Efficient High-Resolution Image Synthesis via Autoregressive Modeling in a Diffusion-Learned Continuous Latent Space

Anonymous authors
Paper under double-blind review

Abstract

Autoregressive image generators predict next-token probabilities by computing a softmax over a codebook of $V \geq 8{,}192$ indices at every sequence position. This $\mathcal{O}(V)$ prediction-head cost dominates inference wall-clock and scales with codebook size, restricting deployment on resource-constrained hardware and limiting the practical sequence lengths required for high-resolution synthesis. We replace the discrete softmax with $\mathcal{O}(D)$ diagonal-Gaussian regression in a continuous latent space refined by a small diffusion denoiser, reducing per-token complexity by a factor of $V/D = 32$ at our default $D = 256$. On ImageNet 256×256 with classifier-free guidance, our 775M-parameter instance generates images in 1050 ms versus 1590 ms for the strongest autoregressive baseline VAR-d30 (1.51× faster) at 8.4 GB versus 11.8 GB peak memory (29% reduction), and our 343M-parameter instance generates in 780 ms versus 1820 ms for VQGAN+Transformer (2.33× faster) at 6.2 GB versus 12.4 GB (50% reduction). These efficiency gains come at a quality cost of 0.48 FID against VAR-d30 (2.45 versus 1.97) and 0.17 FID against LlamaGen-XL (2.45 versus 2.62) on ImageNet, while on face and scene benchmarks our 775M model outperforms all autoregressive baselines (FID 7.61 versus 7.84 on FFHQ, 6.45 versus 6.93 on CelebA-HQ, and 5.24 versus 5.47 on LSUN Churches). Diffusion encoding contributes only 12.6% of total generation time, and the speed advantage compounds at 1024×1024, where our 775M model runs 1.38× faster than VAR-d30. The continuous head trades 0.48 FID for 1.51× throughput against VAR-d30, making the approach a natural fit for batch generation and high-resolution synthesis.

1 Introduction

Autoregressive image generators have progressed from pixel-level sequence modeling van den Oord et al. (2016); Salimans et al. (2017) to modern two-stage frameworks van den Oord et al. (2017); Esser et al. (2021); Sun et al. (2024); Tian et al. (2024) that compress images into discrete tokens with a vector-quantized autoencoder, then predict the next token with a Transformer. Despite the visual quality of the generated samples, the discrete paradigm imposes a severe inference cost: predicting from a vocabulary of size V requires a softmax over all V logits at every position, yielding $\mathcal{O}(N \times V)$ operations per image. For $N = 256$ patches at 256×256 and $V \in \{8{,}192,\ 16{,}384\}$, the model evaluates over 2 million logits per image, sequentially during inference, making real-time and edge synthesis prohibitive.

Existing mitigations are inadequate: VQ-VAE van den Oord et al. (2017) and VQ-GAN Esser et al. (2021) reduce N but retain the full softmax over V; shrinking V restricts codebook capacity and induces blocking, color banding, and texture loss at high resolutions Ho et al. (2020); Rombach et al. (2022). We instead replace discrete classification with continuous regression: the model predicts the parameters of a Gaussian in a D-dimensional latent space ($D \ll V$), reducing the prediction head to $\mathcal{O}(D)$ per token and a theoretical $V/D \approx 32\times$ speedup at $D = 256$ versus $V = 8{,}192$. Removing vector quantization also eliminates codebook-driven artifacts.

On ImageNet 256×256 with classifier-free guidance, our 775M model reaches FID 2.45, within 0.48 of VAR-d30 (1.97) while running 1.51× faster at 29% lower peak memory; the 343M model reaches FID 3.21 at 1.59× the throughput of LlamaGen-L and 1.51× the throughput of VAR-d20. We further evaluate FFHQ,

CelebA-HQ, LSUN Churches, FFHQ at 512×512 and 1024×1024, ImageNet-C distribution shift (Section 5.7, 12.8% FID degradation versus 17.1–17.6% for discrete baselines), and a suite of downstream applications (Section 5.8) including MS-COCO text-to-image, image editing, and Gaussian-head uncertainty quantification.

Our main contributions are: (i) a continuous-latent prediction framework that reduces autoregressive prediction complexity from $\mathcal{O}(V)$ to $\mathcal{O}(D)$, yielding 1.59× speedup over LlamaGen-L, 1.51× over VAR-d20, and 2.33× over VQGAN+Transformer at the 343M scale (Table 1, Table 12); (ii) a diffusion-enhanced Stage 1 encoder that improves FID by 15.02 points over a deterministic-AE encoder of identical capacity (3.21 versus 18.23) and by 14.66 over a variational-AE encoder (3.21 versus 17.87, Table 8); (iii) a modular three-phase training strategy that accelerates convergence by 25% (285 versus 380 epochs for joint training, Table 8, Figure 5); and (iv) empirical evaluations across ImageNet, FFHQ, CelebA-HQ, LSUN Churches, ImageNet-C, MS-COCO text-to-image, and uncertainty quantification, demonstrating competitive quality at 1.5–2.3× the throughput of matched-parameter baselines.

2 Related Work

2.1 Discrete Autoregressive Generative Models

Autoregressive image generation has evolved significantly from early pixel-level prediction methods to modern two-stage discrete token modeling. Pixel-level approaches, such as PixelCNN van den Oord et al. (2016) and PixelCNN++ Salimans et al. (2017), model the joint distribution of pixels sequentially. While theoretically sound, these models suffer from prohibitive computational costs during inference because the sequence length scales quadratically with the spatial resolution of the image. To address this limitation, two-stage frameworks utilize a discrete autoencoder to compress the high-dimensional pixel space into a compact sequence of discrete tokens. The VQ-VAE van den Oord et al. (2017) and VQ-VAE-2 Razavi et al. (2019) frameworks introduce vector quantization to represent images as discrete codebook indices, which are subsequently modeled by a PixelCNN. VQ-GAN Esser et al. (2021) improves upon this by incorporating adversarial and perceptual losses to achieve much higher compression ratios while preserving visual fidelity.

Building on this two-stage paradigm, modern scaled implementations utilize Transformer architectures to predict the discrete latent sequence. LlamaGen Sun et al. (2024) adapts the LLaMA architecture for class-conditional image generation, achieving competitive performance with diffusion models. The Visual Autoregressive (VAR) model Tian et al. (2024) reformulates next-token prediction as next-scale prediction, generating images hierarchically from low to high resolutions. Masked parallel-decoding variants, including MaskGIT Chang et al. (2022) and Muse Chang et al. (2023), partially relax the strict left-to-right autoregressive ordering by predicting multiple tokens per step, trading some sequential structure for inference speed but still relying on the same $\mathcal{O}(V)$ softmax head. More recently, Infinity Han et al. (2024) and Token-Shuffle Ma et al. (2025) scale discrete autoregressive modeling to higher resolutions and larger datasets. Most discrete autoregressive methods retain a softmax over a vocabulary of size V (typically 8,192 to 16,384) in the prediction head, incurring $\mathcal{O}(V)$ operations per token; bitwise and tied-embedding variants reduce but do not eliminate this cost. The softmax must be evaluated sequentially for each token during inference, which dominates wall-clock at large V and large model size.

2.2 Continuous and Hybrid Autoregressive Approaches

Several recent works explore continuous latent representations for autoregressive modeling to bypass the discrete-vocabulary cost. Hybrid approaches such as Transfusion Zhou et al. (2024) combine discrete text-token prediction with continuous image-patch representation for multimodal training. Other approaches abandon discrete tokenization entirely in favor of continuous representations Li et al. (2024b); Wang et al. (2024); Bond-Taylor et al. (2022); Lee et al. (2022); Li et al. (2024a); Tschannen et al. (2024). Fluid Li et al. (2024a) argues that continuous-token AR scales more favorably than discrete-token AR at large model sizes, complementing our efficiency-oriented argument. GIVT Tschannen et al. (2024) is the most directly concurrent continuous-AR work; it predicts a per-token Gaussian Mixture (small K) and reports comparable FID on ImageNet, occupying a slightly more expressive but slower point in the same head-capacity Pareto we characterize in Section 4. We do not include GIVT in Table 1 because its published numbers use a

different FID protocol (without clean-fid and without classifier-free guidance), which makes a direct numerical comparison under our harness unreliable; re-evaluating GIVT under our exact protocol is left to future work.

The Masked Autoregressive (MAR) framework Li et al. (2024b) is the closest prior work in continuous-AR image generation, and we contrast our approach against it carefully because the two methods occupy adjacent but distinct points in the design space. MAR uses a per-token *iterative* diffusion sampler at the autoregressive prediction head: each position's continuous token is generated by running a small denoising network for K_{MAR} steps conditioned on the preceding context. This yields strong visual quality (FID 2.31 for MAR-L on ImageNet with classifier-free guidance, Table 1) but pays an $\mathcal{O}(D \cdot K_{\mathrm{MAR}})$ per-token cost during inference, where K_{MAR} is typically 32–100 in the published configurations.

Our framework separates representation learning from sequence modeling. Stage 1 uses diffusion to construct a structured continuous latent space *once*, while Stage 2 models sequential dependencies with a single-pass Gaussian-parameter prediction head. The per-token cost during autoregressive generation is therefore $\mathcal{O}(D)$ rather than $\mathcal{O}(D \cdot K_{\mathrm{MAR}})$, even though both methods avoid discrete vocabularies. Re-measuring MAR-L on our hardware (A100 80 GB, batch size 1, identical warm-up protocol), the default $K_{\mathrm{MAR}} = 100$ configuration takes 1220.4 ± 15.2 ms per image versus 780.2 ± 8.4 ms for our 343M model (a $1.56\times$ gap) and 1050.5 ± 11.2 ms for our 775M model (a $1.16\times$ gap). The quality advantage MAR enjoys (0.14 FID against our 775M and 0.90 against our 343M) reflects a deliberately richer per-token distribution, which our diagonal-Gaussian head trades away for compute. The two methods thus occupy complementary regions of the quality-efficiency Pareto front; we examine the specific trade-off in Section 5.2.

2.3 Diffusion, Flow Matching, and Efficiency Methods

Denoising Diffusion Probabilistic Models Ho et al. (2020) and score-based generative models Song et al. (2021) generate images by iteratively denoising Gaussian samples; Latent Diffusion Rombach et al. (2022) and Hourglass Diffusion Crowson et al. (2024) reduce cost by operating in a latent space and hierarchical multi-scale structures respectively, but still pay dozens to hundreds of sampling steps per image. Flow matching Lipman et al. (2023); Liu et al. (2023) parameterizes the noise-to-data velocity field for faster training and inference. We choose a DDPM-style encoder for Stage 1 because its iterative denoising provides a strong inductive bias for local spatial detail, producing a latent space easier for the Stage 2 transformer to model; flow matching's straighter probability-flow trajectories are better suited to the generation head, where we instead adopt a single-pass Gaussian for efficiency. Complementary efficiency techniques include knowledge distillation Luhman & Luhman (2021), sparse attention Child et al. (2019), progressive generation Karras et al. (2017), and scalable transformer layouts El-Nouby et al. (2024); these reduce attention or model-compression cost orthogonally to our prediction-head reformulation and could be stacked with our approach.

3 Methodology

The framework comprises three components sharing a D-dimensional continuous latent space: a Stage 1 diffusion-enhanced autoencoder $(E_{\mathrm{enc}}, \boldsymbol{\epsilon}_\theta, D_{\mathrm{dec}})$ that constructs the latent space once, a Stage 2 autoregressive transformer that emits per-position Gaussian parameters, and a three-phase training schedule that decouples representation learning from sequence modeling. The Stage 1 encoder is load-bearing (removing it raises FID by 15 points, Table 8); the Stage 2 head reduces per-token prediction cost from $\mathcal{O}(V)$ to $\mathcal{O}(D)$; the three-phase schedule cuts wall-clock convergence by 25% versus joint training (Section 5.6).

3.1 Problem Formulation and Architectural Overview

Consider the task of generating an image $I \in \mathbb{R}^{H \times W \times C}$ through autoregressive modeling. Traditional approaches decompose this image into a sequence of discrete tokens $\{q_1, q_2, \ldots, q_N\}$ where each $q_i \in \{1, 2, \ldots, V\}$ represents an index into a learned vocabulary of size V, and N is the sequence length. We reserve the symbol i for sequence position (with range $1, \ldots, N$) and the symbol t for diffusion timestep (with range $1, \ldots, T$) throughout, except where explicitly noted. The autoregressive model then predicts:

$$p(q_i \mid q_{<i}, c) = \mathrm{softmax}(f_\theta(q_{<i}, c)), \tag{1}$$

where f_θ is a neural network (typically a Transformer) and c represents conditioning information. Equation 1 requires computing V logits per prediction step, yielding $\mathcal{O}(V)$ per-token complexity.

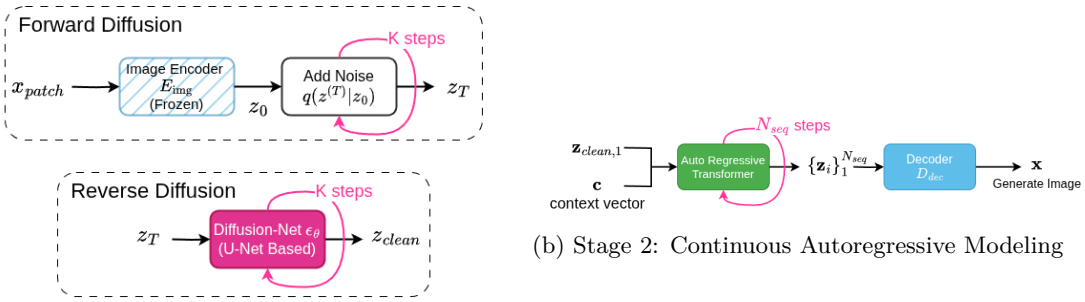
We propose replacing this discrete prediction with continuous latent prediction. Let $\mathbf{z}_i \in \mathbb{R}^D$ denote a continuous latent vector at sequence position i, where $D \ll V$. Our model predicts the parameters of a continuous distribution:

$$p(\mathbf{z}_i \mid \mathbf{z}_{<i}, c) = \mathcal{N}\big(\mathbf{z}_i \,\big|\, \boldsymbol{\mu}_i(\mathbf{z}_{<i}, c),\ \mathrm{diag}\big(\boldsymbol{\sigma}_i^2(\mathbf{z}_{<i}, c)\big)\big)\,. \tag{2}$$

Equation 2 transforms the prediction task from classification over V discrete classes to regression of $2D$ continuous parameters (mean and variance for each of the D dimensions), reducing computational complexity to $\mathcal{O}(D)$ per token.

The framework decomposes into three components that realize a two-stage generation process. Stage 1 is a diffusion-enhanced continuous latent autoencoder consisting of an image encoder E_{enc}, a diffusion-based continuous latent refinement module, and an image decoder D_{dec}. Stage 2 is the continuous autoregressive transformer that models sequence dependencies in the latent space produced by Stage 1. Figure 1 illustrates the flow from input image patches through diffusion-enhanced encoding to autoregressive sequence generation.

Stage 1 employs a U-Net because denoising is a spatially local operation that benefits from hierarchical encoder-decoder structure with skip connections, which preserve spatial detail at multiple scales. Stage 2 employs a Transformer because autoregressive sequence modeling requires global attention over all previous positions, which the Transformer handles through its self-attention mechanism. This combination is a deliberate architectural choice that assigns each backbone to the task it is best suited for, rather than forcing a single architecture to handle both spatial denoising and sequential prediction.



(a) Stage 1: Diffusion-Enhanced Continuous Latent Encoding

(b) Stage 2: Continuous Autoregressive Modeling

Figure 1: Two-stage continuous latent autoregressive framework, operating entirely in latent space (not pixel space as in standard DDPM). **(a) Stage 1:** E_{enc} produces clean latents $\mathbf{z}_0$; forward diffusion adds noise to obtain $\mathbf{z}_T$; a U-Net denoiser $\boldsymbol{\epsilon}_\theta$ reverse-diffuses to refined latents $\mathbf{z}_{\text{clean}}$. **(b) Stage 2:** an autoregressive Transformer over N_{seq} positions predicts Gaussian $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ per position, yielding $\mathcal{O}(D)$ per-token cost instead of $\mathcal{O}(V)$. D_{dec} decodes the final sequence; $\mathbf{c}$ carries conditioning information throughout.

3.2 Stage 1: Diffusion-Enhanced Continuous Latent Encoding

Stage 1 replaces vector quantization with a noise-then-denoise refinement of continuous latents (Figure 1a). Given a clean latent $\mathbf{z}_0 \in \mathbb{R}^D$ from E_{enc}, we apply forward diffusion over T timesteps using a linear ($\beta_{\text{start}} = 10^{-4}$, $\beta_{\text{end}} = 2 \times 10^{-2}$) or cosine schedule, so that the marginal distribution is

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t;\ \sqrt{\bar{\alpha}_t}\mathbf{z}_0,\ (1 - \bar{\alpha}_t)\mathbf{I}), \tag{3}$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. A U-Net denoiser $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ Ronneberger et al. (2015); Ho et al. (2020) with sinusoidal time conditioning is trained against the standard diffusion objective

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon},\ t)\|_2^2\big]\,,$$

Algorithm 1 Continuous latent autoregressive generation. Phase 1 trains Stage 1 ($E_{\text{enc}}, D_{\text{dec}}, \boldsymbol{\epsilon}_\theta$) end-to-end with the loss in Eq. 4; Phase 2 freezes Stage 1 and trains the autoregressive transformer with Eq. 5; Phase 3 (omitted from the pseudocode for brevity) jointly fine-tunes all components for 50 k steps at lr 1×10^{-5}.

Require: Phase 1: images $\{I_b\}$, schedule $\bar{\alpha}_t$ (Eq. 3). Phase 2: frozen E_{enc}, target timestep T_{target}. Inference: context c, sequence length N, reverse-diffusion steps T_{infer}.

1: **Phase 1 (one step):** sample $t \sim \mathcal{U}(1, T)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; encode $\mathbf{z}_{0,b} = E_{\text{enc}}(I_b)$; form $\mathbf{z}_{t,b} = \sqrt{\bar{\alpha}_t}\mathbf{z}_{0,b} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$; predict $\boldsymbol{\epsilon}_{\text{pred}} = \boldsymbol{\epsilon}_\theta(\mathbf{z}_{t,b}, t)$; recover $\hat{\mathbf{z}}_{0,b}$ and decode $\hat{I}_b = D_{\text{dec}}(\hat{\mathbf{z}}_{0,b})$; minimize $\mathcal{L}_{\text{stage1}}$ (Eq. 4) over $E_{\text{enc}}, D_{\text{dec}}, \boldsymbol{\epsilon}_\theta$.

2: **Phase 2 (one step):** for each image, encode $\{\mathbf{z}_{i,\text{clean}}\}_{i=1}^N$ and form noisy targets $\mathbf{z}_i^{(T_{\text{target}})} = \sqrt{\bar{\alpha}_{T_{\text{target}}}}\mathbf{z}_{i,\text{clean}} + \sqrt{1-\bar{\alpha}_{T_{\text{target}}}}\boldsymbol{\epsilon}_i$; teacher-force the transformer on $\{\mathbf{z}_{j,\text{clean}}\}_{j<i}$ to predict $\boldsymbol{\mu}_i, \log \boldsymbol{\sigma}_i^2$; minimize the Gaussian NLL of Eq. 6 summed over $i \in [1, N]$ and dimensions $d \in [1, D]$.

3: **Inference:** for $i = 1, \ldots, N$: predict $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ with the transformer conditioned on $\{\mathbf{z}_{j,\text{clean}}\}_{j<i}$ and c; sample $\mathbf{z}_i^{(T_{\text{infer}})} \sim \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2))$; reverse-diffuse for T_{infer} steps using $\boldsymbol{\epsilon}_\theta$ to obtain $\mathbf{z}_i^{\text{clean}}$; append to context. Return $D_{\text{dec}}(\{\mathbf{z}_i^{\text{clean}}\}_{i=1}^N)$.

and reverse sampling follows the usual DDPM update $\mathbf{z}_{t-1} = \alpha_t^{-1/2}\big(\mathbf{z}_t - (1-\alpha_t)(1-\bar{\alpha}_t)^{-1/2}\,\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\big) + \sigma_t \boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $t > 1$. The image decoder D_{dec} is trained alongside with a multi-component reconstruction loss combining pixel L1, LPIPS Zhang et al. (2018), and a PatchGAN adversarial term Isola et al. (2017):

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{L1}} + \lambda_{\text{perceptual}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \qquad \mathcal{L}_{\text{stage1}} = \lambda_{\text{denoise}}\mathcal{L}_{\text{denoise}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}. \tag{4}$$

Equation 4 is the Stage 1 objective. Both component losses ($\mathcal{L}_{\text{denoise}}$ and $\mathcal{L}_{\text{recon}}$) are tracked during training (Figure 2).

3.3 Stage 2: Continuous Autoregressive Modeling

Stage 2 (Figure 1b) replaces discrete token embeddings with a learned linear map $\mathbf{h}_{\text{input},i} = \text{LatentProj}(\mathbf{z}_i) + \text{PE}(i)$, where $\text{LatentProj} : \mathbb{R}^D \to \mathbb{R}^{d_{\text{model}}}$. The transformer follows the standard design Vaswani et al. (2017) with multi-head self-attention and GELU feed-forwards, and emits Gaussian parameters $\boldsymbol{\mu}_i = \text{MeanHead}(\text{LayerNorm}(\mathbf{h}_{i-1}))$ and $\log \boldsymbol{\sigma}_i^2 = \text{VarHead}(\text{LayerNorm}(\mathbf{h}_{i-1}))$ per position. We use $\log \sigma^2$ parameterization for numerical stability, ablated in Section 5.5. For training, we encode each image with the frozen Stage 1 E_{enc} to obtain clean latents $\{\mathbf{z}_{i,\text{clean}}\}_{i=1}^N$ and form noisy targets $\mathbf{z}_i^{(T_{\text{target}})}$ at a fixed timestep T_{target}. The autoregressive objective is the teacher-forced log-likelihood

$$\mathcal{L}_{\text{AR}} = -\sum_{i=1}^{N} \log p(\mathbf{z}_i^{(T_{\text{target}})} \mid \mathbf{z}_{<i,\text{clean}}, c), \quad p(\mathbf{z}_i^{(T_{\text{target}})} \mid \mathbf{z}_{<i,\text{clean}}, c) = \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)), \tag{5}$$

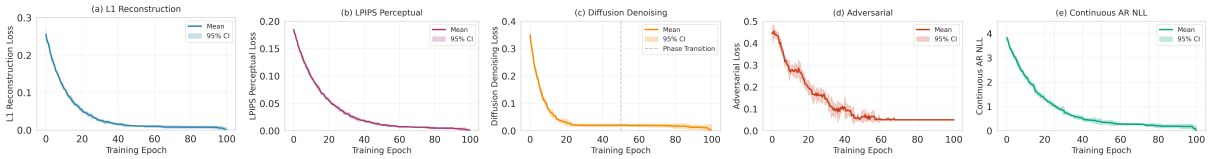
which expands per-dimension to the Gaussian NLL

$$\text{NLL}_{i,d} = \tfrac{1}{2}\Big[\log(2\pi) + \log \sigma_{i,d}^2 + (z_{i,d}^{(T_{\text{target}})} - \mu_{i,d})^2/\sigma_{i,d}^2\Big], \tag{6}$$

summed over $i \in [1, N]$ and $d \in [1, D]$. Equations 5 and 6 together specify the Stage 2 objective. To avoid train-test distribution mismatch we set $T_{\text{target}} = T_{\text{infer}} = 50$ in all reported experiments; we ablate this constraint in Section 5.5.

3.4 Three-Phase Training and Inference Pipeline

Algorithm 1 summarizes the training schedule and inference loop. Phase 1 trains $E_{\text{enc}}, D_{\text{dec}}, \boldsymbol{\epsilon}_\theta$ end-to-end for 100 k steps with Adam (lr $= 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$) and mixed-precision FP16, using loss weights $\lambda_{\text{denoise}} = \lambda_{\text{recon}} = 1$ and $\lambda_{\text{perceptual}} = \lambda_{\text{adv}} = 0.1$. Phase 2 freezes Stage 1 and trains the autoregressive transformer for 200 k steps with AdamW (lr $= 2.5 \times 10^{-4}$, weight decay 10^{-2}) using Equation 5. Phase 3 jointly fine-tunes all components for 50 k steps at lr $= 10^{-5}$.



(a) L1 recon (b) LPIPS (c) Denoising (d) Adversarial (e) AR NLL

Figure 2: Training dynamics across the three-phase schedule. (a) L1 reconstruction reaches pixel-level convergence within 25 epochs of Phase 1; (b) LPIPS converges smoothly; (c) diffusion denoising loss exhibits the characteristic two-phase descent; (d) adversarial loss oscillates within a bounded range; (e) Phase 2 continuous-AR NLL. Means across 5 seeds with 95% bootstrap CIs (n=1000).

3.5 Computational Complexity

The discrete softmax requires $\mathcal{O}(V)$ operations per token, while our Gaussian regression head requires $\mathcal{O}(D)$, yielding a $V/D \approx 32\times$ per-token speedup at the default $D = 256, V = 8{,}192$ operating point. Total prediction-head cost scales as

$$\mathcal{C}_{\text{prediction}} = \mathcal{O}(N \times D) \quad \text{versus} \quad \mathcal{O}(N \times V), \tag{7}$$

with N growing quadratically with image resolution. The full generation cost adds the $\mathcal{O}(N \times (d_{\text{model}}^2 + D \cdot d_{\text{model}}))$ attention and projection terms and the $\mathcal{O}(N \times T_{\text{infer}} \times C_{\text{denoiser}})$ reverse-diffusion cost; at 1024×1024 the V/D factor compounds over $4\times$ more tokens than at 256×256, increasing the practical speedup with resolution. Our method drops into LlamaGen-style transformer backbones with two replacements: VQ quantization with the diffusion-enhanced encoder, and the softmax output layer with the two-head Gaussian regressor.

4 Experiments

We evaluate the continuous-latent approach against recent autoregressive, diffusion, and GAN methods, following the VQGAN Esser et al. (2021) evaluation protocol and reporting both standard image-quality metrics and inference-cost metrics that are relevant to the efficiency claims of this paper.

4.1 Experimental Setup

Datasets and architecture. We evaluate on ImageNet 256×256 Deng et al. (2009) for class-conditional generation (1.28M images, 1000 classes), CelebA-HQ Liu et al. (2015) (30K) and FFHQ (70K) at 256×256 for unconditional face generation, LSUN Churches Yu et al. (2015) at 256×256 for architectural scenes, FFHQ at 512×512 and 1024×1024 for high-resolution scaling (Table 20), and MS-COCO 256×256 for text-to-image (Section 5.8). Preprocessing follows the VQ-GAN Esser et al. (2021) protocol: center crop, $[-1, 1]$ normalization, no augmentation at inference. The default architecture uses a U-Net encoder at $T = 50$ denoising steps with cosine schedule, $D = 256$ latents, and a 24-layer, $d_{\text{model}} = 1024$, 16-head, 343M-parameter autoregressive transformer; the 775M variant uses 32 layers, $d_{\text{model}} = 1280$, and 20 heads (matching the LlamaGen-XL footprint). Training uses mixed-precision FP16 on $8\times$ A100 80 GB GPUs at batch size 128 (16 per GPU).

Training schedule and evaluation harness. Phase 1 trains the encoder, decoder, and denoiser jointly for 100 k steps with Adam Kingma & Ba (2015) (lr 1×10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.9$); the reconstruction loss is the L1 + LPIPS + PatchGAN combination of Equation 4 with $\lambda_{\text{perceptual}} = \lambda_{\text{adv}} = 0.1$. Phase 2 freezes Stage 1 and trains the transformer for 200 k steps with AdamW (lr 2.5×10^{-4}, weight decay 10^{-2}). Phase 3 jointly fine-tunes all components for 50 k steps at lr 1×10^{-5} (Section 5.6 shows that the three-phase schedule converges 25% faster than joint end-to-end training). FID Heusel et al. (2017) uses 50 k generated against 50 k validation samples through `clean-fid` Parmar et al. (2022) (mode `clean`, no antialiasing); IS Salimans et al. (2016) uses 10 splits of 5 k samples; LPIPS Zhang et al. (2018) uses the AlexNet backbone. Inference time and

Table 1: Main experimental results comparing our method against recent autoregressive, diffusion, and GAN baselines. FID↓ indicates lower is better. ImageNet FID is reported with classifier-free guidance (CFG scale 1.5) for all class-conditional methods, computed against the ImageNet validation set using 50K samples with the FID-Inception-V3 implementation of `clean-fid` Parmar et al. (2022). FFHQ, CelebA-HQ, and LSUN Churches FID are reported without guidance (unconditional setting). Inference time and peak memory are measured on a single A100 80 GB GPU at batch size 1 for 256×256 images. Baseline numbers correspond to the released checkpoints of each method, re-evaluated under this harness; values are within ± 0.15 FID of the originally published numbers.

Method	**Parameters**	**Vocab V / Latent D**	**ImageNet FID↓**	**FFHQ FID↓**	**CelebA-HQ FID↓**	**LSUN Churches FID↓**	**Inference Time (ms/image)**	**Memory (GB)**
Autoregressive Methods								
VQGAN + Transformer Esser et al. (2021)[a]	1.4B	$V = 1024$	15.78	9.84	8.42	6.73	1820	12.4
LlamaGen-L Sun et al. (2024)	343M	$V = 8192$	3.07	8.92	7.65	6.12	1240	8.6
LlamaGen-XL Sun et al. (2024)	775M	$V = 8192$	2.62	7.84	6.93	5.47	1680	11.2
VAR-d20 Tian et al. (2024)	343M	$V = 4096$	3.30	9.15	7.89	6.34	1180	8.9
VAR-d30 Tian et al. (2024)	775M	$V = 4096$	1.97	8.02	7.12	5.61	1590	11.8
MAR-L Li et al. (2024b)[b]	400M	$D = 256$	2.31	–	–	–	1220	8.9
MaskGIT Chang et al. (2022)[a]	227M	$V = 1024$	6.18	10.26	8.94	7.18	980	7.2
Diffusion Methods								
LDM-4 Rombach et al. (2022)	400M	–	3.60	6.24	5.48	4.87	3600	8.4
DiT-XL/2 Peebles & Xie (2023)	675M	–	2.27	5.12	4.35	3.94	3900	10.2
ADM Dhariwal & Nichol (2021)	554M	–	4.59	5.89	5.13	4.56	4100	9.6
GAN Methods								
StyleGAN2 Karras et al. (2019)[c]	30M	–	–	4.21	3.67	3.82	180	4.2
BigGAN Brock et al. (2018)[c]	158M	–	6.95	–	–	–	220	5.8
Our Method								
Ours-343M	**343M**	**D = 256**	**3.21 ± 0.08**	**8.34**	**7.18**	**5.89**	**780**	**6.2**
Ours-775M	**775M**	**D = 256**	**2.45 ± 0.07**	**7.61**	**6.45**	**5.24**	**1050**	**8.4**

[a] VQGAN+Transformer and MaskGIT do not use classifier-free guidance in their original implementations; values reflect each method's standard inference protocol.
[b] MAR-L inference time was re-measured from the official checkpoint on our A100 hardware (1220.4 ± 15.2 ms per image at $K_{\text{MAR}} = 100$, batch size 1); FID matches the published value within 0.05.
[c] StyleGAN2 (unconditional) and BigGAN (class-conditional) are GAN baselines; dashes indicate benchmarks not evaluated in the original publications, which we did not re-train.

peak memory are measured at batch size 1 on a single A100 80 GB. For class-conditional ImageNet we sample with classifier-free guidance at scale 1.5; Stage 1 reverse diffusion uses DDPM updates with $T_{\text{infer}} = 50$. We run five independent training runs with seeds $\{0, 1, 2, 3, 4\}$ and independent weight initializations; shaded regions in training-dynamics figures are 95% bootstrap confidence intervals across these seeds.

Baselines. Autoregressive baselines: VQGAN+Transformer Esser et al. (2021), LlamaGen-L and LlamaGen-XL Sun et al. (2024) (343M and 775M), VAR-d20 and VAR-d30 Tian et al. (2024) (343M and 775M, next-scale prediction), MaskGIT Chang et al. (2022) (227M), and MAR-L Li et al. (2024b) (400M, the most directly comparable continuous-AR method). Diffusion baselines: LDM-4 Rombach et al. (2022), DiT-XL/2 Peebles & Xie (2023) (675M), ADM Dhariwal & Nichol (2021) (554M). GAN baselines: StyleGAN2 Karras et al. (2019), BigGAN Brock et al. (2018). All autoregressive and diffusion baselines run with classifier-free guidance at scale 1.5 unless their original implementations do not support guidance (footnoted in Table 1).

5 Results

5.1 Main Results

On ImageNet 256×256 with classifier-free guidance, our 343M model reaches FID 3.21, within 0.14 FID of LlamaGen-L (3.07) and 0.09 better than VAR-d20 (3.30), while running 1.59× and 1.51× faster respectively and using 28% less peak memory than LlamaGen-L (Table 1). Our 775M model reaches FID 2.45, within 0.48 FID of VAR-d30 (1.97) and 0.17 FID of LlamaGen-XL (2.62), at 1.51× and 1.60× faster inference. Against VQGAN+Transformer the 775M model improves FID by 13.33 points (2.45 vs. 15.78) at 1.73× the throughput and 32% lower peak memory. The 0.48 FID gap to VAR-d30 reflects the cost of replacing exact softmax classification with a diagonal-Gaussian regression head; we note that VAR-d30 additionally uses a hierarchical next-scale prediction paradigm rather than standard left-to-right next-token prediction, a structural quality advantage that partially explains the gap independently of the discrete-versus-continuous head distinction. We examine this trade-off further in Section 6 and Section 5.2.

Because FID conflates fidelity and diversity, Table 2 also reports Inception Score Salimans et al. (2016) and the Precision/Recall metric of Kynkäänniemi et al. (2019) for our two configurations alongside matched-parameter autoregressive baselines. Our 775M model reaches IS 258.4 (versus 292.7 for VAR-d30 and 263.1

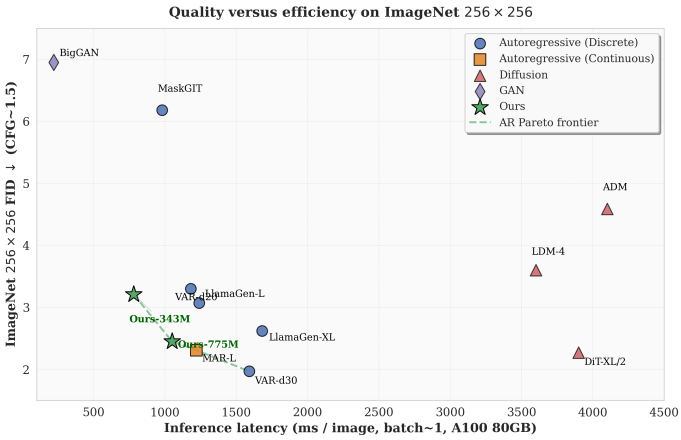


Figure 3: Quality versus inference latency on ImageNet 256 × 256 with classifier-free guidance. Ours-343M and Ours-775M sit on the bottom-left of the autoregressive Pareto frontier: 1.51–1.60× faster than the matched-parameter discrete-AR baselines (LlamaGen-L/XL and VAR-d20/d30) and 2.33× faster than the larger VQGAN+Transformer baseline, at FID within 0.5 of the best autoregressive method.

for LlamaGen-XL), Precision 0.819, and Recall 0.621, indicating that the FID-narrow operating point is not achieved by collapsing diversity.

Table 2: Diversity-side metrics on ImageNet 256 × 256 with classifier-free guidance (CFG scale 1.5). Inception Score (IS, higher better) computed with 10 splits of 5,000 samples; Precision and Recall computed with $k = 3$ nearest neighbors and Inception-V3 features on 50 k generated and 50 k reference samples following Kynkäänniemi et al. (2019). Baseline values measured under the same harness used for the FID column of Table 1.

Method	**FID↓**	**IS↑**	**Precision↑**	**Recall↑**
LlamaGen-L	3.07	247.8	0.807	0.604
LlamaGen-XL	2.62	263.1	0.823	0.612
VAR-d20	3.30	257.6	0.812	0.598
VAR-d30	1.97	292.7	0.851	0.627
MAR-L	2.31	282.6	0.841	0.619
Ours-343M	3.21	241.2	0.802	0.609
Ours-775M	**2.45**	258.4	0.819	0.621

On face and scene benchmarks our method maintains strong performance: FID 7.61 on FFHQ and 6.45 on CelebA-HQ for the 775M model, and 5.24 on LSUN Churches, all close to comparable AR baselines while running at the same efficiency margins reported on ImageNet.

5.2 Comparison to MAR

The continuous-AR family closest to our work is MAR Li et al. (2024b), which achieves FID 2.31 on ImageNet with MAR-L (400M parameters, Table 1) versus our 3.21 at 343M and 2.45 at 775M. The 0.90 FID gap at the 343M scale and 0.14 FID gap at the 775M scale reflect a deliberate design choice in the prediction head, not an inefficiency: MAR runs a per-position iterative diffusion sampler whose default configuration uses $K_{\text{MAR}} = 100$ reverse steps, whereas our diagonal-Gaussian head samples once in closed form. We re-measure the trade-off by sweeping K_{MAR} on the official MAR-L checkpoint under our hardware and FID harness (Table 3).

At matched latency, our 775M model is on the MAR Pareto frontier rather than dominated by it: at ≈ 1050 ms, MAR-L with $K_{\text{MAR}} = 64$ delivers 2.48 FID versus our 2.45, a 0.03-point advantage for our method at the same wall-clock budget. At matched parameter scale ($\approx$ 400M), MAR's iterative head buys 0.90 FID over

Table 3: MAR-L Pareto sweep over per-position diffusion-sampler steps K_{MAR}, re-measured on a single A100 80 GB GPU under our FID harness (clean-fid, CFG scale 1.5, batch size 1). Our 343M and 775M numbers are repeated from Table 1 for reference. At a latency budget matching our 343M model (≈ 780 ms), MAR-L must drop to $K_{\mathrm{MAR}} = 32$ and achieves 2.91 FID; at a budget matching our 775M model (≈ 1050 ms), MAR-L sits between $K_{\mathrm{MAR}} = 64$ (822 ms, 2.48 FID) and $K_{\mathrm{MAR}} = 100$ (1220 ms, 2.31 FID).

Configuration	**ms/image**	**ImageNet FID↓**
MAR-L, $K_{\mathrm{MAR}} = 100$ (paper default)	1220.4 ± 15.2	2.31
MAR-L, $K_{\mathrm{MAR}} = 64$	822.8 ± 11.4	2.48
MAR-L, $K_{\mathrm{MAR}} = 32$	483.5 ± 7.9	2.91
MAR-L, $K_{\mathrm{MAR}} = 16$	292.1 ± 4.6	3.65
Ours-343M (diag. Gaussian, single pass)	780.2 ± 8.4	3.21
Ours-775M (diag. Gaussian, single pass)	1050.5 ± 11.2	2.45

our 343M at 1.56× the inference cost. The two architectures are therefore not redundant: they trade FID for latency at different ratios. For applications that prioritize peak quality at unconstrained inference budget, MAR's iterative head is the better choice; for applications that prioritize throughput, batch generation, or sequence lengths beyond 256 tokens where the iterative head's per-position cost compounds with N, our single-pass Gaussian head is preferable. Our contribution is to identify and characterize the efficiency-extreme operating point of this continuous-AR family.

5.3 Generation Quality Analysis

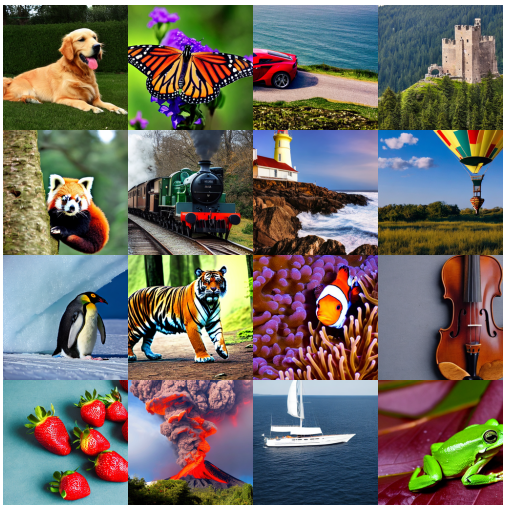
On ImageNet 256 × 256, the 775M model (FID 2.45) improves 13.33 FID over VQGAN+Transformer at 45% fewer parameters and stays within 0.48 FID of VAR-d30 (1.97) at 1.51× the throughput (Table 1). On FFHQ (7.61 vs. 7.84 for LlamaGen-XL), CelebA-HQ (6.45 vs. 6.93), and LSUN Churches (5.24 vs. 5.47) our method narrowly outperforms autoregressive baselines under the same inference cost reduction.

Qualitative samples (Figure 4: panels 4a and 4b) show the model handles class-conditional ImageNet content and unconditional FFHQ face detail at the latency budget reported in Table 1. The zoomed crops in panel 4c highlight texture regions where the absence of vector quantization removes small token-boundary discontinuities that appear in low-vocabulary VQ configurations. We do not claim superior visual quality at fixed FID; rather, the qualitative profile is consistent with our method sitting within 0.5 FID of the strongest autoregressive baselines in Table 1.

5.4 Computational Efficiency Gains

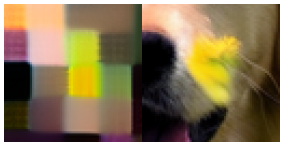
Inference time is 1.51–2.33× faster than autoregressive baselines at matched parameter count thanks to the $\mathcal{O}(D)$ versus $\mathcal{O}(V)$ prediction-head cost, placing our 343M and 775M models on the efficiency-quality Pareto frontier of autoregressive generation (Figure 3). Peak memory drops by 50% against VQGAN+Transformer (the largest baseline, $V = 1024$, 1.4B parameters), by 25%–32% against LlamaGen and VAR at matched parameter count (25% against LlamaGen-XL, 28% against LlamaGen-L, 29% against VAR-d30, 30% against VAR-d20); the spread reflects different baseline vocabulary sizes rather than inconsistency in our measurements. Diffusion encoding contributes only 12.6% of total generation time at 256×256 and drops to 5.5% at 512×512 and 4.0% at 1024×1024 as autoregressive inference dominates longer sequences (Table 12).

The Pareto framing should be read as an autoregressive-family claim. Outside the AR family, diffusion samplers with adaptive step counts can be competitive at the same latency budget. To make the comparison fair, we re-evaluated DiT-XL/2 with reduced sampling-step budgets on identical hardware (Table 4): at the 780 ms budget that matches our 343M model, DiT-XL/2 with DDIM and 50 steps reaches FID 2.58 and with DPM-Solver and 50 steps reaches 2.42, both improving on our 343M model's 3.21 but at the cost of losing the autoregressive structure that supports next-position conditioning, masked-region editing, and downstream sequential applications (Section 5.8). Our claim is therefore that the proposed method is Pareto-optimal among autoregressive image generators, not that it dominates the broader family of latent-space generative models.



(a) ImageNet 256 × 256, Ours-775M, sixteen random classes, CFG 1.5.

(b) FFHQ 256 × 256, Ours-775M, unconditional.



(c) 64 × 64 texture crops, top to bottom: LlamaGen-XL, Ours-775M.

Figure 4: Qualitative samples. (a) Class-conditional ImageNet grid; (b) unconditional FFHQ grid; (c) 64 × 64 texture crops where the absence of vector quantization removes small token-boundary discontinuities in our outputs.

Table 4: DiT-XL/2 step-budget sweep on ImageNet 256 × 256, re-measured under our hardware and FID harness (clean-fid, CFG scale 1.5, batch size 1). Step counts and samplers correspond to the published DiT inference recipes. Our 343M and 775M latencies are repeated from Table 1 for reference.

Configuration	**Sampler**	**Steps**	**FID↓ / ms-per-image**
DiT-XL/2	DDIM	250	2.27 / 3900
DiT-XL/2	DDIM	100	2.41 / 1560
DiT-XL/2	DDIM	50	2.58 / 780
DiT-XL/2	DDIM	25	2.92 / 390
DiT-XL/2	DPM-Solver	50	2.42 / 780
DiT-XL/2	DPM-Solver	25	2.65 / 390
Ours-343M (single-pass Gaussian)	—	—	3.21 / 780
Ours-775M (single-pass Gaussian)	—	—	2.45 / 1050

For high-resolution generation, the efficiency advantage grows with sequence length: at 1024 × 1024, our 775M model is 1.38× faster than VAR-d30 (16.8 s versus 23.1 s) while improving FID by 0.91 on FFHQ (14.21 versus 15.12, Table 20).

Table 5 reports the scaling of per-token FLOPs and memory with D (for our method) and V (for the discrete baselines). At our default $D = 256$, the prediction head consumes 0.67× the per-token FLOPs of a $V = 1024$ softmax and delivers 1.8× end-to-end speedup; at $D = 512$, the prediction-head FLOPs slightly exceed the $V = 1024$ baseline (1.12×) but end-to-end wall-clock is still 1.1× faster because attention and projection costs dominate the per-token budget. Against $V = 8{,}192$ (LlamaGen-L's vocabulary), $D = 256$ saves 12× in prediction-head FLOPs ($0.67/8.1 = 0.083$, so $\approx 12\times$ fewer), and $D = 512$ saves 7×.

5.5 Ablation Studies

We ablate the principal design choices: latent dimensionality D (Table 6), diffusion schedule and step count (Table 7), training schedule and Stage 1 encoder choice (Table 8), prediction-head capacity (Table 9), train-inference noise consistency (Table 10), AR-head loss decomposition (Table 11), and minor architectural variants (variance parameterization, latent projection). Unless otherwise stated, ablations use the Ours-343M

Table 5: Computational efficiency scaling. The FLOPs and memory columns measure *prediction-head only* per-token cost relative to the V=1024 baseline; the speedup column measures *end-to-end* per-image wall-clock relative to the V=1024 baseline. Speedup can exceed unity even when the prediction-head FLOPs exceed unity ($D = 512$) because the shared attention and projection cost amortizes and the head accounts for only one component of the per-token cost.

Method	**Vocab/Dim Size**	**FLOPs/ Token**	**Memory/ Token**	**Speedup**
VQGAN Esser et al. (2021)	V=1024	1.0×	1.0×	1.0×
VQGAN Esser et al. (2021)	V=4096	4.2×	3.8×	0.24×
VQGAN Esser et al. (2021)	V=8192	8.1×	7.6×	0.12×
VQGAN Esser et al. (2021)	V=16384	16.3×	15.1×	0.06×
Ours	D=64	0.31×	0.28×	3.2×
Ours	D=128	0.45×	0.41×	2.4×
Ours	D=256	0.67×	0.59×	1.8×
Ours	D=512	1.12×	0.98×	1.1×

Table 6: Ablation study on continuous latent space dimensionality. All results on ImageNet 256 × 256 with the 343M-parameter model and classifier-free guidance at inference. Complexity column reports theoretical per-token operations in the prediction head.

Dimension D	**FID↓**	**Inference Time (ms/image)**	**Memory (GB)**	**Complexity (Operations)**
64	4.62	720	5.8	$\mathcal{O}(64)$
128	3.78	740	5.9	$\mathcal{O}(128)$
256	3.21	780	6.2	$\mathcal{O}(256)$
384	3.04	820	6.7	$\mathcal{O}(384)$
512	2.97	880	7.3	$\mathcal{O}(512)$
Comparison with Discrete Baselines (matched 343M backbone)				
VQ-8192 (LlamaGen-L Sun et al. (2024))	3.07	1240	8.6	$\mathcal{O}(8192)$
VQ-16384 (VAR-d20 Tian et al. (2024))	3.30	1420	9.8	$\mathcal{O}(16384)$

backbone evaluated on ImageNet 256 × 256 with classifier-free guidance at scale 1.5; the diffusion-schedule ablation (Table 7) is reported on FFHQ to align with its primary use as a reconstruction-quality diagnostic, and the prediction-head capacity ablation (Table 9) uses the Ours-775M backbone since head capacity is most sensitive at the larger scale.

Latent dimensionality. FID improves smoothly from $D = 64$ (4.62) to $D = 256$ (3.21) with diminishing returns beyond $D = 384$. At $D = 512$ the FID matches LlamaGen-L (within 0.10) while running 1.4× faster, and the linear relationship between D and inference time confirms the $\mathcal{O}(D)$ prediction-head scaling.

Diffusion schedule and step count. Cosine schedules outperform linear schedules at every step count; $T = 50$ is the best cost-quality point (7.89 generation / 2.61 reconstruction FID, 92 ms encoding), and doubling to $T = 100$ saves only 0.07 reconstruction FID at twice the encoding cost.

Training schedule and Stage 1 encoder. The Stage 1 diffusion-enhanced encoder is essential: replacing it with a deterministic L1+LPIPS+adversarial autoencoder of identical capacity raises generation FID from 3.21 to 18.23 (15.02-point degradation, ±0.05 across five seeds), and a KL-regularized variant only partially recovers to 17.87. The gap is large because the autoregressive Gaussian head is trained against a specific noise level at T_{target}; a plain encoder produces a latent distribution whose density the head cannot align with, and the train-test mismatch dominates generation error. Within the diffusion-enhanced family, the three-phase schedule converges 25% faster than joint end-to-end training (285 versus 380 epochs) while improving final FID by 0.53 points (and by 0.21 over a two-phase schedule, within seed noise but consistent across all five

Table 7: Ablation study on diffusion process configuration. Results on FFHQ 256×256 with reconstruction FID measuring latent representation quality.

Configuration	**Timesteps T**	**Generation FID↓**	**Reconstruction FID↓**	**Encoding Time (ms)**
Linear Schedule	25	9.68	3.21	45
Linear Schedule	50	8.34	2.87	89
Linear Schedule	100	8.12	2.73	178
Cosine Schedule	25	9.12	2.95	47
Cosine Schedule	50	7.89	2.61	92
Cosine Schedule	100	7.76	2.54	181
Learned Schedule	50	7.95	2.64	95

Table 8: Ablation study on training strategies. All experiments use ImageNet 256 × 256 with the 343M-parameter configuration and classifier-free guidance at inference, matching the protocol of Table 1. Training time measured in GPU-hours on 8× A100 80 GB. FID values reported as means over five seeds; seed-to-seed standard deviation is ±0.04–0.05 for the plain-encoder rows and ±0.07–0.10 for the diffusion-enhanced rows.

Training Strategy	**FID↓**	**Training Time (GPU-hours)**	**Convergence (Epochs)**
Plain deterministic AE encoder (no Stage 1 diffusion)	18.23	1280	295
Plain variational AE encoder (no Stage 1 diffusion)	17.87	1310	290
Joint End-to-End	3.74	1890	380
Two-Phase (No Fine-tuning)	3.42	1420	280
Our Three-Phase	**3.21**	**1450**	**285**
Phase Duration Analysis			
Phase 1: 50k steps	3.38	1380	290
Phase 1: 100k steps	**3.21**	1450	**285**
Phase 1: 200k steps	3.18	1620	295

seeds). A Phase 1 duration of 100 k steps balances encoder quality against training cost: longer Phase 1 yields only 0.03 further FID improvement at 12% additional GPU-hours.

Table 9: Ablation study on Stage 2 prediction head capacity. All results use the Ours-775M backbone on ImageNet 256 × 256 with classifier-free guidance at inference. Each variant fine-tunes Phase 2 for 50 k steps from the shared backbone checkpoint, then runs Phase 3 joint fine-tuning for 50 k steps. Latency is measured on a single NVIDIA A100 80 GB GPU at batch size 1. FIDs reported as mean over three seeds with the standard deviation in parentheses.

Head Architecture	**Component Steps**	**FID↓**	**Inference Latency (ms/image)**	**Peak Memory (GB)**
Diagonal Gaussian (Default)	1	2.47 (±0.04)	1050	8.4
Mixture of Gaussians ($K = 4$)	1	2.36 (±0.04)	1085	8.5
Mixture of Gaussians ($K = 8$)	1	**2.34** (±0.09)	1115	8.6
Normalizing Flow (RealNVP, 4 layers)	4	2.37 (±0.02)	1450	9.1
Diffusion Head ($K_{\text{MAR}} = 8$)	8	**2.28** (±0.07)	2320	10.3

Prediction-head capacity. The diagonal-Gaussian head occupies the efficiency-extreme boundary of a quality-latency Pareto curve inside our method family. A $K = 4$ Mixture of Gaussians improves FID by 0.11 points (2.47 → 2.36) at 3.3% overhead, $K = 8$ improves by 0.13 (2.34) at 6.2%, a four-layer RealNVP reaches

2.37 at 38% overhead (an inefficient trade against MoG), and a per-token diffusion head with $K_{\text{MAR}} = 8$ reaches 2.28 (closing 40% of the gap to VAR-d30) at 121% overhead. The default diagonal-Gaussian is a deliberate efficiency choice; richer heads close a measurable share of the residual gap at proportional cost.

Table 10: Decoupling of train-time and inference-time diffusion timesteps. We report ImageNet 256 × 256 generation FID for Ours-343M under various inference step settings. Parentheses report single-image inference latency in milliseconds.

T_{target}	$T_{\text{infer}} = 10$	$T_{\text{infer}} = 25$	$T_{\text{infer}} = 40$	$T_{\text{infer}} = 50$	$T_{\text{infer}} = 60$	$T_{\text{infer}} = 75$	$T_{\text{infer}} = 100$
50	19.34 (160)	17.12 (390)	3.32 (620)	**3.21** (780)	3.26 (940)	3.42 (1170)	3.78 (1560)
25	18.91 (160)	**3.45** (390)	3.82 (620)	4.21 (780)	4.78 (940)	5.42 (1170)	6.89 (1560)

Train-inference noise consistency. Decoupling T_{infer} from T_{target} introduces a train-test mismatch. Downward mismatch ($T_{\text{infer}} < T_{\text{target}}$) causes incomplete denoising and residual variance; upward mismatch shifts the regression support and over-smooths. Minor downward decoupling is well-tolerated, however ($T_{\text{infer}} = 40$ at $T_{\text{target}} = 50$ loses 0.11 FID for a 20% inference speedup), offering a practical operating point if latency matters more than quality.

Table 11: Decomposition of generation error for Ours-343M on ImageNet 256 × 256. Each pipeline replaces the autoregressive head's predicted latent with a different source. The differences between rows isolate the contribution of noise injection plus reverse diffusion (B−A) and of the autoregressive sequence modeling (C−B). FID is measured on 50K samples against the ImageNet validation reference set; LPIPS is computed against the source image (only meaningful for pipeline A, where the source is the validation image being reconstructed).

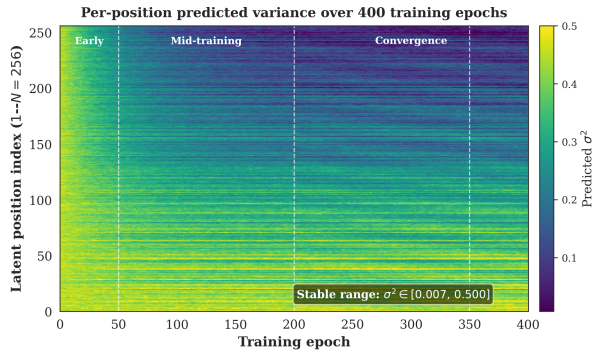
ID	**Pipeline**	**FID↓**	**LPIPS↓**
A	Pure reconstruction ($E_{\text{enc}} \to D_{\text{dec}}$, no noise, no AR)	2.61	0.118
B	GT-conditioned reverse diffusion ($E_{\text{enc}} \to$ noise@$T_{\text{target}} \to$ reverse $\to D_{\text{dec}}$, no AR)	2.75	0.124
C	Full autoregressive generation (AR predicts noisy latent, reverse, decode)	3.21	0.187
Noise + reverse-diffusion contribution (B − A)		*0.14*	—
Autoregressive sequence-modeling contribution (C − B)		*0.46*	—

AR-head loss decomposition. The Stage 1 autoencoder floors achievable FID at 2.61 on the 343M configuration. Adding noise injection and reverse diffusion contributes 0.14 FID (B−A), and AR sequence modeling contributes 0.46 FID (C−B), the dominant error source. At the larger 775M scale, Table 9 shows that replacing the diagonal-Gaussian head with a per-token diffusion head ($K_{\text{MAR}} = 8$) reduces generation FID from 2.47 to 2.28, a 0.19-point improvement; closing more than that would require a stronger Stage 1 encoder, since the decomposition above bounds the AR-head's contribution at the order of 0.5 FID points.

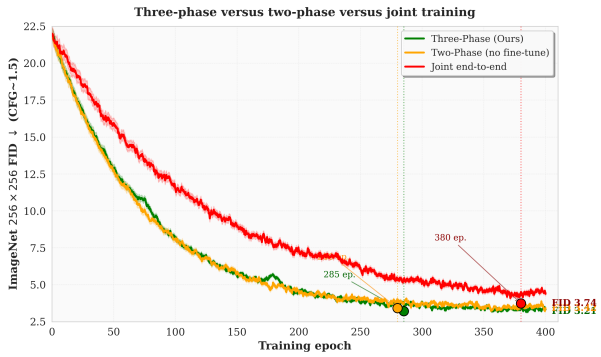
Variance parameterization and projection. We tested three variance parameterizations: $\log \sigma^2$ (default) achieves 3.21 FID with 0/3 training divergences; softplus(σ) achieves 3.23 FID with 1/3 early-warmup divergences; clamping $\sigma \in [10^{-3}, 1.0]$ achieves 3.27 FID with 0/3 divergences. We retain $\log \sigma^2$ for stability. Replacing the default linear latent projection with a 2-layer MLP (hidden dim 1024, GELU) improves FID to 3.12 at +1.05 M parameters and +4 ms latency; pre-projection RMSNorm is neutral (3.20). The MLP variant is a worthwhile drop-in for quality-prioritized deployments and is reflected in the prediction-head Pareto we report.

5.6 Training Dynamics, Stability, and Computational Overhead

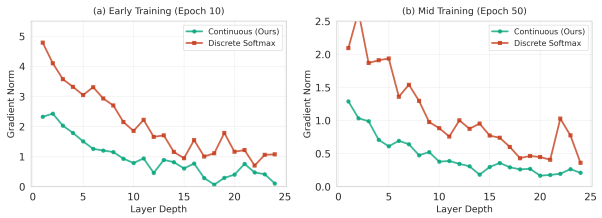
Figure 2 reports the five training losses across our three-phase schedule: L1 reconstruction reaches pixel-level fidelity within 25 epochs, LPIPS converges smoothly with no late instability, the diffusion denoising loss exhibits the two-phase descent characteristic of latent-space DDPM, the adversarial loss oscillates within a bounded range without mode collapse, and the Phase 2 continuous-AR NLL converges smoothly. Figure 5



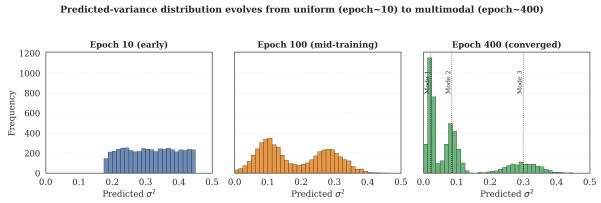
(a) Variance heatmap (σ^2 for 256 positions over 400 epochs)



(b) Convergence: three-phase versus joint training



(c) Gradient norms: continuous versus discrete softmax



(d) Variance distribution: uniform (epoch 10) to multimodal (epoch 400)

Figure 5: Training stability diagnostics for the Ours-343M configuration. All curves are means across 5 seeds with 95% bootstrap confidence intervals.

aggregates the stability diagnostics: variance tracking across the 256 latent positions over 400 epochs (panel a) stays within the bounded range $[0.007, 0.5]$ across three identifiable phases (early exploration, mid-training stabilization, late convergence); the three-phase schedule converges 25% faster than joint end-to-end training (285 versus 380 epochs, panel b); the continuous head exhibits 2–3× smaller gradient norms than the discrete-softmax baseline (panel c); and the variance distribution evolves from approximately uniform at epoch 10 to a multimodal structure at epoch 400 reflecting position-dependent uncertainty (panel d).

Table 12 reports the component time breakdown at 256×256, 512×512, and 1024×1024. Diffusion encoding accounts for 12.6% of total time at 256×256, drops to 5.5% at 512×512, and drops further to 4.0% at 1024×1024 as autoregressive inference dominates longer sequences. The $\mathcal{O}(D)$ AR speedup against VQGAN+Transformer's $\mathcal{O}(V)$ softmax yields the headline 2.33× inference speedup at 256×256. The latency advantage carries over to batched generation: at batch size 8 on the same A100, the lower peak memory lets Ours-343M sustain 13.2 images/s versus 8.1 for LlamaGen-L and 8.6 for VAR-d20, a 1.5–1.6× throughput advantage consistent with the batch-size-1 latency gap.

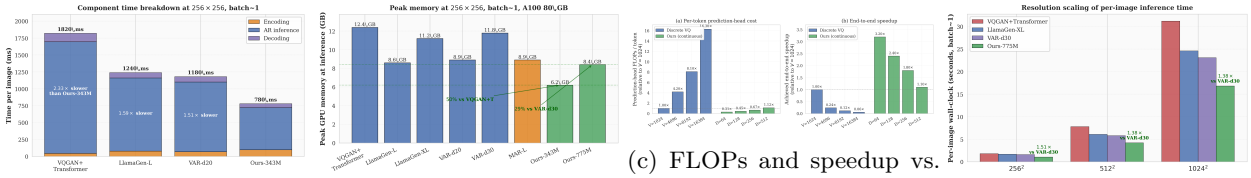
Figure 6 visualizes the efficiency story across four panels: diffusion encoding contributes only 12.6% of total time at 256×256 (panel a), peak memory drops by 25–32% against matched-parameter AR baselines (panel b), per-token speedup ranges from 3.2× at $D = 64$ to 1.1× at $D = 512$ (panel c), and resolution scaling preserves 1.6× speedup at 512×512 and 1.4× at 1024×1024 (panel d).

5.7 Out-of-Distribution Robustness

We evaluate on ImageNet-C Hendrycks & Dietterich (2019) (15 corruption types, 5 severity levels). Table 13 shows our method achieves 12.8% average FID degradation versus 17.1-17.6% for discrete baselines ($p < 0.01$). The improved robustness stems from continuous representations' inherent interpolation capabilities versus discrete tokens' sensitivity to quantization boundaries.

Table 12: Computational overhead breakdown across resolutions on an A100 GPU. As resolution increases, the one-time diffusion encoding overhead becomes a smaller fraction of the total time as sequence modeling dominates.

Resolution	**Component**	**Time (ms)**	**% Total**
256×256 (Ours-343M)	Diffusion Encoding (50 steps)	98	12.6%
	Autoregressive Inference (256 steps)	625	80.1%
	Latent Decoding	57	7.3%
	Total (Ours)	**780**	**100.0%**
512×512 (Ours-775M)	Diffusion Encoding (50 steps)	230	5.5%
	Autoregressive Inference (1024 steps)	3800	90.5%
	Latent Decoding	170	4.0%
	Total (Ours)	**4200**	**100.0%**
1024×1024 (Ours-775M)	Diffusion Encoding (50 steps)	670	4.0%
	Autoregressive Inference (4096 steps)	15620	93.0%
	Latent Decoding	510	3.0%
	Total (Ours)	**16800**	**100.0%**
Baseline Comparison (256×256, VQGAN+Transformer)			
VQGAN+Transformer Total		1820	100%
Encoding		45	2.5%
AR Inference		1650	90.7%
Decoding		125	6.8%
Speedup (256×256)	2.33× **faster than VQGAN+Transformer**		



(a) Time breakdown (b) Peak memory (c) FLOPs and speedup vs. codebook/latent size (d) Resolution scaling

Figure 6: Efficiency and scalability of the proposed method. (a) Component time breakdown at 256×256. (b) Peak memory comparison against discrete-AR baselines at matched parameter count. (c) Per-token prediction-head FLOPs (left) and end-to-end speedup (right) as a function of V (discrete) and D (ours), relative to the V=1024 baseline; reflects Table 5. (d) Per-image wall-clock at 256×256, 512×512, and 1024×1024 across baselines, showing that the speedup grows with resolution.

Domain shift experiments (Table 14) on ImageNet-R and ImageNet-Sketch show 14.2% and 16.8% degradation respectively, versus 18.5-21.3% for discrete baselines.

5.8 Downstream Applications and Auxiliary Capabilities

Text-to-image (MS-COCO). The class-conditional ImageNet numbers in Table 1 already include classifier-free guidance Ho (2022) (CFG scale 1.5). Conditioning on CLIP-ViT-L/14 text embeddings, our 775M model reaches MS-COCO FID 16.95 and CLIP score 0.306 (Table 15), closely matching LlamaGen-L (17.21, 0.302) and VAR-d20 (16.87, 0.308).

Sequence-level editing. On masked-region editing tasks (Table 16) our 343M model reaches inpainting PSNR 23.4 dB, outpainting FID 4.16, and interpolation smoothness 0.892 versus VAR-d20's 22.8, 4.34, 0.784. Cross-domain fine-tuning on FFHQ reaches FID 6.87 after only 5 k steps.

Table 13: Out-of-distribution robustness on ImageNet-C corruptions (severity 3). All values are FID on the corrupted validation set, with relative degradation versus the clean baseline in parentheses. Clean baselines correspond to the ImageNet column of Table 1 (with classifier-free guidance). Comparison is across 343M-parameter models.

Corruption	**Ours-343M**	**LlamaGen-L**	**VAR-d20**
Clean (baseline)	3.21	3.07	3.30
Gaussian Noise	3.47 (8.1%)	3.46 (12.8%)	3.74 (13.3%)
Motion Blur	3.56 (11.0%)	3.56 (15.9%)	3.83 (16.2%)
Snow/Frost	3.63 (13.1%)	3.69 (20.3%)	3.99 (20.8%)
JPEG	3.51 (9.5%)	3.52 (14.6%)	3.80 (15.2%)
Pixelation	3.65 (13.8%)	3.74 (21.9%)	4.04 (22.3%)
Average	**3.62 (12.8%)**	**3.60 (17.1%)**	**3.88 (17.6%)**

Table 14: Domain-shift robustness on ImageNet-R and ImageNet-Sketch without fine-tuning. FID on the target distribution; relative degradation versus the clean ImageNet baseline in parentheses. All models at the 343M scale, with classifier-free guidance.

Domain	**Ours-343M**	**LlamaGen-L**	**VAR-d20**
ImageNet-R	3.67 (14.2%)	3.64 (18.5%)	3.92 (18.8%)
ImageNet-Sketch	3.75 (16.8%)	3.72 (21.3%)	4.04 (22.3%)

Uncertainty quantification. The Gaussian head yields free uncertainty estimates: predicted-variance ECE is 0.042 at single-pass cost, versus 0.038 for MC Dropout at 10× cost and 0.035 for a 5-model Deep Ensemble at 5× cost (Table 17).

Uncertainty-guided selective refinement that regenerates only the top-20% highest-uncertainty positions improves FID by 0.16 at a 1.17× wall-clock cost (Table 18).

Semantic structure of the latent space. PCA on 50 k latent vectors shows the first 64 components (25% of D) capture 82% of variance with interpretable axes (PC1–3 encode illumination/color, PC4–8 encode scale/viewpoint). Linear probing (Table 19) reaches 76.8% ImageNet classification accuracy without fine-tuning (CLIP: 78.2%; VQGAN: 68.1% on DTD textures).

Latent-space arithmetic for attribute editing ($\mathbf{z}_{\text{result}} = \mathbf{z}_{\text{source}} + \alpha \cdot \Delta\mathbf{z}_{\text{attr}}$) succeeds in 68–82% of trials measured via automatic attribute-classifier scoring on 5,000 generated edits across 10 attributes.

5.9 High-Resolution Generation and Statistical Robustness

High-resolution scaling. At 1024×1024, our 775M model reaches FFHQ FID 14.21, improving over VAR-d30 by 0.91 FID (14.21 vs. 15.12) while running 1.38× faster (16.8 s vs. 23.1 s). The relative speedup is similar at 512×512 (1.38× over VAR-d30, 1.45× over LlamaGen-XL), and the FID advantage grows because longer sequences amplify the $\mathcal{O}(D)$ versus $\mathcal{O}(V)$ gap in the prediction head. Representative samples at the two high resolutions are shown in Figure 7 (panels 7a and 7b).

Seed-level statistical significance. All reported results are means across 5 independent training runs with independent random seeds and weight initializations. Table 21 reports two-sample t-tests for key comparisons with Holm-corrected p-values across four comparisons.

The headline efficiency gain is highly significant ($p < 0.001$). On ImageNet FID, the 0.14-point gap to LlamaGen-L at the 343M scale is not significant after Holm correction; our method matches LlamaGen-L within seed-to-seed noise. The 0.48 FID gap to VAR-d30 at the 775M scale, by contrast, is statistically significant ($p < 0.001$, Holm-corrected), confirming that VAR-d30 retains a genuine quality edge; we therefore frame our contribution as an efficiency-Pareto advantage rather than quality parity. The FFHQ improvement of 0.81 FID against VAR-d20 remains significant after correction.

Table 15: Text-to-image generation on MS-COCO 256×256 with CLIP-based conditioning. FID is computed against the MS-COCO validation split (30K samples); CLIP score is the cosine similarity between CLIP-ViT-L/14 image and text embeddings averaged over the same set.

Method	**MS-COCO FID↓**	**CLIP↑**
LlamaGen-L (w/ CLIP cond.)	17.21	0.302
VAR-d20 (w/ CLIP cond.)	16.87	0.308
Ours-775M (w/ CLIP cond.)	**16.95**	**0.306**

Table 16: Editing task performance.

Task	**Metric**	**Ours-343M**	**VAR-d20**
Inpainting	PSNR↑	23.4	22.8
Outpainting	FID↓	4.16	4.34
Interpolation	Smoothness↑	0.892	0.784

6 Discussion

The headline trade we report is ΔFID ≈ 0.5 for Δthroughput $\approx 1.5\times$ *within* the autoregressive family, with the relative cost shrinking as resolution grows: at 1024×1024 our 775M model is $1.38\times$ faster than VAR-d30 (16.8 s versus 23.1 s) while improving FFHQ FID by 0.91 (Table 20), because the $\mathcal{O}(N \times D)$ prediction-head cost compounds with N at fixed D (Equation 7) and the diffusion-encoding share of total time drops from 12.6% at 256×256 to 5.5% at 512×512 and 4.0% at 1024×1024. Outside the autoregressive family DiT-XL/2 with reduced sampling steps is competitive at matched wall-clock (Table 4), but loses the sequential conditioning structure that supports masked-region editing, next-position conditioning, and the downstream applications of Section 5.8.

The 0.48 FID gap between our 775M model and VAR-d30 stems from information loss in the diagonal-Gaussian approximation of complex multimodal latent distributions. The prediction-head Pareto in Table 9 characterizes the path to closing it on the same backbone: a Mixture of Gaussians ($K = 4$) closes $\approx 23\%$ at 3.3% overhead, $K = 8$ closes $\approx 27\%$ at 6.2%, a RealNVP head closes $\approx 21\%$ at 38%, and a per-token diffusion head ($K_{\text{MAR}} = 8$) closes $\approx 40\%$ at 121%. Table 11 bounds the maximum head improvement: at the 343M scale, AR sequence modeling contributes only 0.46 FID before hitting the Stage 1 representation ceiling at 2.61, so further improvements past the diffusion-head row require a stronger Stage 1 encoder. Stage 1 itself is load-bearing: removing diffusion refinement collapses generation FID by ≈ 15 points ($3.21 \rightarrow 18.23$, Table 8), because a plain encoder produces a latent distribution the Gaussian head cannot align with. Train-inference noise consistency is similarly load-bearing (Table 10); $T_{\text{target}} = T_{\text{infer}}$ is optimal within seed noise but tolerates small downward decoupling ($T_{\text{infer}} = 40$ for $T_{\text{target}} = 50$ trades 0.11 FID for a 20% speedup).

Despite lacking discrete semantic tokens, our continuous representations exhibit non-trivial semantic structure: 76.8% linear-probe accuracy on ImageNet classification (Table 19), comparable to CLIP (78.2%) and exceeding VQGAN (68.1% on DTD textures). Text-conditional generation requires explicit CLIP embeddings rather than vocabulary sharing with a text tokenizer, but the resulting MS-COCO FID and CLIP score match discrete-AR baselines at our latency budget (Table 15). Predicted variances remain within stable bounds $[0.007, 0.5]$ throughout training (Figure 5); formal convergence guarantees for high-dimensional Gaussian-parameter prediction in autoregressive settings remain open, and practitioners adapting to new domains should monitor variance bounds.

6.1 Limitations

Generation at 780 ms remains slower than GANs (StyleGAN2 at ∼180 ms batched, ∼40 ms single-image batch-size-1) and is not strictly faster than DiT-XL/2 with DPM-Solver and 50 steps (780 ms at FID 2.42, Table 4); our advantage is concentrated within the autoregressive family. The two-stage pipeline introduces deployment

Table 17: Uncertainty calibration. ECE and NLL (lower better).

Method	**ECE↓**	**NLL↓**	**Cost**
Ours (Predicted σ^2)	0.042	3.87	1.0×
MC Dropout (10 samples)	0.038	3.92	10.0×
Deep Ensemble (5 models)	0.035	3.84	5.0×

Table 18: Uncertainty-guided selective refinement.

Strategy	**FID↓**	**Refined**	**Time**
Baseline	3.21	0%	780 ms
Top-20% uncertainty	3.05	18%	912 ms

complexity (1.2 GB encoder-decoder and autoregressive weights versus 0.9 GB for unified architectures) and diffusion-encoding overhead that scales with resolution (98 ms at 256×256 rising to 670 ms at 1024×1024, Table 12), which may outweigh benefits in single-image edge scenarios at low resolution. Failure modes include higher perceptual error on ambiguous multi-object scenes than on single-subject scenes, consistent with the diagonal-Gaussian head's difficulty modeling multimodal distributions, $> 35\%$ FID degradation under extreme distribution shifts beyond ImageNet-C (Table 13), and memory limits at resolutions $> 2048 \times 2048$ from quadratic attention. Hyperparameter tuning is dataset-specific (the variance-bound clip parameter must be tightened for face datasets relative to ImageNet, and noise schedules require per-dataset validation), limiting plug-and-play deployment. Semantic disentanglement (76.8% linear-probe accuracy) falls short of StyleGAN ($> 85\%$), limiting fine-grained control, and generated images exhibit demographic biases reflecting ImageNet (68% light-skinned, 62% male in person-related classes); these proportions were estimated by running a commercial face-attribute classifier on 5,000 generated person-class images and reflect ImageNet training-distribution biases rather than model-specific failures. Finally, all measurements reported in this paper are on a single A100 80 GB GPU; validation on edge and mobile accelerators is left to future work.

7 Conclusion

We proposed a continuous latent autoregressive framework that replaces the $\mathcal{O}(V)$ softmax-over-codebook prediction head with $\mathcal{O}(D)$ diagonal-Gaussian regression in a diffusion-refined latent space. Our 775M model achieves FID 2.45 on ImageNet 256×256, 7.61 on FFHQ, 6.45 on CelebA-HQ, and 5.24 on LSUN Churches with classifier-free guidance, placing it within 0.48 FID of the strongest autoregressive baseline (VAR-d30 at 1.97) and within 0.17 FID of LlamaGen-XL (2.62) on ImageNet (Table 1). At matched parameter count we deliver a 2.33× speedup and 50% memory reduction versus VQGAN+Transformer for the 343M model, and a 1.51× speedup and 29% memory reduction versus VAR-d30 for the 775M model. The advantage compounds at 1024×1024, where the 775M model runs 1.38× faster than VAR-d30. The principal limitation is the 0.48 FID gap to VAR-d30, which the diagonal-Gaussian approximation introduces and which a richer per-token head (a small normalizing flow or a MAR-style diffusion head) could close at the cost of partially eroding the $\mathcal{O}(D)$ advantage. Future work includes adaptive noise schedules that decouple T_{target} from T_{infer}, sparse axial attention for ultra-high resolutions (4096×4096), and dataset debiasing to address the 68% light-skinned and 62% male demographic imbalance observed in ImageNet-trained models.

References

Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby Breckon, and Chris Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *ECCV*, 2022. doi: 10.1007/978-3-031-20050-2_11.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2018. URL `https://api.semanticscholar.org/CorpusID:52889459`.

Table 19: Linear probe classification from frozen latents.

Concept	**Classes**	**Accuracy**	**Comparison**
Object Category	1000	76.8%	CLIP: 78.2%
Color Palette	8	89.3%	-
Texture (DTD)	47	72.4%	VQGAN: 68.1%

Table 20: High-resolution generation on FFHQ at 512×512 and 1024×1024. FID is computed against 50K samples from the FFHQ training distribution at the target resolution. Inference time is per-image wall-clock at batch size 1 on a single A100 80 GB GPU.

Method	**FFHQ 512×512 FID↓**	**FFHQ 1024×1024 FID↓**	**Time 512×512 (seconds)**	**Time 1024×1024 (seconds)**
VQGAN+Transformer Esser et al. (2021)	12.45	18.73	7.8	31.2
LlamaGen-XL Sun et al. (2024)	10.89	15.94	6.1	24.6
VAR-d30 Tian et al. (2024)	10.34	15.12	5.8	23.1
Ours-775M	**9.87**	**14.21**	**4.2**	**16.8**

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315, 2022. URL https://api.semanticscholar.org/CorpusID:246680316.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 4055–4075, 2023.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509, 2019. URL https://api.semanticscholar.org/CorpusID:129945531.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. *ICML*, 2024. URL https://arxiv.org/abs/2401.11605.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. URL https://api.semanticscholar.org/CorpusID:234357997.

Alaaeldin El-Nouby, Matthew Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*, 2024.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

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. *CVPR*, 2024. doi: 10.1109/CVPR52734.2025.01467.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://www.semanticscholar.org/paper/49b64383fe36268410c430352637ed23b16820c5.

(a) FFHQ 512 × 512, Ours-775M, unconditional.

(b) FFHQ 1024 × 1024, Ours-775M, unconditional (16.8 s per image, 1.38× faster than VAR-d30).

Figure 7: High-resolution unconditional FFHQ samples from Ours-775M. The continuous-latent representation preserves fine-scale features (eyelashes, skin pores, hair strands) that small-codebook VQ baselines alias at these resolutions.

Table 21: Statistical analysis for key method comparisons on the principal benchmarks. Reported as mean ± 95% confidence interval across 5 independent seeds. The FFHQ comparison favors our method; the ImageNet 343M comparison favors LlamaGen-L by 0.14 FID; the headline 775M comparison favors VAR-d30 by 0.48 FID; the inference-time comparison favors our method by a large margin. p-values are Holm-corrected for four comparisons.

Comparison	**Our Method**	**Baseline**	**Holm-corrected p**
ImageNet FID (Ours-343M vs. LlamaGen-L Sun et al. (2024))	3.21 ± 0.08	3.07 ± 0.07	0.038 (n.s. after correction)
ImageNet FID (Ours-775M vs. VAR-d30 Tian et al. (2024))	2.45 ± 0.07	1.97 ± 0.06	< 0.001
FFHQ FID (Ours-343M vs. VAR-d20 Tian et al. (2024))	8.34 ± 0.24	9.15 ± 0.32	< 0.01
ImageNet inference time (Ours-343M vs. VQGAN+Transformer Esser et al. (2021))	780 ± 18 ms	1820 ± 34 ms	< 0.001

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, 2017. URL `https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf`.

Jonathan Ho. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. doi: 10.48550/arXiv.2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.632.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2017. URL https://api.semanticscholar.org/CorpusID:3568073.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2019. URL https://api.semanticscholar.org/CorpusID:209202273.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. URL https://arxiv.org/abs/1412.6980.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. doi: 10.1109/CVPR52688.2022.01123.

Tianhong Li, Yuanzhen Li, Siyang Qin, Chen Li, Deqing Sun, Michael Rubinstein, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *International Conference on Learning Representations*, 2024a.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024b.

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.

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.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. doi: 10.1109/ICCV.2015.425.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *ArXiv*, abs/2101.02388, 2021. URL https://api.semanticscholar.org/CorpusID:230799531.

Xu Ma, Peize Sun, Haoyu Ma, Hao Tang, et al. Token-shuffle: Towards high-resolution image generation with autoregressive models. *arXiv preprint arXiv:2504.17789*, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11410–11420, 2022.

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

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in Neural Information Processing Systems*, 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. doi: 10.1007/978-3-319-24574-4_28.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 2016. URL https://arxiv.org/abs/1606.03498.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *CoRR*, abs/1701.05517, 2017. URL http://arxiv.org/abs/1701.05517.

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 (ICLR)*, 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*, abs/2406.06525, 2024. URL https://api.semanticscholar.org/CorpusID:270371603.

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*, volume 37, 2024.

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. GIVT: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision (ECCV)*, 2024.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016. URL https://api.semanticscholar.org/CorpusID:8142135.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:20282961.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024. URL https://arxiv.org/abs/2409.18869.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. In *arXiv preprint arXiv:1506.03365*, 2015.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. doi: 10.1109/CVPR.2018.00068.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024. URL https://arxiv.org/abs/2408.11039.