LEARNING SPATIAL-PYRAMID VISUAL REPRESENTATIONS FOR AUTOREGRESSIVE IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive image generation relies on two coupled components: visual token representation and token distribution modeling. Existing tokenizers either preserve spatial correspondence through single-scale grids or aggregate global information with flat sequences, without explicitly organizing tokens across spatial granularities. Although end-to-end training aligns representation learning with generation objectives, it does not by itself determine how hierarchical visual information should be organized for conditional prediction. We introduce a ***spatial-pyramid 1D visual representation*** that organizes global tokens and regional tokens at multiple spatial granularities within a single 1D sequence. The corresponding context model generates global tokens causally and predicts each regional level in parallel, conditioned on all coarser groups, yielding a global-to-local factorization. This factorization retains all latent tokens while reducing sequential decoding depth compared with token-wise prediction. We jointly optimize the tokenizer and context model so that the learned representation adapts to the proposed factorization. On ImageNet at 256×256 resolution, our method achieves a gFID of **1.21** with classifier-free guidance using 19 sequential decoding steps.

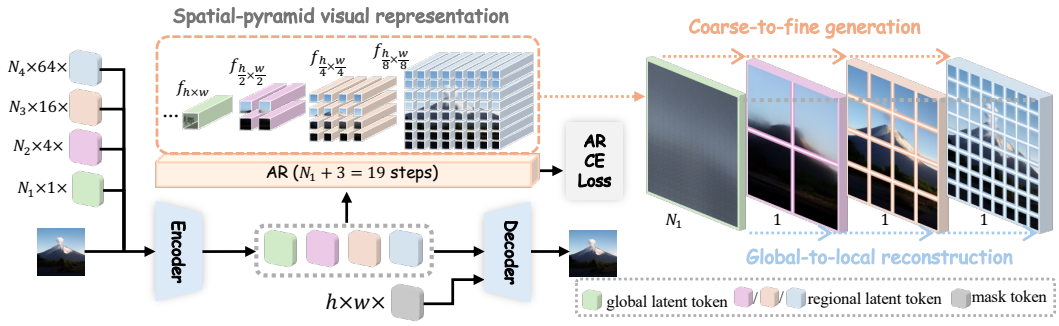


Figure 1: **Spatial-pyramid 1D visual representation for reconstruction and autoregressive generation.** The encoder extracts one global group and three progressively finer regional groups, with N_s tokens per cell. The decoder reconstructs images from these groups and mask tokens. The AR model generates global tokens causally and each regional level in parallel, conditioned on all coarser groups. The right-hand stages illustrate global-to-local reconstruction from encoded prefixes and coarse-to-fine generation from sampled prefixes. This schedule requires N_1+3 sequential steps.

1 INTRODUCTION

Autoregressive (AR) image generation models extend the next-token prediction paradigm from language to visual data and have recently achieved strong generative performance (Van Den Oord et al., 2017; Esser et al., 2021; Yu et al., 2021; Lee et al., 2022; Chang et al., 2022; Sun et al., 2024; Li et al., 2024; Tian et al., 2024). At the core of these models are two tightly coupled design choices: ***how image content is represented as visual tokens and how their joint distribution is modeled.*** The representation determines what each token encodes and how visual information is organized, whereas the context model determines how the joint distribution of these tokens is factorized into conditional predictions. Because the former defines the variables to be predicted and the latter defines their dependency structure, the two components should be designed in concert.

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.

Historically, AR image generators have predominantly adopted a ***two-stage pipeline***, which remains common in recent work (Weber et al., 2024; Ren et al., 2025; Pang et al., 2025; Li et al., 2025; Ai et al., 2026). In this paradigm, a tokenizer is first trained to encode images into latent tokens, typically under a reconstruction-oriented objective, and a separately optimized context model then predicts these tokens through raster-scan autoregression (Esser et al., 2021), iterative masked prediction (Chang et al., 2022), or other prediction strategies (Huang et al., 2025b; Han et al., 2026; Pang et al., 2026). Although modular and effective, this separation causes the learned token space to be optimized primarily for reconstruction rather than for the conditional distributions required by generation. Within this two-stage paradigm, ***grid-based 2D tokenizers*** (Lee et al., 2022) retain explicit spatial correspondence through downsampled latent grids, but their token count remains tied to image resolution and a fixed spatial granularity. ***Sequence-based 1D tokenizers*** (Yu et al., 2024; Xiong et al., 2025; Miwa et al., 2025; Wen et al., 2025; Fu et al., 2026; Chi et al., 2026; Qu et al., 2026) instead use learnable latent tokens without fixed spatial layouts, yet their flat organization does not explicitly associate tokens with image regions or spatial granularities. More structured organizations (Tian et al., 2024; Zheng et al., 2025a; Zhang et al., 2026c) further demonstrate the benefit of hierarchical token structures, but the underlying two-stage optimization still leaves the representation space disconnected from the conditional distributions modeled by the context model.

Recent ***end-to-end approaches*** instead jointly optimize the tokenizer and the context model (Chu et al., 2026; Lin et al., 2026b; Zheng et al., 2026a; Jiang et al., 2026), allowing generator-derived supervision to update the tokenizer (Wu et al., 2025; Duggal et al., 2026; Zheng et al., 2026b) and thereby reducing the objective mismatch between reconstruction and generation. This objective alignment, however, does not determine how image-wide and regional information should be organized across spatial granularities or how the resulting token dependencies should be factorized for conditional prediction. Without an explicit hierarchy, the learned token space may remain weakly structured, leaving the context model to infer global, regional, and cross-granularity relationships implicitly. Moreover, token-wise autoregression still requires N sequential decoding steps to generate an N-token representation (Sun et al., 2024; Yu et al., 2025a). Thus, beyond joint optimization, the central challenge is to co-design a hierarchical representation that exposes complementary variables across spatial granularities and a matching context model that exploits this structure through an appropriate conditional factorization.

We therefore introduce an end-to-end AR framework that couples a ***spatial-pyramid 1D visual representation with a matching global-to-local context model***. The representation organizes one group of N_1 global tokens and three levels of regional tokens within a single 1D latent sequence. The global tokens capture image-wide content, whereas the regional tokens represent patterns at progressively finer spatial granularities. Within each regional level, tokens are assigned to individual cells of a nested image partition (Figure 1). The same spatial pyramid induces a global-to-local factorization of the token distribution. The context model first generates the global tokens causally, capturing their mutual dependencies and establishing shared image-wide context. It then predicts all tokens within each regional level in parallel, conditioned on the global tokens and all coarser regional levels. We jointly optimize the tokenizer and the context model under this factorization, allowing the representation to adapt directly to the conditional structure it must support.

Our main contributions are as follows:

- We introduce a spatial-pyramid 1D visual representation that assigns complementary roles to global tokens and regional tokens across multiple spatial granularities.
- We develop a matching end-to-end context model that generates global tokens causally and predicts each regional level in parallel, conditioned on all coarser groups. This global-to-local factorization retains all N tokens while reducing sequential decoding to N_1+3 steps.
- On ImageNet at 256×256 resolution, our method achieves a gFID of **1.21** using 19 sequential decoding steps. Controlled comparisons support the hierarchical formulation, while token-group interventions reveal the complementary roles of global and regional tokens.

2 Related Work

Visual token representation learning. Visual tokenization determines how visual information is organized in the latent space for generation. Early grid-based tokenizers (Van Den Oord et al., 2017;

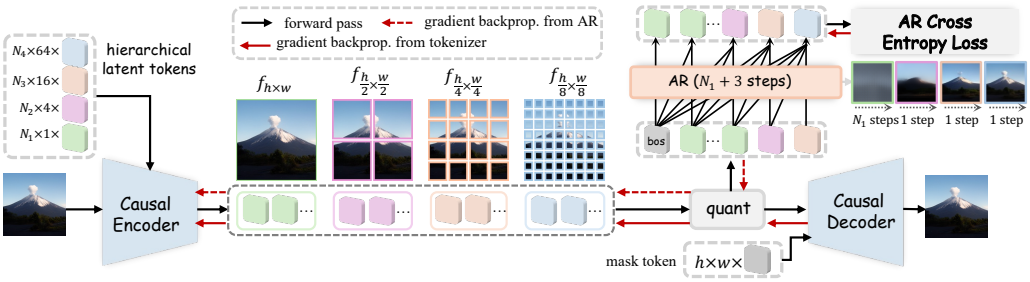


Figure 2: **Overall framework for end-to-end autoregressive image generation with spatial-pyramid 1D representations.** The encoder produces global tokens and three progressively finer levels of region-specific tokens, which are quantized and concatenated into a single sequence. The AR model generates global tokens autoregressively, then predicts each regional level in parallel, conditioned on the global tokens and all coarser regional levels. The decoder reconstructs the image from all quantized tokens using $h \times w$ learned mask tokens.

Esser et al., 2021; Yu et al., 2023b; 2026) represent images as spatial grids of discrete codes, preserving explicit correspondence between token positions and image regions. Sequence-based tokenizers, such as TiTok (Yu et al., 2024), FlexTok (Bachmann et al., 2025), and SpectralAR (Huang et al., 2025a), replace fixed spatial layouts with compact or adaptive token sequences. Beyond flat sequences, hierarchical approaches introduce explicit multi-scale organizations, including multi-scale token maps with scale-causal attention (Chen et al., 2025), next-scale prediction (Tian et al., 2024; Zhang et al., 2026b), and holistic-to-local representations (Zheng et al., 2025a). These approaches demonstrate the importance of structured token organization, but they mainly focus on representation design rather than the conditional factorization used for generation. In contrast, our spatial-pyramid 1D representation jointly defines token roles and prediction groups through nested spatial partitions.

Structured token modeling. The organization of visual tokens determines how their joint distribution is factorized and which dependencies are modeled sequentially or in parallel. Existing models explore different prediction factorization strategies, including causal next-token prediction in LlamaGen (Sun et al., 2024), masked iterative refinement in MaskGIT (Chang et al., 2022), and hierarchical prediction in VAR (Tian et al., 2024) and ResTok (Zhang et al., 2026c). Other structured formulations extend hierarchical prediction to continuous latent spaces through flow matching (Ren et al., 2024) or adaptive token allocation (Wang et al., 2025). Different from scale-based or residual-level grouping, our prediction groups are defined by nested spatial assignments, enabling causal global generation and parallel regional prediction conditioned on coarser groups.

End-to-end generation-aware training. Beyond representation and factorization design, recent work investigates how to align tokenizer optimization with generation objectives. Joint optimization has been explored in latent diffusion (Duggal et al., 2026), generative compression (Jiang et al., 2026), generative image super-resolution (Li et al., 2026), and AR image generation. For AR generation, EOSTok (Chu et al., 2026), GEAR (Lin et al., 2026b), and Prologue (Zheng et al., 2026a) explore different forms of generation-aware tokenizer optimization, allowing representations to better align with downstream generation. While these methods reduce the objective mismatch between representation learning and generation, how to organize the learned token space remains an open design question. Different from these approaches, we jointly optimize the tokenizer and context model so that the spatial-pyramid 1D representation adapts to its global-to-local factorization.

3 Preliminaries

We briefly review flat 1D tokenization, token-wise AR modeling, and their end-to-end training objective. This formulation serves as the baseline denoted *Vanilla* in the experiments.

Tokenization. Given an image $\boldsymbol{x}$, a patch embedding layer maps its $h \times w$ patch grid to $\boldsymbol{p} = P(\boldsymbol{x}) = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{hw}) \in \mathbb{R}^{hw \times d}$ (Dosovitskiy et al., 2020). The encoder E_ϕ processes this patch sequence together with N learned flat latent inputs $\boldsymbol{L}^{\text{flat}} = (\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_N)$:

$$[\boldsymbol{p}'; \boldsymbol{f}^{\text{flat}}] = E_\phi([\boldsymbol{p}; \boldsymbol{L}^{\text{flat}}]), \qquad \boldsymbol{f}^{\text{flat}} = (\boldsymbol{f}_{h\times w,1}, \ldots, \boldsymbol{f}_{h\times w,N})\,. \tag{1}$$

Only the outputs at latent-token positions are retained as the continuous representation $\boldsymbol{f}^{\text{flat}}$. Here, $\boldsymbol{f}_{a\times b,i}$ denotes the i-th latent feature assigned to an $a \times b$ region of the patch grid. The spatial subscript specifies its assigned support, not its tensor shape. For flat tokenization, $a = h$ and $b = w$, so latent tokens have full-image spatial support without predefined regional assignments.

A vector quantizer Q next maps each feature to a hard code index u_i and a quantized feature $\boldsymbol{f}_i^q$ (Esser et al., 2021; Yu et al., 2023a; Lin et al., 2026a). The indices form $\boldsymbol{u} = (u_1, \ldots, u_N)$, while the quantized features form the decoder input $\boldsymbol{f}^{q,\text{flat}} = (\boldsymbol{f}_1^q, \ldots, \boldsymbol{f}_N^q)$. For reconstruction, the decoder also receives $\boldsymbol{m} \in \mathbb{R}^{hw\times d}$, obtained by repeating a learned mask token over the patch positions. Quantization and reconstruction are thus written as

$$(u_i, \boldsymbol{f}_i^q) = Q(\boldsymbol{f}_{h\times w,i}), \qquad \widehat{\boldsymbol{x}}_{\text{flat}} = D_\psi\big([\boldsymbol{f}^{q,\text{flat}}; \boldsymbol{m}]\big). \tag{2}$$

Factorization. Under a token-wise AR factorization, the generator predicts these indices conditioned on an optional class label y:

$$p_\theta(\boldsymbol{u} \mid y) = \prod_{i=1}^{N} p_\theta(u_i \mid \boldsymbol{u}_{<i}, y). \tag{3}$$

Here, $\boldsymbol{u}_{<i} = (u_1, \ldots, u_{i-1})$ denotes the prefix preceding u_i. Shifted teacher forcing evaluates the next-token losses in parallel during training. Inference under this factorization requires N sequential decoding steps because each prediction depends on previously sampled indices $\boldsymbol{u}_{<i}$.

To let AR supervision update the tokenizer, end-to-end training introduces a differentiable context path through discrete code selection. A probability-based straight-through estimator (STE) provides this path from the AR loss to the encoder (Jang et al., 2017; Yin et al., 2019; Shi et al., 2025). For each normalized encoder feature $\bar{\boldsymbol{f}}_i$, its similarities to the normalized codebook vectors $\{\bar{\boldsymbol{e}}_k\}_{k=1}^K$ define the soft assignment $\pi_{ik} = \text{softmax}_k\big(\bar{\boldsymbol{f}}_i^\top \bar{\boldsymbol{e}}_k/\tau\big)$. The softmax runs over the K codebook entries, with temperature τ. The most probable entry determines the hard assignment $h_{ik} = \mathbf{1}[k = k_i^\star]$, where $k_i^\star = \arg\max_r \pi_{ir}$. Combining the hard and soft assignments yields a straight-through assignment and a differentiable AR context embedding:

$$\widetilde{\boldsymbol{h}}_i = \boldsymbol{\pi}_i + \text{sg}(\boldsymbol{h}_i - \boldsymbol{\pi}_i), \qquad \boldsymbol{v}_i^{\text{AR}} = \sum_{k=1}^{K} \widetilde{h}_{ik} \boldsymbol{v}_k. \tag{4}$$

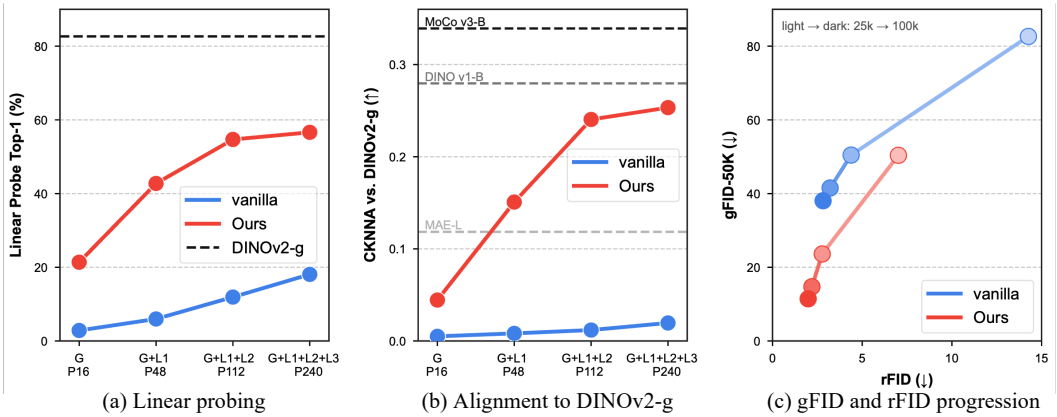
Here, $\text{sg}(\cdot)$ denotes stop-gradient, and $\{\boldsymbol{v}_k\}_{k=1}^K$ is the AR embedding table. The forward pass uses $\widetilde{\boldsymbol{h}}_i = \boldsymbol{h}_i$, so the context and detached target share the same hard index $u_i = k_i^\star$. During backpropagation, gradients flow through the context embedding and $\boldsymbol{\pi}_i$ to the encoder, while the target remains detached. This gradient path enables joint optimization of the tokenizer and AR generator:

$$\mathcal{L}_{\text{E2E}}^{\text{flat}} = \underbrace{\lambda_2 \mathcal{L}_2 + \lambda_{\text{perc}} \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{VQ}} \mathcal{L}_{\text{VQ}} + \lambda_G \mathcal{L}_{\text{GAN}}^G}_{\mathcal{L}_{\text{rec}}} + \lambda_{\text{AR}} \mathcal{L}_{\text{AR}}^{\text{flat}}. \tag{5}$$

Here, $\mathcal{L}_{\text{rec}}$ collects the tokenizer losses. $\mathcal{L}_2$ and $\mathcal{L}_{\text{LPIPS}}$ denote the pixel and perceptual losses (Zhang et al., 2018), respectively. $\mathcal{L}_{\text{VQ}}$, $\mathcal{L}_{\text{GAN}}^G$, and $\mathcal{L}_{\text{AR}}^{\text{flat}}$ denote the vector-quantization, generator adversarial, and cross-entropy losses, respectively. Each λ coefficient controls the relative contribution of its associated loss. Joint optimization adapts the flat representation to the generative objective without introducing explicit regional assignments.

4 METHOD

We extend the flat 1D baseline with a spatial-pyramid representation that assigns global and regional tokens complementary spatial roles (Figure 2). Inspired by image pyramids (Burt & Adelson, 1987), we use nested spatial partitions to specify these tokens' assigned regions and granularities. We use these assignments to define prediction groups and their global-to-local conditioning order. Global tokens are generated causally, whereas each regional level is predicted in parallel conditioned on all coarser groups. The following subsections describe the tokenizer (Section 4.1), AR model (Section 4.2), and joint training and inference (Section 4.3).



(a) Linear probing (b) Alignment to DINOv2-g (c) gFID and rFID progression

Figure 3: **Decoder-feature diagnostics and image-quality trajectories.** Panels (a) and (b) compare Ours with Vanilla using ImageNet linear-probe accuracy and CKNNA against DINOv2-g, respectively. Budgets of 16, 48, 112, and 240 tokens use cumulative global-to-regional prefixes for Ours and equal-length prefixes for Vanilla. Both analyses use third-decoder-block features from the 100K-step checkpoints, and dashed lines in (b) mark external encoders. Panel (c) traces rFID and gFID from 25K to 100K training steps, with darker markers indicating later checkpoints.

4.1 Spatial-Pyramid Hierarchical 1D Representation

For 256×256 image generation, the input is embedded as an $h \times w$ patch grid. We divide this grid into four nested levels with 1×1, 2×2, 4×4, and 8×8 cells. The 1×1 level contains the global tokens, while the remaining levels contain regional tokens assigned to progressively finer cells. At level s, the grid contains $C_s = 4^{s-1}$ non-overlapping cells, each covering a $h/2^{s-1} \times w/2^{s-1}$ region. Each cell c receives N_s learned latent inputs $\boldsymbol{L}_c^{(s)} = (\boldsymbol{\ell}_{c,1}^{(s)}, \ldots, \boldsymbol{\ell}_{c,N_s}^{(s)})$. Concatenating these inputs across cells forms $\boldsymbol{L}^{(s)}$, containing $C_s N_s$ tokens at level s. Thus, N_s counts tokens per cell, giving level totals of N_1, $4N_2$, $16N_3$, and $64N_4$, respectively. The partitions form a quadtree, so each regional cell has a unique ancestor at every coarser level.

The encoder jointly processes the patches and all grouped latent inputs in a single pass:

$$[\boldsymbol{p}'; \boldsymbol{f}^{(1)}; \boldsymbol{f}^{(2)}; \boldsymbol{f}^{(3)}; \boldsymbol{f}^{(4)}] = E_\phi\Big([\boldsymbol{p}; \boldsymbol{L}^{(1)}; \boldsymbol{L}^{(2)}; \boldsymbol{L}^{(3)}; \boldsymbol{L}^{(4)}]\Big). \tag{6}$$

As in Eq. (1), only latent-position outputs are retained. At level s, the retained features form $\boldsymbol{f}^{(s)}$, with $\boldsymbol{f}_{c,j}^{(s)}$ indexing cell $c = 1, \ldots, C_s$ and query $j = 1, \ldots, N_s$. Quantization produces $(z_{c,j}^{(s)}, \boldsymbol{f}_{c,j}^{q,(s)}) = Q(\boldsymbol{f}_{c,j}^{(s)})$, and the quantized features are collected as $\boldsymbol{f}^{q,(s)}$ for decoding. A spatial mask (Appendix E) restricts each latent token's direct patch access to its assigned cell.

The resulting code indices are ordered first by level, from global to local. Within each regional level, cells follow parent-major order, with raster ordering of parents and their children, followed by query index j. Let $\boldsymbol{z}^{(s)}$ denote the resulting sequence at level s, and let $\boldsymbol{z}^{(<s)} = [\boldsymbol{z}^{(1)}; \ldots; \boldsymbol{z}^{(s-1)}]$ denote its complete coarser prefix. The complete hierarchical representation is

$$\boldsymbol{z} = [\boldsymbol{z}^{(1)}; \boldsymbol{z}^{(2)}; \boldsymbol{z}^{(3)}; \boldsymbol{z}^{(4)}], \qquad |\boldsymbol{z}| = N_1 + 4N_2 + 16N_3 + 64N_4. \tag{7}$$

The decoder receives all four quantized feature groups together with the patch-position mask tokens $\boldsymbol{m}$. Each mask token directly attends only to latent features whose assigned cells contain its patch, including all global tokens. Mask tokens also interact bidirectionally during reconstruction. The decoder reconstructs the image as follows.

$$\widehat{\boldsymbol{x}} = D_\psi\Big([\boldsymbol{f}^{q,(1)}; \boldsymbol{f}^{q,(2)}; \boldsymbol{f}^{q,(3)}; \boldsymbol{f}^{q,(4)}; \boldsymbol{m}]\Big). \tag{8}$$

Joint decoding combines image-wide context with regional features at each output position.

4.2 Global-to-Local Autoregressive Modeling

We model the spatial-pyramid 1D sequence using token-wise prediction for global tokens and group-wise prediction for regional levels. A causal factorization models dependencies among global tokens

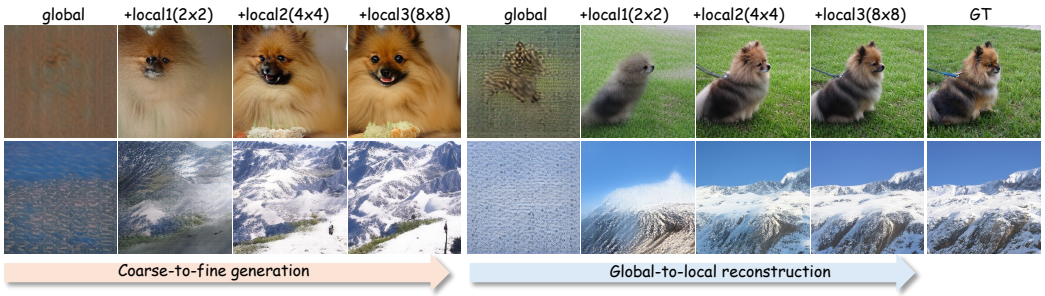


Figure 4: **Coarse-to-fine generation and global-to-local reconstruction.** Each image is decoded from the cumulative prefix indicated above its column. The local-group labels denote the three progressively finer regional levels of the spatial-pyramid representation. The left and right panels visualize generation from sampled prefixes and reconstruction from encoded prefixes, respectively.

as they establish the shared image-wide context for regional prediction. Each regional level is then predicted in parallel, conditioned on the global tokens and all coarser regional levels. For a regional level $s > 1$, let $z_i^{(s)}$ denote the i-th code in the fixed ordering of $\boldsymbol{z}^{(s)}$. The resulting likelihood is

$$p_\theta(\boldsymbol{z} \mid y) = \left[\prod_{j=1}^{N_1} p_\theta\left(z_j^{(1)} \mid \boldsymbol{z}_{<j}^{(1)}, y\right)\right] \prod_{s=2}^{4} p_\theta\left(\boldsymbol{z}^{(s)} \mid \boldsymbol{z}^{(<s)}, y\right), \tag{9}$$

Each regional likelihood is the product of its token-wise conditional probabilities given the same coarser prefix and class label. This conditional independence is a modeling choice, not a consequence of the spatial partition. The prediction queries may interact through self-attention, but cannot access current-level target codes.

Training preserves this conditioning structure through stage-wise teacher forcing. The global stage uses shifted encoded codes, whereas each regional stage combines the encoded coarser prefix with learned prediction queries.

$$\begin{aligned} [y, z_1^{(1)}, \ldots, z_{N_1-1}^{(1)}] &\longrightarrow [z_1^{(1)}, z_2^{(1)}, \ldots, z_{N_1}^{(1)}], s = 1, \\ [y, \boldsymbol{z}^{(<s)}, \boldsymbol{q}_{\mathrm{AR}}^{(s)}] &\longrightarrow \boldsymbol{z}^{(s)}, s \in \{2, 3, 4\}. \end{aligned} \tag{10}$$

In these inputs, code indices stand for their STE context embeddings, and y denotes the class-conditioned start embedding. Positional embeddings distinguish the $C_s N_s$ prediction slots within each regional level. Prefix tokens cannot attend to the current queries, which attend to the prefix and one another without receiving current-level targets. Each training iteration uses four AR stage evaluations that reuse the same encoder outputs, as summarized in Algorithm 1.

The hierarchical AR objective averages cross-entropy over all code positions, using each regional group's factorized likelihood:

$$\mathcal{L}_{\mathrm{AR}}^{\mathrm{hier}} = -\frac{1}{\sum_{s=1}^{4} C_s N_s} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\sum_{j=1}^{N_1} \log p_\theta\left(z_j^{(1)} \mid \boldsymbol{z}_{<j}^{(1)}, y\right) + \sum_{s=2}^{4} \log p_\theta\left(\boldsymbol{z}^{(s)} \mid \boldsymbol{z}^{(<s)}, y\right)\right]. \tag{11}$$

The hard indices remain detached targets, while the STE embeddings in Eq. (4) make the visible prefix differentiable. The AR loss therefore updates both the generator and the tokenizer through the context path.

4.3 Joint Training and Inference

For each training batch, a shared encoder–quantizer pass supplies quantized reconstruction features, detached AR targets, and differentiable STE context embeddings. Joint optimization allows the representation to adapt to the chosen global-to-local factorization through the differentiable context path. The objective retains the baseline reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ in Eq. (5) and replaces its token-wise AR loss with $\mathcal{L}_{\mathrm{AR}}^{\mathrm{hier}}$ in Eq. (11). Following representation-alignment methods (Xiong et al.,

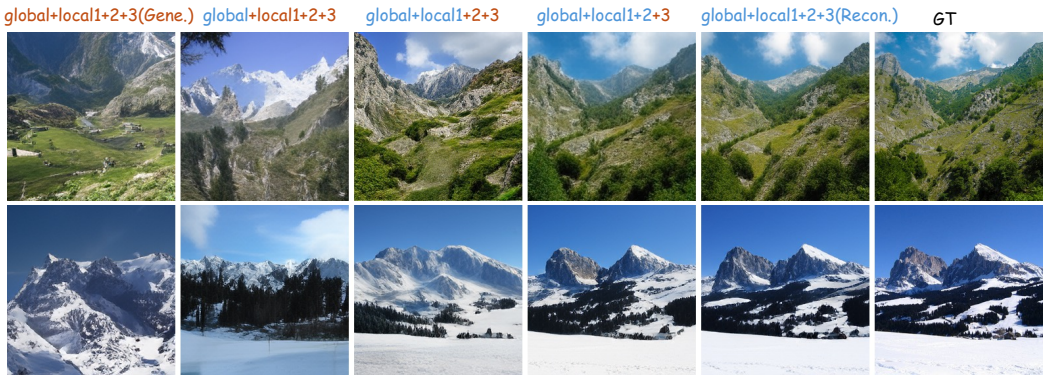


Figure 5: **Progressive replacement with reference-encoded token groups.** Blue and orange labels denote reference-encoded and AR-sampled token groups, respectively. Starting from an all-sampled sequence, we replace the global group and then successively finer regional levels with their reference-encoded counterparts. Unreplaced groups retain their originally sampled codes, and the last two columns show the all-encoded reconstruction and the reference image.

2025; Chu et al., 2026; Yao et al., 2025; Yu et al., 2025b; Wu et al., 2026; Zheng et al., 2025b), we further add decoder-side semantic supervision from frozen DINOv2 ViT-L/14 features (Oquab et al., 2023). The resulting hierarchical end-to-end objective is

$$\mathcal{L}_{\mathrm{E2E}}^{\mathrm{hier}} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{AR}}\mathcal{L}_{\mathrm{AR}}^{\mathrm{hier}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}}. \tag{12}$$

Token-level cross-entropy does not directly match the distribution of decoded images to that of real images. After joint training, we therefore apply distribution-alignment post-training, following recent approaches (Yang et al., 2026; Zhang et al., 2026a; Gao et al., 2026; Liu et al., 2026). This separate stage updates only the AR generator while keeping the tokenizer, including its decoder, frozen. Appendices F.1 and F.2 detail the decoder-side alignment loss and the separate post-training objective, respectively. Inference follows the same global-to-local conditioning order:

$$\underbrace{z_1^{(1)} \to \cdots \to z_{N_1}^{(1)}}_{N_1 \text{ global steps}} \to \underbrace{\boldsymbol{z}^{(2)} \to \boldsymbol{z}^{(3)} \to \boldsymbol{z}^{(4)}}_{3 \text{ local steps}}. \tag{13}$$

The generator first samples global tokens causally with a KV cache. It then samples each regional level in one model evaluation, conditioned on the generated global tokens and all coarser regional levels. After all groups have been sampled, their quantized features are passed to the decoder to reconstruct the image. For a matched token budget $N = N_1 + 4N_2 + 16N_3 + 64N_4$, this schedule requires $N_1 + 3$ sequential decoding steps, compared with N for the flat baseline.

5 Experiments

5.1 Experimental Setup

Implementation details. We evaluate class-conditional image generation on ImageNet at 256×256 resolution. Unless otherwise specified, the tokenizer uses a codebook of size K=16,384 and latent dimension d=64. The default per-cell allocation (N_1, N_2, N_3, N_4)=$(16, 8, 4, 2)$ yields N=240 tokens across the 1×1, 2×2, 4×4, and 8×8 levels. The Base (B) and Large (L) configurations share the same encoder architecture, with a larger decoder and AR generator in L. For the main experiments, we jointly train each configuration for 150 epochs with AdamW, a batch size of 256, and BF16 precision. We then apply one epoch of distribution-alignment post-training to the AR generator while keeping the tokenizer frozen (Appendix F.2). Unless otherwise specified, controlled analyses in Sections 5.2 and 5.4 use 100K end-to-end optimization steps with a batch size of 128. Architecture, optimization, and attention-mask details are provided in Appendices H and E. Appendix A describes the extension to ImageNet at 512×512 resolution.

Evaluation metrics. We assess reconstruction quality using reconstruction Fréchet Inception Distance (rFID) (Heusel et al., 2017), LPIPS (Zhang et al., 2018), PSNR, and SSIM. Generation quality

Table 2: **Class-conditional ImageNet 256×256 generation.** Methods are grouped by grid-based, sequence-based, and spatial-pyramid discrete tokenization. Epochs report tokenizer+generator training for two-stage systems and count joint end-to-end training once. Guidance denotes CFG, "Gen. steps" denotes the number of steps used to generate an image, and "–" indicates unreported values.

Method	Epochs	Tokenizer			Generator		w/o guidance		w/ guidance	
		#Params	#Tokens	rFID↓	#Params	Gen. steps	gFID↓	IS↑	gFID↓	IS↑
Grid-based 2D Discrete Tokenization										
VQGAN	–	23M	16×16	4.98	1.4B	256	15.78	74.3	–	–
RQTrans	–	66M	8×8×4	3.20	1.4B	68	8.71	119.0	3.89	337.5
MaskGIT	–+500	66M	16×16	2.28	227M	8	6.18	182.1	–	–
MAGVIT-v2	270+1080	133M	16×16	1.17	307M	64	3.65	200.5	1.78	319.4
LlamaGen-3B	–+300	72M	16×16	0.94	3B	256	9.38	112.9	2.18	263.3
RAR-XXL	–+400	66M	16×16	2.28	1.5B	256	3.26	193.3	1.48	326.0
xAR-H	–+800	–	16×16	–	1.1B	50	–	–	1.24	301.6
VAR-d30	–+350	109M	680	0.90	2B	10	–	–	1.92	323.1
AliTok-L	–+800	–	16+16×16	0.86	318M	272	1.98	200.8	1.38	326.2
Prologue L–XL	200+950	86+305M	16+16×16	0.99	685M	272	2.26	–	1.46	257.7
GEAR-XL	–+800	72M	16×16	1.62	775M	256	6.28	136.7	2.45	254.3
Bitdance-H-1x	–+800	460M	16×16	–	1.0B	256	–	–	1.24	304.4
Sequence-based 1D Discrete Tokenization										
TiTok-S-128	200+300	22+22M	128	1.71	287M	64	4.44	168.2	1.97	281.8
FlexTok d18-28	–	286M+1.1B	1–256	1.45	1.33B	1–256	–	–	1.86	–
ReTok-S-B	–	232M	1–256	1.01	775M	256	–	–	2.27	245.9
Semanticist	–	–	256	0.72	343M	32	–	–	2.57	254.0
GigaTok-XL-XXL	–+300	2.9B	256	0.79	1.4B	256	–	–	1.98	256.8
VFMTok-3B	–+200	–	256	0.89	3.1B	256	2.04	280.4	2.07	267.6
ResTok	–	662M	128	1.28	326M	9	–	–	2.34	257.8
Hita-L	–+300	–	569	1.03	343M	–	4.56	–	2.86	267.3
EOSTok-H	400	388M	256	0.71	644M	256	1.48	239.5	1.38	265.7
SpectralAR-d24	–	–	64	4.03	1B	64	–	–	2.13	307.7
Spatial-pyrami 1D Discrete Tokenization										
Ours B	150	88+88M	240	1.58	114M	16+3	2.78	178.2	1.59	254.4
Ours L	150	88+308M	240	1.34	691M	16+3	1.96	232.9	**1.21**	283.1

is evaluated using generation FID (gFID) and Inception Score (IS) (Salimans et al., 2016), both computed over 50,000 generated samples. ImageNet linear-probe accuracy measures decoder-feature separability, while CKNNA (Huh et al., 2024) measures neighborhood agreement with DINOv2-g (Appendix G). Detailed protocols for these metrics are provided in Appendix I.

5.2 Hierarchical Representation Analysis

Table 1 compares our spatial-pyramid model with the flat 1D baseline (*Vanilla*) under matched end-to-end training. Our model lowers rFID from 4.82 to 2.63 and gFID from 38.06 to 27.10. At batch size 16, sampling and decoding throughput increases from 4.43 to 49.57 samples/s, a measured 11× gain. These gains support the joint design of the representation and AR factorization.

Table 1: **Flat 1D (Vanilla) and spatial-pyramid (Ours) tokenization under matched end-to-end training.** Throughput is reported in samples per second for sampling and decoding at batch size 16.

Method	rFID↓	LPIPS↓	PSNR↑	SSIM↑	gFID↓	Thro.↑
Vanilla	4.82	0.35	18.14	0.37	38.06	4.43
Ours	**2.63**	**0.24**	**20.31**	**0.58**	**27.10**	**49.57**

Figure 3(a,b) complements this comparison with decoder-feature diagnostics at matched budgets of 16, 48, 112, and 240 tokens. Cumulative hierarchical prefixes yield higher linear-probe accuracy and CKNNA than equal-length Vanilla prefixes. Panel (c) also shows lower rFID and gFID at each evaluated checkpoint. Progressive decoding shows how regional levels refine the global representation (Figure 4). Extending sampled prefixes adds structure and detail, while extending reference-encoded prefixes brings reconstructions closer to the original image. To examine each group's role, we progressively replace sampled groups with their reference-encoded counterparts, from global to finer regional levels (Figure 5). Replacing global tokens changes composition, while subsequent regional

Table 3: **Global-token generation and auxiliary objectives.** Parallel, Masked, and Causal use one, eight, and 16 global-token prediction steps, respectively. The final row successively adds decoder-side DINOv2 alignment and distribution-alignment post-training to causal generation. Post-training updates only the AR generator, leaving reconstruction metrics unchanged.

Setting	rFID↓	LPIPS↓	PSNR↑	SSIM↑	Global steps	gFID↓
Parallel	2.86	0.25	20.26	0.57	1	30.32
Masked	2.58	0.24	20.43	0.58	8	28.55
Causal	2.63	0.24	20.31	0.58	16	27.10
+ Decoder + Distribution alignment	1.98	0.26	19.71	0.56	16	11.44→4.60

replacements restore reference-specific structure and texture. These observations support complementary global and regional roles, but do not imply complete information separation or strictly local effective receptive fields.

5.3 Generation Results on ImageNet

Table 2 compares grid-based, sequence-based, and spatial-pyramid 1D models on ImageNet 256×256. Both Ours-B and Ours-L use 240 tokens and require 19 sequential decoding steps. This compares with 64 steps for SpectralAR-d24 and 256 for GigaTok, VFMTok, and EOSTok-H. Figure 6 and Appendix B complement this comparison by showing generation throughput and gFID across the reported models. Ours-B achieves an rFID of 1.58 and an unguided gFID of 2.78 with a 114M generator. With CFG (Ho & Salimans, 2022), its gFID improves to 1.59, compared with 1.48 for RAR-XXL's 1.5B generator. Ours-L further reduces rFID to 1.34 and guided gFID to 1.21, the lowest guided gFID among the listed models. Together with the matched comparison in Table 1, these results are consistent with the view that better-structured causal dependencies can ease autoregressive learning.

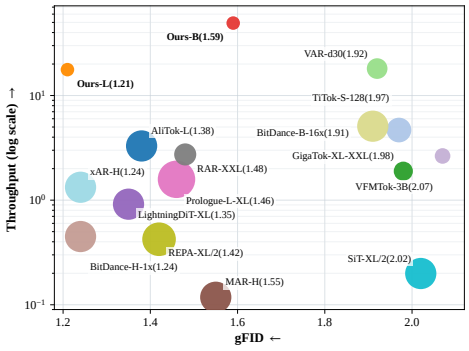


Figure 6: **Generation throughput (samples/s, log scale) vs. gFID.** Bubble area represents the minimum known training epochs.

5.4 Ablation Studies

Within the spatial-pyramid model, Table 3 compares one-shot parallel, eight-step MaskGIT-style masked (Chang et al., 2022), and causal global-token generation. The Causal row corresponds to Ours in Table 1, before auxiliary alignment. gFID decreases from 30.32 with parallel prediction to 28.55 with masked prediction and 27.10 with causal generation. Masked prediction yields the lowest rFID (2.58), while causal generation gives the lowest gFID at the cost of 16 rather than eight global steps. Adding decoder-side DINOv2 alignment to causal generation reduces rFID from 2.63 to 1.98 and gFID from 27.10 to 11.44. These improvements come with higher LPIPS (0.24 to 0.26) and lower PSNR (20.31 to 19.71) and SSIM (0.58 to 0.56). Distribution-alignment post-training further lowers gFID from 11.44 to 4.60 while leaving reconstruction metrics unchanged, because it updates only the AR generator. Further ablations examine the effects of latent-token count and codebook size (Appendix C) and encoder attention masks (Appendix D).

6 Conclusion

We introduced a ***spatial-pyramid 1D visual representation*** that jointly organizes global and regional tokens across multiple spatial granularities within a single 1D sequence. The matching AR model generates global tokens causally and each regional level in parallel, conditioned on all coarser groups. On ImageNet at 256 × 256 resolution, our method achieves a gFID of 1.21 with classifier-free guidance using only 19 sequential decoding steps. Ablation studies further demonstrate the effectiveness of the proposed representation and factorization, while token-group interventions reveal the complementary functions of global and regional tokens.

Reproducibility Statement

We describe the model and experiments in Sections 4 and 5. Additional implementation details and evaluation protocols are provided in Appendices H and I. We plan to release our implementation and trained checkpoints upon publication.

AI Use Statement

We used large language models to assist with manuscript drafting and editing, the presentation of experimental results, and the preparation of LaTeX figures and tables. The authors reviewed the AI-assisted material and take responsibility for the final content of the paper.

References

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

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

Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pp. 671–679. Elsevier, 1987.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

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*, pp. 11315–11325, 2022.

Cong Chen, Ziyuan Huang, Cheng Zou, Muzhi Zhu, Kaixiang Ji, Jiajia Liu, Jingdong Chen, Hao Chen, and Chunhua Shen. Hieratok: Multi-scale visual tokenizer improves image reconstruction and generation. *arXiv preprint arXiv:2509.23736*, 2025.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9640–9649, October 2021.

Haozhe Chi, Jinghan Li, Hao Jiang, Wu Sheng, Yi Ma, Jing Wang, and Yadong Mu. Balancing image compression and generation with bootstrapped tokenization. *arXiv preprint arXiv:2606.05552*, 2026.

Wenda Chu, Bingliang Zhang, Jiaqi Han, Yizhuo Li, Linjie Yang, Yisong Yue, and Qiushan Guo. End-to-end autoregressive image generation with 1d semantic tokenizer. *arXiv preprint arXiv:2605.00503*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Shivam Duggal, Xingjian Bai, Zongze Wu, Richard Zhang, Eli Shechtman, Antonio Torralba, Phillip Isola, and William T Freeman. End-to-end training for unified tokenization and latent denoising. *arXiv preprint arXiv:2603.22283*, 2026.

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

Zixuan Fu, Lanqing Guo, Chong Wang, Binbin Song, Ding Liu, and Bihan Wen. Improving flexible image tokenizers for autoregressive image generation. *arXiv preprint arXiv:2601.01535*, 2026.

Mingju Gao, Jingkai Zhou, Kun Gai, Changqian Yu, and Hao Tang. Advfd: Boosting visual generation via adversarial fréchet distance loss. *arXiv preprint arXiv:2608.11205*, 2026.

Jian Han, Jinlai Liu, Jiahuan Wang, Bingyue Peng, and Zehuan Yuan. Generative refinement networks for visual synthesis. *arXiv preprint arXiv:2604.13030*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

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

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

Yuanhui Huang, Weiliang Chen, Wenzhao Zheng, Yueqi Duan, Jie Zhou, and Jiwen Lu. Spectralar: Spectral autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15842–15852, 2025a.

Zhihao Huang, Xi Qiu, Yukuo Ma, Yifu Zhou, Chi Zhang, and Xuelong Li. Nfig: Autoregressive image generation with next-frequency prediction. *arXiv preprint arXiv:2503.07076*, 2025b.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International conference on learning representations*, 2017.

Shiyin Jiang, Wei Long, Minghao Han, Zhenghao Chen, Ce Zhu, and Shuhang Gu. Differentiable vector quantization for rate-distortion optimization of generative image compression. *arXiv preprint arXiv:2604.10546*, 2026.

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*, pp. 11523–11532, 2022.

Haopeng Li, Jinyue Yang, Guoqi Li, and Huan Wang. Autoregressive image generation with randomized parallel decoding. *arXiv preprint arXiv:2503.10568*, 2025.

Qifan Li, Jiale Zou, Jinhua Zhang, Wei Long, Xingyu Zhou, and Shuhang Gu. Texture vector-quantization and reconstruction aware prediction for generative super-resolution. In *International Conference on Learning Representations*, volume 2026, pp. 63694–63714, 2026.

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

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

Bin Lin, Zongjian Li, Yuwei Niu, Kaixiong Gong, Yunyang Ge, Yunlong Lin, Mingzhe Zheng, JianWei Zhang, Miles Yang, Zhao Zhong, et al. ifsq: Improving fsq for image generation with 1 line of code. *arXiv preprint arXiv:2601.17124*, 2026a.

Bin Lin, Zheyuan Liu, Chenguo Lin, Sixiang Chen, Yunyang Ge, Yunlong Lin, Jianwei Zhang, Miles Yang, Zhao Zhong, Liefeng Bo, et al. Gear: Guided end-to-end autoregression for image synthesis. *arXiv preprint arXiv:2606.32039*, 2026b.

Wenze Liu, Xintao Wang, Pengfei Wan, and Xiangyu Yue. Amortized moment matching for visual generation. *arXiv preprint arXiv:2607.26860*, 2026.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression. *arXiv preprint arXiv:2501.10064*, 2025.

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

Yatian Pang, Peng Jin, Shuo Yang, Bin Zhu, Bin Lin, Chaoran Feng, Zhenyu Tang, Liuhan Chen, Francis EH Tay, Ser-Nam Lim, et al. Next patch prediction for autoregressive visual generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8260–8268, 2026.

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

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

Yunpeng Qu, Kaidong Zhang, Yukang Ding, Ying Chen, and Jian Wang. Semantic one-dimensional tokenizer for image reconstruction and generation. *arXiv preprint arXiv:2603.16373*, 2026.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. *arXiv preprint arXiv:2412.15205*, 2024.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15781–15791, 2025.

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

Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16037–16046, 2025.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Yikai Wang, Zhouxia Wang, Zhonghua Wu, Qingyi Tao, Kang Liao, and Chen Change Loy. Next visual granularity generation. *arXiv preprint arXiv:2508.12811*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024.

Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "principal components" enable a new language of images. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16641–16651. IEEE, 2025.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Ming-Ming Cheng, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *Advances in Neural Information Processing Systems*, 38:7714–7743, 2026.

Pingyu Wu, Kai Zhu, Yu Liu, Longxiang Tang, Jian Yang, Yansong Peng, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Towards sequence modeling alignment between tokenizer and autoregressive model. *arXiv preprint arXiv:2506.05289*, 2025.

Tianwei Xiong, Jun Hao Liew, Zilong Huang, Jiashi Feng, and Xihui Liu. Gigatok: Scaling visual tokenizers to 3 billion parameters for autoregressive image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18770–18780, 2025.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fréchet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Weiyi You, Jinhua Zhang, Xingyu Zhou, Wei Long, Junyu Lou, and Shuhang Gu. Advanced pixel diffusion model with guided sparse global refinement. *arXiv preprint arXiv:2609.00798*, 2026.

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

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10459–10469, 2023a.

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023b.

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

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

Qihang Yu, Qihao Liu, Ju He, Xinyang Zhang, Yang Liu, Liang-Chieh Chen, and Xi Chen. Autoregressive image generation with masked bit modeling. *arXiv preprint arXiv:2602.09024*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025b.

Jinhua Zhang, Yisong Lin, Wei Long, and Shuhang Gu. Beyond token-level cross-entropy: Fréchet distributional post-training for autoregressive image generation. *arXiv preprint arXiv:2608.00562*, 2026a.

Jinhua Zhang, Wei Long, Minghao Han, Weiyi You, and Shuhang Gu. Mvar: Visual autoregressive modeling with scale and spatial markovian conditioning. In *International Conference on Learning Representations*, volume 2026, pp. 32889–32917, 2026b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Xu Zhang, Cheng Da, Huan Yang, Kun Gai, Ming Lu, and Zhan Ma. Restok: Learning hierarchical residuals in 1d visual tokenizers for autoregressive image generation. *arXiv preprint arXiv:2601.03955*, 2026c.

Anlin Zheng, Haochen Wang, Yucheng Zhao, Weipeng Deng, Tiancai Wang, Xiangyu Zhang, and Xiaojuan Qi. Hita: Holistic tokenizer for autoregressive image generation. *arXiv preprint arXiv:2507.02358*, 2025a.

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

Bowen Zheng, Weijian Luo, Guang Yang, Colin Zhang, and Tianyang Hu. Autoregressive visual generation needs a prologue. *arXiv preprint arXiv:2605.06137*, 2026a.

Guangting Zheng, Yiyuan Zhang, Tao Yang, Yunpeng Chen, Rui Zhu, Jiajun Deng, and Yanyong Zhang. Genfirst: Generation before reconstruction for stable end-to-end latent generative modeling. *arXiv preprint arXiv:2608.29335*, 2026b.

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. *arXiv preprint arXiv:2512.24176*, 2025.

Table 4: **Class-conditional ImageNet 512×512 generation.** "Gen. steps" denotes sequential sampling rounds, including denoising iterations for diffusion models.

Method	Epochs	Tokenizer			Generator		w/o guidance		w/ guidance	
		#Params	#Tokens	rFID↓	#Params	Gen. steps	gFID↓	IS↑	gFID↓	IS↑
DiT-XL/2	–+600	–	–	0.19	675M	250	12.03	105.3	3.04	240.8
MaskGIT	–+800	–	32×32	1.97	177M	12	7.32	156.0	–	–
MAGVIT-v2	–	–	–	1.22	307M	64	3.07	213.1	1.91	324.3
TiTok-B-128	–	–	128	1.37	177M	64	4.17	181.0	2.13	261.2
VAR-d36-s	–+350	109M	2240	–	2.3B	10	–	–	2.63	303.2
Ours-B	150	88+88M	496	2.46	98M	16+4	3.10	165.9	1.71	258.2



Figure 7: **Qualitative evaluation on ImageNet 512×512.**

A Additional Results on ImageNet 512×512

For class-conditional ImageNet generation at 512×512, we extend the 256×256 spatial-pyramid hierarchy with an additional 16×16 regional level. The per-cell allocation $(N_1, N_2, N_3, N_4, N_5) = (16, 8, 4, 2, 1)$ yields 496 latent tokens across the resulting five-level hierarchy. Generation requires 20 sequential decoding steps, with 16 causal global-token predictions followed by four parallel regional stages. Table 8 summarizes the architecture, optimization, and sampling settings. Decoder-side DINOv2 alignment and distribution-alignment post-training follow Appendices F.1 and F.2, respectively. We also use Internal Guidance (Zhou et al., 2025) as an auxiliary sampling strategy. Table 4 reports the quantitative results, while Figure 7 shows qualitative samples.

B Sampling Speed Comparison

Figure 6 visualizes the gFID–throughput trade-off on ImageNet 256×256. Table 5 extends this comparison with model size, sequential decoding steps, and peak allocated CUDA memory for diffusion (Peebles & Xie, 2023; Ma et al., 2024; Yao et al., 2025) and autoregressive models. We benchmark each model in FP32 on a single NVIDIA RTX 4090 with batch size 16, using its reported

Table 5: **Sampling throughput and peak allocated CUDA memory on ImageNet-256×256.** Throughput (samples/s) and peak memory (GB) are measured on one NVIDIA RTX 4090 with batch size 16 using FP32 precision.

Method	Params	gFID↓	Steps	Throughput↑	Memory (GB)
Diffusion					
REPA-XL/2	675M	1.42	250	0.42	6.65
SiT-XL/2	675M	2.02	250	0.20	6.62
LightningDiT-XL	675M	1.35	250	0.92	4.92
Autoregressive					
AliTok-L	318M	1.38	272	3.30	4.42
GigaTok-XL-XXL	1.4B	1.98	256	1.89	19.32
MAR-H	943M	1.55	256	0.12	5.61
Prologue-L-XL	685M	1.46	272	1.58	6.44
VAR-d30	2B	1.92	10	18.10	15.46
VFMTok-3B	3.1B	2.07	256	2.65	11.96
xAR-H	1.1B	1.24	50	1.33	8.69
RAR-XXL	1.5B	1.48	256	2.74	10.04
TiTok-S-128	287M	1.97	64	4.68	4.46
BitDance-B-16x	260M	1.91	16	5.09	6.62
BitDance-H-1x	1.0B	1.24	256	0.45	9.31
Ours-B	114M	1.59	16+3	49.27	2.02
Ours-L	691M	1.21	16+3	17.68	6.44

sampling configuration. After one warm-up batch, we record throughput and peak allocated CUDA memory while generating 50,000 class-conditional samples with balanced validation-set class labels.

With $N_1 = 16$, our $N_1 + 3$ schedule generates all 240 latent tokens in 19 sequential decoding steps. Ours-B reaches 49.27 samples/s with 2.02 GB of peak allocated CUDA memory, while Ours-L reaches 17.68 samples/s with 6.44 GB. Both models exceed the listed flat token-wise AR baselines in throughput, and Ours-B has the lowest peak allocated memory among all compared systems.

C Ablation Studies on Latent and Codebook Capacity

Table 6 examines how hierarchical token allocation and codebook size affect reconstruction, generation, and decoding steps. With 32 global tokens, expanding from two levels (64 tokens) to four (256 tokens) lowers rFID from 3.53 to 1.91 but raises gFID from 8.52 to 10.67. At 128 tokens, allocations $(16, 4, 2, 1)$ and $(32, 8, 4, 0)$ give similar reconstruction quality. The four-level allocation achieves lower gFID (8.88 vs. 9.91) with 19 rather than 34 sequential decoding steps. Although increasing the budget from 240 to 256 tokens improves all five quality metrics, we retain 240 tokens for 19 rather than 35 sequential steps. At 176 tokens, $K = 16{,}384$ yields the best reported metrics, with gFID varying non-monotonically across codebook sizes.

D Ablation Studies on Encoder Attention Mask

We compare full and stage-causal encoder latent-to-latent attention masks while keeping spatial patch access fixed (Table 7). Relative to the stage-causal (Figure 8 (a)), full visibility lowers rFID from 2.63 to 2.57 and raises PSNR from 20.31 to 20.37. We retain the stage-causal mask for its lower gFID (27.10 vs. 28.61) and consistency with the AR model's global-to-local prediction order.

E Attention Mask Design

Encoder patch tokens attend to all patches and latent tokens (Figure 8(a)). Latents directly access patches only within their assigned cells, which cover the full image for global tokens. Latent-to-latent attention is causal for global tokens and restricted to the same or coarser levels for regional

Table 6: **Effects of hierarchical token allocation and codebook size.** N_s gives the tokens per cell at level s, with zero indicating an omitted level. Allocations vary at $K = 16{,}384$, and codebook sizes vary at allocation $(16, 8, 4, 1)$, with other settings fixed. "Gen. steps" counts N_1 global steps plus one parallel step per active regional level.

(N_1, N_2, N_3, N_4)	$\lvert z \rvert$	Gen. steps	K	rFID↓	LPIPS↓	PSNR↑	SSIM↑	gFID↓
$(4, 2, 1, 1)$	92	4+3	16,384	2.97	0.32	18.16	0.47	9.30
$(8, 4, 2, 1)$	120	8+3		2.84	0.30	18.63	0.49	9.74
$(16, 4, 2, 1)$	128	16+3		2.56	0.30	18.71	0.50	8.88
$(16, 8, 4, 1)$	176	16+3	1,024	2.85	0.31	18.59	0.50	10.86
			2,048	2.54	0.30	18.80	0.51	10.42
			4,096	2.42	0.29	18.81	0.51	10.96
			16,384	2.26	0.28	19.16	0.53	10.08
$(16, 8, 4, 2)$	240	16+3	16,384	1.98	0.26	19.71	0.56	11.44
$(32, 8, 4, 2)$	256	32+3		1.91	0.25	20.00	0.57	10.67
$(32, 8, 4, 0)$	128	32+2		2.57	0.30	18.73	0.50	9.91
$(32, 8, 0, 0)$	64	32+1		3.53	0.34	17.55	0.44	8.52

Table 7: **Effect of encoder latent-to-latent attention under fixed spatial patch access.**

Pattern	rFID↓	LPIPS↓	PSNR↑	SSIM↑	gFID↓
Full	2.57	0.24	20.37	0.58	28.61
Stage-causal	2.63	0.24	20.31	0.58	27.10

tokens. In the decoder (Figure 8(b)), patch queries interact bidirectionally and directly access only latents whose cells contain their positions. Decoder latent tokens retain full access to all latent tokens and patch queries. AR training evaluates each stage separately, using shifted inputs for global tokens and parallel prediction queries for regional tokens. Regional queries attend to the class-conditioned prefix of all coarser token embeddings and to one another, while prefix positions cannot attend to these queries. Current-level target embeddings are excluded, and only query outputs contribute to that level's predictions. Prefixes use encoded ground-truth tokens during training and previously sampled tokens during inference. Figure 8(c) illustrates a regional stage, where `G` and `L1` denote prefix tokens and `L2 queries` denotes prediction positions. Thus, the global block depicts prefix context rather than shifted global-training inputs. Algorithm 1 specifies the four stage evaluations, mask construction, and token-averaged AR loss.

F Details of Training Loss

F.1 Decoder-Side Representation Alignment

Following prior methods (Xiong et al., 2025; Chu et al., 2026; Yao et al., 2025; You et al., 2026), we supplement reconstruction with decoder-side supervision from a frozen DINOv2 ViT-L/14 encoder (Oquab et al., 2023). Let $h_s(x)$ denote the decoder hidden state at patch position s for image x, computed from quantized encoder features. DINOv2 receives the same image after ImageNet normalization and resizing to 224×224, producing a 16×16 grid of 1,024-dimensional final patch features. A learned linear projection P matches decoder features to the teacher's channel dimension. With 256×256 images and decoder patches of size 16, the spatial grids match directly. Otherwise, the projected decoder feature map is resized by bilinear interpolation to match the teacher grid.

Writing $u_{b,s} = P(h_s(x_b))$ and $v_{b,s}$ for projected decoder and frozen teacher features, respectively, we minimize their spatially averaged cosine distance:

$$\mathcal{L}_{\text{align}} = \frac{1}{BP} \sum_{b=1}^{B} \sum_{p=1}^{P} \left(1 - \frac{u_{b,p}^{\top} v_{b,p}}{\|u_{b,p}\|_2 \|v_{b,p}\|_2} \right), \tag{14}$$

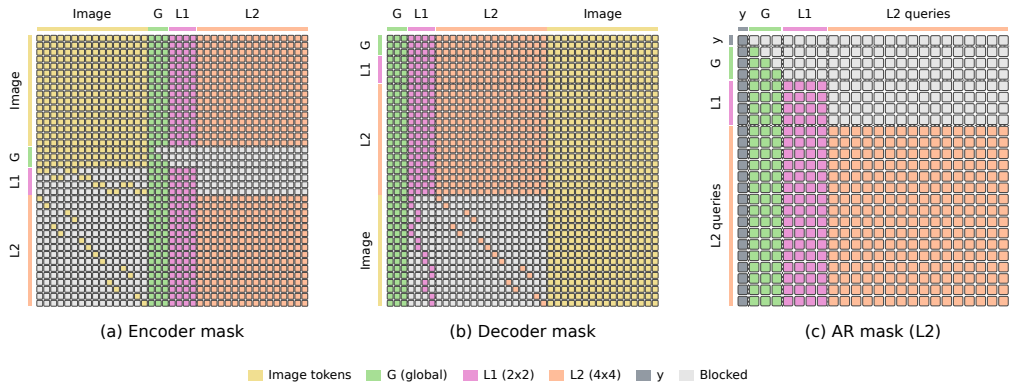


Figure 8: **Encoder (a), decoder (b), and L_2-stage AR (c) attention masks.** The schematic uses 4×4 patches and $[3, 4, 16]$ tokens at grid sizes $[1, 2, 4]$. Colored and gray cells indicate allowed and blocked attention, respectively. In (c), y is the class-conditioned start embedding, `G`/`L1` are prefix tokens, and `L2 queries` are parallel prediction queries.

Algorithm 1: Stage-wise Teacher Forcing for Global-to-Local AR Training at 256×256

Inputs : Detached targets $\{\boldsymbol{z}^{(s)}\}_{s=1}^{S}$, differentiable STE embeddings $\{\boldsymbol{e}^{(s)}\}_{s=1}^{S}$ (Eq. 4), and class-conditioned start embedding y for the same batch of B images.

Settings: $S = 4$ levels, with $C_s = 4^{s-1}$ cells and N_s tokens per cell, giving $T_s = C_s N_s$ tokens per level. All stages share the AR model F_θ and learnable query vector $\boldsymbol{q}$. The coarser prefix is $\boldsymbol{e}^{(<s)} = [\boldsymbol{e}^{(1)}; \ldots; \boldsymbol{e}^{(s-1)}]$.

$\ell \leftarrow 0$;
for $s \leftarrow 1$ **to** S **do**
  if $s = 1$ **then**
    $\boldsymbol{u} \leftarrow [y, \boldsymbol{e}_1^{(1)}, \ldots, \boldsymbol{e}_{T_1-1}^{(1)}]$;
    $\boldsymbol{M} \leftarrow \mathrm{CausalMask}(T_1)$; *// Current and earlier positions.*
  else
    $\boldsymbol{q}_{\mathrm{AR}}^{(s)} \leftarrow \mathrm{Repeat}(\boldsymbol{q}, T_s)$;
    $\boldsymbol{u} \leftarrow [y, \boldsymbol{e}^{(<s)}, \boldsymbol{q}_{\mathrm{AR}}^{(s)}]$; *// No current-level target embeddings.*
    // Prefix: y sees itself; global tokens see y and causal global context.
    // Regional prefix tokens see y, their level, and all coarser levels.
    // Queries see the prefix and one another; the prefix cannot see queries.
    $\boldsymbol{M} \leftarrow \mathrm{StageMask}(T_1, \ldots, T_s)$;
  $\boldsymbol{a} \leftarrow F_\theta(\boldsymbol{u}; \boldsymbol{M})$; *// Code logits with positional encoding.*
  $\boldsymbol{a}^{(s)} \leftarrow \mathrm{Tail}_{T_s}(\boldsymbol{a})$; *// Select the last T_s prediction positions.*
  // Accumulate per-token cross-entropy over this batch and level.
  $\ell \leftarrow \ell + \sum_{b=1}^{B} \sum_{i=1}^{T_s} \mathrm{CE}(\boldsymbol{a}_{b,i}^{(s)}, z_{b,i}^{(s)})$;
return $\mathcal{L}_{\mathrm{AR}}^{\mathrm{hier}} = \ell / \big(B \sum_{s=1}^{S} T_s\big)$;

where B is the batch size and P is the number of spatial positions. Teacher features are detached, while gradients update the projection and decoder and reach the encoder through the quantizer's straight-through estimator. During joint training, the weight in Eq. (12) follows $\lambda_{\mathrm{align}}(t) = \min(t/20{,}000, 1)$, where t is the training step. Neither the teacher nor the projection is needed for reconstruction or generation at inference.

F.2 Distribution-Alignment Post-Training

After joint training, Fréchet distribution-alignment post-training (Yang et al., 2026; Zhang et al., 2026a) updates only the AR model to match decoded and real-image distributions. The tokenizer and representation network remain frozen throughout this stage. At each iteration, uniformly sampled

class labels condition hierarchical token sampling. The detached sequence is replayed through the AR model to recompute logits with the corresponding classifier-free guidance.

Let s_i denote the sampled token at position i, e_k a frozen projected codebook embedding, and $\ell_{i,k}$ the replay logit for code k. We pass image-level gradients through discrete tokens using the straight-through assignment

$$p_{i,k} = \mathrm{softmax}_k(\ell_i/\tau), \qquad \bar{e}_i = \sum_{k=1}^{K} p_{i,k} e_k, \qquad \widetilde{e}_i = \bar{e}_i + \mathrm{sg}(e_{s_i} - \bar{e}_i). \tag{15}$$

Here, sg denotes stop-gradient, preserving the sampled hard embedding in the forward pass. Backward gradients use full-codebook soft assignments with $\tau = 1$, ignoring any top-k or top-p truncation used during hard sampling. Sampled contexts remain fixed, so gradients do not traverse the full sampling trajectory.

The frozen decoder maps $\widetilde{e}$ to images, which undergo differentiable preprocessing before a frozen Inception-v3 (Szegedy et al., 2016) extracts 2,048-dimensional pooled features. We initialize running first and second moments from 50,000 class-balanced generated images and update them across batches using exponential moving averages:

$$\mu_t = \beta\ \mathrm{sg}(\mu_{t-1}) + (1-\beta)\frac{1}{B}\sum_{b=1}^{B} f_b, M_t = \beta\ \mathrm{sg}(M_{t-1}) + (1-\beta)\frac{1}{B}\sum_{b=1}^{B} f_b f_b^\top, \Sigma_t = M_t - \mu_t \mu_t^\top. \tag{16}$$

Here, f_b is a generated-image feature and B the global batch size. Gradients pass only through current-batch features, with historical moments detached. The Fréchet distance and normalized optimization objective are then given by

$$d_t = \|\mu_t - \mu_r\|_2^2 + \mathrm{tr}\left[\Sigma_t + \Sigma_r - 2\left(\Sigma_r^{1/2}\Sigma_t\Sigma_r^{1/2}\right)^{1/2}\right], \mathcal{L}_{\mathrm{dist}} \quad = \frac{d_t}{\mathrm{sg}(d_t) + \epsilon}, \epsilon = 0.01. \tag{17}$$

The detached denominator rescales Fréchet gradients, which reach the replay logits through the frozen feature extractor and decoder. Table 8 summarizes the post-training hyperparameters and inference configuration for each model.

G Representation Analysis

We characterize the representations in Figure 3(a,b) through ImageNet linear separability and alignment with DINOv2-g (ViT-g/14) (Oquab et al., 2023). Here, *Vanilla* denotes the flat 1D baseline in Section 3, whereas *Ours* uses the spatial-pyramid representation. At 100K training steps, both models are evaluated with decoder inputs of 16, 48, 112, and 240 tokens. These inputs comprise cumulative groups for *Ours* and equal-length prefixes for *Vanilla*.

Feature extraction. For each frozen model and token budget, we spatially average image patch-query hidden states from the third decoder block, before alignment projection. Images are deterministically resized and center-cropped to 256×256, and features are extracted under null-class conditioning to exclude ground-truth labels. This diagnostic setting introduces a conditioning shift relative to class-conditioned training. For the external reference, we use frozen DINOv2-g features obtained by averaging its final normalized patch tokens, excluding the class token. DINOv2-g receives the same image crops, further resized to 224×224 using bicubic interpolation.

Linear probing. Following REPA (Yu et al., 2025b), we train a linear classifier for each model and token budget, preceded by batch normalization without learnable affine parameters. Classifiers are trained on the ImageNet training split using cross-entropy loss and Adam for 90 epochs with a batch size of 16,384. The learning rate follows cosine decay from 10^{-3}, with no warmup or weight decay. Figure 3(a) reports final-epoch validation top-1 accuracy, with higher values indicating greater linear separability. The dashed DINOv2-g reference is evaluated using the same probing protocol.

Alignment with DINOv2-g. Following REPA, we quantify local neighborhood agreement with DINOv2-g using CKNNA (Huh et al., 2024), with higher values indicating stronger alignment. All comparisons use the same fixed random subset of 10,000 ImageNet validation images. Inner-product

kernels on ℓ_2-normalized features define each image's $k = 10$ nearest neighbors, excluding self-matches. The cross-representation term uses shared neighbor pairs, while normalization uses each space's own neighborhood graph. We compute normalized alignment using the Platonic Representation implementation of the unbiased Hilbert–Schmidt independence criterion estimator. Dashed references in Figure 3(b) use frozen MoCo v3-B (Chen et al., 2021), DINO v1-B (Caron et al., 2021), and MAE-L (He et al., 2022) features evaluated under the same protocol.

H Hyperparameter and More Implementation Details

The main experiments use eight NVIDIA RTX PRO 6000 GPUs, while ablations use 4090-series GPUs. At a global batch size of 256 on these eight GPUs, Ours-B and Ours-L average 0.719 and 1.05 seconds per training iteration at 256×256, respectively. Under the same setup, the 512×512 Base configuration averages 1.15 seconds per iteration. Table 8 summarizes the model architectures, training hyperparameters, and sampling settings for each model variant.

I Evaluation Details

We evaluate image generation using gFID and IS, and assess image reconstruction using rFID, LPIPS, PSNR, and SSIM. We compute FID and IS using `torch-fidelity`. For generation, we sample 50,000 images with balanced class conditioning, yielding 50 images per ImageNet class, and use the reference statistics provided by JiT (Li & He, 2026) for gFID computation. For reconstruction, validation images are resized and center-cropped to 256×256 or 512×512 pixels before encoding and decoding.

- **gFID** (Heusel et al., 2017) measures the discrepancy between the distributions of real and generated images in the Inception-v3 feature space. It models both feature distributions as multivariate Gaussians and computes their Fréchet distance. Lower gFID indicates closer agreement with the reference distribution.
- **IS** (Salimans et al., 2016) measures the confidence and diversity of Inception predictions for generated images. Specifically, it is computed as $\exp(\mathbb{E}_x[D_{\mathrm{KL}}(p(y \mid x) \,\|\, p(y))])$, where $p(y \mid x)$ is the predicted class distribution for image x and $p(y)$ is its marginal over generated images. Higher IS indicates more confident predictions and greater class diversity.
- **rFID** uses the same FID formulation to compare reconstructed images with the original validation images. It evaluates reconstruction quality at the distribution level, with lower values indicating better preservation of the reference feature distribution.
- **LPIPS** (Zhang et al., 2018) measures perceptual differences between original and reconstructed images using a pretrained VGG-16 network. Squared differences between normalized features are weighted by learned linear layers, spatially averaged, and summed across network layers. Lower LPIPS indicates greater perceptual similarity.
- **PSNR** measures pixel-level reconstruction fidelity based on the mean squared error (MSE). With image intensities mapped and clamped to $[0, 1]$, it is computed as $\mathrm{PSNR} = -10 \log_{10}(\mathrm{MSE})$ in decibels. Higher PSNR indicates smaller reconstruction errors.
- **SSIM** (Wang et al., 2004) measures similarity in luminance, contrast, and structure between original and reconstructed images. We use the TorchMetrics implementation with a data range of 1. Higher SSIM indicates better structural preservation.

J More Qualitative Results

Figures 9 and 10 extend the token-group intervention analysis with additional examples of progressive replacement using reference-encoded groups. Figures 11 and 12 show class-conditional ImageNet samples generated at 256×256 resolution. Additional samples at 512×512 resolution are presented in Figures 13 and 14.

Table 8: **Model architectures, training hyperparameters, and sampling settings on ImageNet.**

Setting	Ours-B 256×256	Ours-L 256×256	Ours-B 512×512
Tokenizer Configs			
Patch size	16	16	16
Encoder depth / width / heads	12 / 768 / 12	12 / 768 / 12	12 / 768 / 12
Decoder depth / width / heads	12 / 768 / 12	24 / 1024 / 16	12 / 768 / 12
Encoder parameters	88.13M	88.13M	87.54M
Decoder parameters	87.90M	307.67M	88.49M
Tokenizer GFLOPs	93.71 + 93.99	93.71 + 324.75	344.80 + 345.38
Latent tokens	240	240	496
Total tokens per level	16 / 32 / 64 / 128	16 / 32 / 64 / 128	16 / 32 / 64 / 128 / 256
Reconstruction L2 / LPIPS weights	1 / 1	1 / 1	1 / 1
GAN generator loss weight	0.1	0.1	0.1
Aligned decoder block / Alignment weight	3 / 1	5 / 1	3 / 1
Quantizer Configs			
Codebook size / Code dimension	16,384 / 64	16,384 / 64	4,096 / 64
Commitment coefficient	0.25	0.25	0.25
Loss weight	1	1	1
AR Model Configs			
Depth / width / heads	12 / 768 / 12	32 / 1280 / 20	12 / 768 / 12
AR parameters	113.90M	690.88M	98.17M
AR GFLOPs	49.81	327.91	101.38
AR cross-entropy weight	0.1	0.1	0.1
Training Details			
Global training batch	256	256	256
Epochs	150	150	150
Optimizer	AdamW	AdamW	AdamW
Peak / Final AdamW learning rate	10^{-4} / 10^{-5}	10^{-4} / 10^{-5}	10^{-4} / 10^{-5}
AdamW (β_1, β_2), Tokenizer	(0,0.99)	(0,0.99)	(0,0.99)
AdamW (β_1, β_2), AR	(0.9,0.99)	(0.9,0.99)	(0.9,0.99)
Distribution-alignment Post-training			
Global post-training batch	256	128	128
Optimizer	AdamW	AdamW	AdamW
Peak / Minimum learning rate	10^{-6} / 10^{-7}	5×10^{-7} / 5×10^{-8}	5×10^{-7} / 5×10^{-8}
AdamW (β_1, β_2)	(0.9,0.95)	(0.9,0.95)	(0.9,0.95)
Learning-rate schedule	Warmup + cosine	Warmup + cosine	Warmup + cosine
Feature-statistics EMA decay	0.999	0.9995	0.9995
Autoregressive image generation			
Sequential AR sampling rounds	19	19	20
CFG scale	1.5	1.5	1.6

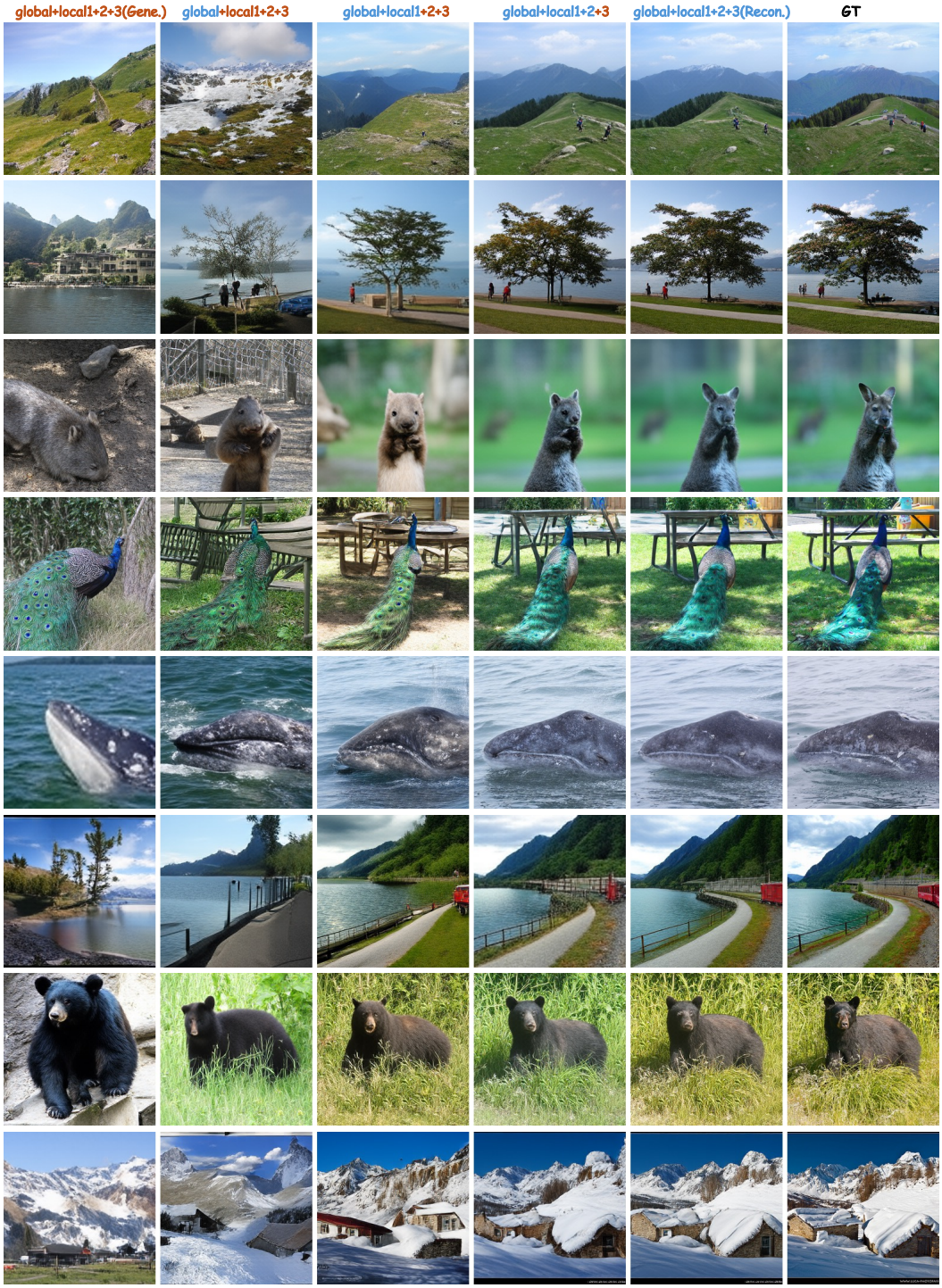


Figure 9: **Additional examples of progressive replacement with reference token groups.** Blue and orange labels denote reference-encoded and AR-sampled token groups, respectively.

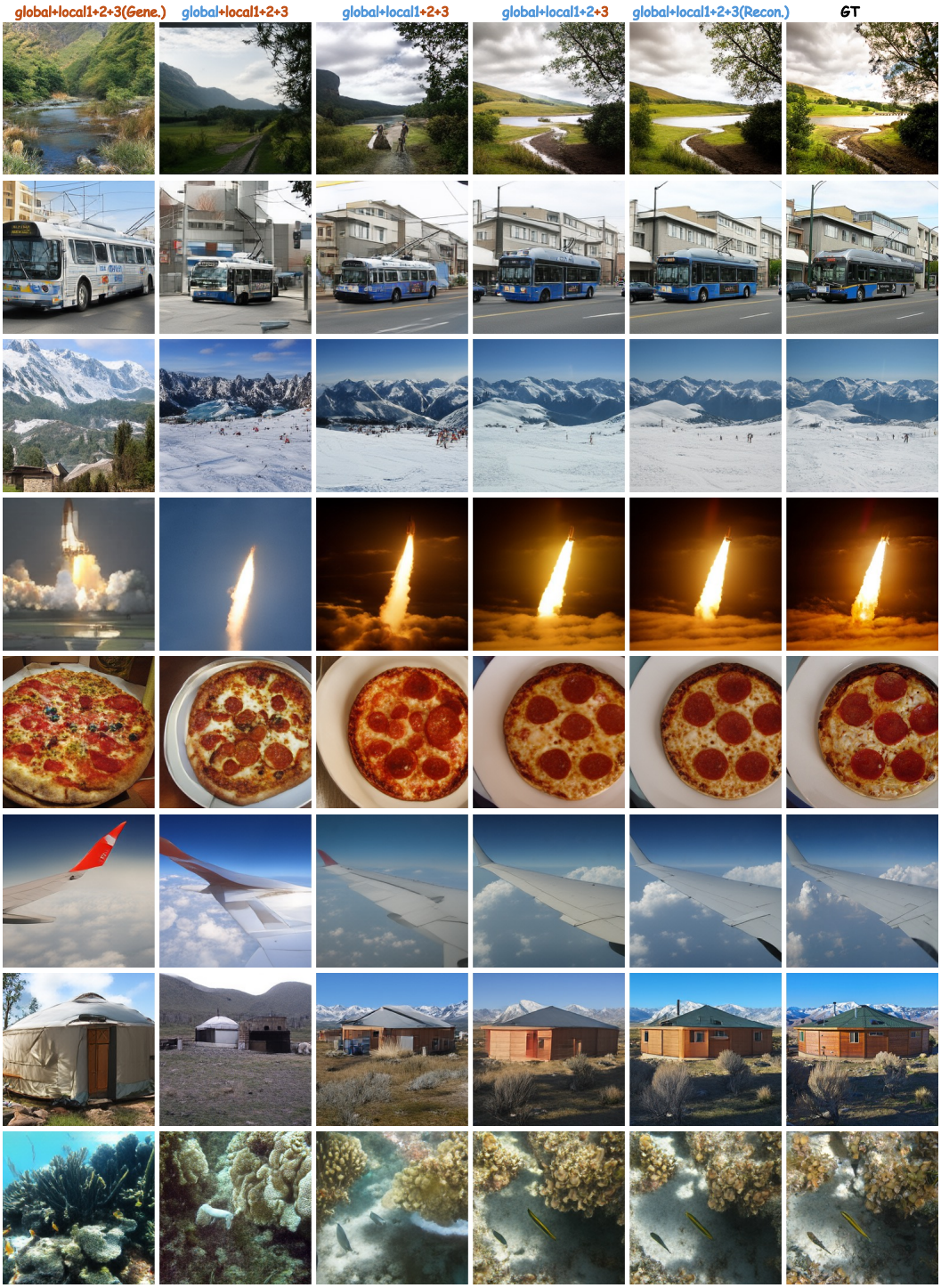


Figure 10: **Additional examples of progressive replacement with reference token groups.** Blue and orange labels denote reference-encoded and AR-sampled token groups, respectively.

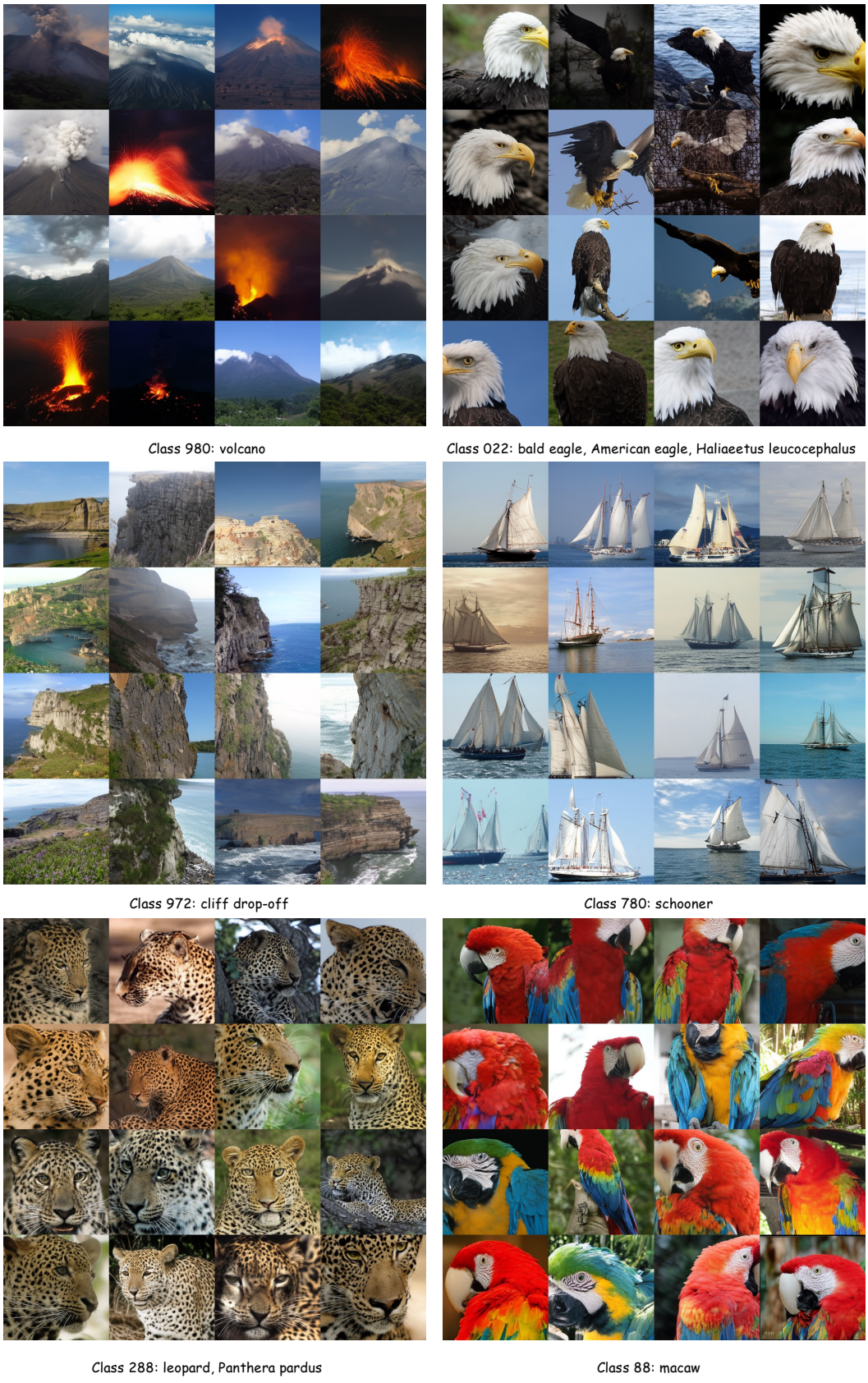


Figure 11: **Class-conditional samples generated by our method on ImageNet at** 256×256 **resolution.** Each block contains 16 samples from the indicated class.

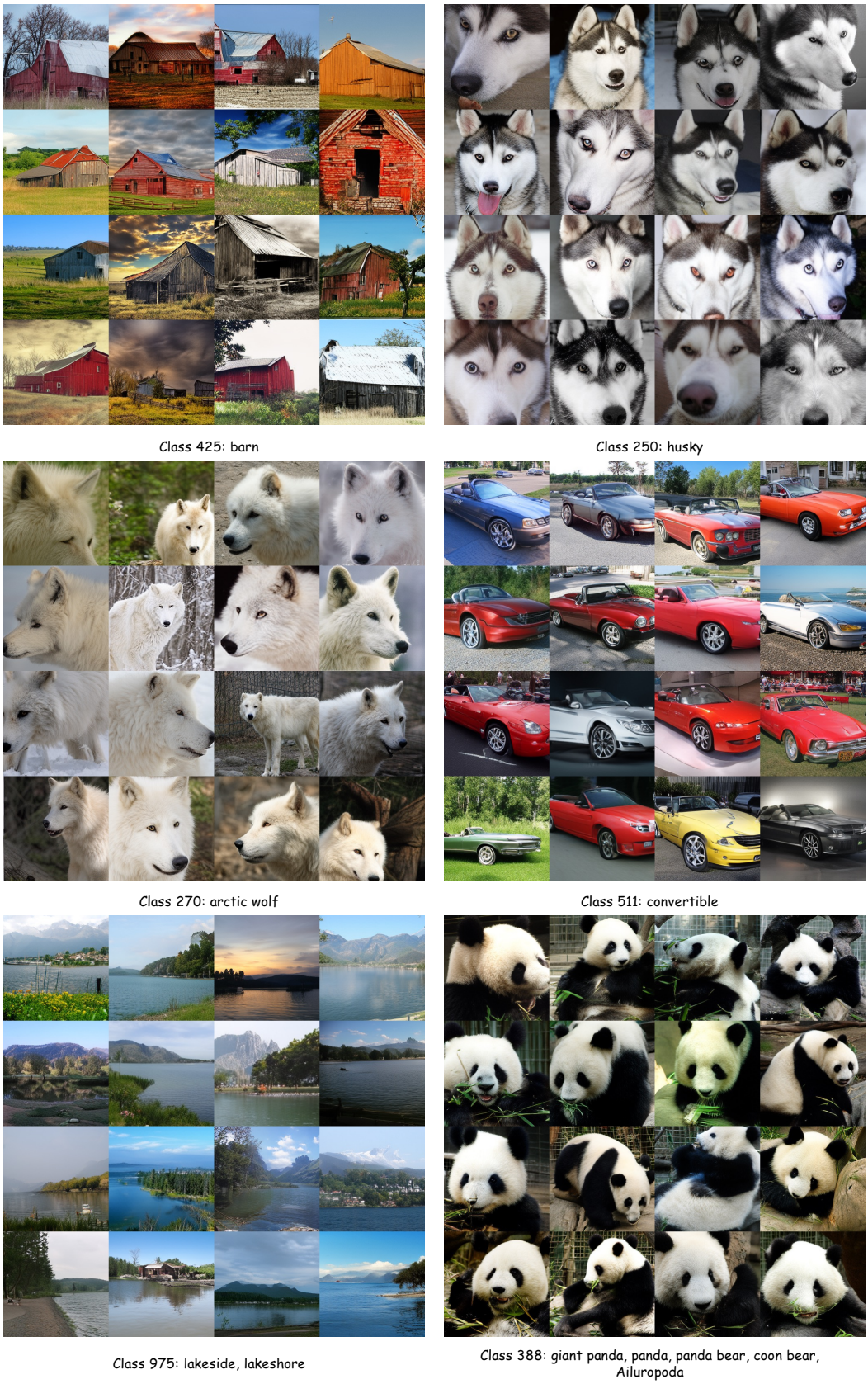


Figure 12: **Class-conditional samples generated by our method on ImageNet at** 256×256 **resolution.** Each block contains 16 samples from the indicated class.



Class 970: alp

Class 933: cheeseburger



Class 1:goldfish, carassius auratus



Class 992: agaric



Class 974: geyser



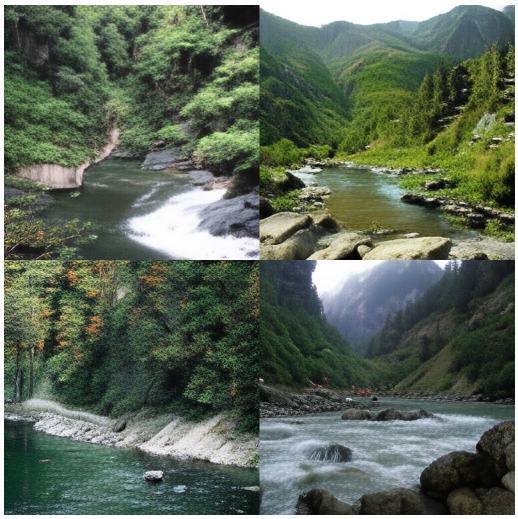
Class 291: lion, king of beasts, Panthera leo

Figure 13: **Class-conditional samples generated by our method on ImageNet at 512×512 resolution.** Each block contains 4 samples from the indicated class.



Class 9: ostrich, Struthio camelus

Class 449: boathouse



Class 979: valley, vale



Class 105: koala bear



Class 24: great grey owl, great gray owl, Strix nebulosa



Class 562: fountain

Figure 14: **Class-conditional samples generated by our method on ImageNet at 512×512 resolution.** Each block contains 4 samples from the indicated class.