD, CUDiT reduces estimated backbone compute by up to two orders of magnitude while maintaining comparable generation quality (Figures 1a and 1b).

Taken together, our contributions are summarized as follows:

- We analyze the modeling error induced by parallel factorization in high-dimensional discrete code spaces and relate its joint-distribution discrepancy to conditional total correlation.
- We introduce CUDiT, a cubic uniform diffusion framework that combines a full-support uniform path with CTMC sampling for efficient parallel generation of representation cubes.
- On ImageNet-256, CUDiT-XL reaches gFID 1.99 in 64 steps, matching CubiD-XL with $24\times$ fewer inference steps. Across model-scale comparisons, CUDiT reduces estimated backbone compute by up to two orders of magnitude at comparable generation quality.

2 Related Work

Visual tokenization. Visual tokenization provides images with a discrete interface for generative modeling, allowing image synthesis to be formulated as token prediction. It involves two design stages: constructing a reconstruction representation and discretizing that representation into a token space. For the first stage, classical VAE and VQ-based tokenizers use convolutional encoders and decoders to compress images into low-dimensional feature maps (Kingma & Welling, 2013; Van Den Oord et al., 2017; Esser et al., 2021). Transformer-based tokenizers replace or augment these components with vision transformers and more token-efficient bottlenecks (Yu et al., 2021; 2024b). RAE takes a different route by using the native output of a frozen semantic encoder as the representation space and training a decoder to reconstruct images from those features (Zheng et al., 2026). For the second stage, vector quantization maps each latent vector to a learned codeword (Van Den Oord et al., 2017; Esser et al., 2021), while residual quantization composes multiple discrete codes at each location (Lee et al., 2022). Lookup-free and binary schemes replace explicit nearest-neighbor codebook lookup with channel-wise discrete codes (Yu et al., 2024a; Zhuang et al., 2026b;a; Ai et al., 2026). Unlike conventional quantizers that learn their codebooks or discretization parameters during tokenizer training, dimension-wise quantization independently maps each scalar to a fixed, small codebook and requires no quantizer training (Wang et al., 2025; 2026). These two stages determine the geometry of the representation and the structure of the resulting discrete space.

Autoregressive visual generation. Autoregressive visual generators factorize the distribution of discrete image tokens into causal predictions. Early systems model compact VQ hierarchies or rasterized VQGAN tokens, with residual quantization providing a deeper code hierarchy per spatial position (Razavi et al., 2019; Esser et al., 2021; Yu et al., 2021; Lee et al., 2022). Recent work scales decoder-only next-token models (Sun et al., 2024; Yu et al., 2024a), shortens the token sequence through stronger tokenizers (Yu et al., 2024b), or changes the prediction order through next-scale and randomized generation (Tian et al., 2024; Pang et al., 2025; Yu et al., 2025). Representation-based tokenizers and binary product codes further extend AR generation toward semantic features and much larger effective vocabularies (Zheng et al., 2025; Wang et al., 2025; Ai et al., 2026). These designs improve token quality, ordering, or sequence length, but causal decoding still scales with the number of autoregressive units.

Discrete diffusion visual generation. Discrete diffusion reduces sequential depth by updating many visual tokens in parallel. These models learn to reverse a prescribed token corruption process. Common choices use absorbing-mask, uniform, or structured transition kernels (Austin et al., 2021; Lou et al., 2023). Absorbing-state transitions are the predominant choice in visual generation because they map naturally to masked-token prediction. MaskGIT and related methods start from an all-mask state and iteratively reveal subsets of low-dimensional visual tokens (Chang et al., 2022; Lou et al., 2023; Nie et al., 2026). CubiD extends this formulation to high-dimensional representation cubes by masking individual elements across both spatial and channel axes, allowing partial observations to expose within-position and cross-position dependencies (Wang et al., 2026). These formulations differ in their source distributions and intermediate-state support. An absorbing path begins from a dedicated mask state and trains on mixtures of clean content tokens and masks. A uniform path instead begins from independent samples in the content vocabulary and retains full support over content-token configurations throughout the path (Austin et al., 2021; Gat et al., 2024). The corresponding denoiser parameterizes token-wise CTMC probability velocities that transport the uniform source toward the data distribution (Gat et al., 2024; Campbell et al., 2024).

3 Efficient Generation of High-Dimensional Discrete Visual Representations

3.1 Background: From higher dimension to larger codebooks

Discrete visual representations are scaling along two distinct axes: higher-dimensional features retain more semantic information, while larger codebooks reduce quantization error.

Higher dimension. Classical visual tokenization pipelines encode images into compressed reconstruction latents with only 8–32 channels (Chang et al., 2022; Sun et al., 2024; Tian et al., 2024; Yu et al., 2024b). One complementary direction distills semantic teachers into reconstruction tokenizers (Zhuang et al., 2026a; Zheng et al., 2025; Du et al., 2026). These methods introduce semantic supervision while retaining a compressed tokenizer space whose ambient channel dimension remains limited relative to native foundation-model features. Although this compact interface makes generation tractable, reconstruction-oriented compression can discard semantic information needed for visual understanding (Zheng et al., 2026; Wang et al., 2026). This limitation has motivated shared feature spaces for understanding and reconstruction that directly preserve pretrained vision features (Zheng et al., 2026). RAE follows this direction by reconstructing images directly from native semantic features with 768–1024 channels, one to two orders of magnitude wider than conventional generative latents (Oquab et al., 2023; Tschannen et al., 2025; Zheng et al., 2026). More recently, S-AE regularizes end-to-end fine-tuning of a semantic encoder and reaches rFID 0.06 while retaining 81.9% ImageNet classification accuracy, further illustrating the potential of high-dimensional latent spaces to support both understanding and near-lossless reconstruction (Li et al., 2026).

Larger codebooks. A parallel trend enlarges the effective vocabulary to reduce quantization error. Classical VQ tokenizers typically use codebooks of size 2^{10}–2^{14} (Van Den Oord et al., 2017; Esser et al., 2021; Sun et al., 2024). Lookup-free and group-wise binary schemes push far beyond this regime: WeTok scales the codebook to 2^{32} and reports ImageNet rFID as low as 0.12 under a high-fidelity setting (Zhuang et al., 2026b), while UniWeTok further expands to 2^{128} and, at $32\times$ spatial compression with only 8×8 tokens, reaches rFID 0.79 versus 4.99 for a 16384-entry VQGAN baseline (Zhuang et al., 2026a; Esser et al., 2021). Larger codebooks generally reduce quantization error, but also produce enormous code spaces that are typically highly redundant. Dimension-wise quantization discretizes each scalar with only 3–4 bits while preserving the full channel axis (Wang et al., 2025; 2026). For a 768-channel feature vector, the resulting code space reaches size $2^{768\times3}$. Experiments show that the resulting discrete features achieve nearly lossless quantization relative to their continuous counterparts.

3.2 Few-Step Inference Amplifies Parallel Factorization Error

The enormous product space induced by dimension-wise quantization cannot be represented by a single categorical head. Practical generators therefore factorize it into small coordinate-wise vocabularies and predict their categorical distributions in parallel (Zhuang et al., 2026b;a; Wang et al., 2026). This approximation introduces a distinct modeling error: the product of coordinate-wise marginals generally differs from the joint distribution of the correlated coordinates. We refer to this discrepancy as *parallel factorization error*.

To separate parallel factorization error from ordinary prediction error, assume that every coordinate marginal is exact. Given resolved coordinates C, independently sampling a block U uses

$$q(X_U \mid X_C) = \prod_{i \in U} p(X_i \mid X_C) \tag{1}$$

in place of the joint conditional. We quantify the parallel factorization error by its average KL divergence, which equals the conditional total correlation:

$$\mathcal{E}_{\mathrm{PF}}(U \mid C) := \mathbb{E}_{X_C}[D_{\mathrm{KL}}(p(X_U \mid X_C) \,\|\, q(X_U \mid X_C))] = \mathrm{TC}(X_U \mid X_C). \tag{2}$$

Thus, exact marginal prediction does not guarantee an accurate joint sample. The error is controlled by dependence within the simultaneously sampled block rather than by marginal accuracy alone.

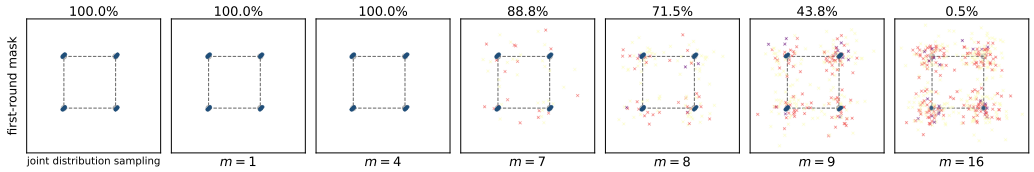


Figure 2: **Parallel factorization error grows with parallel block size** (k=8, d=16). Columns compare joint sampling with exact-marginal parallel sampling at $m \in \{1, 4, 7, 8, 9, 16\}$. Despite exact marginals, first-round validity follows the 2^{k-m} bound once m exceeds k.

Few-step inference directly increases the scale of this error. A decoder that generates d coordinates in S steps must update roughly $m \approx d/S$ coordinates per step. Reducing S therefore enlarges the parallel block, incorporates more cross-coordinate dependencies into the product approximation, and can sharply increase $\mathcal{E}_{\mathrm{PF}}$. This relation explains why parallel factorization can be relatively benign at large sampling budgets but become substantial under aggressive few-step inference. Appendix B provides the full derivation and scope.

The linear-code model below captures the relevant geometry in a controlled form. Prior work on pixel and representation-space generation finds that image data and high-dimensional pretrained visual features concentrate near low-dimensional manifolds (Li & He, 2026; Zhang et al., 2026). Consequently, an RAE feature space can have a large ambient dimension while remaining highly redundant, because relatively few intrinsic degrees of freedom constrain many feature coordinates. We use a linear code to model this setting: it embeds a k-dimensional latent variable into a d-dimensional redundant representation while making the consistency cost of parallel factorization exactly tractable.

Let $Z \sim \mathrm{Unif}(\mathbb{F}_2^k)$ and $X = GZ \bmod 2$, with $\mathrm{rank}(G) = k$. When m non-deterministic coordinates are sampled independently in one parallel update, exact counting gives

$$P_{\mathrm{valid}}(U) = 2^{r_U - m} \leq 2^{k-m}. \tag{3}$$

where $r_U = \mathrm{rank}(G_U)$. Although every marginal is exact, the probability of obtaining a jointly valid block can decay exponentially once the parallel block size exceeds the intrinsic dimension. Because m grows as the step budget decreases, the result gives a direct mechanism by which parallel factorization error is amplified during few-step inference (Figure 2). Appendix B gives the counting argument. The analysis and controlled experiment isolate the contribution of parallel factorization itself: increasing the number of correlated coordinates sampled per update enlarges the mismatch between the factorized proposal and the target joint distribution. Although the consistency probability in this stylized first update can decay exponentially, the generalization of a learned neural generator means that generation quality does not typically deteriorate exponentially over a complete trajectory.

3.3 CUDiT: Cubic Uniform Diffusion for High-Dimensional Discrete Tokens

Because parallel factorization error grows sharply as the inference budget shrinks, we hypothesize that it is an important source of CubiD's few-step degradation. CubiD is trained on states containing only clean tokens and masks, yet later inference steps condition on content values sampled by earlier parallel updates. We first augment mask-diffusion training with random perturbations to the visible content tokens, but observe only limited improvement. Treating noisy content as an auxiliary perturbation to an otherwise absorbing path is therefore insufficient in our experiments, motivating a diffusion process in which noisy content states are part of the training path itself.

Uniform diffusion addresses this problem with a full-support corruption process. Training states contain alphabet-valid noisy content rather than only clean tokens and masks, allowing the denoiser to learn directly on the type of state produced during parallel generation. This principle motivates CUDiT, a cubic uniform diffusion model that transports a uniform token source toward the data distribution through continuous-time Markov chain (CTMC) denoising for few-step generation of dimension-wise quantized RAE features. Concretely, a frozen encoder yields $\boldsymbol{z} = E(\boldsymbol{x}) \in \mathbb{R}^{h \times w \times d}$ (default DINOv2-B, h=w=16, d=768), which we map to a discrete cube by

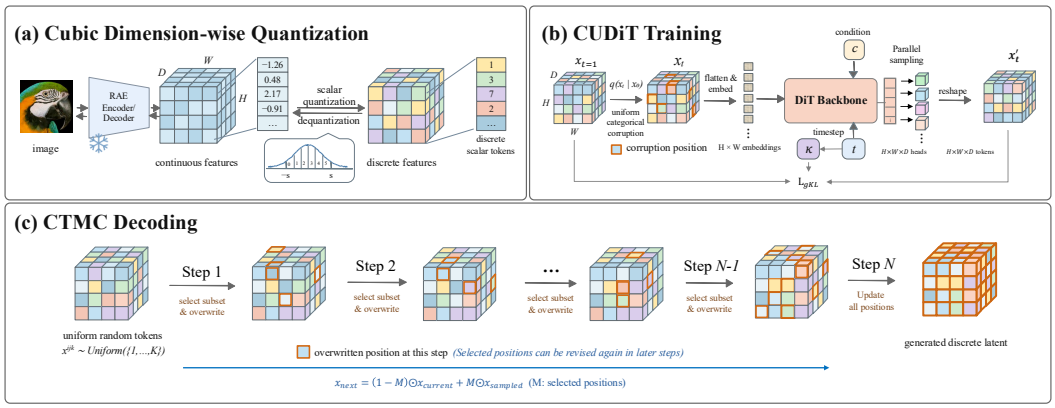


Figure 3: **Overview of CUDiT.** (a) High-dimensional token discretization: a frozen RAE encoder yields continuous $h\times w\times d$ features that are mapped to discrete cubes via dimension-wise quantization (Zheng et al., 2026; Wang et al., 2026). (b) Full-support probability-path training interpolates between random content tokens and the quantized cube. A LightningDiT denoiser predicts per-site categorical heads and updates tokens by parallel CTMC sampling. (c) Few-step inference starts from a uniform source and iteratively denoises the content-token cube (schedule details in Section 4.2).

training-free dimension-wise quantization (Wang et al., 2025; 2026)

$$q_{x,y,i} = \mathrm{Quantize}(z_{x,y,i}; L) \in \{0, \dots, L-1\}, \tag{4}$$

with $L{=}2^b{=}8$ and Gaussian reconstruction centroids at decode time. The generative target is $\boldsymbol{q} \in \{0, \dots, L-1\}^{h\times w\times d}$. CubiD models this cube with absorbing mask diffusion, using masked prediction $p_\theta(q_i \mid \boldsymbol{q}_{\bar{M}})$ and progressive unmasking from an all-mask state (Wang et al., 2026; Chang et al., 2022). This formulation makes high-dimensional discrete modeling tractable, but typically requires $T \in [512, 1536]$ steps and is strongly affected by the parallel factorization error of Section 3 under few-step budgets.

Figure 3 summarizes CUDiT. Its core is a full-support probability path (Gat et al., 2024) that couples independent uniform tokens with data tokens. The denoiser learns the clean-token posterior along this path, which determines the CTMC transition rates used at inference time. Detailed path and rate derivations are deferred to Appendix A.

Identifying the cube with a sequence in $[L]^N$ ($N{=}hwd$), we couple data $x_1 = \boldsymbol{q}$ to an independent uniform source

$$x_0^i \sim \mathrm{Unif}[L], \tag{5}$$

for all positions i and define the factorized mixture path

$$p_t(x^i \mid x_0, x_1) = (1-\kappa_t)\,\delta_{x_0^i}(x^i) + \kappa_t\,\delta_{x_1^i}(x^i), \tag{6}$$

with monotone scheduler $\kappa_0{=}0$, $\kappa_1{=}1$ (noise-schedule choices in Section 4.2). A LightningDiT-style denoiser $p_\theta(\cdot \mid x_t, t, y)$ approximates the posterior $p_{1|t}$ under the rate-weighted generalized KL objective of Gat et al. (2024):

$$\mathcal{L}_{\mathrm{gKL}}(\theta) = -\mathbb{E}\left[\sum_{i=1}^{N} \frac{\dot\kappa_t}{1-\kappa_t}\left(p_\theta(x_t^i \mid x_t, t, y) - \mathbf{1}_{x_t^i = x_1^i} + \mathbf{1}_{x_t^i \neq x_1^i} \log p_\theta(x_1^i \mid x_t, t, y)\right)\right]. \tag{7}$$

The expectation samples $t \sim \mathcal{U}(0, 1-\varepsilon)$, $x_0 \sim \mathrm{Unif}[L]^N$, and $x_t \sim p_t(\cdot \mid x_0, x_1)$ for a data token cube x_1. The rate factor $\dot\kappa_t/(1-\kappa_t)$ aligns training with the CTMC jump intensity, and the objective has the clean-token posterior $p_{1|t}$ as its population minimizer. Every training state contains only alphabet-legal content tokens, with no special mask symbol, exposing the denoiser to noisy content contexts during training.

Generation follows an Euler discretization of the learned CTMC probability velocity (Gat et al., 2024). Starting from $x_0 \sim \mathrm{Unif}[L]^N$, at time t the denoiser predicts the clean-token posterior and forms coordinate-wise jump rates

$$u_t^i(x^i, z) = \frac{\dot\kappa_t}{1-\kappa_t}\Big(p_{1|t}(x^i \mid z) - \delta_{z^i}(x^i)\Big). \tag{8}$$

Table 2: **Performance comparison of discrete generation methods on ImageNet (Russakovsky et al., 2015)** 256×256**.**

Method	Latent Dim	#Params	Steps↓	TFlops↓	gFID↓	IS↑	Prec.↑	Rec.↑
Discrete Diffusion Models with Low-dimensional Tokens								
MaskGIT (Chang et al., 2022)	16	227M	8	0.74	6.18	182.1	0.80	0.51
VQ-Diffusion (Gu et al., 2022)	16	518M	100	24.8	11.89	-	-	-
Token-Critic (Lezama et al., 2022a)	16	368M	36	3.04	4.69	174.5	0.76	0.51
DPC (Lezama et al., 2022b)	16	454M	180	16.6	4.45	244.8	-	-
TiTok-S-128 (Yu et al., 2024b)	16	287M	64	4.46	-	-	-	-
Discrete Autoregressive Models with Low-dimensional Tokens								
VQVAE2 (Razavi et al., 2019)	64	13.5B	5120	–	31.11	45	0.36	0.57
VQGAN (Esser et al., 2021)	128	1.4B	256	0.71	15.78	74.3	-	-
ViT-VQGAN (Yu et al., 2021)	32	1.7B	1024	–	4.17	175.1	-	-
RQTran. (Lee et al., 2022)	16	3.8B	64	0.67	7.55	134.0	-	-
LlamaGen-XXL (Sun et al., 2024)	8	1.4B	576	1.62	14.6	86.3	0.63	0.68
VAR (Tian et al., 2024)	32	2.0B	10	4.71	2.16	288.7	0.81	0.61
VFMTok-XXL (Zheng et al., 2025)	12	1.4B	256	0.71	1.95	259.3	0.82	0.62
VFMTok-3B (Zheng et al., 2025)	12	3.1B	256	1.54	2.04	267.6	0.82	0.61
Discrete Models with High-dimensional Tokens								
VQRAE (Du et al., 2026)	1536	0.6B	–	–	–	–	–	–
CubiD-L (Wang et al., 2026)	768	946M	1536	921.5	2.38	213.1	0.84	0.57
CubiD-XL (Wang et al., 2026)	768	1.4B	1536	1,430.2	2.06	216.4	0.83	0.58
CubiD-XXL (Wang et al., 2026)	768	3.7B	1536	3,624.3	2.02	214.8	0.81	0.61
CUDiT-B (ours)	768	0.6B	64	14.9	2.39	227.6	0.81	0.61
CUDiT-L (ours)	768	1.0B	64	22.7	2.11	236.0	0.80	0.61
CUDiT-XL (ours)	768	1.6B	64	35.3	1.99	249.1	0.80	0.62

For a step of size h, coordinate i is sampled from $\delta_{x_t^i} + h\, u_t^i(\cdot, x_t)$, yielding the next denoised content cube x_{t+h}. We optionally apply temperature τ via $\mathrm{softmax}(\ell/\tau)$ before forming rates. Unless otherwise stated, we use $T{=}64$ steps.

The network reuses CubiD's spatial packing (Wang et al., 2026): each spatial location is one transformer token of width d, obtained by dequantizing the d discrete codes at that site. A LightningDiT backbone (Peebles & Xie, 2023; Zheng et al., 2026) processes the $h{\times}w$ sequence with bidirectional attention, conditioned on class label and time via adaLN. It emits $d{\times}L$ logits per site, with one categorical head per channel. The sequence length is hw. The high channel dimension widens the input and output layers without increasing the attention length to hwd.

4 Experiments

4.1 Main Results

We train CUDiT-B/L/XL for class-conditional generation on ImageNet-1K (Russakovsky et al., 2015) at 256×256, following the training recipe in Appendix C (Table 7). The models use a frozen DINOv2-B representation encoder, dimension-wise quantization, and LightningDiT backbones at three parameter scales. Their architecture configurations are listed in Table 3. Unless otherwise specified, we evaluate the final 800-epoch checkpoints on 50k samples using 64 inference steps, temperature $\tau{=}0.75$, and no classifier-free guidance or auxiliary sampling modules. We report gFID (Heusel et al., 2017), Inception Score (Salimans et al., 2016), and Precision/Recall (Kynkäänniemi et al., 2019).

Table 3: **Model sizes and architecture configurations of CUDiT.**

Model	Hidden	Blocks	#Params
CUDiT-B	1152	28	0.6B
CUDiT-L	1536	24	1.04B
CUDiT-XL	1920	24	1.62B

Table 2 presents the main comparison. Methods are grouped into discrete diffusion with low-dimensional tokens, discrete autoregressive models with low-dimensional tokens, and discrete models with high-dimensional tokens. Both CubiD and CUDiT generate native 768d representation

tokens. For a fair comparison, baseline results are likewise reported without CFG. Latent Dim denotes the dimensionality of the latent representation produced by the encoder. TFlops are estimated backbone FLOPs per image where available.

CUDiT attains comparable or better gFID than CubiD with substantially fewer evaluations in the high-dimensional token regime. At the XL scale, it reaches gFID 1.99 in 64 steps, compared with gFID 2.06 in 1536 steps for CubiD-XL. This corresponds to 24× fewer inference steps and about 40.5× lower estimated backbone compute per image (35.3 vs. 1430.2 TFlops). What's more, CUDiT-XL attains gFID 1.99 with about 102.7× lower estimated backbone compute (35.3 vs. 3624.3 TFlops) compared with CubiD-XXL (gFID 2.02). CUDiT also achieves higher IS and recall than CubiD, while attaining comparable or better distribution-level quality as reflected by its lower gFID. Qualitative samples generated by CUDiT-XL with 64 inference steps are provided in Appendix D (Figure 5).

4.2 Studies of Training and Inference Recipes

We ablate the noise schedule and sampling temperature, and study scaling with inference steps and model size.

Noise scheduler. The noise scheduler κ_t controls the weighting of noise levels during training and sampling, which strongly affects quality in our high-dimensional setting (Zheng et al., 2026; Zhang et al., 2026). We compare three probability-path schedules: linear and quadratic interpolation (Gat et al., 2024), and dimension-aware time shifting (Zheng et al., 2026; Zhang et al., 2026). For this schedule ablation, we train CUDiT-B for 100 epochs and evaluate every variant with 64 inference steps at temperature τ=1. Time shifting uses $\alpha = \sqrt{hwd/m}$ with m=4096 ($\alpha \approx 6.9$) and reaches gFID 3.95 and IS 192.2, compared with gFID 10.53 and 8.01 for the linear and quadratic schedules (Table 4).

Table 4: **Effect of different noise schedules on CUDiT.**

Method	Formula	gFID↓	IS↑
Linear	$\kappa_t = t$	10.53	136.3
Quadratic	$\kappa_t = t^2$	8.01	137.5
Time-shift	$\kappa_t = \frac{t}{\alpha-(\alpha-1)t}$	3.95	192.2

Table 5: **Effect of inference steps and model size on CUDiT.**

(a) Inference steps.

T	gFID↓	IS↑
32	2.35	235.1
64	2.11	236.0
128	1.99	239.2

(b) Model size.

Model	#Params	gFID↓	IS↑
B	0.6B	2.39	227.6
L	1.0B	2.11	236.0
XL	1.6B	1.99	249.1

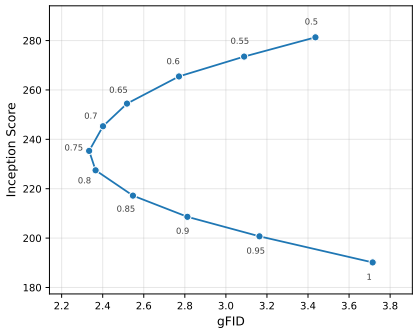


Figure 4: **Temperature sweep.**

Inference temperature. We evaluate how sampling temperature affects generation quality. Using a fixed CUDiT-L checkpoint trained for 400 epochs, we sweep a constant temperature $\tau \in [0.5, 1.0]$ without retraining. All settings use 64 inference steps and 50k evaluation samples. Figure 4 reports the results. gFID follows a U-shaped trend. Relative to τ=1.0 (gFID 3.71), moderate cooling improves generation quality and reaches its best value of gFID 2.33 near τ=0.75. Further lowering the temperature to τ=0.5 degrades gFID to 3.44. In contrast, IS rises monotonically as τ decreases, from 190 at τ=1.0 to 235 at τ=0.75 and 281 at τ=0.5. This pattern is consistent with moderate cooling tempering noisy parallel updates, while excessive cooling makes sampling overly concentrated.

Scaling inference steps. Table 5a varies the number of steps T for CUDiT-L after 800 epochs at τ=0.75. The model reaches gFID 2.35 at T=32 and 1.99 at T=128. This limited sensitivity is consistent with full-support noisy-content training making the denoiser more robust to the factorization errors induced by large parallel updates, and contrasts with the sharp low-budget degradation observed for CubiD.

Scaling model size. Table 5b compares CUDiT-B/L/XL on a fixed DINOv2-B interface under the same few-step protocol (T=64, τ=0.75). gFID improves from 2.39 (B) to 1.99 (XL) as capacity

grows, and IS rises from 227.6 to 249.1. Across these three scales, both gFID and IS improve monotonically with model capacity.

4.3 Revisability Is Not the Primary Driver of CUDiT's Gains

One possible explanation for CUDiT's improvement is that CTMC sampling can rewrite previously sampled tokens, rather than that uniform diffusion provides a better training distribution. We directly test this alternative explanation with a three-way controlled ablation that separates revisability from corruption support. All variants use CUDiT-B trained from scratch for 100 epochs on the same DINOv2-B tokens, with the same LightningDiT-B backbone, data pipeline, optimizer, time-shift schedule, T=64 sampling budget, and native temperature τ=1. No additional inference enhancements are enabled. They differ only in the source distribution and whether a sampled content token may be revised (Table 6).

Irreversible Mask uses standard progressive unmasking: once a coordinate receives a content value, it remains fixed. *CTMC-Mask* retains the same all-mask source and clean-plus-mask training states, but uses CTMC updates that may rewrite a previously revealed coordinate. The comparison between these two variants holds the training corruption fixed and isolates the effect of revisability. *Uniform* retains the revisable CTMC sampler but replaces the mask source with full-support content corruption. Its comparison with CTMC-Mask holds the CTMC sampler and revisability fixed, isolating the effect of replacing clean-plus-mask training states with full-support noisy content states. Thus, if CTMC-Mask and Irreversible Mask perform similarly while Uniform improves, the gain cannot primarily be attributed to the ability to revise sampled tokens.

Table 6: **Ablation of revisability and corruption support.**

Method	Source	Revisable	gFID↓	IS↑
Irreversible Mask	Mask	No	26.64	87.7
CTMC-Mask	Mask	Yes	25.31	95.19
CTMC-Uniform	Uniform	Yes	3.95	192.2

Table 6 shows that enabling content revisions under otherwise unchanged mask corruption improves gFID from 26.64 to 25.31 and IS from 87.7 to 95.19. This gain is modest compared with CTMC-Uniform, which reaches gFID 3.95 and IS 192.2 under the same protocol. Because CTMC-Mask and CTMC-Uniform use the same revisable sampler, their large performance gap cannot primarily be attributed to revisability. The effective difference is the training distribution: Uniform learns from full-support noisy content states, whereas both mask variants learn from clean tokens and masks. This result supports full-support corruption as the important component and is consistent with improved robustness to noisy inference states. It does not require a revisability-based explanation.

5 Conclusion

We study the high sampling cost of generating dimension-wise quantized semantic representations. Our analysis identifies parallel factorization error as a central obstacle to few-step generation: reducing the step budget enlarges each parallel update and can sharply increase the mismatch between factorized predictions and the target joint distribution. These sampling errors create noisy content states that clean-plus-mask training does not cover. Guided by this analysis, CUDiT uses full-support corruption and CTMC denoising to train directly on content-valued noisy states. Our controlled ablation finds little difference between mask variants with and without content rewriting, whereas uniform corruption retains a clear advantage. This supports the viewpoint we advance: by covering the noisy content states induced by parallel factorization errors, full-support training can reduce the distribution gap between training and inference. The result makes revisability unlikely to explain the gain. On ImageNet-256, CUDiT-XL reaches gFID 1.99 in 64 steps, matching CubiD-XL with 24× fewer inference steps. Across model-scale comparisons, CUDiT reduces estimated backbone compute by up to two orders of magnitude at comparable generation quality.

AI USE STATEMENT

We used generative AI tools for drafting and polishing manuscript text. We did not use generative AI tools to invent experimental results. Reported numbers come from completed runs or cited sources. We reviewed all AI-assisted prose for technical correctness and take responsibility for the final content.

ETHICS STATEMENT

This work studies class-conditional image generation on ImageNet. It does not involve human subjects, sensitive personal data, or dual-use biological/chemical applications. Generated images may reflect biases present in ImageNet. We discourage deployment in high-stakes decision systems without additional fairness auditing.

REPRODUCIBILITY STATEMENT

We will release code, configs, and checkpoints. Section 3.3 specifies the path, loss, and sampler. Section 3 defines parallel factorization error and analyzes how larger parallel updates amplify it. Section 4.3 tests whether CUDiT's gain can instead be explained by revisability. Section 4.2 studies training and inference recipes, and Appendix B contains the full factorization analysis. Evaluation follows standard ImageNet 50k gFID protocols with fixed seeds where applicable.

REFERENCES

Yuang Ai, Jiaming Han, Shaobin Zhuang, Weijia Mao, Xuefeng Hu, Ziyan Yang, Zhenheng Yang, Yali Wang, Huaibo Huang, Xiangyu Yue, et al. Bitdance: Scaling autoregressive generative models with binary tokens. *arXiv preprint arXiv:2602.14041*, 2026.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, et al. Vqrae: Representation quantization autoencoders for multimodal understanding, generation and reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30322–30334, 2026.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10686–10696. IEEE, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11513–11522. IEEE, 2022.

José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *European Conference on Computer Vision*, pp. 70–86. Springer, 2022a.

Jose Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *The Eleventh International Conference on Learning Representations*, 2022b.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Xiaojie Li, Yang Zhao, Ming Li, Yancheng Zhang, Zonglin Lyu, Yunpeng Chen, Rui Wang, and Daquan Zhou. Unified latent space for understanding and generation via semantic auto-encoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2115–2124, 2026.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 45–55. IEEE, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18596–18605. IEEE, 2025.

Yuqing Wang, Chuofan Ma, Zhijie Lin, Yao Teng, Lijun Yu, Shuai Wang, Jiaming Han, Jiashi Feng, Yi Jiang, and Xihui Liu. Cubic discrete diffusion: Discrete visual generation on high-dimensional representation tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36072–36081, June 2026.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024a.

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, 2024b.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18431–18441. IEEE, 2025.

Le Zhang, Ning Mang, and Aishwarya Agrawal. Rit: Vanilla diffusion transformers suffice in representation space. *arXiv preprint arXiv:2605.21981*, 2026.

Anlin Zheng, Xin Wen, Xuanyang Zhang, Chuofan Ma, Tiancai Wang, Gang Yu, Xiangyu Zhang, and Xiaojuan Qi. Vision foundation models as effective visual tokenizers for autoregressive image generation. *arXiv preprint arXiv:2507.08441*, 2025.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Shaobin Zhuang, Yuang Ai, Jiaming Han, Weijia Mao, Xiaohui Li, Fangyikang Wang, Xiao Wang, Yan Li, Shanchuan Lin, Kun Xu, et al. Uniwetok: An unified binary tokenizer with codebook size 2^{128} for unified multimodal large language model. *arXiv preprint arXiv:2602.14178*, 2026a.

Shaobin Zhuang, Yiwei Guo, Fangyikang Wang, Canmiao Fu, Zhipeng Huang, Zeyue Tian, Xiaohui Li, Ying Zhang, Chen Li, and Yali Wang. Wetok: Powerful discrete tokenization for high-fidelity visual reconstruction. In *International Conference on Learning Representations*, volume 2026, pp. 6284–6314, 2026b.

A Probability Paths and CTMC Denoising

We provide the probability-path and CTMC-rate details used in Section 3.3, following the discrete transport formulation of Gat et al. (2024).

Setup and couplings. Let $[L] := \{0, \dots, L-1\}$ and $\mathcal{D} := [L]^N$. A source p and target q are coupled by $\pi(x_0, x_1)$ with $(X_0, X_1) \sim \pi$. The independent coupling $\pi(x_0, x_1) = p(x_0)q(x_1)$ is the default in our work. Common sources include the all-mask (absorbing) distribution and the factorized uniform p_{u}.

Probability paths. We construct a path $(p_t)_{t\in[0,1]}$ with $p_0{=}p$ and $p_1{=}q$. The marginal path mixes a conditional path under the coupling,

$$p_t(x) = \sum_{x_0,x_1\in\mathcal{D}} p_t(x \mid x_0, x_1)\, \pi(x_0, x_1), \qquad p_t(x \mid x_0, x_1) = \prod_{i=1}^{N} p_t(x^i \mid x_0, x_1), \tag{9}$$

where each coordinate path satisfies the boundary conditions $p_0(x^i \mid x_0, x_1) = \delta_{x_0^i}(x^i)$ and $p_1(x^i \mid x_0, x_1) = \delta_{x_1^i}(x^i)$. The convex mixture path used in the main text is Eq. equation 6. Sampling $X_t \sim p_t(\cdot \mid X_0, X_1)$ keeps X_0^i with probability $1 - \kappa_t$ and X_1^i otherwise.

Probability velocity and Euler updates. Generation is a CTMC on $\mathcal{D}$. Given $X_t \sim p_t$, each token updates independently by

$$X_{t+h}^i \;\sim\; \delta_{X_t^i}(\cdot) + h\, u_t^i(\cdot, X_t), \tag{10}$$

for step size $h > 0$. The field u_t *generates* p_t if $X_{t+h} \sim p_{t+h} + o(h)$ whenever $X_t \sim p_t$. A valid rate must satisfy

$$\sum_{x^i\in[L]} u_t^i(x^i, z) = 0, \qquad u_t^i(x^i, z) \geq 0 \quad (x^i \neq z^i). \tag{11}$$

A conditional probability velocity generating $p_t(\cdot \mid x_0, x_1)$ can be marginalized with the posterior $p_t(x_0, x_1 \mid z)$ to obtain a velocity generating the marginal path p_t. For equation 6, this yields the denoiser parameterization equation 8, where

$$p_{1|t}(x^i \mid z) = \sum_{x_0,x_1} \delta_{x_1^i}(x^i)\, p_t(x_0, x_1 \mid z). \tag{12}$$

This expression converts the predicted clean-token posterior $p_{1|t}$ into valid token-wise jump rates whose probability flow follows the prescribed path (Gat et al., 2024; Lipman et al., 2022).

Training objectives. Learning $p_\theta(\cdot \mid x_t, t) \approx p_{1|t}(\cdot \mid x_t)$ can be done by cross-entropy

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\sum_{i=1}^{N} \mathbb{E}_{t,(X_0,X_1),X_t}\big[\log p_\theta(X_1^i \mid X_t, t)\big], \tag{13}$$

which is minimized by the true denoiser (Gat et al., 2024). Our implementation uses the rate-weighted mixture-path generalized KL stated in Eq. equation 7, which is aligned with the jump rate in Eq. equation 8 and shares the same population minimizer $p_{1|t}$.

B Parallel Factorization Error in High-Dimensional Code Spaces

This appendix expands the parallel-factorization analysis of Section 3 and clarifies its scope. We consider representations $X = (X_1, \dots, X_d)$ whose coordinates are determined by a lower-complexity latent variable Z with $H(Z) = k$. If every coordinate carries nonzero information, $H(X_i) \geq h_0 > 0$, then their total correlation satisfies

$$\mathrm{TC}(X) = \sum_{i=1}^{d} H(X_i) - H(X) \geq dh_0 - k. \tag{14}$$

Thus, fixed intrinsic information can induce increasing aggregate dependence as the ambient dimension grows. This statement concerns aggregate redundancy and does not require every conditional block to be strongly dependent.

Parallel factorization gap. Suppose coordinates C have been resolved and a block U is sampled in parallel. Even with exact single-coordinate conditionals, the proposal

$$q(X_U \mid X_C) = \prod_{i \in U} p(X_i \mid X_C) \tag{15}$$

need not reproduce the joint conditional. Expanding the KL divergence and averaging over X_C gives

$$\mathbb{E}_{X_C}[D_{\mathrm{KL}}(p(X_U \mid X_C) \,\|\, q(X_U \mid X_C))] = \mathrm{TC}(X_U \mid X_C). \tag{16}$$

Exact marginal prediction is insufficient for correct parallel joint sampling. With only S generation steps, roughly $m = \Theta(d/S)$ coordinates must be updated per step, so few-step generation exposes the sampler to increasingly large simultaneous dependence.

To make the support error explicit, consider $Z \sim \mathrm{Unif}(\mathbb{F}_2^k)$ and $X = GZ \bmod 2$, where G has rank k. After conditioning on a valid partial assignment $X_C = x_C$, let m be the number of non-deterministic coordinates sampled in U and let $r_A = \mathrm{rank}(G_A)$. Only $2^{r_{C\cup U} - r_C}$ of the 2^m independently sampled assignments remain compatible with the code, so

$$P_{\mathrm{valid}}(U \mid X_C = x_C) = 2^{r_{C\cup U} - r_C - m}, \qquad P_{\mathrm{valid}}(U) \le 2^{k-m} \tag{17}$$

at the first step. Hence, validity can fall exponentially once the parallel block contains more unresolved decisions than the intrinsic dimension. Since a smaller step budget increases the number of coordinates sampled per update, this construction isolates how few-step inference can amplify parallel factorization error even when every coordinate marginal is exact. The result concerns the first parallel update rather than exponential accumulation throughout the entire trajectory: after the sampled rows determine Z, the remaining coordinates become deterministic. Moreover, validity measures support consistency, not whether valid codewords are sampled with the correct probabilities.

Scope of the analysis. The analysis establishes a mismatch between a factorized proposal and the correlated joint distribution, and shows how that mismatch can grow with the size of the parallel block. It does not establish that uniform corruption eliminates the factorization error or exactly matches the errors of a learned sampler. The trained-model ablation in Section 4.3 instead provides complementary evidence: changing only the ability to rewrite mask-based predictions has little effect, while replacing mask corruption with full-support noisy-content training yields a clear improvement. This result is consistent with reducing the train–inference gap caused by parallel factorization errors, but does not claim exact distributional matching.

C Experimental Setup and Training Recipe

Table 7: **Default experimental recipe.**

Setting	Value
Representation encoder	frozen DINOv2-B (Oquab et al., 2023)
Representation decoder	official RAE decoder (Zheng et al., 2026)
Discrete cube shape	16×16×768
Quantization granularity	3 bits (Wang et al., 2026)
Backbone	LightningDiT
Model scales	B/L/XL (Table 3)
Epochs	800
Warmup epochs	40
Optimizer	AdamW
lr	1×10^{-4}
lr scheduler	cosine decay
Batch Size	1024
Noise scheduler	time-shift scheduler
EMA rate	0.9995

D QUALITATIVE RESULTS OF CLASS-CONDITIONAL GENERATION



Figure 5: **Some generated** 256×256 **samples by CUDiT-XL trained on ImageNet.**