SUBDIVISION MEASURE MATCHING: GENERATIVE MODELING IN INTEGRAL FORM

Anonymous authors
Paper under double-blind review

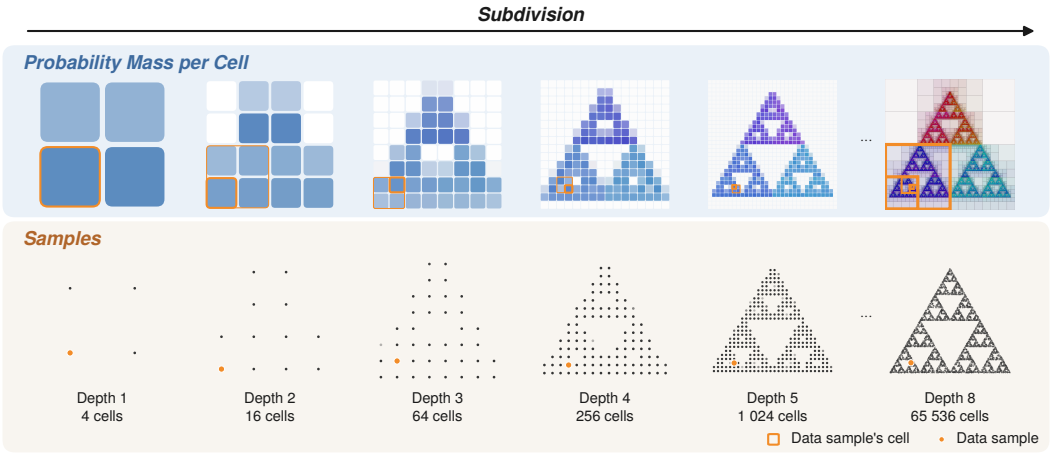


Figure 1: Subdivision Measure Matching represents a distribution through hierarchical subdivision of its domain. From left to right, top: probability mass on a Sierpinski target is progressively localized across subdivision depth; bottom: samples at each depth.

ABSTRACT

Generating samples from distributions with fractal, sharp, sparse, or disconnected support requires reproducing fine-scale structure while preserving distinct modes and gaps. While learned quantizers suffer from codebook collapse or depth saturation, increasing the resolution of scalar grid quantizers leads to exponential growth in the prediction head. To address these limitations, we introduce Subdivision Measure Matching (SMM), which models probability mass through codebook-free recursive subdivision of the domain. SMM progressively refines coarse regions into fine samples, increasing precision exponentially with depth while keeping the categorical prediction dimension fixed. For faster sampling, we introduce parallel coordinate refinement at finer levels, requiring only one model evaluation per level. Theoretically, SMM achieves optimal reconstruction accuracy for its code length and codebook size, with a strict worst-case advantage over shared additive residual quantization, and vanishing factorization-error bounds under standard smoothness assumptions. Experiments across 7 toy distributions demonstrate superior recall, coverage, and depth scaling compared to residual quantization baselines. On D3IL robot imitation tasks, SMM outperforms baselines, including diffusion policies, in 8 of 9 settings and achieves higher behavioral diversity on 3 tasks. On high-dimensional image synthesis, SMM improves ImageNet FID from 15.48 to 13.17 over a Flow Matching baseline.

1 INTRODUCTION

Generative modeling of continuous-valued distributions has seen remarkable progress through flow- and score-based methods, which learn velocity fields or scores to transport a Gaussian prior to the data distribution (Lipman et al., 2023; Liu et al., 2023; Ho et al., 2020; Song et al., 2021). Following

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

the success of denoising diffusion (Ho et al., 2020), subsequent work has shown that flow matching and diffusion models enable generation across image synthesis, scientific simulation, and other domains (Tong et al., 2024; Albergo & Vanden-Eijnden, 2023). At the same time, discrete tokenization has emerged as a parallel paradigm (van den Oord et al., 2017; Mentzer et al., 2024; Esser et al., 2021), where data is mapped to a finite codebook and a separate model learns a categorical distribution over indices (Zhu et al., 2025). However, learned vector quantization can leave many codes unused as the codebook grows, requiring additional mechanisms to maintain code utilization (Mentzer et al., 2024). Residual-Quantized VAE (Lee et al., 2022) (RQ-VAE) increases reconstruction accuracy through successive residual codes but retains a learned codebook (Lee et al., 2022). Finite scalar quantization (FSQ) removes codebook learning and improves utilization through a fixed scalar grid (Mentzer et al., 2024). However, making this grid denser exponentially increases the number of classes required for joint-token prediction with the number of jointly quantized coordinates. These limitations motivate a representation that increases grid resolution through depth while avoiding learned-codebook collapse.

We propose **Subdivision Measure Matching (SMM)**, a generative framework that learns how probability mass is distributed by recursively subdividing the domain. Concretely, SMM recursively subdivides $[0, 1]^D$ into K-ary cells along each coordinate axis. For each group of P coordinates, the model predicts a categorical distribution over the K^P joint child cells, conditioned on the preceding coarse predictions and autoregressive context. Repeating this subdivision provides K^T reconstruction values per coordinate after T levels, without learning residual codebooks or enlarging the prediction dimensionality.

The probability assigned to each child cell corresponds to its share of the parent cell's mass, expressed as a ratio of density integrals when a density exists. These probabilities remain defined for distributions without a density, including those supported on fractals. Such distributions contain occupied regions separated by gaps at successive scales, requiring increasing precision to resolve their structure. Through experiments on seven low-dimensional distributions, we examine whether generated samples preserve these regions and whether additional depth improves reconstruction. SMM recovers the recursive structure of the Sierpinski triangle in 10 localization levels, whereas the evaluated Flow Matching model fails to recover it within 1000 function evaluations. We also observe diminishing reconstruction gains and reduced effective code usage at deeper levels within the evaluated RQ-VAE baseline, while SMM continues to improve reconstruction performance. Across all seven distributions, SMM achieves higher recall and coverage than the evaluated baselines.

We further evaluate SMM on D3IL, extending the study to simulated robotics tasks. The same task admits several valid behaviors, such as different routes around obstacles or different orders of manipulation. Combining actions from these behaviors can produce an unsuccessful trajectory. Collision and placement constraints create sharp boundaries between successful trajectories and failures: a small deviation near a constraint boundary can change the outcome of an otherwise valid behavior, creating sharp boundaries in the valid behavior distribution. Success tests whether the generated behavior satisfies task constraints, while behavioral entropy tests whether this success is accompanied by preservation of different strategies. SMM achieves higher reported success rates than the diffusion baselines across eight out of nine settings while also achieving higher behavioral entropy on three of those.

We also study whether our method remains effective on high-dimensional image data, where VQ and Flow Matching are commonly used. Image-generation experiments demonstrate competitive performance, with ImageNet FID decreasing from 15.48 to 13.17 relative to a Flow Matching baseline. Reconstruction experiments on pixel-space datasets show better reconstruction scaling with depth than the RQ-VAE baseline. Reconstruction error, reconstruction FID, effective code usage, and distortion versus autoregressive coding cost assess both the accuracy gained from additional depth and its encoding cost. For latent-space encoder–decoder training through quantization, we show that increasing subdivision depth tightens a worst-case bound on the discrepancy between the straight-through gradient and the gradient of the corresponding unquantized objective. The bound is largest at depth one, where SMM is comparable to a single-stage FSQ quantizer, and decreases geometrically with each additional level.

For faster image generation, we introduce parallel refinement at deeper levels following autoregressive generation of coarse subdivisions. The model predicts coordinates in parallel within each finer level, requiring one model evaluation per level. Experiments on MNIST and KMNIST demonstrate that this reduction in sequential evaluations can preserve sample quality while cutting sequential network

evaluations by up to 7.7× and sampling time by up to 4.6×. Additional experiments show that temperature control, classifier-free guidance, and reward-based post-training carry over naturally to our framework.

Our main contributions are as follows.

1. We introduce **Subdivision Measure Matching,** a generative method, that models probability mass through codebook-free recursive subdivision of the domain into a tree of nested regions, with each parent's mass equal to the sum of the masses of its non-overlapping children.
2. We provide **exact sampling at finite precision** from the learned quantized distribution in T localization steps, with per-coordinate precision K^T and logarithmic complexity in output precision
3. We prove **optimal worst-case reconstruction** with a strict advantage over shared, unscaled, sum-decoded RQ-VAE at equal codebook size and depth
4. We propose a faster sampling approach via level-autoregression at deeper levels, predicting coordinates in parallel with one model evaluation per level and a vanishing factorization KL error bound under the stated smoothness and interaction assumptions.

2 Related Work

2.1 Continuous-Transport Generative Models

Flow- and score-based methods have emerged as the dominant paradigm for continuous generative modeling. Flow matching trains a velocity field $v_\theta(x, t)$ that defines an ODE transporting a Gaussian prior to the data distribution (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Tong et al., 2024), while score-based diffusion learns the score $\nabla_x \log p_t(x)$ that drives an SDE from noise to data (Ho et al., 2020; Song et al., 2021; Sohl-Dickstein et al., 2015).

One of the limitations of Flow Matching is its reliance on Lipschitz regularity of the velocity field for stability and convergence guarantees, which creates transport difficulties on targets with topology mismatch (Verine et al., 2023). For smooth target distributions, error bounds depend on how rapidly the learned field changes across nearby inputs, as measured by its Lipschitz constant (Cornish et al., 2020; Kunkel, 2025). When the target is not smooth, this becomes a limitation.

(Cornish et al., 2020) characterize topological obstructions to flows on supports whose topology differs from that of the prior support; Verine et al. (2023) extend the analysis for bi-Lipschitz flows; when source and target differ in topology, flows can leak probability mass outside the target support (Kelly et al., 2023; Bevins et al., 2023). Flow matching smooths over sharp and discontinuous targets (Utkarsh et al., 2025), and sparse target regions leave the velocity field under-supervised and lead to generation failures (Lee et al., 2025). A growing body of work has aimed to relax the smoothness assumptions and extend transport models to targets with different support geometry: Mathieu & Nickel (2020) develop flows on Riemannian manifolds, and Rezende et al. (2020) construct normalizing flows on tori, spheres, and Lie groups. These constructions require the support geometry to be known in advance and do not cover fractal, sparse, or disconnected supports, which motivates our approach.

2.2 Discrete and Hierarchical Token-Based Generation

A parallel line of work has focused on discrete tokenization, which replaces continuous outputs with discrete representations from a finite codebook (van den Oord et al., 2017; Esser et al., 2021; Razavi et al., 2019). VQ-VAE (van den Oord et al., 2017) learns discrete latent representations by replacing encoder outputs with vectors from a learned codebook; VQ-VAE-2 (Razavi et al., 2019) extends this to coarse and fine spatial latent maps; VQGAN (Esser et al., 2021) trains a quantized encoder–decoder with perceptual and adversarial losses. In these approaches, a discrete tokenizer first quantizes continuous inputs into integer indices, after which a separate autoregressive or masked transformer learns a distribution over token sequences (Esser et al., 2021; Chang et al., 2022). Since one code is allocated per location, improving reconstruction requires increasing a larger codebook. Subsequent work has focused on improving tokenizer quality and mitigating codebook collapse through techniques such as finite scalar quantization (FSQ) (Mentzer et al., 2024), look-up free

quantization, such as MAGVIT-v2 (Yu et al., 2024), and stochastic variants such as SQ-VAE (Takida et al., 2022). Takida et al. (2022) formulate quantization as variational inference with a self-annealing posterior to improve codebook utilization, while Huh et al. (2023) address the issue via affine reparameterization.

To achieve a superior reconstruction, Lee et al. (2022) introduce RQ-VAE, which applies vector quantization to the reconstruction residuals in a greedy manner. However, learned codebooks can be underutilized, while encoder training through discrete assignments is challenging to optimize (Mentzer et al., 2024).

A separate line of work generates samples in a coarse-to-fine, multi-scale manner (Han et al., 2025). VAR (Tian et al., 2024) organizes image generation through token maps of increasing spatial resolution, predicting the tokens within each scale in parallel. MaskGIT (Chang et al., 2022) predicts subsets of tokens at each step in a non-autoregressive masked-prediction framework.

The idea of representing a probability distribution through its mass on a hierarchical partition is common in nonparametric statistics. Pólya-tree priors (Lavine, 1992; Mauldin et al., 1992) place Bayesian nonparametric priors on dyadic partitions of the domain; classical histogram density estimation (Scott, 1979) estimates densities through axis-aligned binning; and decision-tree-based density estimators (Ram & Gray, 2011) learn data-adaptive partitions. These methods establish the theoretical and statistical foundations of partition-based density modeling but operate in low-dimensional regimes and use simple, often hand-designed, splitting rules. A separate line of work has explored quadtrees and adaptive grids in image and 3D modeling, but for representational rather than generative purposes: spatial indexing (Finkel & Bentley, 1974), occupancy networks (Mescheder et al., 2019), and neural radiance fields with hierarchical sampling (Liu et al., 2020a). None of these approaches model the full data distribution over the partition hierarchy.

2.3 Generative Policies for Robot Imitation Learning

Distributions with separated modes and sharp support boundaries also arise in robot imitation learning. Demonstrations may follow different routes around an obstacle, while the target distribution of successful trajectories excludes routes that collide with it (Jia et al., 2024; Zhai et al., 2025). Generating trajectories between valid modes can therefore cause failures. Precision also matters within a selected mode as a small deviation across a collision or placement boundary can turn a successful trajectory into a failed one. We therefore evaluate Subdivision Measure Matching on D3IL (Jia et al., 2024) that includes obstacle avoidance, alignment, pushing, sorting, and stacking tasks with human demonstrations. Our work studies recursive subdivision as an alternative representation of the conditional action distribution, comparing Subdivision Measure Matching with policy baselines on both task success and behavioral diversity.

3 Method

We first introduce the K-ary subdivision representation of the data domain in Section 3.1. Subsequently, we present Subdivision Measure Matching in Section 3.2, describing the depth and position factorizations and the KP3Attention that implements them. We then describe the training objective and sampling procedure in Section 3.3, including conditioning, temperature control and classifier-free guidance.

3.1 Subdivision Representations

We normalize the data via marginal cumulative distribution function (CDF). We model probability measures on unit cube $[0,1]^{D_x}$, where D_x is the number of data dimensions (*e.g.*, number of pixels). We recursively partition the domain into axis-aligned cells of equal volume by dividing each coordinate interval into K equal subintervals: for every coordinate, the depth-d localization step selects one of the K children of its depth-$(d-1)$ interval. Applied recursively from the root, this splits every coordinate interval into K^d equal subintervals at depth d. We denote the depth-(d) interval containing coordinate j's value as $B_j^{(d)}$. The root interval is $B_j^{(0)} = [0,1]$, coordinates are indexed by $j = 1, \ldots, D_x$, and subdivision steps by $d = 1, \ldots, T$.

Cell midpoint and child indices. For each coordinate j and depth d, we define a cell midpoint and a cell child index. The *midpoint* $x_j^{(d)}$ is the midpoint of the depth-d interval $B_j^{(d)}$; together with d, it identifies that interval. The *child index* $b_j^{(d)} \in \{0, \dots, K-1\}$ is the index of the child interval that contains coordinate x_j at depth d. Selecting child $b_j^{(d)}$ updates the midpoint according to

$$x_j^{(d)} = x_j^{(d-1)} + K^{-d}\big(b_j^{(d)} - \tfrac{K-1}{2}\big), \qquad x_j^{(0)} = \tfrac{1}{2}. \tag{1}$$

Following prior autoregressive models (van den Oord et al., 2017; Agarwal et al., 2025; Défossez et al., 2022; Ng et al., 2022), we group the distribution dimensions into $N = D_x/P$ non-overlapping patches containing P dimensions each, reducing the sequence length from D_x to $N = \frac{D_x}{P}$. As a result, each token represents P coordinates jointly, and the child index at every depth ranges over the K^P joint-child cells of the patch (Setting $P = 1$ recovers individual one dimension-per-token setting). We index patches by $i = 1, \dots, N$ and for simplicity of notation reuse $x_i^{(d)}$ and $b_i^{(d)}$ for the midpoints and joint child index of patch i. The path $b_i^{(1)}(x), \dots, b_i^{(T)}(x)$ is a deterministic function of the data, and the leaf midpoint $x_i^{(T)}$ recovers patch i up to the cell size K^{-T}; we write $x_i := x_i^{(T)}$ and $x_{<i} = (x_1, \dots, x_{i-1})$.

3.2 Subdivision Measure Matching

Subdivision Measure Matching factorizes subdivision across positions and, within each position, across depths. At the core of Subdivision Measure Matching is the K^P Tree Attention Block (KP3Attention), with our model structured as a stack of these blocks.

Depth factorization. Given the preceding patches $x_{<i}$, the path of patch i is factorized from the root to the leaf at finite resolution,

$$p_\theta\big(b_i^{(1:T)}(x) \mid x_{<i}, y\big) = \prod_{d=1}^{T} p_\theta\big(b_i^{(d)}(x) \mid x_{<i}, x_i^{(d-1)}, y, d\big), \tag{2}$$

where $x_i^{(d-1)}$ is the current cell of patch i, determined by the prefix $b_i^{(1:d-1)}(x)$.

Position factorization. The path probability of the data point is the product over completed patches,

$$p_\theta\big(b^{(1:T)}(x) \mid y\big) = \prod_{i=1}^{N} p_\theta\big(b_i^{(1:T)}(x) \mid x_{<i}, y\big). \tag{3}$$

Both decompositions are exact chain rules and impose no independence assumption; Eq. 3 is the probability of the leaf cell containing x.

KP3Attention. Following autoregressive image models (Esser et al., 2021; Van Den Oord et al., 2016), we use a transformer to realize factorization. The KP3Attention is a transformer block that processes an interleaved sequence of complete and coarse tokens in a causal manner. Our KP3Attention introduces a custom attention mask, shown in Figure 2. Complete tokens thus form a causal decoder over finished patches, as in a language model, and coarse tokens read its per-layer keys and values. Coarse token of patch i attends to the complete tokens $x_{<i}$ and to itself, and the complete token x_j attends to $x_{<j}$ and to itself. The output of the coarse token of patch i is mapped by a linear head to the K^P logits of $b_i^{(d)}$. During training all tokens are processed in parallel with the ground-truth $x_{<i}$ as complete tokens.

Block stacking and conditioning. Subdivision Measure Matching consists of a stack of L KP3Attentions. Spatial positions are encoded with a learned 2D row-column positional embedding shared across depths, and token-type information (complete vs. coarse) is added through a learned embedding. The depth index d and conditioning signal y are combined into a single vector and applied at every block via adaLN-Zero (Peebles & Xie, 2022). This design shares the transformer-block and output-head parameters across localization depths and spatial positions.

3.3 Training and Sampling

Subdivision Measure Matching is trained by minimizing the per-patch categorical cross-entropy between the predicted logits and the ground-truth child indices, summed across patches and averaged

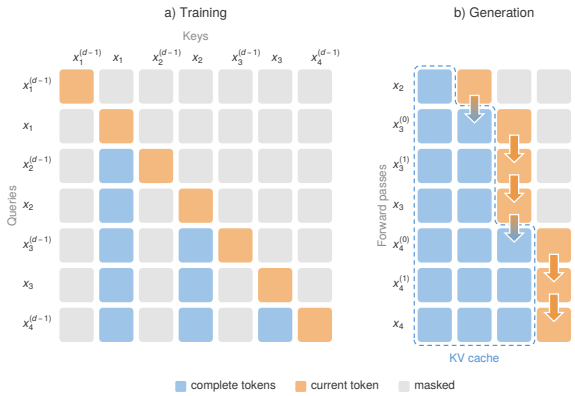


Figure 2: **(a)** Attention mask of the KP3Attention at depth d. Rows are queries, columns are keys. The coarse token $x_i^{(d-1)}$ and the complete token x_i of patch i both attend to the complete tokens of the preceding patches (blue) and to themselves (orange); everything else is masked (grey). The complete token of the last patch is never attended and is omitted. **(b)** Generation. Each pass refines the current token (orange); only complete tokens enter the KV cache (blue).

across depths. Formally, for a data point $x \sim p_{\text{data}}$ and a depth d drawn uniformly from $\{1, \dots, T\}$, we compute the midpoints $x_i^{(d-1)}$, the leaves $x_{<i}$, and the ground-truth child indices $b_i^{(d)}(x)$ from x's deterministic localization path, and minimize

$$\mathcal{L}(\theta) = \mathbb{E}_{x,d}\Big[\sum_{i=1}^{N} \mathrm{CE}\Big(\hat{\pi}_\theta\big(x_{<i}, x_i^{(d-1)}, y, d\big), b_i^{(d)}(x)\Big)\Big], \tag{4}$$

where $\hat{\pi}_\theta$ denotes the predicted categorical distribution over K^P joint children and CE is the categorical cross-entropy. Because d is sampled uniformly, the loss in Eq. 4 is the expected path negative log-likelihood divided by T:

$$\mathcal{L}(\theta) = \tfrac{1}{T}\, \mathbb{E}_{x \sim p_{\text{data}}}\Big[-\log p_\theta\big(b^{(1:T)}(x) \mid y\big)\Big]. \tag{5}$$

Training Subdivision Measure Matching is therefore exact maximum likelihood estimation of the finite-resolution discretization of the target distribution. More training details are provided in Appendix A.1.

Sampling. Patches are generated autoregressively, each by a root-to-leaf descent: for $i = 1, \dots, N$ and $d = 1, \dots, T$, draw $b_i^{(d)} \sim \hat{\pi}_\theta(x_{<i}, x_i^{(d-1)}, y, d)$ and apply Eq. 1. After depth T, the leaf $x_i = x_i^{(T)}$ is passed once through the network as a complete token and its per-layer keys and values are cached, exactly as a generated token in a language model, Figure 2; the cache never exceeds N tokens. The resulting sample is $(x_1, \dots, x_N)$. Because every step of Subdivision Measure Matching is a categorical draw, temperature and classifier-free guidance can be applied directly to the prediction logits. The formulations and ablations are provided in Appendix A.2.

Conditioning signal. Subdivision Measure Matching optionally accepts a conditioning signal, for example, a class label or environment state features - to enable controllable generation. We denote the encoded conditioning representation as y. Following prior work (Dosovitskiy et al., 2021), we encode the conditioning signal using a learned class embedding for class-conditional generation or pass state vector for robotics tasks.

4 Experiments

4.1 Experimental Settings

Toy distributions. We evaluate SMM on a suite of low-dimensional distributions to cover multimodal, sparse-discrete, fractal, and irregular-boundary supports. The 2D distributions include `sierpinski` (the Sierpinski triangle fractal generated), `mandelbrot` (samples from the Mandelbrot-set), `grid` (a sparse-support discrete distribution where each cell of a grid is independently filled with probability

0.5), and `shannon` (the pixel-brightness density of a natural image treated as a 2D probability density) (Durkan et al., 2019; Nash & Durkan, 2019). The 3D distributions include `gaussians` (a mixture of Gaussians at fixed lattice centers), `sierpinski` (the 3D Sierpinski tetrahedron), and `Swiss_roll` (the Swiss roll manifold with uniform thickness along one axis). All distributions are rescaled to $[-4, 4]^N$ and we draw 100000 training samples and 100000 validation points per dataset.

Robot imitation learning. We evaluate SMM on D3IL (Jia et al., 2024), a MuJoCo benchmark of human demonstrations on a 7-DoF Franka Panda in which every task has several distinct successful behaviors shown. We use the state-based variant in all nine published settings: Avoiding, Aligning, Pushing, Sorting-X for $X \in \{2, 4, 6\}$ blocks, and Stacking-X for $X \in \{1, 2, 3\}$. We follow data preprocessing and input representations format of the D3IL benchmark for fair comparison. Following Jia et al. (2024), we report success rate, the fraction of rollouts that meet the task's completion criterion, and behavior entropy. Success rate is the primary metric, while entropy is reported alongside it as a behavior diversity measure. We compare against the eleven policies reported on D3IL (Jia et al., 2024). Baseline numbers are taken from Jia et al. (2024) (six-seed means); we follow the benchmark's training and evaluation protocol for Subdivision Measure Matching. Task details, model configuration, the entropy definition, and baseline descriptions are provided in Sec. A.4.

Image generation. We evaluate pixel-space generation on six 28×28 datasets (MNIST, FashionMNIST, KMNIST, EMNIST, Omniglot, QuickDraw) and 32×32 dataset CIFAR-10. We further evaluate latent-space generation on ImageNet using the latent representation detailed in Appendix A.7.

Baselines. We compare SMM against Flow Matching (Lipman et al., 2023) as our primary baseline, with the architecture and parameter count matched to SMM across all settings; the two models are identical except for the attention mask, input, and final output layer (a velocity field for Flow Matching and per-axis K-way categorical logits for SMM). We further compare against three methods that discretize the data via a learned encoder rather than via recursive subdivision: VQ-VAE (van den Oord et al., 2017), FSQ-VAE (Mentzer et al., 2024), and RQ-VAE (Lee et al., 2022).

For toy distributions, we report Fréchet distance, Precision and Recall (Kynkäänniemi et al., 2019), Coverage and Density (Naeem et al., 2020), and per-dimension Wasserstein-1 distance to the ground-truth distribution. For pixel-space images, we report PCA FID (Jung & Keuper, 2021; Liu et al., 2020b; Thomas et al., 2025) along with pixel-histogram L1 distance: the mean over pixel positions of the L1 distance between the 64-bin intensity histograms of the real and generated images at that position. For latent-space images, we report standard FID, Inception Score, and Precision/Recall on ImageNet.

Implementation Details. For pixel-space images, we use AdaLN-Zero transformers with $d_{\text{model}} = 128$, 6 blocks, and 4 attention heads (∼1.3M parameters) for the 28×28 group, and $d_{\text{model}} = 512$, 8 blocks, and 16 attention heads (∼34M parameters) for the 32×32 group. We use joint-patch logits head $K = 4$ throughout the main results unless specified, with localization depth T chosen so that K^T matches the bit-precision of the data. We use the Adam optimizer with a learning rate of $1e-4$ and train for 500,000 steps on each dataset.

4.2 Results on Toy Distributions

Table 1 reports the comparison on the toy distribution suite. SMM outperforms Flow Matching on distributions with fractal (sierpinski2d, sierpinski3d) or sparse-discrete (grid) supports, which are challenging to represent through continuous transport. On the smooth multimodal distribution (Gaussians), SMM also outperforms Flow Matching in marginal Wasserstein-1 (0.025 vs. 0.058), while improving in coverage and recall. The discrete-token baselines VQ-VAE, FSQ-VAE, and RQ-VAE are limited by the resolution of their learned codebooks and underperform both SMM and Flow Matching on the smooth distributions, while exhibiting comparable performance to Flow Matching on the fractal supports. Additional details and results on Grid, Shannon, and Swiss Roll are provided in Appendix A.3, and a qualitative comparison on the Sierpinski triangle is in Appendix A.3.

4.3 Results on Robot Imitation Learning

Our method outperforms baselines (Table 2) in success rate across 8 of 9 task settings, while also achieving highest state entropy in 3 of them (Table 5), showing the ability of our approach to generalize on real-world imitation learning tasks with sharp distribution boundaries. For qualitative

Distribution	Method	Metrics					
		FD↓	W1↓	Precision↑	Recall↑	Coverage↑	Density↑
Sierpinski2d	Flow Matching	0.018	0.079	0.560	0.909	0.670	0.441
	VQ-VAE	0.009	0.066	0.819	0.655	0.570	0.719
	FSQ-VAE	0.017	0.338	0.081	0.000	0.000	0.062
	RQ-VAE	0.006	0.052	0.678	0.997	0.810	0.552
	SMM	**0.004**	**0.040**	**0.857**	**0.998**	**0.906**	**0.755**
Mandelbrot	Flow Matching	0.015	0.066	**0.989**	0.983	0.905	0.977
	VQ-VAE	0.010	0.087	0.975	0.202	0.303	0.971
	FSQ-VAE	0.008	0.326	0.883	0.000	0.004	0.853
	RQ-VAE	0.011	0.076	0.955	0.992	0.905	0.923
	SMM	**0.002**	**0.034**	0.985	**0.998**	**0.963**	**0.980**

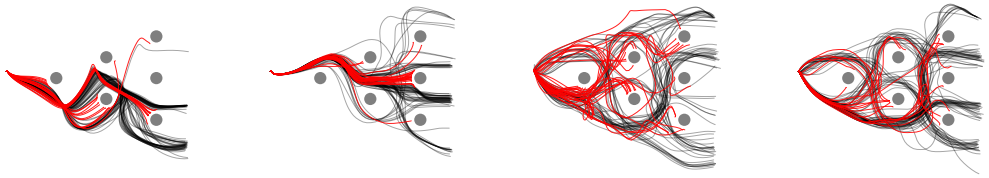
Distribution	Method	Metrics					
		FD↓	W1↓	Precision↑	Recall↑	Coverage↑	Density↑
Gaussians	Flow Matching	0.011	0.058	0.986	0.966	0.895	**0.998**
	VQ-VAE	0.016	0.217	0.960	0.025	0.061	0.907
	FSQ-VAE	0.004	0.327	0.902	0.211	0.007	0.910
	RQ-VAE	0.044	0.125	0.963	0.906	0.759	0.909
	SMM	**0.002**	**0.025**	**0.992**	**0.993**	**0.962**	0.995
Sierpinski3d	Flow Matching	0.016	0.067	0.544	0.920	0.641	0.388
	VQ-VAE	0.020	0.144	0.524	0.171	0.167	0.341
	FSQ-VAE	0.075	0.366	0.089	0.033	0.000	0.035
	RQ-VAE	0.013	0.066	0.642	0.975	0.711	0.466
	SMM	**0.003**	**0.032**	**0.873**	**0.996**	**0.909**	**0.770**

Table 1: Quantitative comparison on the toy-distribution suite (`Sierpinski2d`, `Mandelbrot`, `Gaussians`, `Sierpinski3d`). We report FD, per-dimension Wasserstein-1 distance, Precision, Recall, Coverage, and Density. Best results are shown in bold, and second-best results are underlined. `Grid`, `Shannon`, and `Swiss Roll` are in Table 4 (Appendix A.3).

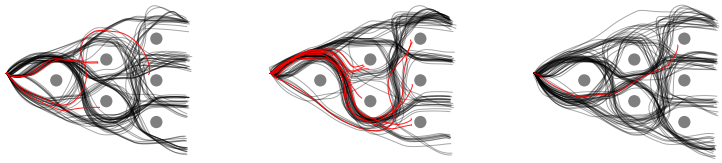
Method	Avoiding	Aligning	Pushing	Sort-2	Sort-4	Sort-6	Stack-1	Stack-2	Stack-3
BC-MLP	66.6 ± 51.2	70.8 ± 5.2	52.2 ± 16.5	44.4 ± 6.9	5.8 ± 4.8	1.6 ± 1.4	3.0 ± 2.2	0.0 ± 0.0	0.0 ± 0.0
VAE-State	71.6 ± 26.5	57.9 ± 13.8	60.4 ± 8.7	45.1 ± 5.9	7.9 ± 4.2	$\mathbf{3.0 \pm 1.6}$	3.6 ± 2.3	0.1 ± 0.1	0.0 ± 0.0
BeT-MLP	66.5 ± 13.6	50.7 ± 8.1	42.0 ± 4.9	31.7 ± 7.0	0.3 ± 0.1	0.0 ± 0.0	10.2 ± 5.9	0.3 ± 0.1	0.0 ± 0.0
IBC	76.0 ± 4.6	63.8 ± 2.7	57.4 ± 4.8	45.9 ± 3.3	8.0 ± 1.6	0.7 ± 0.5	1.0 ± 1.6	0.0 ± 0.0	0.0 ± 0.0
DDPM-MLP	63.7 ± 5.5	76.3 ± 3.9	56.9 ± 4.7	46.0 ± 3.9	3.2 ± 0.9	0.0 ± 0.0	24.4 ± 16.1	4.2 ± 4.5	0.1 ± 0.1
VAE-ACT	85.1 ± 10.9	89.1 ± 2.2	95.1 ± 3.3	85.8 ± 7.3	56.5 ± 10.8	2.1 ± 1.9	30.8 ± 9.6	8.3 ± 6.7	0.0 ± 0.0
BC-GPT	83.3 ± 40.8	83.3 ± 4.3	85.5 ± 5.4	80.0 ± 4.0	7.5 ± 6.3	0.7 ± 0.3	24.1 ± 5.5	2.5 ± 1.6	0.0 ± 0.0
BeT	74.7 ± 3.0	64.5 ± 6.9	72.4 ± 5.1	68.0 ± 6.0	1.8 ± 1.0	0.0 ± 0.0	16.3 ± 6.6	2.5 ± 1.5	0.1 ± 0.1
DDPM-GPT	92.7 ± 2.2	83.9 ± 2.0	84.7 ± 5.0	85.4 ± 1.3	33.7 ± 8.4	0.5 ± 0.3	89.5 ± 1.3	70.8 ± 3.1	12.0 ± 1.8
BESO	95.0 ± 1.5	86.1 ± 1.6	79.4 ± 9.5	78.4 ± 2.6	18.8 ± 4.2	2.0 ± 0.2	90.9 ± 0.7	68.5 ± 3.4	9.5 ± 0.6
DDPM-ACT	89.9 ± 0.6	84.9 ± 2.3	92.0 ± 2.7	88.2 ± 2.3	30.0 ± 6.0	0.0 ± 0.0	63.4 ± 4.9	30.0 ± 5.3	9.6 ± 2.1
Ours	$\mathbf{97.0 \pm 0.2}$	$\mathbf{91.4 \pm 2.1}$	$\mathbf{95.4 \pm 1.9}$	$\mathbf{89.4 \pm 0.3}$	$\mathbf{64.4 \pm 1.1}$	2.8 ± 0.2	$\mathbf{93.4 \pm 0.5}$	$\mathbf{72.6 \pm 0.5}$	$\mathbf{14.4 \pm 0.4}$

Table 2: D3IL success rates (%), state observations, across all nine settings. Our and published results are six-seed means; Bold and underline mark the best and second-best displayed success rate mean.

comparison of generated trajectories on Avoiding task, we ran our and baseline methods 100 times and visualized the trajectory of each run in Fig. 3. Our method has less collisions with obstacles, while maintaining the diversity of the paths.



(a) VAE-State (b) VAE-ACT (c) IBC (d) BeT



(e) BESO (f) DDPM-ACT (g) **Ours**

Figure 3: Visualization of 100 end-effector trajectories on the Avoiding task (red=failure). Panels (a)–(f) are the baselines published by Jia et al. (2024), panel (g) is our SMM policy, achieving 97% success with 0.98 behavioral entropy, above the best published success - BESO - of 95.0% and corresponding 0.856 behavioral entropy.

4.4 Results on Image Generation

Pixel-space generation. Table 3 reports the comparison on four pixel-space image datasets. Additional results on FashionMNIST, Omniglot, and QuickDraw are provided in Appendix A.6. SMM achieves the lowest FID of all evaluated methods on the 28×28 group, where it matches Flow Matching at equal backbone parameter count on one dataset (QuickDraw) and exceeds it on the remaining

Dataset	Method	Pixel-Space Metrics			
		FID↓	Hist. L1↓	Precision↑	Recall↑
MNIST	Flow Matching	0.807	0.053	0.836	0.937
	VQ-VAE	11.468	0.378	**0.997**	0.006
	FSQ-VAE	3.211	1.979	0.704	0.130
	RQ-VAE	2.820	0.347	0.970	0.367
	Ours	**0.422**	**0.047**	0.810	**0.949**
KMNIST	Flow Matching	2.636	**0.106**	0.731	0.796
	VQ-VAE	2.318	0.603	0.377	0.940
	FSQ-VAE	6.243	1.965	**0.950**	0.133
	RQ-VAE	2.376	0.476	0.449	**0.941**
	Ours	**1.693**	0.182	0.643	0.810
EMNIST	Flow Matching	4.970	0.286	0.594	0.875
	VQ-VAE	1.620	0.601	0.522	**0.953**
	FSQ-VAE	8.600	1.970	0.724	0.417
	RQ-VAE	1.593	0.497	**0.876**	0.703
	Ours	**0.730**	**0.170**	0.712	0.901
CIFAR-10	Flow Matching	0.546	**0.131**	**0.719**	0.390
	VQ-VAE	1.135	1.854	0.479	0.429
	FSQ-VAE	0.315	1.845	0.338	**0.627**
	RQ-VAE	0.390	0.255	0.477	0.503
	Ours	**0.163**	0.191	0.421	0.543

Table 3: Quantitative comparison on pixel-space image generation (MNIST, KMNIST, EMNIST, CIFAR-10). We report FID, pixel-histogram L1 distance, Precision, and Recall. **Bold** is best, underline is second best. FashionMNIST, Omniglot, and QuickDraw are in Appendix A.6).

four (MNIST, FashionMNIST, KMNIST, and EMNIST). On the 32×32 group, SMM achieves a FID of 0.163 on CIFAR-10, compared to 0.546 for Flow Matching with the matched transformer backbone.

Latent-space generation. Table 7 reports the comparison on ImageNet latent-space generation. SMM achieves an FID of 13.17 at $\sim$85M parameters, improving on the 15.48 of Flow Matching at the same parameter count. The Inception Score and Precision/Recall results, also reported in Table 7, show that SMM obtains a higher Inception Score (97.8 vs. 80.5) and higher recall (0.465 vs. 0.386) at the cost of slightly higher sFID (8.75 vs. 7.89) and lower precision (0.753 vs. 0.788). For SMM @ CFG we use guidance weight $w = 3.0$ (Eq. 7) and temperature $\tau = 1$ (Eq. 6); full sweeps of both on ImageNet are provided in the supplementary material (Appendix A.8).

4.5 Sampling Cost and Scaling

For low-dimensional distributions, drawing a sample at K-ary precision K^T requires $D \times T$ forward passes. Thus, sampling cost scales linearly with spatial dimension D and logarithmically with target precision. On 2D and 3D toy distributions, SMM uses fewer forward passes than Flow Matching at all tested precisions while achieving better sample quality on fractal and sparse supports (cf. Table 1). On robot learning tasks, SMM also uses fewer forward passes than the diffusion baseline. Scaling to higher-dimensional structured data such as images requires the autoregressive per-coordinate loop introduced in Section 3.2, whose generation cost is linear in the number of spatial positions N. In practice, joint-patch prediction reduces the number of sequential predictions by a factor of P, the number of spatial positions per patch.

5 Conclusion

We introduce Subdivision Measure Matching, a generative modeling framework based on measure decomposition: the data distribution is represented as a recursive factorization of conditional probability mass over a K-ary axis-aligned partition of the domain. Unlike prior methods that rely on continuous transport between a base and target distribution, Subdivision Measure Matching formulates generation as a sequence of categorical draws over the decomposition tree, enabling exact maximum-likelihood training via a per-step categorical cross-entropy loss and supporting principled temperature control and classifier-free guidance in logit space. At its core, the autoregressive depths factorization realized by our KP3Attention captures spatial correlations across positions while preserving the categorical structure that makes Subdivision Measure Matching applicable to distributions with fractal, sparse, or otherwise irregular support. To validate the formulation across scales, we evaluated Subdivision Measure Matching on a suite of low-dimensional toy distributions, pixel-space images, and latent-space ImageNet generation, where it consistently matches or exceeds the performance of Flow Matching at equal parameter count and substantially outperforms it on distributions whose supports violate the smoothness assumptions underlying continuous-transport methods. This work positions measure decomposition as a practical alternative to continuous-transport generative modeling.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Ricardo Baptista, Rebecca Morrison, Olivier Zahm, and Youssef Marzouk. Learning non-Gaussian graphical models via Hessian scores and triangular transport. *Journal of Machine Learning Research*, 25(85):1–46, 2024. URL https://jmlr.org/papers/v25/21-0022.html.

Harry Bevins, Will Handley, and Thomas Gessey-Jones. Piecewise normalizing flows. *arXiv preprint arXiv:2305.02930*, 2023.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11315–11325, 2022.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Rob Cornish, Anthony L. Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International Conference on Machine Learning*, pp. 2133–2143. PMLR, 2020.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Alexandre Ern and Jean-Luc Guermond. Scalar second-order elliptic PDEs. Chapter 27 of the course manuscript *Finite Elements* (Math 661, Texas A&M University, Fall 2017), https://people.tamu.edu/~guermond/M661_FALL_2017/chap27.pdf. Theorem 27.26(ii): homogeneous Neumann H^2 regularity on convex domains. Published version: Ern and Guermond, *Finite Elements II: Galerkin Approximation, Elliptic and Mixed PDEs*, Texts in Applied Mathematics 73, Springer, 2021, Chapter 31, 2017.

Alexandre Ern and Jean-Luc Guermond. *Finite Elements II: Galerkin Approximation, Elliptic and Mixed PDEs*, volume 73 of *Texts in Applied Mathematics*. Springer, 2021.

Patrick Esser, Robin Rombach, and Björn 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.

Raphael A. Finkel and Jon Louis Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Whan Ghang, Zane Martin, and Steven Waruhiu. The sharp log-Sobolev inequality on a compact interval. *Involve, a Journal of Mathematics*, 7(2):181–186, 2014. doi: 10.2140/involve.2014.7.181. Theorem 2.2 and Corollary 2.4: interval LSI and scaling.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15733–15744. IEEE, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14096–14113. PMLR, 2023. URL https://proceedings.mlr.press/v202/huh23a.html.

Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, and Gerhard Neumann. Towards Diverse Behaviors: A Benchmark for Imitation Learning with Human Demonstrations. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 41666–41686, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b63cb503757f9f31e5d74e3567dd160a-Paper-Conference.pdf.

Steffen Jung and Margret Keuper. Internalized biases in fréchet inception distance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Keegan Kelly, Lorena Piedras, Sukrit Rao, and David Roth. Variations and relaxations of normalizing flows. *arXiv preprint arXiv:2309.04433*, 2023.

Lea Kunkel. Distribution estimation via Flow Matching with Lipschitz guarantees. *arXiv preprint arXiv:2509.02337*, 2025.

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.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. arXiv:2510.25544, 2025. URL https://arxiv.org/abs/2510.25544.

Michael Lavine. Some aspects of Pólya tree distributions for statistical modelling. *The Annals of Statistics*, 20(3):1222–1235, 1992.

Michel Ledoux. Logarithmic Sobolev inequalities for unbounded spin systems revisited. In *Séminaire de Probabilités XXXV*, volume 1755 of *Lecture Notes in Mathematics*, pp. 167–194. Springer, Berlin, 2001. doi: 10.1007/978-3-540-44671-2_13. URL http://www.numdam.org/item/SPS_2001__35__167_0/.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, 2022.

Junho Lee, Kwanseok Kim, and Joonseok Lee. Is there a better source distribution than Gaussian? Exploring source distributions for image flow matching. *arXiv preprint arXiv:2512.18184*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 15651–15663, 2020a.

Ran Liu, Cem Subakan, Aishwarya H Balwani, Jennifer Whitesell, Julie Harris, Sanmi Koyejo, and Eva L Dyer. A generative modeling approach for interpreting population-level variability in brain structure. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 257–266. Springer, 2020b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in Neural Information Processing Systems*, 33:2503–2515, 2020.

R. Daniel Mauldin, William D. Sudderth, and S. C. Williams. Pólya trees and random distributions. *The Annals of Statistics*, 20(3):1203–1221, 1992.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4460–4470, 2019.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pp. 7176–7185. PMLR, 2020.

Charlie Nash and Conor Durkan. Autoregressive energy machines. In *International Conference on Machine Learning*, pp. 1735–1744. PMLR, 2019.

Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20363–20373. IEEE, 2022.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Parikshit Ram and Alexander G. Gray. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 627–635, 2011.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 14837–14847, 2019.

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

Danilo Jimenez Rezende, George Papamakarios, Sebastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, pp. 8083–8092. PMLR, 2020.

Guillaume Sanchez, Alexander Spangher, Honglu Fan, Elad Levi, and Stella Biderman. Stay on topic with classifier-free guidance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43197–43234. PMLR, 2024.

David W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22955–22968, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

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.

Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. SQ-VAE: Variational Bayes on discrete representation with self-annealed stochastic quantization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20987–21012, 2022.

Aaron Mark Thomas, Harry Youel, and Sharu Theresa Jose. Vae-qwgan: addressing mode collapse in quantum gans via autoencoding priors. *Quantum Machine Intelligence*, 7(2):91, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Best Paper Award.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. Expert Certification.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems*, 2025.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Alexandre Verine, Benjamin Negrevergne, Yann Chevaleyre, and Fabrice Rossi. On the expressivity of bi-Lipschitz normalizing flows. In *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, pp. 1054–1069. PMLR, 2023.

Roman Vershynin. High-dimensional probability. *Cambridge Series in Statistical and Probabilistic Mathematics*, 47, 2019.

Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019. doi: 10.3150/18-BEJ1065.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion — tokenizer is key to visual generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Xuanran Zhai, Qianyou Zhao, Qiaojun Yu, and Ce Hao. Vfp: Variational flow-matching policy for multi-modal robot manipulation. *arXiv preprint arXiv:2508.01622*, 2025.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Xiaoxu Zhu, Jiakui Li, Ken Zheng, Guiping Zhong, Huimeng Wang, Shiyin Kang, and Dahua Lin. Robust residual finite scalar quantization for neural compression. *arXiv preprint arXiv:2508.15860*, 2025.

A Additional experiments

A.1 Training Details

Monte Carlo estimator over levels. Computing the full inner sum $\sum_{k=1}^{R}$ on every gradient step is expensive for large R because each level requires a separate forward pass with different current-patch midpoints.

In practice, we use a standard Monte Carlo estimator. For each data point x in the mini-batch, we set the terminal depth $R = T$ with probability 0.8 and randomly sample R otherwise. All complete tokens are assigned this terminal depth R and represented by their depth-R midpoints. Thus, complete tokens are not always at the maximum depth T; they are complete at the selected terminal depth R.

For each coarse token i, we independently sample a level $k_i \sim \mathcal{U}\{1, \dots, R\}$, compute its midpoint $x_i^{(k_i-1)}$ from x's localization path, and evaluate

$$\mathcal{L}_{\mathbf{k},R}(x;\theta) = \sum_{i=1}^{N} \mathrm{CE}\Big(\hat{\pi}_\theta\big(x_{<i}^{(R)}, x_i^{(k_i-1)}, y, k_i\big), b_i^{(k_i)}(x)\Big)\,,$$

where $\mathbf{k} = (k_1, \dots, k_N)$. The prediction levels are sampled independently per coarse token, while the terminal depth of the complete tokens is shared within each example.

For fixed R, the expectation over x and $\mathbf{k}$ is the expected depth-R path negative log-likelihood divided by R. This way, the model learns to sample the distribution at different depths (precisions). This becomes useful for parallel sampling, described in Section A.9.

This resembles diffusion model training (Ho et al., 2020), where a timestep is sampled uniformly per data point; here, a depth is sampled independently for each coarse token.

A.2 Temperature and classifier-free guidance.

Because every step of Subdivision Measure Matching is a categorical draw, temperature and classifier-free guidance can be applied directly to the prediction logits. Let $\ell_\theta = \ell_\theta(x_{<i}, x_i^{(d-1)}, y, d)$ denote the pre-softmax logits. For temperature control, we sample from

$$p_\theta^{(\tau)} \propto \exp(\ell_\theta/\tau)\,, \tag{6}$$

which recovers the learned categorical distribution at $\tau = 1$ and produces more concentrated or more uniform child probabilities at $0 < \tau < 1$ or $\tau > 1$, respectively.

For classifier-free guidance (Ho & Salimans, 2021), we train Subdivision Measure Matching with conditioning dropout with probability p_{drop} and combine conditional and unconditional logits at sampling time as

$$\hat{\ell} = (1 + w)\,\ell_\theta(x_{<i}, x_i^{(d-1)}, y, d) - w\,\ell_\theta(x_{<i}, x_i^{(d-1)}, \varnothing, d), \tag{7}$$

where $w \geq 0$ is the guidance weight following (Sanchez et al., 2024).

A.3 Additional toy-distribution experiments.

The additional distributions in Table 4 test whether the gains observed on fractal supports also extend to sparse-discrete supports, complex probability densities, and thin manifolds. SMM achieves the highest recall and coverage on all three, while the advantage in distributional distance depends on the support: it improves marginal Wasserstein distance on Grid and Shannon and remains comparable to Flow Matching on Swiss Roll. Shannon also illustrates why FD alone is insufficient: FSQ-VAE achieves the lowest FD despite failing to recover the distribution's support, as reflected by its near-zero recall and coverage. For toy-distribution experiments, we use $K = 3$ and subdivision depth $T = 8$. Models are trained for 500k iterations. Experiments that vary subdivision depth are reported in Sec. A.10.

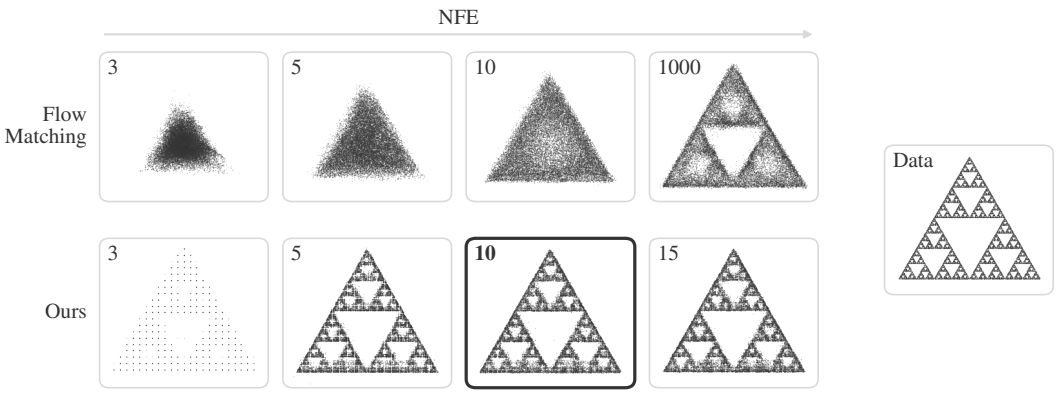


Figure 4: Samples from Flow Matching (top) and Subdivision Measure Matching (bottom) on the `Sierpinski` target as a function of the number of function evaluations (NFE). Subdivision Measure Matching recovers the fractal structure at NFE $= 5$, whereas Flow Matching still blurs the fine-scale gaps at NFE $= 1000$.

Distribution	Method	Metrics					
		FD↓	W1↓	Precision↑	Recall↑	Coverage↑	Density↑
Grid	Flow Matching	0.032	0.087	**0.926**	0.986	0.924	**0.847**
	VQ-VAE	0.014	0.123	0.862	0.769	0.764	0.820
	FSQ-VAE	0.005	0.333	0.891	0.018	0.073	0.807
	RQ-VAE	0.016	0.076	0.869	0.989	0.900	0.792
	SMM	**0.005**	**0.034**	0.896	**0.998**	**0.931**	0.782
Shannon	Flow Matching	0.010	0.060	0.997	0.991	0.946	0.988
	VQ-VAE	0.014	0.101	0.995	0.507	0.578	0.965
	FSQ-VAE	**0.001**	0.333	**1.000**	0.000	0.019	**1.060**
	RQ-VAE	0.015	0.076	0.995	0.992	0.938	0.982
	SMM	0.004	**0.038**	0.998	**0.999**	**0.962**	0.993

Distribution	Method	Metrics					
		FD↓	W1↓	Precision↑	Recall↑	Coverage↑	Density↑
Swiss Roll	Flow Matching	0.014	**0.052**	0.943	0.996	0.956	0.934
	VQ-VAE	0.023	0.208	0.469	0.339	0.219	0.306
	FSQ-VAE	0.024	0.342	0.381	0.119	0.017	0.236
	RQ-VAE	0.019	0.080	0.962	0.990	0.919	0.878
	SMM	**0.013**	0.056	**0.997**	**0.998**	**0.966**	**1.009**

Table 4: Additional toy-distribution results on `Grid`, `Shannon`, and `Swiss Roll`, complementing Table 1. We report FD, per-dimension Wasserstein-1 distance, Precision, Recall, Coverage, and Density. Best results are bold; second-best results are underlined.

Qualitative comparison on the Sierpinski triangle. Figure 4 compares samples from Flow Matching and SMM on the 2D Sierpinski triangle as the number of function evaluations (NFE) increases. Flow Matching, even at high step counts, smooths over the recursive holes that define the fractal structure of the support; only the outermost levels of the triangle are recovered, and the deeper self-similar holes remain filled. In contrast, SMM produces samples that lie cleanly on the fractal support after only a few localization levels, with finer self-similar structure emerging as T increases. This mirrors the quantitative results in Table 1: continuous transport methods struggle to represent supports whose connected-component structure changes at every scale, while a hierarchical-subdivision formulation can.

A.4 Robot Imitation Learning Details

We follow the D3IL setup, where the baselines use task-dependent model sizes, and use a KP3Attention-based transformer with parameter counts matched to the benchmark configuration for each task. Following the official implementation, we train for 500 epochs with batch size 1024. We use subdivision depth $T = 6$ and $K = 4$.

Tasks and evaluation. We use the state-based variant in all nine published settings: Avoiding (96 demonstrations, 24 valid routes past six obstacles), Aligning (1k demonstrations, push a box to a target pose from inside or outside), Pushing (2k demonstrations, two blocks to two zones in any order), Sorting-X for $X \in \{2, 4, 6\}$ blocks ($1.6k$ demonstrations at X=6, distinct behaviors by sorting order), and Stacking-X for $X \in \{1, 2, 3\}$ (1k demonstrations, six stacking orders; X counts blocks stacked in the evaluation). States are features of dimension 4-20; actions are 2-3D end-effector velocities, except Stacking that has 8D joint velocities plus gripper width. Object poses are resampled every episode. Following Jia et al. (2024), we report success rate, the fraction of rollouts that meet the task's completion criterion, and behavior entropy, $\mathbb{E}_{s_0}\left[-\sum_{\beta\in\mathcal{B}} \pi(\beta \mid s_0) \log_{|\mathcal{B}|} \pi(\beta \mid s_0)\right]$, where

Method	Avoiding	Aligning	Pushing	Sort-2	Sort-4	Sort-6	Stack-1	Stack-2	Stack-3
BC-MLP	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
VAE-State	0.195	0.030	0.170	0.106	0.090	**0.086**	0.064	0.000	0.000
BeT-MLP	0.836	0.485	0.628	0.309	0.021	0.000	0.120	0.000	0.000
IBC	0.850	0.300	0.816	0.370	0.097	0.024	0.013	0.000	0.000
DDPM-MLP	0.801	<u>0.712</u>	0.796	0.327	0.071	0.000	0.113	0.011	0.096
VAE-ACT	0.224	0.025	0.070	0.012	0.114	0.057	0.095	0.037	0.000
BC-GPT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
BeT	0.844	0.536	0.788	**0.473**	0.059	0.000	0.234	0.051	0.000
DDPM-GPT	<u>0.898</u>	0.664	<u>0.862</u>	<u>0.431</u>	<u>0.372</u>	0.031	0.222	0.146	0.120
BESO	0.856	0.711	**0.871**	0.387	0.232	0.055	<u>0.382</u>	<u>0.177</u>	0.131
DDPM-ACT	0.863	**0.749**	0.859	0.347	0.300	0.000	**0.762**	**0.503**	<u>0.190</u>
Ours	**0.980**	0.631	0.612	0.360	**0.421**	<u>0.062</u>	0.145	0.144	**0.284**

Table 5: Behavior entropy on D3IL state-based tasks. All results are means over six random seeds following baseline methodology; baseline results are taken from the published benchmark. Bold indicates highest entropy; underline indicates second highest.

β is the discrete behavior of a rollout and s_0 is the initial state; entropy is computed per initial state and averaged. An entropy of 0 means a single behavior per initial state; 1 means all behaviors occur equally often for each initial state. Success rate is the primary metric, while entropy is reported alongside it as a behavior diversity measure.

Baselines. We compare against the eleven policies reported on D3IL (Jia et al., 2024). Early behavioral cloning regresses actions directly from states or a history of states with an MLP or a transformer (BC-MLP, BC-GPT); a deterministic regressor averages over demonstrated modes and cannot represent their multiplicity. Subsequent work models the conditional action distribution explicitly. VAE-State uses a conditional variational autoencoder (Sohn et al., 2015) to sample actions from a latent variable given the current state, and VAE-ACT (Zhao et al., 2023) uses a transformer that predicts a chunk of future actions, motivated by tasks that require precision and careful coordination of contact forces. Shafiullah et al. (2022) propose Behavior Transformers (BeT), which cluster actions with k-means and predict a cluster together with a continuous offset, in order to learn multiple behavioral modes from unlabeled demonstrations. BeT and BeT-MLP use a transformer or an MLP backbone, respectively. Florence et al. (2022) introduce Implicit Behavioral Cloning (IBC), which learns an energy function over state–action pairs and samples actions by Langevin dynamics. More recently, diffusion policies generate actions by iterative denoising: Pearce et al. (2023) and Chi et al. (2023) use discrete-time DDPMs with MLP, GPT, and action-chunking backbones (DDPM-MLP, DDPM-GPT, DDPM-ACT), citing the multimodality of human behavior and the need for high precision, and Reuss et al. (2023) propose a continuous-time score-based policy with a transformer backbone (BESO) to capture the multiple solutions.

A.5 Behavior entropy on D3IL

Table 5 reports behavior entropy on all nine D3IL state-based settings, complementing the success rates in Section 4.

A.6 Additional pixel-space results.

All pixel-space methods operate directly on raw pixels without an encoder-decoder pair. Since pixel intensities are 8-bit, each coordinate has 256 possible values; the main SMM configuration therefore uses $K = 4$ and $T = 4$, giving $K^T = 256$ reconstruction values per coordinate. Models are trained for 500k iterations with batch size 128 and evaluated on 10,000 generated samples for each of three random seeds.

For MNIST, FashionMNIST, KMNIST, EMNIST, Omniglot, and QuickDraw, the transformer backbone has width 128, 6 blocks, 4 attention heads, and feed-forward width 512. For CIFAR-10, it has width 512, 8 blocks, 16 attention heads, and feed-forward width 2048; the SMM model has 34.29M parameters. SMM, FSQ, VQ-VAE, and RQ-VAE use the same autoregressive transformer backbone, with VQ and RQ additionally using learned embedding tables. Flow Matching uses the matched

Dataset	Method	Pixel-Space Metrics			
		FID↓	Hist. L1↓	Precision↑	Recall↑
FashionMNIST	Flow Matching	0.677	**0.092**	**0.778**	0.746
	VQ-VAE	0.709	0.921	0.489	**0.865**
	FSQ-VAE	3.075	1.921	0.492	0.323
	RQ-VAE	0.599	0.422	0.674	0.787
	Ours	**0.512**	0.135	0.508	0.832
Omniglot	Flow Matching	0.898	0.103	0.706	**0.677**
	VQ-VAE	21.057	0.359	**1.000**	0.000
	FSQ-VAE	4.281	1.970	0.787	0.007
	RQ-VAE	16.154	0.326	**1.000**	0.000
	Ours	**0.279**	**0.094**	0.688	0.652
QuickDraw	Flow Matching	0.421	**0.060**	0.775	0.847
	VQ-VAE	1.377	0.565	0.853	0.689
	FSQ-VAE	3.773	1.966	**0.995**	0.089
	RQ-VAE	0.842	0.490	0.619	**0.947**
	Ours	**0.420**	0.108	0.708	0.882

Table 6: Additional pixel-space generation results on FashionMNIST, Omniglot, and QuickDraw, complementing Table 3. We report FID, pixel-histogram L1 distance, Precision, and Recall. Best results are bold; second-best results are underlined.

Method	ImageNet Metrics				
	FID↓	sFID↓	IS↑	Precision↑	Recall↑
Flow Matching	15.48	7.89	80.5	0.788	0.386
MAR	19.37	7.7	70.6	0.718	0.438
RQ-VAE	25.54	9.26	57.3	0.692	0.435
VQ-VAE	27.13	8.98	55.2	0.672	0.440
FSQ-VAE	17.35	8.28	86.5	0.685	**0.482**
SMM	13.17	8.75	97.8	0.753	0.465
SMM @ CFG 3.0	**8.88**	**7.7**	**154.4**	**0.846**	0.345

Table 7: Quantitative comparison on ImageNet latent-space generation. We report standard FID, sFID, Inception Score, and Precision/Recall.

transformer backbone without the autoregressive categorical factorization and conditions on time instead of subdivision depth.

Table 6 reports additional results on FashionMNIST, Omniglot, and QuickDraw. SMM achieves lower FID than all evaluated baselines on FashionMNIST and Omniglot, while Flow Matching retains higher precision on both datasets and higher recall on Omniglot. On QuickDraw, SMM achieves comparable FID and higher recall than Flow Matching at the cost of lower precision.

A.7 IMAGENET LATENT-SPACE GENERATION

For ImageNet at 256×256 resolution, SMM operates on a 32×32 spatial grid of seven-channel autoencoder latents, following the LlamaGen VQ-8 recipe (Sun et al., 2024), with the additional quantizer losses and ℓ_2 normalization removed and the latent dimensionality changed from 8 to 7 channels. The seven latent channels at each spatial position are predicted jointly, giving K^7 categorical outputs. We use $K = 4$, yielding $4^7 = 16{,}384$ classes. We use subdivision depth $T = 2$.

SMM uses the HGPT-B (Sun et al., 2024) transformer backbone with 12 blocks, 12 attention heads, and hidden width 768, with total of 85M parameters. We train for 50 epochs with batch size 256 and evaluate generation using 50,000 samples. Table 7 reports the ImageNet latent-space comparison discussed in Section 4. Table 8 shows the effect of scaling SMM from B to L and XL. SMM-L and SMM-XL follow the LlamaGen-L and LlamaGen-XL configurations (24 blocks, 16 heads, width 1024, and 36 blocks, 20 heads, width 1280, respectively) and are trained for 50 epochs with the same recipe as SMM-B. Figure 5 shows samples from SMM-XL. Table 9 reports the effect of GRPO post-training on SMM and on the Flow Matching baseline.

A.8 GUIDANCE AND TEMPERATURE ABLATIONS

Here we ablate the two sampling-time controls of SMM — the classifier-free guidance weight w (Eq. 7) and the sampling temperature τ (Eq. 6) — on ImageNet latent-space generation, using the same FID, sFID, Inception Score, and Precision/Recall metrics as the main comparison in Table 7. Results are summarized in Tables 10 and 11.

Method	ImageNet Metrics				
	FID↓	**sFID↓**	**IS↑**	**Precision↑**	**Recall↑**
SMM-B	8.88	7.70	154.4	0.846	0.345
SMM-L	4.47	6.31	245.1	0.835	0.480
SMM-XL	**4.30**	**6.00**	227.4	0.838	**0.475**

Table 8: Effect of model scaling on class-conditional ImageNet generation at 256×256 resolution. Scaling SMM from B to L and XL improves FID and sFID. We report FID, sFID, Inception Score, and Precision/Recall using 50k generated samples per model, with CFG 3 for SMM-B, SMM-L, and SMM-XL.

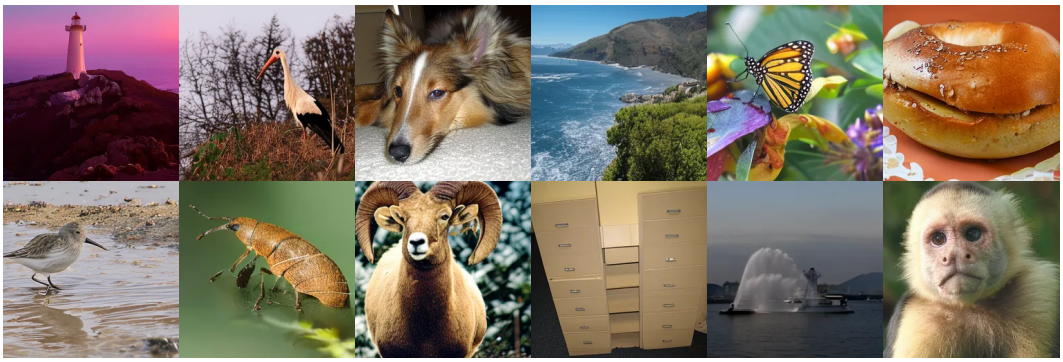


Figure 5: Qualitative results of SMM-XL on class-conditional ImageNet generation at 256×256 resolution, showing fine details and textures across scenes, animals, and objects.

Method	CFG	ImageNet Metrics					Reward-Model Metrics				
		FID↓	**sFID↓**	**IS↑**	**Precision↑**	**Recall↑**	**CLIP↑**	**HPSv2↑**	**PickScore↑**	**ImageReward↑**	**Aesthetic↑**
SMM	3.0	**8.88**	**7.7**	154.4	0.846	0.345	1.445	1.110	0.762	−0.542	4.645
SMM + GRPO	3.0	15.34	10.42	**187.6**	**0.913**	0.129	**1.486**	**1.203**	**0.775**	**−0.187**	**5.026**
Flow Matching	3.0	15.269	11.984	111.1	0.860	0.264	1.447	1.114	0.764	−0.495	4.722
Flow Matching + Flow-GRPO	3.0	18.979	16.472	125.1	0.907	0.148	1.474	1.171	0.773	−0.257	5.018

Table 9: GRPO post-training on ImageNet latent-space generation (50k samples, temperature 1.0). CLIP and HPSv2 are used as training rewards; PickScore, ImageReward, and Aesthetic are held out. In both model families GRPO raises Inception Score, Precision, and all reward scores while degrading FID and Recall. SMM + GRPO scores best on all five reward models and keeps a lower absolute FID than Flow-GRPO, though Flow-GRPO degrades FID less relative to its own pre-RL model.

Effect of classifier-free guidance. Following the logit-space classifier-free guidance formulation of Eq. 7, at fixed temperature $\tau = 1$ (Eq. 6), we sweep the guidance weight w on ImageNet latent-space generation (Section 4); results are shown in Table 10. FID improves sharply with guidance, dropping to a minimum of 8.88 at $w = 3.0$, beyond which it degrades; Inception Score increases monotonically while recall decreases, recovering the standard fidelity–recall trade-off of classifier-free guidance directly in logit space, as anticipated in Section 3.3. This monotone trade-off arises because every step of SMM is a genuine categorical draw rather than a continuous-transport update. This finding motivates our choice of $w = 3.0$ as the default reported as SMM @ CFG 3.0 in Table 7.

Effect of temperature. At the default guidance weight $w = 3.0$ (Eq. 7), we vary the sampling temperature τ of Eq. 6 and report sample quality in Table 11. Consistent with Eq. 6, $\tau = 1$ recovers the data distribution while $\tau < 1$ and $\tau > 1$ yield sharper and smoother samples respectively. FID is minimized exactly at $\tau = 1$ and degrades on either side, while recall increases monotonically with τ.

Table 10: Classifier-free guidance (cfg) ablation on ImageNet latent-space generation. We fix the sampling temperature to $\tau = 1$ and vary the guidance weight w. FID and sFID are lower-is-better; Inception Score, Precision, and Recall are higher-is-better.

w	FID↓	sFID↓	IS↑	Precision↑	Recall↑
no cfg	13.17	8.75	97.8	0.753	0.465
$w = 2.5$	9.74	8.08	129.5	0.811	0.390
$w = 3.0$	**8.88**	7.7	154.4	0.846	0.345
$w = 4.0$	9.81	8.14	184.8	0.882	0.267

Table 11: Sampling-temperature ablation on ImageNet latent-space generation. We fix the classifier-free guidance weight to $w = 3.0$ and vary the temperature τ. FID and sFID are lower-is-better; Inception Score, Precision, and Recall are higher-is-better.

τ	FID↓	sFID↓	IS↑	Precision↑	Recall↑
$\tau = 0.90$	11.35	10.40	137.0	0.828	0.314
$\tau = 0.95$	9.71	8.51	149.8	0.843	0.335
$\tau = 1.00$	**8.88**	**7.7**	**154.4**	**0.846**	0.345
$\tau = 1.05$	9.20	8.34	150.0	0.834	**0.351**

A.9 Parallel refinement experiments

The parallel-refinement model reuses the SMM backbone with two changes: the attention mask is replaced by full attention among coarse tokens, so every patch sees the current coarse image, and complete tokens are dropped, since no patch is generated before another. It is initialized from the pretrained SMM weights and trained for 100k further steps with Adam at learning rate 10^{-4} and batch size 128. Each training step draws a level k uniformly from $\{1, \ldots, T-1\}$, quantizes the image to its depth-k cells to form the input, supplies k through adaLN, and applies per-patch categorical cross-entropy to the depth-$(k{+}1)$ children of all patches at once, conditioned on the entire depth-k coarse image.

At inference, the autoregressive model generates each patch through levels $1, \ldots, r$ before proceeding to the next patch, using preceding patches completed to depth r as context. Once all patches reach depth r, the parallel model refines the remaining levels with one forward pass per level, sampling each patch independently from its predicted softmax. Refinement proceeds to the pretrained depth T in Tables 12 and 13, or beyond T when additional refinement levels are trained, as in the KMNIST depth-8 experiment in Table 13. We compare this approach to regression-based residual prediction: the regression model shares the architecture and recipe but replaces the classifier head by a linear projection trained with mean-squared error against the ground-truth residual, and produces the final sample in a single step.

For the MNIST parallel-refinement ablation, we use $K = 2, T = 8$ instead of the main $K = 4, T = 4$ configuration. Both yield 256 reconstruction values per pixel, while the binary subdivision provides more refinement levels for evaluating the parallel tail. A single autoregressive level followed by seven parallel levels cuts sequential steps from 1,568 to 203 ($7.7\times$) and sampling time from 17.5 to 3.8 s per 1,000 images ($4.6\times$) with no loss in quality (FID 0.407 vs. 0.422). On KMNIST the same costs a small FID increase (1.894 vs. 1.776) at $3.9\times$ fewer steps; two autoregressive levels recover full quality (1.762) at $2\times$ fewer steps. Refining beyond the original model depth improves the pixel histogram (Hist. L1 0.109 at depth 8 on KMNIST) without affecting FID. MSE regression of the residual suffers from mean collapse: with one autoregressive level the MNIST FID is 4.49 (precision 0.92, recall 0.19), and it matches autoregressive quality only when the residual is small enough that the speed-up is gone. Categorical parallel refinement avoids mean collapse.

A.10 Representation quality and depth scaling

We evaluate the quantized representations independently of sampling quality by encoding and reconstructing inputs without sampling the autoregressive prior. Reconstruction MSE measures the

Table 12: Comparison of SMM residual tail modeling approaches on MNIST data.

	Residual head	FID↓	Hist. L1↓	Precision↑	Recall↑	Coverage↑	Seq. steps	Time / 1k img
SMM pretrained (AR to depth 8)		0.422	0.047	0.810	0.943	0.842	1568	17.51 s
AR to depth 1, followed by residual	MSE (regress)	4.491	0.770	**0.916**	0.193	0.693	197	2.47 s
	Parallel CE, to depth 8	**0.407**	**0.043**	0.841	0.943	**0.880**	203	3.80 s
AR to depth 2, followed by residual	MSE (regress)	0.964	0.536	0.872	0.856	0.829	393	4.11 s
	Parallel CE, to depth 8	0.409	0.048	0.833	0.940	0.868	398	5.79 s
AR to depth 4, followed by residual	MSE (regress)	0.420	0.168	0.815	0.945	0.847	785	7.61 s
	Parallel CE, to depth 8	0.419	0.050	0.803	**0.949**	0.842	788	9.78 s

Table 13: Comparison of SMM residual tail modeling approaches on KMNIST data

	Residual head	FID↓	Hist. L1↓	Precision↑	Recall↑	Coverage↑	Seq. steps	Time / 1k img
SMM pretrained (4 AR levels)		1.776	0.183	0.642	0.812	0.558	784	9.04 s
AR level 0, residual from level 1	MSE (regress)	2.082	0.286	**0.839**	0.601	**0.689**	197	2.51 s
	Parallel CE, to depth 4	1.894	0.160	0.638	**0.861**	0.571	199	2.97 s
AR levels 0–1, residual from level 2	MSE (regress)	1.772	0.267	0.659	0.800	0.574	393	4.75 s
	Parallel CE, to depth 4	**1.762**	0.185	0.652	0.803	0.569	394	5.00 s
	Parallel CE, to depth 8	1.778	**0.109**	0.637	0.814	0.561	398	5.96 s

squared error introduced by quantization, while reconstruction FID measures the discrepancy between reconstructed and real images. We report nominal capacity as the sum of logarithms of codebook sizes, expressed in bits per point or pixel. To quantify the benefit of increasing depth, we measure the geometric-average error ratio $(D_b/D_a)^{1/(b-a)}$ between consecutive evaluated depths $a < b$; smaller values indicate greater reconstruction improvement per added level. Following the code-utilization analysis of FSQ, we additionally measure effective codebook usage, $2^{H(Z_d)}/K_d$, where $H(Z_d)$ is the marginal code entropy (estimated by trained AR model) and K_d is the codebook size. This measures assignment concentration rather than reconstruction fidelity. Finally, we compute compression cost as the negative log-probability of encoded representations under the autoregressive prior, in bits, and plot distortion against this cost to assess coding efficiency.

Toy distributions. Figure 6 shows that SMM's reconstruction advantage grows with depth. Residual quantization (RQ) achieves lower distortion at shallow depths, but the curves cross at depth 6; beyond it, SMM's MSE keeps falling while RQ's flattens. Accordingly, SMM's per-level error ratio is constant across depths whereas RQ's rises at deeper levels, and SMM's effective codebook usage saturates at one while RQ's degrades at the deepest levels. Measured against coding cost under the learned autoregressive prior, SMM's advantage persists at the high depths.

Pixel-space representations. Figure 7 compares the trained pixel-space SMM and RQ across different capacities, expressed in bits per pixel. SMM's reconstruction MSE continues decreasing as capacity increases, while RQ plateaus beyond four bits per pixel. At 8 bits per pixel, SMM achieves approximately $1530\times$ and $709\times$ lower MSE on MNIST and KMNIST, respectively. Reconstruction FID likewise favors SMM, reaching 0.0012 versus 2.297 on MNIST and 0.0011 versus 0.1463 on KMNIST.

B Reconstruction accuracy and representation cost

We represent a vector $x \in [0,1]^D$ through recursive subdivision. At each level, SMM divides every coordinate interval into $K \geq 2$ equal parts and predicts the corresponding digit. A depth-t sequence $a_{1:t}$ identifies a product cell with midpoint

$$g_t(a_{1:t}) = \sum_{s=1}^{t} K^{-s} a_s + \frac{K^{-t}}{2}\mathbf{1}, \qquad a_s \in \{0, \ldots, K-1\}^D. \tag{1}$$

We use half-open cells, including the outer faces at 1, to resolve boundaries. Let $\mathcal{Q}_t$ denote the resulting midpoint quantizer. Predicting groups of g coordinates, where g divides D, uses $S = DT/g$ categorical symbols with $J = K^g$ choices each. All reconstruction comparisons below use the same

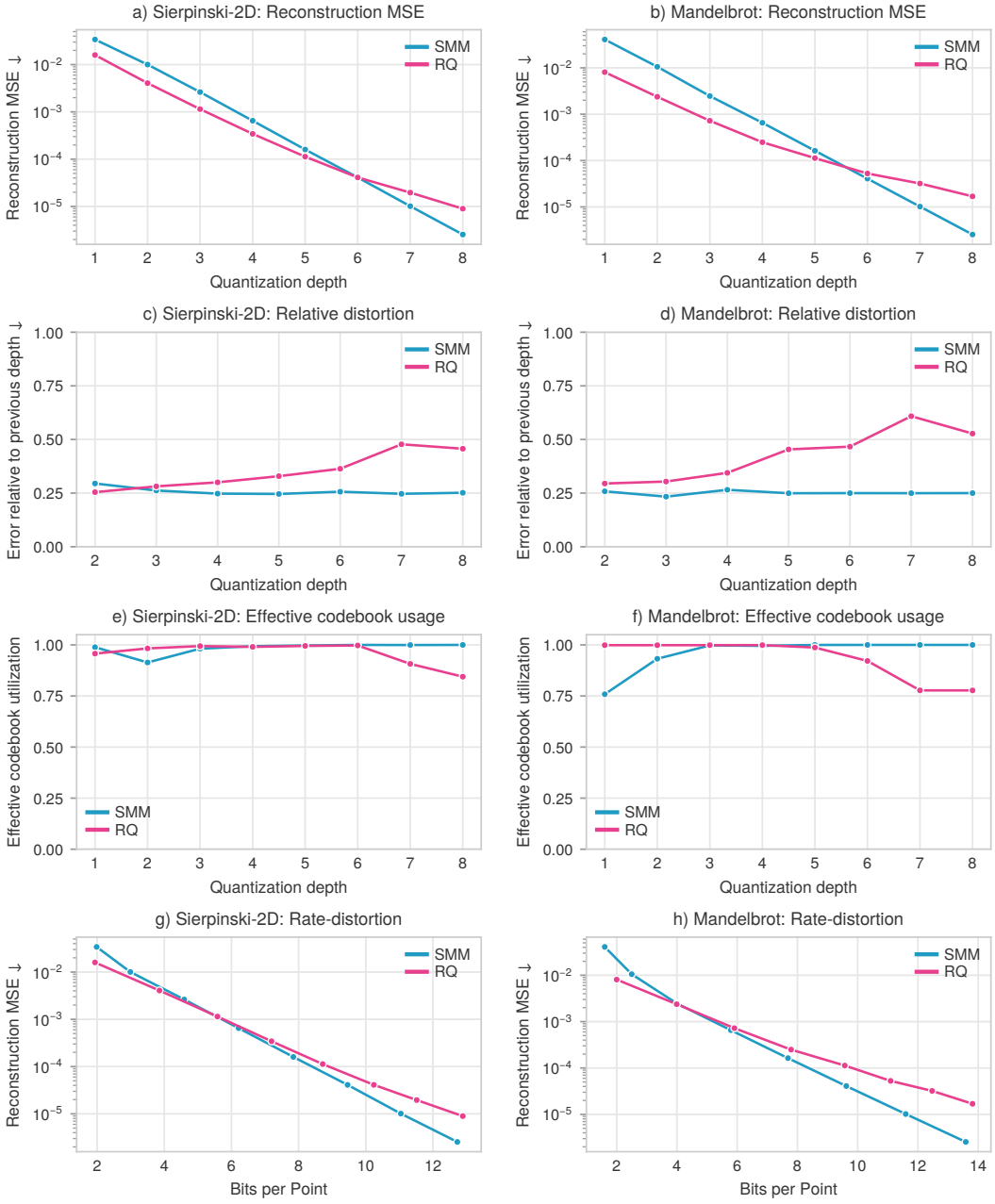


Figure 6: Representation quality and depth scaling on Sierpinski-2D (left) and Mandelbrot (right). We compare SMM and residual quantization (RQ) in terms of (a,b) reconstruction MSE, (c,d) geometric-average error ratio per added level, (e,f) final-level effective codebook usage, and (g,h) rate–distortion under the learned autoregressive prior. Reconstruction uses encoded held-out inputs rather than samples from the prior.

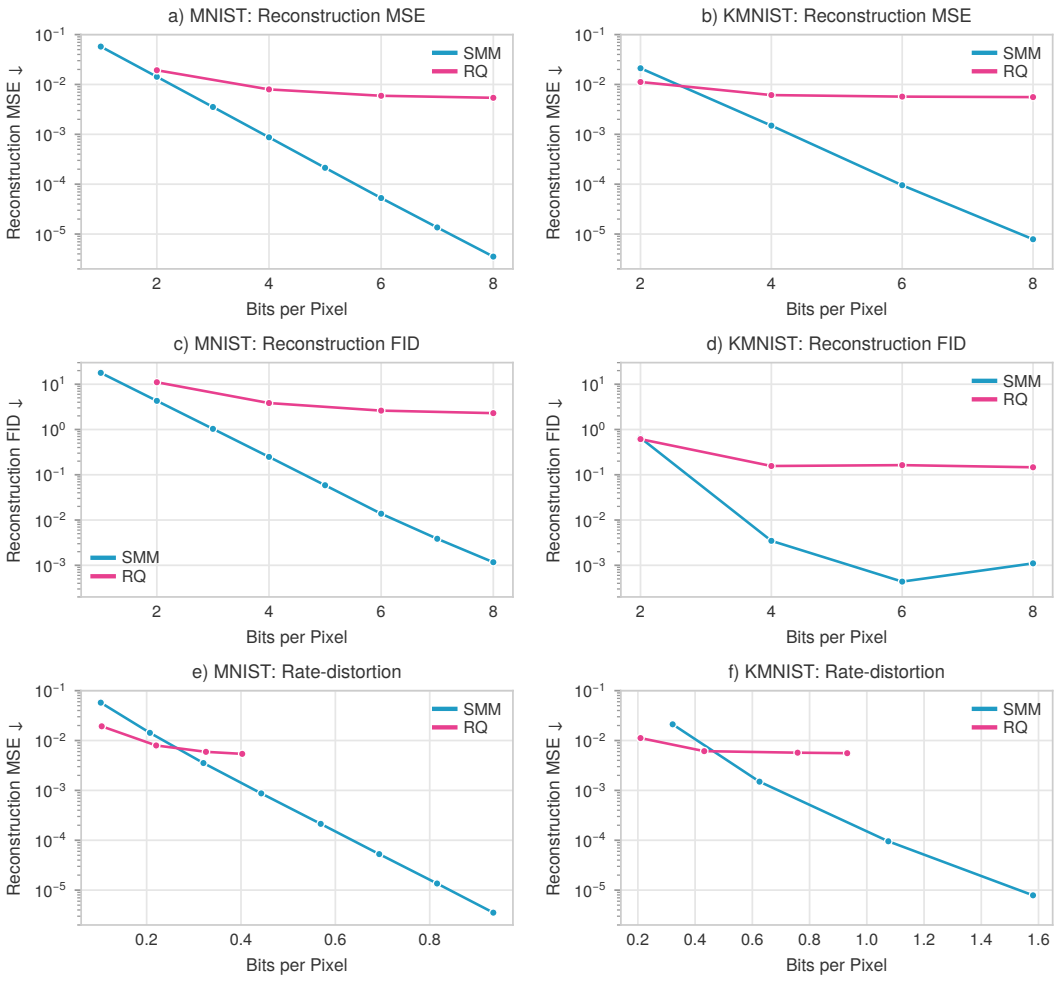


Figure 7: Pixel-space representation quality on MNIST (left) and KMNIST (right). We compare SMM and residual quantization (RQ) using (a,b) reconstruction MSE and (c,d) reconstruction FID versus nominal capacity, and (e,f) reconstruction MSE versus autoregressive coding cost. Both capacity and coding cost are expressed in bits per pixel. These metrics evaluate reconstruction from held-out inputs, not prior-sample quality.

D-dimensional cube and this finite symbol budget. The deterministic decoder receives only the code, without input-dependent side information.

The following applies the standard volumetric covering bound (Vershynin, 2019, Proposition 4.2.12) to the subdivision representation.

Proposition 1 (optimal worst-case reconstruction). For any encoder e and decoder d using S symbols from a J-element codebook,

$$R_\infty(e,d) := \sup_{x\in[0,1]^D} \|x - d(e(x))\|_\infty \geq \frac{1}{2J^{S/D}}. \tag{2}$$

SMM has exactly $K^{DT} = J^S$ distinct midpoint reconstructions and attains equality:

$$R_\infty^{\mathrm{SMM}} = \frac{1}{2K^T}, \qquad \sup_x \|x - \mathcal{Q}_T(x)\|_2 = \frac{\sqrt{D}}{2K^T}. \tag{3}$$

Consequently, every RQ representation with the same symbol budget satisfies $R_\infty^{\mathrm{RQ}} \geq R_\infty^{\mathrm{SMM}}$, for any choice of codebooks. For a single unscaled shared codebook decoded only through the sum, as in (Lee et al., 2022, §§3.1.2–3.1.3), the stronger comparison is

$$R_\infty^{\text{shared RQ}} \geq \frac{1}{2\binom{S+J-1}{J-1}^{1/D}} > R_\infty^{\mathrm{SMM}}, \qquad S, J \geq 2. \tag{4}$$

Separate or scaled codebooks can achieve equality in (2).

Corollary 1.1 (worst-case Wasserstein error). Write $W_{p,\ell}$ for Wasserstein distance with ground norm ℓ. Let $\mathcal{P}$ be all Borel probability measures on $[0,1]^D$ and $\mathcal{A}_M$ the deterministic Borel quantizers with at most $M = J^S = K^{DT}$ outputs. Then

$$\begin{aligned}\inf_{Q\in\mathcal{A}_M} \sup_{P\in\mathcal{P}} W_{1,\ell_\infty}(P, Q_\# P) &= \sup_{P\in\mathcal{P}} W_{1,\ell_\infty}(P, (\mathcal{Q}_T)_\# P) \\ &= \frac{1}{2M^{1/D}}.\end{aligned} \tag{5}$$

For Euclidean ground distance, with v_D the volume of the unit Euclidean ball,

$$\begin{aligned}\frac{1}{(Mv_D)^{1/D}} &\leq \inf_{Q\in\mathcal{A}_M} \sup_{P\in\mathcal{P}} W_{1,\ell_2}(P, Q_\# P) \\ &\leq \sup_{P\in\mathcal{P}} W_{1,\ell_2}(P, (\mathcal{Q}_T)_\# P) = \frac{\sqrt{D}}{2M^{1/D}}.\end{aligned} \tag{6}$$

Thus SMM is exactly minimax optimal for W_{1,ℓ_∞} and attains the optimal order $M^{-1/D}$ for Euclidean W_1 at fixed D. These statements concern reconstructed data distributions. Separate or scaled RQ codebooks reproducing the grid share both guarantees; learned generation probabilities are addressed next.

Precision and prediction cost. SMM achieves maximum-norm error at most $\varepsilon \in (0, 1/2)$ with $T = \lceil \log_K(1/(2\varepsilon)) \rceil$ levels while keeping the prediction size fixed. A shared affine head of hidden width H has $(H+1)J$ parameters. A single flat FSQ/VQ token matching the attained error $1/(2K^T)$ over the same block requires at least J^S classes and $(H+1)J^S$ parameters in an affine head. FSQ predicts the product index (Mentzer et al., 2024, §§3.2, 4.1); in the joint-token FSQ regime considered here, the per-channel ceiling is eight levels ($L_i \leq 8$), with $\prod_i L_i \leq M$ for an M-class head, following the configurations in (Mentzer et al., 2024, Table 1 and Appendix A.4.1). SMM increases the number of scalar reconstruction values to K^T through depth.

C DISTRIBUTION ERROR AND CONTINUATION

Let P be any Borel probability measure on $[0,1]^D$. Let A_t contain all depth-t digits, with $p_t(\cdot \mid A_{<t})$ and $\widehat{p}_t(\cdot \mid A_{<t})$ denoting the conditional distributions induced by P and the model's joint law, respectively. Let Q_T be the law of the model's depth-T midpoint output. Define the population conditional prediction error

$$e_t = \mathbb{E}_{A_{<t}\sim P}\,\mathrm{KL}(p_t(\cdot \mid A_{<t}) \,\|\, \widehat{p}_t(\cdot \mid A_{<t}))\,. \tag{7}$$

All logarithms are natural. Thus e_t equals population cross-entropy minus the target conditional entropy; it includes the factorization used by $\widehat{p}_t$.

Theorem 2 (distribution error). In this section W_p uses Euclidean ground distance. If all e_t are finite, then

$$W_1(P, Q_T) \le \frac{\sqrt{D}}{2K^T} + \sqrt{D}\sum_{t=1}^{T} K^{-(t-1)}\sqrt{\frac{e_t}{2}}. \tag{8}$$

The first term is the finite-resolution error. The second weights each conditional prediction error by the size of the cell refined at that level. Our conditional-KL derivation follows the multiscale transport principle of Weed and Bach (Weed & Bach, 2019, Definition 1 and Proposition 1); the proof is in Appendix F.

Corollary 2.1 (continuation). Suppose two samplers have the same depth-r prefix law and keep their outputs inside the selected prefix cells. For $0 \le r \le T$ and $1 \le p \le \infty$,

$$W_p(Q_T, \widetilde{Q}_T) \le \sqrt{D}K^{-r}, \qquad W_p(Q_T, Q_r) \le \frac{\sqrt{D}}{2K^r}, \tag{9}$$

where Q_r returns the prefix midpoint. Apply the same-cell coupling of (Weed & Bach, 2019, Lemma A.2): couple identical prefixes and bound the displacement by the cell diameter, or half that diameter for the midpoint. The almost-sure displacement bounds also cover $p = \infty$.

For a general hierarchical representation, the proof of (8) gives $\rho_T + \sum_t d_{t-1}\sqrt{e_t/2}$, where ρ_T is expected quantization error and d_{t-1} is the maximal diameter of decoded continuations after a prefix. SMM fixes both quantities geometrically through (1). In additive RQ space, a suffix diameter is at most $\sum_{s>r}\mathrm{diam}(C_s)$; depth alone does not prescribe its contraction. Under a fixed L_G-Lipschitz decoder, the transport bounds acquire a factor L_G.

D PARALLEL COORDINATE REFINEMENT

We next bound the cost of predicting the new coordinate digits independently, conditional on the preceding approximation of all coordinates. Before level t, this approximation identifies a product cell B_t of side $h_t = K^{-(t-1)}$. One model evaluation produces all coordinate probabilities, followed by independent coordinate sampling.

Assume that P has a strictly positive density q with $\varphi = \log q \in C^2([0,1]^D)$. Define

$$G = \sup_z \max_j |\partial_j \varphi(z)|, \qquad \mathsf{M}_{jk} = \begin{cases} \sup_z |\partial_j\partial_k\varphi(z)|, & j \ne k, \\ 0, & j = k, \end{cases} \qquad \kappa = \|\mathsf{M}\|_{\mathrm{op}},$$

$$\Omega_{\mathrm{off}} = \mathbb{E}_q \sum_{j\ne k} |\partial_j\partial_k\varphi(Z)|^2, \qquad c_h = \frac{h^2 e^{2Gh}}{\pi^2}, \qquad \theta_h = c_h\kappa. \tag{10}$$

The mixed-derivative score follows (Baptista et al., 2024, Eq. (4)). We use it to control the total correlation of the refinement vector.

Theorem 3 (cost of parallel refinement). If $\theta_{h_t} < 1$, then

$$\begin{aligned}\Delta_t &:= \mathbb{E}_{B_t \sim P}\,\mathrm{KL}\left(q(A_t \mid B_t) \,\middle\|\, \prod_{j=1}^{D} q(A_{t,j} \mid B_t)\right) \\ &\le C(h_t) h_t^4 \Omega_{\mathrm{off}}, \qquad C(h) = \frac{e^{4Gh}}{2\pi^4(1-\theta_h)}.\end{aligned} \tag{11}$$

Here Δ_t is exactly the excess optimal expected negative log-likelihood of independent coordinate predictors over a joint predictor. The proof combines interval functional inequalities with the mixed-derivative argument of Baptista et al. (Baptista et al., 2024, Theorem 2 and Appendix A); Appendix G supplies the required rectangular-domain spectral gap and the whole-vector bound.

Prediction error. For learned positive marginals $\widehat{q}_{t,j}$, set

$$\mathcal{E}_{\mathrm{learn},t} = \sum_j \mathbb{E}_{B_t \sim P}\,\mathrm{KL}(q(A_{t,j} \mid B_t) \,\|\, \widehat{q}_{t,j}(A_{t,j} \mid B_t))\,.$$

For $\widehat{p}_t = \prod_j \widehat{q}_{t,j}$, the fixed-order learning/factorization decomposition (Lavenant & Zanella, 2025, Proposition 2) gives

$$e_t = \Delta_t + \mathcal{E}_{\mathrm{learn},t} \le C(h_t) h_t^4 \Omega_{\mathrm{off}} + \mathcal{E}_{\mathrm{learn},t}. \tag{12}$$

All prefix expectations use the true law. The bound controls the cost of independent sampling with exact marginals; prediction error is accounted for separately.

If $\widetilde{P}_{1:T}$ uses exact joint transitions before level r and exact products of marginals from r onward, then, whenever $\theta_{h_r} < 1$,

$$\mathrm{KL}(P_{1:T} \| \widetilde{P}_{1:T}) = \sum_{t=r}^{T} \Delta_t \le \frac{C(h_r)}{1 - K^{-4}} h_r^4 \Omega_{\mathrm{off}}. \tag{13}$$

Using learned marginals instead adds exactly $\sum_{t=r}^{T} \mathcal{E}_{\mathrm{learn},t}$. Moreover, $\Omega_{\mathrm{off}} \le D\kappa^2$, so $\Delta_t / D \le C(h_t) h_t^4 \kappa^2$. This per-coordinate bound is uniform in dimension when G, κ are uniformly bounded and θ_{h_t} stays below a fixed constant smaller than one. The result applies to the stated smooth-density setting and nested product partition; it does not assume independent coordinates.

E Reconstruction bounds

Proof of Proposition 1. The volumetric covering argument (Vershynin, 2019, Proposition 4.2.12), applied with ℓ_∞ balls, gives $1 \le N(2r)^D$ for at most $N \le J^S$ outputs. This proves (2). SMM's K^{DT} cells have side K^{-T} and distinct midpoints, giving (3) and attaining (2).

For shared additive RQ, the sum depends only on the multiplicities of the J codewords. There are $\binom{S+J-1}{J-1}$ multiplicity vectors. For $S, J \ge 2$, this is strictly smaller than J^S because different ordered strings have identical multiplicities. A deterministic decoder of the sum cannot distinguish those strings. Applying the covering bound proves (4). □

Proof of Corollary 1.1. For any fixed quantizer Q and either ground norm ℓ,

$$\sup_{P \in \mathcal{P}} W_{1,\ell}(P, Q_{\#}P) = \sup_{x \in [0,1]^D} \|x - Q(x)\|_\ell. \tag{E1}$$

The coupling $(X, Q(X))$ gives the upper bound; taking $P = \delta_x$ gives the reverse inequality. Proposition 1 proves (5). The Euclidean volumetric bound (Vershynin, 2019, Proposition 4.2.12) gives

$1 \le M v_D r^D$, while (3) gives SMM's radius, proving (6). The same argument applies to $W_{p,\ell}$ for every $1 \le p \le \infty$. □

Equation (E1) also transfers the strict comparison (4) to W_{1,ℓ_∞}. For the shared-additive architecture, the reconstruction count gives a Euclidean worst-case Wasserstein lower bound

$$[v_D \binom{S+J-1}{J-1}]^{-1/D}.$$

At fixed $J \ge 2$ and D, this is $\Omega(S^{-(J-1)/D})$, whereas SMM's error is $\frac{\sqrt{D}}{2} J^{-S/D}$. This separation in the number of symbols requires sharing, no stage-dependent scaling, and decoding only through the sum.

F TRANSPORT BOUNDS

Proof of Theorem 2. Let H_t use target conditionals through level t and model conditionals afterward, followed by the midpoint decoder. Then $H_0 = Q_T$ and $H_T = (\mathcal{Q}_T)_{\#}P$. Consecutive hybrids share a target-distributed prefix of length $t-1$. Couple their next digits maximally. If these agree, use identical model suffixes; otherwise, both outputs remain in the same parent cell of diameter $\sqrt{D}K^{-(t-1)}$. Hence

$$W_1(H_{t-1}, H_t) \le \sqrt{D}K^{-(t-1)} \mathbb{E}_P \, \mathrm{TV}(p_t, \widehat{p}_t) \le \sqrt{D}K^{-(t-1)} \sqrt{e_t/2}, \tag{F1}$$

where the last step is Pinsker's inequality and Jensen's inequality. Sum over t and use

$$W_1(P, (\mathcal{Q}_T)_{\#}P) \le \sqrt{D}/(2K^T).$$

□

G PARALLEL-REFINEMENT BOUND

G.1 FUNCTIONAL INEQUALITIES ON A CELL

We normalize the logarithmic Sobolev inequality (LSI) as

$$\mathrm{Ent}_\mu(f) \le \frac{c}{2} \mathbb{E}_\mu \frac{\|\nabla f\|_2^2}{f}, \qquad \mathrm{Ent}_\mu(f) = \mathbb{E}_\mu[f \log f] - \mathbb{E}_\mu[f] \log \mathbb{E}_\mu[f]. \tag{G1}$$

The uniform law on a length-h interval satisfies (G1) with $c = h^2/\pi^2$ (Ghang et al., 2014, Theorem 2.2 and Corollary 2.4). If a positive interval density has log-derivative bounded by G, its density relative to the uniform law lies in $[e^{-Gh}, e^{Gh}]$. The bounded-perturbation comparison (Ledoux, 2001, Lemma 1.2) therefore gives the admissible constant $c_h = h^2 e^{2Gh}/\pi^2$. Linearizing at $f = 1$ gives $\mathrm{Var}_\mu(v) \le c_h \mathbb{E}_\mu |v'|^2$. Products of these interval laws satisfy (G1) with the same c_h by tensorization (Ledoux, 2001, Lemma 1.1).

Lemma G.1 (rectangular spectral gap). Let μ have density proportional to e^ψ on a bounded rectangle $R \subset \mathbb{R}^n$, with $\psi \in C^2(\overline{R})$. Suppose its single-coordinate conditionals have Poincaré constants at most c. If $|\psi_{ij}| \le N_{ij}$ for $i \ne j$, where N is symmetric, entrywise nonnegative, has zero diagonal, and $\|N\|_{\mathrm{op}} \le \kappa < c^{-1}$, then

$$\mathrm{Var}_\mu(v) \le (c^{-1} - \kappa)^{-1} \mathbb{E}_\mu \|\nabla v\|_2^2, \qquad v \in H^1(R). \tag{G2}$$

Proof. We adapt the single-site argument in (Ledoux, 2001, Proposition 3.1) to the rectangle. Set $L_i u = u_{ii} + \psi_i u_i$ and $L = \sum_i L_i$. For a finite product-cosine polynomial u, conditional integration by parts and conditional Poincaré give

$$\mathbb{E}_\mu (L_i u)^2 \ge c^{-1} \mathbb{E}_\mu u_i^2, \qquad \mathbb{E}_\mu (L_i u)(L_j u) = \mathbb{E}_\mu u_{ij}^2 - \mathbb{E}_\mu \psi_{ij} u_i u_j \quad (i \ne j). \tag{G3}$$

For the first inequality, apply Cauchy–Schwarz to $\mathbb{E}_i u_i^2 = -\mathbb{E}_i[(u - \mathbb{E}_i u)L_i u]$. For the second identity, integrate first in i, then in j; the boundary terms vanish since $u_i = 0$ on the i-faces and $u_{ij} = \partial_i u_j = 0$ on the j-faces. No boundary condition is imposed on ψ. Expanding the square and using $a^\top N a \le \kappa \|a\|_2^2$, $a_i = |u_i|$, yields

$$\mathbb{E}_\mu (Lu)^2 \ge (c^{-1} - \kappa)\mathbb{E}_\mu \|\nabla u\|_2^2. \tag{G4}$$

The density w of μ is bounded above and away from zero. Compactness of $H^1(R) \hookrightarrow L^2(R)$ therefore gives a minimizer of the Dirichlet energy subject to $\mathbb{E}_\mu u = 0$ and $\mathbb{E}_\mu u^2 = 1$, with minimum $\lambda_1 > 0$. Its weak equation is $\int w \nabla u \cdot \nabla v = \lambda_1 \int wuv$. Testing with v/w gives the ordinary weak Neumann problem

$$-\Delta u = F, \qquad F = \lambda_1 u + \nabla \psi \cdot \nabla u \in L^2(R). \tag{G5}$$

Neumann H^2 regularity follows from (Ern & Guermond, 2017, Theorem 27.26(ii)), which cites (Ern & Guermond, 2021). For the approximation needed in (G4), let e_k be the normalized product-cosine basis, $-\Delta e_k = \nu_k e_k$, and $a_k = \langle u, e_k \rangle$. Equation (G5) gives $\nu_k a_k = \langle F, e_k \rangle$, with $\int F = 0$. Thus $\sum_k \nu_k^2 |a_k|^2 = \|F\|_2^2$. For any finite cosine sum $p = \sum_k b_k e_k$,

$$\|p\|_2^2 + \|\nabla p\|_2^2 + \sum_{i,j} \|\partial_{ij} p\|_2^2 = \sum_k (1 + \nu_k + \nu_k^2)|b_k|^2. \tag{G6}$$

Parseval's identity and (G6) imply that the cosine partial sums of u converge in H^2. Boundedness of $\nabla\psi$ and w allows passage to the limit in (G4). Since $Lu = -\lambda_1 u$ and $\mathbb{E}_\mu \|\nabla u\|_2^2 = \lambda_1 \mathbb{E}_\mu u^2$, (G4) gives $\lambda_1 \ge c^{-1} - \kappa$. The variational characterization proves (G2) for all H^1 functions; the test function v requires no Neumann condition. □

G.2 Conditional dependence

Proof of Theorem 3. Fix a side-h cell B, let $\mu = q(\cdot \mid B)$, denote its coordinate marginals by μ_j, and set $\nu = \bigotimes_j \mu_j$. Every one-coordinate conditional of μ has log-derivative bounded by G. Write $B = \prod_{k=1}^D I_k$ with intervals I_k of length h, and $z = (z_j, z_{-j})$ with $z_{-j} \in B_{-j} := \prod_{k \ne j} I_k$. Then μ has density $q(z \mid B) = e^{\varphi(z)}/P(B)$ on B, $P(B) = \int_B e^\varphi$, and its j-th marginal is

$$\mu_j(z_j) = P(B)^{-1} \int_{B_{-j}} e^{\varphi(z_j, z_{-j})}\, dz_{-j}.$$

The domain B_{-j} does not depend on z_j, and $e^\varphi \partial_j \varphi$ is continuous on the compact cell since $\varphi \in C^2$, so differentiation under the integral sign gives

$$\partial_j \mu_j(z_j) = P(B)^{-1} \int_{B_{-j}} e^{\varphi(z)}\, \partial_j \varphi(z)\, dz_{-j}.$$

Dividing by $\mu_j(z_j)$ cancels $P(B)$, and $e^{\varphi(z_j,\cdot)} \big/ \int_{B_{-j}} e^{\varphi(z_j,\cdot)}$ is the conditional density of Z_{-j} given $Z_j = z_j$, hence

$$\partial_j \log \mu_j(z_j) = \frac{\int_{B_{-j}} e^\varphi\, \partial_j \varphi\, dz_{-j}}{\int_{B_{-j}} e^\varphi\, dz_{-j}} = \mathbb{E}_\mu[\partial_j \varphi(Z) \mid Z_j = z_j]. \tag{G7}$$

Since $|\partial_j \varphi| \le G$, also $|\partial_j \log \mu_j| \le G$. Each marginal is therefore a bounded perturbation of the uniform law on I_j with density ratio in $[e^{-Gh}, e^{Gh}]$ and satisfies (G1) with c_h; by tensorization ν satisfies (G1) with the same c_h. For fixed z_j, the law of the remaining coordinates has single-coordinate Poincaré constants at most c_h and interaction matrix a principal submatrix of M. Lemma G.1 yields

$$\mathrm{Var}_\mu(v \mid Z_j) \le \frac{c_h}{1 - \theta_h} \mathbb{E}_\mu \left[\sum_{k \ne j} |\partial_k v|^2 \,\middle|\, Z_j \right]. \tag{G8}$$

For $D = 1$, both sides are zero. Let $f = d\mu/d\nu = q(\cdot \mid B)/\prod_k \mu_k$, so that

$$\log f(z) = \varphi(z) - \sum_k \log \mu_k(z_k) - \log P(B).$$

Only the $k = j$ term depends on z_j, hence by (G7)

$$\partial_j \log f = \partial_j \varphi - \partial_j \log \mu_j = \partial_j \varphi - \mathbb{E}_\mu[\partial_j \varphi \mid Z_j], \qquad \mathbb{E}_\mu |\partial_j \log f|^2 = \mathbb{E}_\mu \operatorname{Var}_\mu(\partial_j \varphi \mid Z_j).$$

. Applying the product LSI followed by (G8) gives

$$\begin{aligned} \mathrm{KL}(\mu \| \nu) &\le \frac{c_h}{2} \sum_j \mathbb{E}_\mu \operatorname{Var}_\mu(\partial_j \varphi \mid Z_j) \\ &\le \frac{c_h^2}{2(1-\theta_h)} \mathbb{E}_\mu \sum_{j \neq k} |\partial_j \partial_k \varphi|^2. \end{aligned} \tag{G9}$$

This is the LSI-to-mixed-derivative argument of (Baptista et al., 2024, Appendix A, Eqs. (44)–(45)), using the higher-dimensional conditional slice (G8). Coordinatewise quantization pushes μ to $q(A_t \mid B)$ and ν to the product of its digit marginals. The data-processing inequality for KL therefore bounds the digit total correlation by (G9). Averaging over the cells recovers Ω_{off} and gives (11). The difference between the optimal product and joint log-losses is $\sum_j H(A_{t,j} \mid B_t) - H(A_t \mid B_t) = \Delta_t$. □

G.3 Learning error and accumulated refinement

At each fixed $B_t = b$, insert the product of true marginals into the likelihood ratio:

$$\log \frac{q(a \mid b)}{\prod_j \widehat{q}_{t,j}(a_j \mid b)} = \log \frac{q(a \mid b)}{\prod_j q(a_j \mid b)} + \sum_j \log \frac{q(a_j \mid b)}{\widehat{q}_{t,j}(a_j \mid b)}. \tag{G10}$$

Taking target expectations gives (12), the deterministic-level specialization of (Lavenant & Zanella, 2025, Proposition 2). This identity needs no density regularity. The KL chain rule gives (13) because the transitions agree before r. For $t \ge r$, $C(h_t) \le C(h_r)$ and $\sum_{t=r}^{T} h_t^4 \le h_r^4/(1 - K^{-4})$. The same chain rule adds the marginal learning errors for learned transitions. Finally, $\Omega_{\mathrm{off}} \le \|\mathsf{M}\|_F^2 \le D\kappa^2$, proving the dimension statement.

H STE discrepancy

We consider encoder training through quantization with the identity straight-through estimator. For a fixed composed decoder-and-task loss F, define $\Gamma_Q^F(z) = \|\nabla F(Q(z)) - \nabla F(z)\|_2$. Following the gradient-gap argument in (Huh et al., 2023, §3.2, Eq. (14)), L-Lipschitz continuity of ∇F gives $\Gamma_Q^F(z) \le L\|Q(z) - z\|_2$.

To state the worst-case comparison, let

$$\mathfrak{G}_L(Q) = \sup_{\substack{F \in C^1(\mathbb{R}^D) \\ \mathrm{Lip}(\nabla F) \le L}} \sup_{z \in [0,1]^D} \Gamma_Q^F(z). \tag{H1}$$

The quadratic loss $F_*(z) = L\|z\|_2^2/2$ attains the smoothness bound at every input. Therefore

$$\mathfrak{G}_L(Q) = L \sup_z \|Q(z) - z\|_2, \qquad \mathfrak{G}_L(\mathcal{Q}_T) = \frac{L\sqrt{D}}{2K^T}. \tag{H2}$$

For an equal-width coordinatewise midpoint grid with N_i cells, the first quantity is

$$\frac{L}{2} \left(\sum_i N_i^{-2} \right)^{1/2}.$$

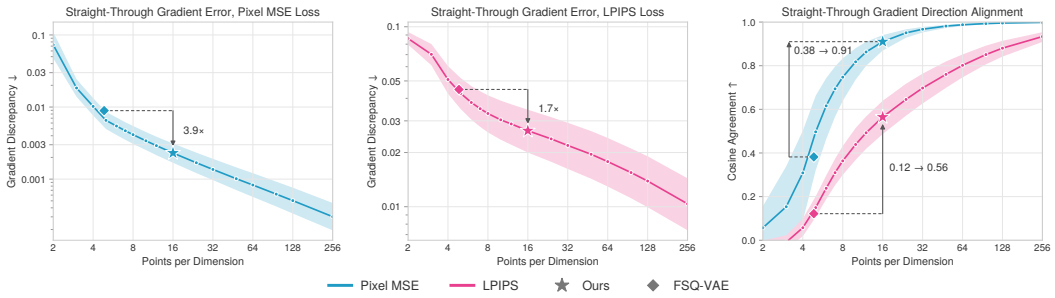


Figure 8: **Straight-through gradient discrepancy and alignment.** Results on 10 000 ImageNet validation images with a frozen VQ-8 decoder (Sun et al., 2024). Lines show medians; shaded regions show interquartile ranges. Left and middle: gradient discrepancy Γ_Q^F for pixel MSE and LPIPS, respectively. Right: cosine similarity between the straight-through gradient $\nabla F(Q(z))$ and the unquantized gradient $\nabla F(z)$. The horizontal axis gives the number of grid cells per coordinate. Stars mark two-level SMM ($K = 4$, $N = 16$); diamonds mark the single-level FSQ grid (the depth-one special case of SMM) with $L_i = (5, 5, 5, 5, 5, 5, 4)$ and 62 500 codes.

Under uniform inputs, $\mathbb{E}(\Gamma_Q^{F_*})^2 = \frac{L^2}{12}\sum_i N_i^{-2}$. If a joint-token FSQ quantizer on the same block has at most M reconstruction values, (2) and $\|\cdot\|_2 \geq \|\cdot\|_\infty$ give

$$\mathfrak{G}_L(Q_{\mathrm{FSQ}}) \geq \frac{L}{2M^{1/D}}, \qquad \frac{\mathfrak{G}_L(\mathcal{Q}_T)}{\mathfrak{G}_L(Q_{\mathrm{FSQ}})} \leq \sqrt{D}\, M^{1/D} K^{-T} \quad (L > 0). \tag{H3}$$

Thus subdivision reduces the worst-case discrepancy bound through depth at fixed head size. In the $L_i \leq 8$ joint-token regime of Appendix B, any scalar quantizer covering $[0, 1]$ has worst-case error at least $1/(2L_i) \geq 1/16$; scalar SMM with an eight-way head reaches $1/(2 \cdot 8^T)$ after T levels. The equality and lower bound in (H2)–(H3) are over the loss class in (H1), not every fixed loss. For an encoder Jacobian of operator norm at most B, the fixed-loss parameter-gradient discrepancy is at most $BL\sqrt{D}/(2K^T)$. These statements concern the smooth surrogate gradient, not the almost-everywhere-zero derivative of $F \circ Q$.

The same bound controls gradient direction. For a fixed F and any z with $\|\nabla F(z)\|_2 > \Gamma_Q^F(z)$, both gradients are nonzero and the angle θ between them satisfies

$$\sin\theta \leq \frac{\Gamma_Q^F(z)}{\|\nabla F(z)\|_2} \leq \frac{L\|Q(z) - z\|_2}{\|\nabla F(z)\|_2}. \tag{H4}$$

Thus, for any fixed z with $\nabla F(z) \neq 0$, increasing subdivision depth aligns the straight-through gradient with the unquantized gradient: $\theta = O(K^{-T})$ and $1 - \cos\theta = O(K^{-2T})$.

Measurement. We measure Γ_Q^F and $\cos\theta$ on 10 000 ImageNet validation images with the frozen VQ-8 decoder, for pixel MSE and LPIPS (Figure 8). We use coordinatewise midpoint grids with N cells per CDF coordinate. Increasing N reduces the discrepancy and improves gradient alignment for both losses. For the evaluated single-level FSQ grid with 62 500 codes, the straight-through gradient has cosine similarity of only 0.38 with the unquantized gradient for pixel MSE and 0.12 for LPIPS. Two-level SMM ($K = 4$, $N = 16$) increases these values to 0.91 and 0.56, respectively, while reducing Γ_Q^F by factors of 3.9 and 1.7. Subdivision therefore provides the encoder with a gradient closer to that of the unquantized reconstruction objective, reducing the distortion of its training signal caused by quantization.

H.1 AI Use Statement

We used generative AI tools to assist with writing and improve the clarity, grammar and consistency of the manuscript. All AI-text has been reviewed and revised by the authors, who take full responsibility for the final manuscript.

I LIMITATIONS AND FUTURE WORK.

Despite its effectiveness across the evaluated settings, Subdivision Measure Matching has certain limitations, particularly in the autoregressive loop over patches, which scales linearly with the number of spatial positions N and dominates the sampling cost on high-resolution data. We partially mitigate this through the joint-patch predictions and parallel sampling starting from some depth level, but a fully parallel sampling procedure remains an open problem. Future work includes non-causal within-level factorizations, scale-wise generation order, and non-axis-aligned partitions.