

Hyperspherical Latents Improve Continuous-Token Autoregressive Generation

Guolin Ke^{1*}, Hui Xue²

¹DP Technology, ²Peking University

Code: <https://github.com/guolinke/SphereAR>

Abstract

Autoregressive (AR) models are promising for image generation, yet continuous-token AR variants often trail latent diffusion and masked-generation models. The core issue is heterogeneous variance in VAE latents, which is amplified during AR decoding, especially under classifier-free guidance (CFG), and can cause variance collapse. We propose *SphereAR* to address this issue. Its core design is to constrain all AR inputs and outputs—including after CFG—to lie on a fixed-radius hypersphere (constant ℓ_2 norm), leveraging hyperspherical VAEs. Our theoretical analysis shows that hyperspherical constraint removes the scale component (the primary cause of variance collapse), thereby stabilizing AR decoding. Empirically, on ImageNet generation, *SphereAR-H* (943M) sets a new state of the art for AR models, achieving FID 1.34. Even at smaller scales, *SphereAR-L* (479M) reaches FID 1.54 and *SphereAR-B* (208M) reaches 1.92, matching or surpassing much larger baselines such as MAR-H (943M, 1.55) and VAR-d30 (2B, 1.92). To our knowledge, this is the first time a pure next-token AR image generator with raster order surpasses diffusion and masked-generation models at comparable parameter scales.

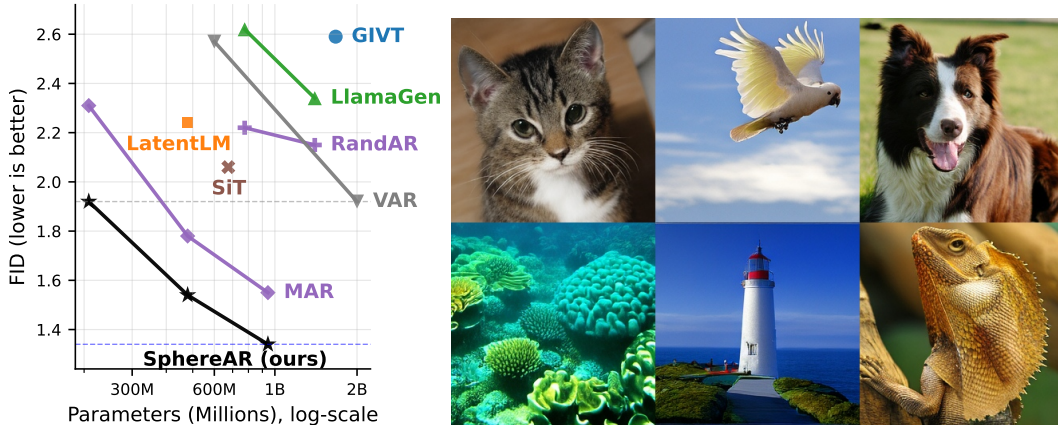


Figure 1: **Left:** FID vs. parameters on ImageNet 256×256 class-conditional generation, *SphereAR* attains lower FID with fewer parameters. **Right:** 256×256 samples generated by *SphereAR-L* (479M).

1 Introduction

Autoregressive (AR) models have achieved remarkable success in text [1, 2] and have been extended to images [3, 4], speech [5], video [6], and other modalities [7]. Early multimodal AR systems discretized latents with vector quantization (VQ) [8, 9]; more recently, *continuous*-token AR dispenses

*Corresponding author: kegl@dp.tech

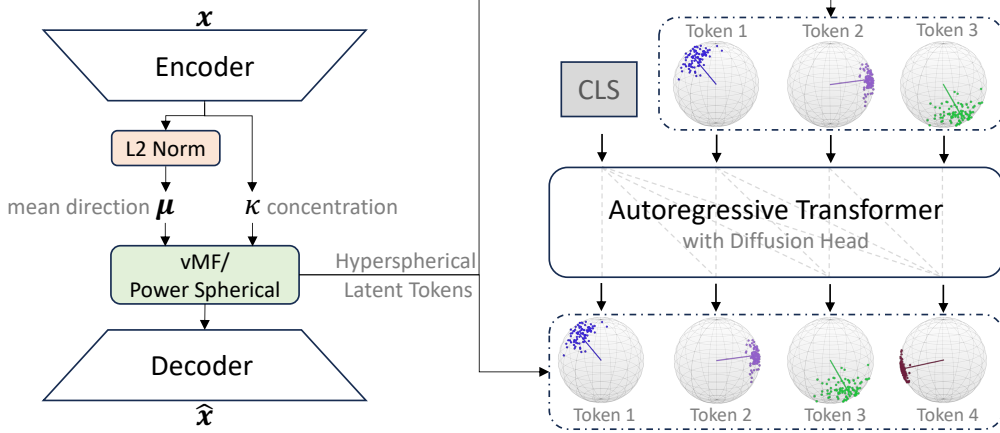


Figure 2: Overview of *SphereAR*. **Left:** A hyperspherical VAE (S-VAE) encodes raw data into a sequence of latent tokens constrained to a fixed-radius hypersphere $\mathbb{S}^{d-1}$. The encoder outputs a unit mean direction μ and a concentration κ that parameterize a von Mises–Fisher (vMF) or Power Spherical posterior. **Right:** A causal Transformer with a token-level diffusion head models the next-token distribution over the hyperspherical token sequence. At inference, the AR model’s predictions, including CFG-rescaled ones, are projected back onto the fixed-radius hypersphere. The VAE decoder then reconstructs the image from the predicted hyperspherical latents.

with codebooks: a VAE [10] emits token-level latents and the AR model predicts the next latent in continuous space (e.g., Gaussian mixtures [11] or diffusion objectives [12, 13]). Yet, when built on the same VAE latents, continuous-token AR models often trail latent diffusion and masked-generation models.² Prior analyses attribute this gap to variance pathologies during AR decoding [13, 17]: latent variances are heterogeneous across dimensions/tokens and are amplified due to exposure bias and classifier-free guidance (CFG) [18], causing stepwise variance drift and collapse. Strengthening the KL term [11] or fixing a large variance [13] improves stability but leaves the root cause intact: scale heterogeneity remains and can still drift during AR decoding with CFG.

We address this with a more principled solution: make all AR inputs and outputs *scale-invariant*. As illustrated in Fig. 2, the proposed *SphereAR* couples a hyperspherical VAE (S-VAE) [19, 20] with an autoregressive Transformer [21] and a token-level diffusion head [12]. The S-VAE constrains each latent token to a fixed-radius hypersphere (constant ℓ_2 norm), parameterizing only direction via a unit mean direction vector μ and a scalar concentration κ . During training, the AR model consumes these hyperspherical latents under teacher forcing. During inference, AR model’s predictions—including those after CFG rescaling—are projected back onto the fixed-radius hypersphere to remove the radial (scale) component. Thus, every signal provided to or produced by the AR model is ℓ_2 -normalized to the same radius. A concise theoretical justification supports these design choices, showing why scale-invariant inputs/outputs stabilize AR decoding and why a hyperspherical posterior is preferable to Gaussian alternatives.

Empirically, *SphereAR-H* (943M) sets a new state of the art for AR models on ImageNet 256×256 class-conditional generation, achieving FID 1.34. Even at smaller scales, *SphereAR-L* (479M) attains FID 1.54, outperforming comparably sized diffusion (DiT-XL/2, FID 2.27) and bidirectional masked-generation (MAR-L, FID 1.78) baselines, while matching MAR-H (943M, FID 1.55) with roughly half the parameters. At the base scale, *SphereAR-B* (208M) achieves FID 1.92, surpassing VAR-d20 (600M, FID 2.57) and the prior continuous-token AR model LatentLM-L (479M, FID 2.24), while matching VAR-d30 (2B, FID 1.92) with $\sim 10\times$ fewer parameters. Ablations show that AR models with hyperspherical VAEs consistently outperform diagonal-Gaussian and fixed-variance σ -VAE [13] baselines; moreover, applying post-hoc normalization to diagonal-Gaussian latents helps but still underperforms S-VAE. To our knowledge, this is the first time a pure next-token AR image generator with raster order surpasses diffusion and masked-generation models at comparable parameter scales.

²In this paper, “autoregressive” denotes token-by-token generation with unidirectional (causal) self-attention, excluding bidirectional masked/next-scale methods such as MaskGIT [14], MAR [12], and VAR [15]. With the same VAE latents, [12] reports an AR model at FID 4.69 vs. 1.98 for MAR and 2.27 for DiT [16].

2 Related Work

Image Tokenizers A large body of work improves the performance of image tokenizers by enhancing reconstruction fidelity and semantic alignment. Typical ingredients include (i) refined training objectives [3, 22, 23], (ii) CLIP-aligned distillation for better text guidance [24, 25], and (iii) various decoder improvements [26, 23]. These techniques are orthogonal to our approach and can be combined with it. In parallel, a complementary line of work targets the *quantization* mechanism itself—improving codebook utilization, training stability, and the rate-distortion tradeoff. Building on VQ-VAE [9], extensions include hierarchical VQ-VAE-2 [27], residual/hierarchical quantization [28], and multi-codebook schemes [29]. Some methods also adopt spherical or normalized feature geometry in the quantizer: for instance, ViT-VQGAN [4] normalizes latent features before computing codebook distances, and BSQ [30] constructs binarized *spherical* latents for bit-efficient quantization.

By contrast, comparatively less work targets *continuous* image tokenizers tailored to autoregressive modeling. Most prior approaches follow latent diffusion practice [16] and employ diagonal-Gaussian VAEs. GIVT [11] and LatentLM [13] mitigate instability by inflating or fixing latent variance (e.g., β -VAE, σ -VAE), which helps but does not remove scale degrees of freedom. NextStep-1 [17] instead normalizes Gaussian-posterior latents to a constant norm, achieving scale invariance. However, both our theoretical analysis and empirical results indicate that hyperspherical posteriors are preferable to post-hoc normalization of diagonal-Gaussian latents.

Autoregressive Image Generation Autoregressive image generation can be grouped into three families: *next-scale*, *next-set*, and *next-token* prediction. In *next-scale* prediction (e.g., VAR [15]), images are generated coarse-to-fine across resolutions; within each scale, context is modeled bidirectionally. In *next-set* prediction (also called masked generation; e.g., MaskGIT [14], MAR [12]), a single scale is used and a *set* of tokens is updated in parallel under bidirectional attention. In *next-token* prediction (e.g., VQGAN [3], LlamaGen [31]), the model follows language-style sequence modeling: one token is predicted at a time with strictly unidirectional (causal) attention. We focus on next-token models because they align naturally with autoregressive language modeling and offer headroom for unified multimodal models.

A wide range of next-token variants has been explored: discrete tokens [3, 4, 31] vs. continuous tokens [11, 13]; raster order [31, 11] vs. randomized order [32, 33]; and more [34]. However, at comparable parameter scales, these models have often trailed next-set and next-scale approaches. A key reason is the variance collapse that emerges during autoregressive decoding [13, 17]. We address this by enforcing *scale-invariant* latents via hyperspherical VAEs, thereby removing scale degrees of freedom. Empirically, this yields substantial gains for sequential AR decoding, with performance that matches or surpasses state-of-the-art next-set and next-scale methods at comparable model budgets.

3 Method

We observe that with *discrete* tokens, next-token autoregressive (AR) models can outperform bidirectional masked-generation (MG) approaches. For example, LlamaGen-L (343M, FID 3.07; [31]) vs. MaskGIT (207M, FID 4.02; [14]). A system-level study [35] further reports that, with discrete tokens, AR consistently achieves better FID than MG across model sizes from 166M to 3.1B. By contrast, with *continuous* tokens, MG is consistently stronger than AR. This divergence—discrete tokens thriving under AR while continuous tokens do not—motivates us to probe what truly differentiates the two. As illustrated in Fig.3, discrete tokens (Fig.3 a) are *normalized* on the probability simplex (components sum to 1), yielding *scale-invariant* inputs and outputs that stabilize AR decoding. In contrast, diagonal-Gaussian latents (Fig.3 b) are unconstrained and can destabilize multi-step AR decoding due to scale drift that compounds across steps. We hypothesize this scale sensitivity is the key issue, and therefore constrain continuous latents to a fixed-radius hypersphere to enforce a constant norm (Fig.3 c). This idea underpins *SphereAR*: a hyperspherical VAE paired with a causal Transformer equipped with a token-level diffusion head; we detail these components below.

3.1 From VAE to Hyperspherical VAE

A variational autoencoder (VAE) [10] is widely used to compress raw data into a lower-dimensional latent vector. It consists of an encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ that parameterizes an approximate posterior over $\mathbf{z}$

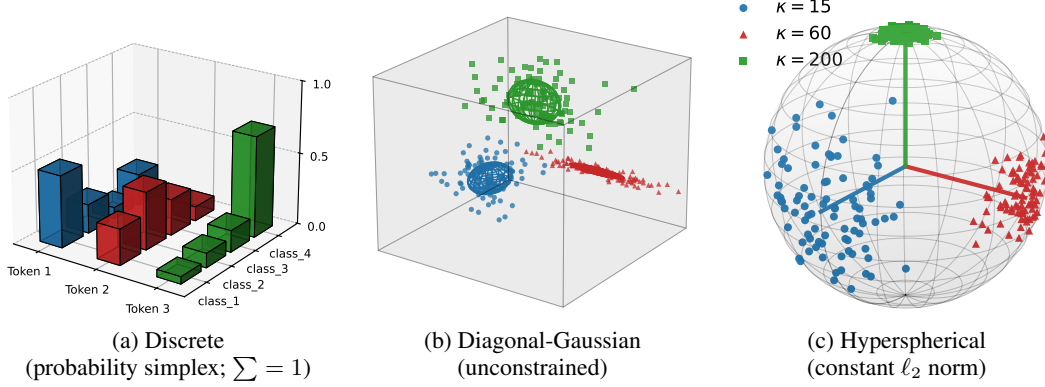


Figure 3: Visualization of token distributions. Each panel shows one token type, with three tokens in different colors. (a) Discrete tokens lie on the probability simplex and are intrinsically scale-invariant. (b) Diagonal-Gaussian latents are unconstrained in scale; despite a KL prior, per-dimension/token variances remain heterogeneous. (c) Hyperspherical latents constrain each token to a fixed norm (e.g., $\|\mathbf{z}\|_2 = R$), yielding scale-invariant representations. In practice, (a) and (c) are robust under AR decoding, whereas (b) is prone to scale drift and occasional variance collapse (e.g., with CFG).

and a decoder $p_\psi(\mathbf{x} | \mathbf{z})$ that reconstructs $\mathbf{x}$ from $\mathbf{z}$. We train the model by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\phi, \psi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\psi(\mathbf{x} | \mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})). \quad (1)$$

By default, both the prior $p(\mathbf{z})$ and the approximate posterior $q_\phi(\mathbf{z} | \mathbf{x})$ are parameterized as Gaussians with diagonal covariance; the prior is the isotropic standard Normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Using the reparameterization trick, $\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, makes the sampling step differentiable so that gradients backpropagate from the decoder to the encoder.

With this diagonal-Gaussian posterior, the encoder’s scale $\boldsymbol{\sigma}_\phi(\mathbf{x})$ is data-dependent and per-dimension, yielding *heterogeneous* variances across dimensions and tokens. This imbalance amplifies exposure bias and can trigger variance collapse in AR decoding, particularly under CFG [13, 17].

Hyperspherical VAE (S-VAE) To fully address this issue, we remove the *scale* degree of freedom in the latent representation, rendering the AR model’s inputs and outputs scale-invariant. Specifically, leveraging hyperspherical VAEs (S-VAEs) [19, 20], we constrain each latent token to lie on a fixed-radius hypersphere.

For each token, the S-VAE encoder parameterizes a *directional* posterior on the unit sphere by outputting a unit mean *direction* $\boldsymbol{\mu}_\phi(\mathbf{x}) \in \mathbb{S}^{d-1}$ (via ℓ_2 normalization; d is the latent dimension) and a nonnegative *concentration* $\kappa_\phi(\mathbf{x}) \in \mathbb{R}_{\geq 0}$. For notational convenience, let $\boldsymbol{\mu} = \boldsymbol{\mu}_\phi(\mathbf{x})$ and $\kappa = \kappa_\phi(\mathbf{x})$. S-VAE adopts a von Mises–Fisher (vMF) distribution [19] for the directional approximate posterior:

$$q_\phi(\mathbf{u} | \mathbf{x}) = C_d(\kappa) \exp(\kappa \boldsymbol{\mu}^\top \mathbf{u}), \quad \mathbf{u} \in \mathbb{S}^{d-1}, \quad (2)$$

where $C_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)}$ is the normalizing constant and $I_\nu(\cdot)$ is the modified Bessel function

of the first kind. Intuitively, $\boldsymbol{\mu}$ sets the preferred direction and κ controls concentration: $\kappa = 0$ gives the uniform distribution on $\mathbb{S}^{d-1}$, and larger κ yields tighter mass around $\boldsymbol{\mu}$. Because $\boldsymbol{\mu}^\top \mathbf{u}$ is the cosine similarity on the sphere, the density in equation 2 increases as $\mathbf{u}$ aligns with $\boldsymbol{\mu}$.

We take the prior over directions to be uniform on the sphere, $p(\mathbf{u}) = \text{Unif}(\mathbb{S}^{d-1})$, and use a fixed radius $R > 0$ (hyperparameter) so that $\mathbf{z} = R\mathbf{u}$ is fed to the decoder. The ELBO becomes

$$\mathcal{L}_{\text{S-VAE}}(\phi, \psi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{u} | \mathbf{x})} [\log p_\psi(\mathbf{x} | \mathbf{z} = R\mathbf{u})] - D_{\text{KL}}(q_\phi(\mathbf{u} | \mathbf{x}) \| p(\mathbf{u})). \quad (3)$$

While vMF is principled for spherical latents, it can be less efficient due to the need for rejection sampling. For efficiency, we adopt the *Power Spherical* posterior [20] on $\mathbb{S}^{d-1}$,

$$q_\phi(\mathbf{u} | \mathbf{x}) \propto (1 + \boldsymbol{\mu}^\top \mathbf{u})^\kappa, \quad \mathbf{u} \in \mathbb{S}^{d-1}, \quad (4)$$

which preserves spherical support and rotational symmetry yet admits a fully reparameterizable sampler *without* rejection sampling. For convenience, define the axial projection (cosine similarity) $c = \boldsymbol{\mu}^\top \mathbf{u} \in [-1, 1]$ with the affine transform $C = (c + 1)/2 \in [0, 1]$. Under equation 4, the marginal of C is a Beta distribution with parameters determined by d and κ :

$$C \sim \text{Beta}\left(\alpha = \frac{d-1}{2} + \kappa, \beta = \frac{d-1}{2}\right), \quad \text{so that} \quad c = 2C - 1. \quad (5)$$

Sampling proceeds by drawing C from the Beta and setting $c = 2C - 1$, then sampling a unit vector $\mathbf{v}_\perp$ uniformly in the tangent space orthogonal to $\boldsymbol{\mu}$ and composing

$$\mathbf{u} = c\boldsymbol{\mu} + \sqrt{1 - c^2} \mathbf{v}_\perp, \quad (6)$$

optionally implemented via a Householder transform to align a reference basis with $\boldsymbol{\mu}$. This inverse-CDF construction yields low-variance, fully reparameterizable gradients and improved numerical stability; the spherical ELBO in equation 3 remains unchanged with q_ϕ taken as Power Spherical. In downstream autoregressive models, we keep the radius fixed and renormalize latent inputs/outputs back to $\|\mathbf{z}\|_2 = R$ (also after CFG rescaling) to remove scale degrees of freedom.

Why Scale-Invariant Inputs and Outputs Matter in AR We normalize each provisional next-token prediction by the radius- R projection $N_R(\mathbf{z}) = R\mathbf{z}/\|\mathbf{z}\|_2$ onto the hypersphere. At a reference token on the sphere, the differential of N_R is exactly the orthogonal projector onto the tangent space; thus, to first order, normalization removes radial (scale) perturbations and preserves only tangential (directional) ones. Consequently, composing normalization with the next-token predictor removes the radial (scale) component of the linearized one-step error *prior* to refeeding, so scale errors cannot accumulate across autoregressive steps. See Appendix A for the formal statement and proof.

Limitations of Gaussian Posterior with Post-hoc Normalization A tempting alternative to achieve scale invariance is to retain a diagonal-Gaussian posterior $q_\phi(\mathbf{z} | \mathbf{x})$ and normalize the sampled latents (via N_R), before feeding them to the decoder (henceforth “Gaussian+norm”). However, this choice is theoretically suboptimal: it optimizes a *strictly looser* variational bound than a spherical posterior (see Appendix B). Intuitively, the decoder discards radius by normalization, yet the ELBO still incurs an extra nonnegative *radial* KL term that does not arise with a hyperspherical posterior. By contrast, a hyperspherical posterior aligns the training objective with the constant-norm constraint and avoids this mismatch. Moreover, hyperspherical posteriors are axially symmetric about $\boldsymbol{\mu}$ and governed by a single concentration parameter κ , whereas Gaussian+norm induces a projected-normal (Angular Central Gaussian) directional law whose level sets are elliptical and generally not axially symmetric; this geometric mismatch makes it a poorer fit to purely directional structure (details in Appendix B). Empirically (Sec. 4.3), S-VAE outperforms Gaussian+norm, corroborating this analysis.

3.2 Continuous-Token Autoregressive Transformer

Given an image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$, S-VAE encodes it into a latent tensor $\mathbf{Z} \in \mathbb{R}^{h \times w \times d}$ with a *fixed per-token norm*: for every spatial location (i, j) , $\|\mathbf{Z}_{i,j}\|_2 = R$ (each $\mathbf{Z}_{i,j} \in \mathbb{R}^d$). For sequential autoregressive modeling, we flatten $\mathbf{Z}$ in *raster-scan* (row-major) order to obtain a sequence $\{\mathbf{z}_1, \dots, \mathbf{z}_l\}$ of length $l = hw$, where $\mathbf{z}_k$ is simply the latent at the k -th position in row-major order.

We employ a causal (unidirectional) Transformer to model the conditional distribution of the next token in the flat sequence. At position $k - 1$, the model takes the prefix $\{\mathbf{z}_1, \dots, \mathbf{z}_{k-1}\}$ as input and produces a hidden state $\mathbf{h}_{k-1} = f(\mathbf{z}_{<k}; \theta)$, where θ denotes the Transformer parameters. Optionally, discrete class labels or text prompts are *prepended* as conditioning tokens to the prefix and included in the causal context.

To predict the next continuous token $\mathbf{z}_k$, we follow MAR [12] and attach a *token-level diffusion head*. Conditioned on $\mathbf{h}_{k-1}$, the head progressively transforms a simple prior (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$) into the data distribution of the next token $\mathbf{z}_k$.

We train the diffusion head with *Rectified Flow* [36, 37]. Given a prior $\mathbf{z}_k^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, target $\mathbf{z}_k^1 = \mathbf{z}_k$, and a continuous time $t \in (0, 1)$, we form the linear interpolation

$$\mathbf{z}_k^t = (1 - t)\mathbf{z}_k^0 + t\mathbf{z}_k^1. \quad (7)$$

The diffusion head, parameterized by ω , takes the noisy interpolation $\mathbf{z}_k^t$, the scalar time t , and the condition $\mathbf{h}_{k-1}$ as inputs, and predicts a velocity, $\mathbf{v}_\omega(\mathbf{z}_k^t, t, \mathbf{h}_{k-1}) \in \mathbb{R}^d$. The training target is the

flow velocity along the straight path, $\frac{dz_k^t}{dt} = \mathbf{z}_k^1 - \mathbf{z}_k^0$, and the objective is mean-squared error:

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{\mathbf{z}_k^0, \mathbf{z}_k^1, t} \left[\left\| \mathbf{z}_k^1 - \mathbf{z}_k^0 - \mathbf{v}_\omega(\mathbf{z}_k^t, t, \mathbf{h}_{k-1}) \right\|_2^2 \right]. \quad (8)$$

At inference, we initialize $\mathbf{z}_k^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at $t = 0$ and integrate the learned velocity field $\mathbf{v}_\omega(\mathbf{z}_k^t, t, \mathbf{h}_{k-1})$ up to $t = 1$ using N uniform steps $\Delta = 1/N$ (e.g., explicit Euler):

$$\mathbf{z}_k^{t+\Delta} \leftarrow \mathbf{z}_k^t + \Delta \mathbf{v}_\omega(\mathbf{z}_k^t, t, \mathbf{h}_{k-1}). \quad (9)$$

After N steps, we enforce the constant-norm constraint with a single projection onto the radius- R hypersphere: $\mathbf{z}_k \leftarrow R \mathbf{z}_k^1 / \|\mathbf{z}_k^1\|_2$. The resulting token $\mathbf{z}_k$ is then fed to the next AR step and ultimately to the VAE decoder. When using classifier-free guidance (CFG), the velocity at each step is obtained from a guided (rescaled) combination of conditional and unconditional predictions; we perform no intermediate normalization and apply a single constant-norm projection only after N steps.

3.3 Model Architectures

S-VAE Although VQGAN-style CNN backbones [3] are effective for latent VAEs, their throughput is limited by large convolutional activation maps. To improve efficiency without sacrificing quality, we adopt a *hybrid* backbone: the encoder uses a lightweight CNN stem with downsampling for *patchification*, followed by a stack of Transformer blocks; the decoder mirrors this with a Transformer stack that refines latent tokens and a lightweight CNN with upsampling for *unpatchification* and pixel reconstruction. This preserves the CNN’s strong local inductive bias while leveraging the Transformer’s efficient global modeling at token resolution, yielding a favorable speed–quality trade-off. As shown in Appendix D, the hybrid matches CNN baselines in quality while being about $2.6\times$ faster.

Autoregressive Transformer Following prior work [31, 13], we adopt a modern causal Transformer. Concretely, we use pre-norm Transformer blocks with RMSNorm [38, 39], FlashAttention for efficient attention computation [40], and SwiGLU feed-forward layers [41]. For image positional encoding, we employ 2D rotary embeddings (RoPE) [42] applied in raster-scan order. All self-attention is strictly unidirectional (causal mask). For the diffusion head, we follow MAR [12] and use an MLP architecture.

4 Experiments

We evaluate *SphereAR* on ImageNet-1K [43] class-conditional generation of a resolution of 256×256 , comparing against previous strong baselines. Beyond end-to-end comparisons, we include targeted studies to substantiate our design choices, focusing on the following questions: (1) **S-VAE vs. diagonal-Gaussian**: Does S-VAE outperform diagonal-Gaussian VAEs for continuous-token AR? (2) **Post-hoc normalization**: If we apply ℓ_2 normalization to latents from a diagonal-Gaussian VAE, how does it compare with S-VAE? (3) **Component contributions**: Which parts of S-VAE drive the gains—(i) the hyperspherical posterior, (ii) normalization applied to the VAE decoder input, or (iii) normalization applied to AR inputs/outputs?

4.1 Implementation Details

S-VAE We adopt a Power Spherical [20] directional posterior with latent dimensionality $d = 16$ and fix the radius to $R = \sqrt{d}$. Complete setting for the S-VAE’s backbone is provided in Appendix D. We train S-VAE from scratch on ImageNet-1K [43] with random-crop augmentation, optimizing a weighted sum of ELBO (reconstruction + KL), perceptual [44, 45], and adversarial [46] losses. Optimization uses AdamW [47, 48] for 100 epochs (batch size 256, learning rate 1×10^{-4} , $\beta = (0.9, 0.95)$, weight decay 0.05).

Autoregressive Transformer Following MAR [12], we instantiate three model sizes for *SphereAR*. *SphereAR-B* uses 24 Transformer blocks (hidden size 768) and a diffusion head with 6 feed-forward blocks (hidden size 768). *SphereAR-L* uses 32 Transformer blocks (hidden size 1024) and a diffusion head with 8 feed-forward blocks (hidden size 1024). *SphereAR-H* uses 40 Transformer blocks (hidden size 1280) and a diffusion head with 12 feed-forward blocks (hidden size 1280). As in

Table 1: Overall comparison on ImageNet 256×256 class-conditional generation. Abbreviations: AR = next-token (causal) autoregression; Mask. = masked generation (next-set); N.S. = next-scale; Diff. = diffusion. An asterisk (*) indicates models trained at 384×384 and evaluated at 256×256 by resizing.

Model	Type	Order	#Params	#Epochs	FID↓	IS↑	Pre.↑	Rec.↑
<i>Discrete Tokens</i>								
VQGAN [3]	AR	raster	1.4B	240	5.20	280.3	-	-
ViT-VQGAN [4]	AR	raster	1.7B	240	3.04	227.4	-	-
LlamaGen-L [31]	AR	raster	343M	300	3.07	256.1	0.83	0.52
LlamaGen-XL* [31]	AR	raster	775M	300	2.62	244.1	0.80	0.57
LlamaGen-XXL* [31]	AR	raster	1.4B	300	2.34	253.9	0.80	0.59
RandAR-L [32]	AR	random	343M	300	2.55	288.8	0.81	0.58
RandAR-XL [32]	AR	random	775M	300	2.22	314.2	0.80	0.60
RandAR-XXL [32]	AR	random	1.4B	300	2.15	322.0	0.79	0.62
RAR-B [33]	AR	hybrid	261M	400	1.95	290.5	0.82	0.58
RAR-L [33]	AR	hybrid	461M	400	1.70	299.5	0.81	0.60
MaskGIT [14]	Mask.	random	227M	300	4.02	355.6	0.78	0.50
MAGVIT-v2 [49]	Mask.	random	307M	270	1.78	319.4	-	-
VAR-d20 [15]	N.S.	-	600M	350	2.57	302.6	0.83	0.56
VAR-d30 [15]	N.S.	-	2B	350	1.92	323.1	0.82	0.59
<i>Continuous Tokens</i>								
LDM-4 [50]	Diff.	-	400M	-	3.60	247.7	0.87	0.48
DiT-XL/2 [16]	Diff.	-	675M	400	2.27	278.2	0.83	0.57
SiT-XL/2 [51]	Diff.	-	675M	400	2.06	277.5	0.83	0.59
GIVT [11]	AR	raster	1.67B	500	2.59	-	0.81	0.57
LatentLM-L [13]	AR	raster	479M	400	2.24	253.8	-	-
MAR-B [12]	Mask.	random	208M	800	2.31	281.7	0.82	0.57
MAR-L [12]	Mask.	random	479M	800	1.78	296.0	0.81	0.60
MAR-H [12]	Mask.	random	943M	800	1.55	303.7	0.81	0.62
<i>SphereAR-B</i> (Our)	AR	raster	208M	400	1.92	277.8	0.81	0.61
<i>SphereAR-L</i> (Our)	AR	raster	479M	400	1.54	295.9	0.80	0.63
<i>SphereAR-H</i> (Our)	AR	raster	943M	400	1.34	300.0	0.80	0.64

MAR, we employ multiple class-conditioning tokens (16 in our models vs. 64 in MAR) and apply class-token dropout with probability 0.1 during training to enable classifier-free guidance (CFG) at inference. Models are trained on ImageNet-1K for 400 epochs using S-VAE latents with AdamW (batch size 512, $\beta = (0.9, 0.95)$, weight decay 0.05), a cosine learning-rate schedule with 20k linear warmup steps and peak learning rate 3×10^{-4} , and an exponential moving average (EMA) of weights with decay 0.9999. Under these settings, *SphereAR-B*, *SphereAR-L* and *SphereAR-H* contain ~ 208 M, ~ 479 M and ~ 943 M parameters, respectively.

Inference Settings For next-token prediction we integrate the learned velocity field with a fixed-step Euler scheme (100 steps). We use the linear CFG schedule from MAR. We enable a KV cache to improve autoregressive decoding efficiency.

4.2 Image Generation Result

We report Fréchet Inception Distance (FID) [52] as the primary metric, computed on 50k samples drawn with a fixed random seed using the ADM evaluation code [53]. The optimal CFG scale is determined through a sweep with a step size of 0.1. Following MAR [12], we additionally report Inception Score (IS) [54] and Precision/Recall (Pre./Rec.) [55].

From the results summarized in Table 1, we observe: (1) *SphereAR-H* (943M) achieves state-of-the-art FID **1.34**, outperforming VAR-d30 (2B, **1.92**) and MAR-H (943M, **1.55**). (2) *SphereAR* is *parameter-efficient*. At large scale, *SphereAR-L* (479M) matches MAR-H (943M, **1.55**) with roughly half the parameters. Even at the base scale, *SphereAR-B* (208M) reaches FID **1.92**, outperforming 2B-parameter VAR, diffusion baselines (DiT and SiT), prior continuous-token AR models (GIVT

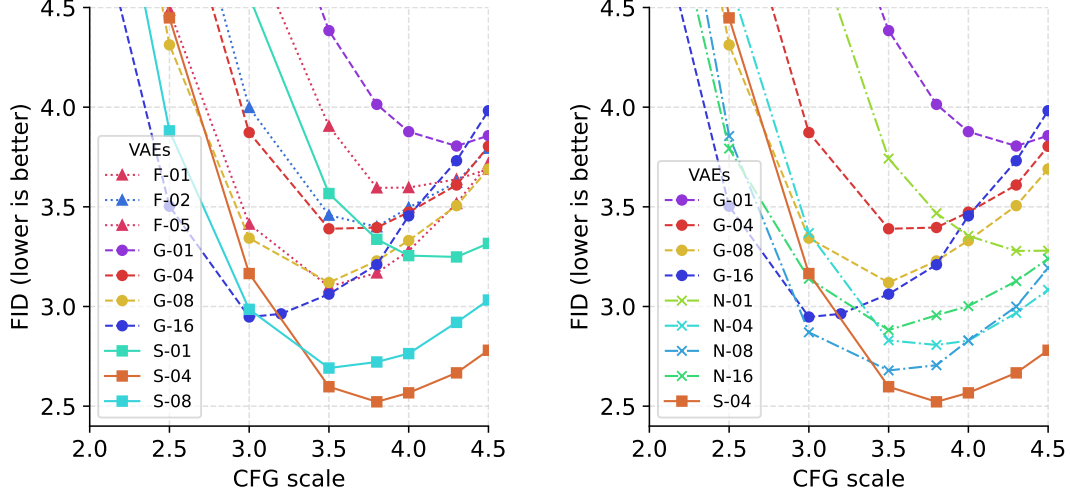


Figure 4: Impact of VAE variants on generation performance (FID vs. CFG). All variants share the same backbone and training/evaluation setup; only the VAE objective/posterior differs. **Left:** diagonal-Gaussian with enlarged KL weight (G-01/04/08/16), σ -VAE with fixed scale (F-01/02/05), and S-VAE with a Power Spherical posterior (S-01/04/08). **Right:** additionally includes diagonal-Gaussian with post-hoc normalization (N-01/04/08/16).

and LatentLM-L), and larger discrete AR models (LlamaGen and RandAR). (3) *Hyperspherical latents are critical*. The key difference from LatentLM is the latent parameterization—fixed-variance diagonal-Gaussian vs. hyperspherical. The large gap (*SphereAR-L*: **1.54** vs. LatentLM-L: **2.24**) indicates that constant-norm, directional latents are crucial for high-quality AR decoding.

Overall, *SphereAR* delivers a scale-invariant AR model that sets the best reported FID with far fewer parameters and outperforms diffusion, masked-generation, and next-scale baselines. Appendix E shows qualitative results.

4.3 Ablation Study

All variants in this ablation use the same model backbone and training/evaluation configuration; only the VAE *objective/posterior* differs. To reduce compute, each VAE is trained on ImageNet for 50 epochs. For the AR stage we use the *SphereAR-L* backbone, trained for 50 epochs with a constant learning rate 1×10^{-4} and batch size 256; all other settings follow Sec. 4.2.

S-VAE vs. Diagonal-Gaussian The core design of *SphereAR* is to constrain latents on a hypersphere via S-VAE. We compare it with prior VAEs. In particular, we evaluate three settings: (1) *Diagonal-Gaussian* (β -VAE). A standard diagonal-Gaussian posterior trained with an up-weighted KL term.³ We sweep four KL weights $\{0.01, 0.04, 0.08, 0.16\}$, denoted G-01, G-04, G-08, and G-16. (2) σ -VAE (*fixed variance*). Following LatentLM [13], we fix the posterior scale with a *fixed, non-learned* scalar $\sigma \sim \mathcal{N}(0, C_\sigma)$, and sample $\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We sweep $C_\sigma \in \{0.1, 0.2, 0.5\}$ (denoted F-01, F-02, F-05). (3) *S-VAE (hyperspherical)*. A Power Spherical posterior on $\mathbb{S}^{d-1}$ with a KL-weight sweep $\{0.001, 0.004, 0.008\}$, denoted S-01, S-04, S-08.

Fig. 4 (left) plots FID versus CFG for these posteriors. We observe: (1) *S-VAE is consistently best and most stable across CFG*. In particular, S-04 attains the lowest FID and S-08 is a close second. (2) *Stronger regularization helps diagonal-Gaussian but saturates*. Increasing β (or C_σ) improves the curves, yet they become unstable at larger CFG and remain below S-VAE. (3) *Fixed variance offers no advantage*. σ -VAE variants achieve performance on par with standard diagonal-Gaussian VAEs (e.g., F-02 vs. G-04; F-05 vs. G-08), indicating that fixing the posterior scale does not help.

³Most prior VAEs compute the KL by taking a sum over spatial-channel dimensions ($h \times w \times d$) and then a batch mean. We instead take a mean over *all* elements (batch and spatial-channel), which lowers the numerical KL value; to match the effective regularization strength, we therefore use larger KL weights (our 10^{-2} roughly matches a prior 2×10^{-6}).

Overall, the above ablation isolates the tokenizer’s role: hyperspherical VAEs yield the most robust and best final AR performance.

Post-hoc Normalization on Diagonal-Gaussian As discussed in Sec. 3.1, a simple alternative to achieve scale invariance is to apply a post-hoc normalization to latents from a diagonal-Gaussian posterior. We therefore run an empirical ablation. Starting from G-x models, we project each latent onto the radius- R hypersphere via $R\mathbf{z}/\|\mathbf{z}\|_2$, yielding N-01, N-04, N-08, and N-16. Fig. 4 (right) plots FID versus CFG for these variants.

From these results, we observe: (1) *Post-hoc normalization helps*. Each N-x improves over its G-x counterpart and is more stable at higher CFG scales, supporting our motivation that scale-invariant inputs/outputs stabilize AR decoding. (2) *S-VAE remains the best*. S-04 outperforms the best post-hoc-normalized Gaussian (N-08). This aligns with our analysis: Gaussian with post-hoc normalization optimizes a *strictly looser* variational bound than the hyperspherical ELBO (Appendix B) and induces a non-axially symmetric directional law on $\mathbb{S}^{d-1}$.

S-VAE’s Component Contributions The above ablations indicate that both the normalization on latent tokens and the hyperspherical posterior are important. Because the normalization can affect two interfaces—the VAE decoder’s input and the AR model’s inputs/outputs—we further isolate their effects by conducting a variant that normalizes only the VAE decoder’s input. As summarized in Table 2, *normalization applied to the AR inputs and outputs is more critical*: normalizing only the VAE decoder’s input yields a modest gain (FID 2.97 $\rightarrow$ 2.89), whereas additionally normalizing AR inputs/outputs produces a larger improvement (FID 2.89 $\rightarrow$ 2.68). This matches our analysis: the AR pathway re-feeds tokens step by step, so scale errors would otherwise accumulate, while the VAE decoder consumes its input once and does not induce cascading scale drift. Finally, replacing the diagonal-Gaussian posterior with a hyperspherical one gives a further boost (FID 2.68 $\rightarrow$ 2.52), confirming that aligning the posterior with constant-norm geometry is beneficial.

Table 2: Ablation of normalization (applied to the VAE decoder and AR) and posterior family.

Norm. on VAE Decoder	Norm. on AR	Posterior	FID↓	IS↑
✗	✗	Gaussian	2.97	240.2
✓	✗	Gaussian	2.89	254.3
✓	✓	Gaussian	2.68	257.3
✓	✓	Spherical	2.52	258.4

5 Conclusion

To address variance collapse in continuous-token AR models, we propose *SphereAR*, whose core idea is to make all AR inputs and outputs scale-invariant. Concretely, it consists of (1) a hyperspherical VAE (S-VAE) that produces latent tokens constrained to a fixed-radius hypersphere; and (2) an autoregressive Transformer with a token-level diffusion head modeling the next-token distribution over hyperspherical latents. During AR training and inference, all inputs and outputs—including those after CFG rescaling—are normalized onto this hypersphere. Our theoretical analysis demonstrates that scale-invariant inputs and outputs are critical to AR modeling. On ImageNet class-conditional generation, *SphereAR-H* (943M) achieves FID 1.34 and *SphereAR-L* (479M) achieves FID 1.54, surpassing prior diffusion and masked-generation baselines. Ablations point to two key factors: constant-norm AR refeeding and a hyperspherical posterior. With both, S-VAE is best, exceeding diagonal-Gaussian, σ -VAE, and even diagonal-Gaussian with post-hoc normalization.

Future work While our results substantiate the motivation and design choices of *SphereAR*, several extensions would further strengthen this work: (i) exploring Riemannian Flow Matching (RFM) [56], which may better align with hyperspherical latent geometry since trajectories of RFM remain on the hypersphere; and (ii) extending *SphereAR* to multimodal applications. We leave these to future work.

Acknowledgments

We are grateful to Prof. Di He for his careful reading and helpful comments on earlier drafts of this manuscript.

References

- [1] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [3] 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*, pages 12873–12883, 2021.
- [4] 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.
- [5] Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. Autoregressive speech synthesis without vector quantization. *arXiv preprint arXiv:2407.08551*, 2024.
- [6] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.
- [7] Shuqi Lu, Haowei Lin, Lin Yao, Zhifeng Gao, Xiaohong Ji, Linfeng Zhang, Guolin Ke, et al. Uni-3dar: Unified 3d generation and understanding via autoregression on compressed spatial tokens. *arXiv preprint arXiv:2503.16278*, 2025.
- [8] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984.
- [9] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [11] Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pages 292–309. Springer, 2024.
- [12] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.
- [13] Yutao Sun, Hangbo Bao, Wenhui Wang, Zhiliang Peng, Li Dong, Shaohan Huang, Jianyong Wang, and Furu Wei. Multimodal latent language modeling with next-token diffusion. *arXiv preprint arXiv:2412.08635*, 2024.
- [14] 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*, pages 11315–11325, 2022.
- [15] 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.
- [16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [17] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, et al. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025.
- [18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

- [19] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyper-spherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.
- [20] Nicola De Cao and Wilker Aziz. The power spherical distribution. *arXiv preprint arXiv:2006.04437*, 2020.
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [22] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.
- [23] Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good visual tokenizers. *arXiv preprint arXiv:2507.15856*, 2025.
- [24] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.
- [25] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025.
- [26] Yinbo Chen, Rohit Girdhar, Xiaolong Wang, Sai Saketh Rambhatla, and Ishan Misra. Diffusion autoencoders are scalable image tokenizers. *arXiv preprint arXiv:2501.18593*, 2025.
- [27] 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.
- [28] 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*, pages 11523–11532, 2022.
- [29] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024.
- [30] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.
- [31] 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.
- [32] 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*, pages 45–55, 2025.
- [33] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024.
- [34] Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *arXiv preprint arXiv:2502.17437*, 2025.
- [35] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*, 2024.
- [36] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

- [38] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.
- [39] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.
- [40] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.
- [41] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
- [42] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [44] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [46] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [47] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [48] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [49] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vignesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion-tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.
- [50] 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*, pages 10684–10695, 2022.
- [51] 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*, pages 23–40. Springer, 2024.
- [52] 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.
- [53] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [54] 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.
- [55] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.
- [56] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

A First-Order Stability of Radius Projection in AR

Let $N_R : \mathbb{R}^d \setminus \{\mathbf{0}\} \rightarrow \mathbb{S}_R^{d-1}$ be the radial projection $N_R(\mathbf{z}) = R\mathbf{z}/\|\mathbf{z}\|_2$, where $\mathbb{S}_R^{d-1} = \{\mathbf{z} \in \mathbb{R}^d : \|\mathbf{z}\|_2 = R\}$. All linearizations are taken at reference tokens $\bar{\mathbf{z}}_k = R\bar{\mathbf{u}}_k$ with $\|\bar{\mathbf{u}}_k\|_2 = 1$.

Let g denote the *pre-normalization next-token map* implemented by our model: given the prefix $\mathbf{z}_{<k}$, the causal Transformer produces a condition $\mathbf{h}_{k-1}$, and the diffusion head returns a provisional latent $\tilde{\mathbf{z}}_k = g(\mathbf{z}_{<k}) \in \mathbb{R}^d$, which is then projected by N_R . We assume g is *continuously differentiable* in a neighborhood of $\bar{\mathbf{z}}_{<k}$ (i.e., C^1 : its Jacobian $\partial g/\partial \mathbf{z}_{<k}$ exists and varies continuously there), which holds for our architecture with smooth activations and fixed-step explicit ODE solvers.⁴

Lemma 1 (Jacobian is the tangent-space projector). For $\|\bar{\mathbf{z}}_k\|_2 = R$,

$$DN_R(\bar{\mathbf{z}}_k) = \mathbf{P}_k := \mathbf{I} - \frac{\bar{\mathbf{z}}_k \bar{\mathbf{z}}_k^\top}{R^2}. \quad (10)$$

Moreover, $\mathbf{P}_k^\top = \mathbf{P}_k$, $\mathbf{P}_k^2 = \mathbf{P}_k$, $\mathbf{P}_k \bar{\mathbf{z}}_k = \mathbf{0}$, and $\mathbf{P}_k \mathbf{v} = \mathbf{v}$ for all $\mathbf{v} \in T_{\bar{\mathbf{z}}_k} \mathbb{S}_R^{d-1} = \{\mathbf{v} : \bar{\mathbf{z}}_k^\top \mathbf{v} = 0\}$; in particular, $\|\mathbf{P}_k\|_2 = 1$. *Proof.* Differentiate $N_R(\mathbf{z}) = R\mathbf{z}/\|\mathbf{z}\|_2^{-1}$ and evaluate at $\bar{\mathbf{z}}_k$.

Lemma 2 (First-order scale invariance). For any small $\Delta \mathbf{z}$,

$$N_R(\bar{\mathbf{z}}_k + \Delta \mathbf{z}) = \bar{\mathbf{z}}_k + \mathbf{P}_k \Delta \mathbf{z} + o(\|\Delta \mathbf{z}\|), \quad (11)$$

so radial derivatives vanish and tangential derivatives are preserved (eigenvalues 0 and 1). *Proof.* First-order Taylor expansion with Lemma 1.

Proposition (One-step AR refeeding error, linearized). Let g be the (unnormalized) next-token predictor and define $\tilde{\mathbf{z}}_k = g(\mathbf{z}_{<k})$, $\mathbf{z}_k = N_R(\tilde{\mathbf{z}}_k)$, and $F_k = N_R \circ g$. Linearizing at $\bar{\mathbf{z}}_{<k}$ (with $\bar{\mathbf{z}}_k = F_k(\bar{\mathbf{z}}_{<k})$) yields

$$\mathbf{e}_k := \mathbf{z}_k - \bar{\mathbf{z}}_k \approx \mathbf{P}_k \left(J_k \mathbf{e}_{<k} + \boldsymbol{\eta}_k \right), \quad J_k = \left(\frac{\partial g}{\partial \mathbf{z}_{<k}} \right)_{\bar{\mathbf{z}}_{<k}}, \quad (12)$$

where $\mathbf{e}_{<k}$ stacks the prefix errors and $\boldsymbol{\eta}_k$ collects local modeling/integration error. *Proof.* Chain rule with Lemma 2.

Corollary (No scale-channel cascade; norm bound). Writing $J_k \mathbf{e}_{<k} + \boldsymbol{\eta}_k = \alpha_k \bar{\mathbf{z}}_k + \mathbf{t}_k$ with $\mathbf{t}_k \in T_{\bar{\mathbf{z}}_k} \mathbb{S}_R^{d-1}$,

$$\mathbf{e}_k \approx \mathbf{P}_k (\alpha_k \bar{\mathbf{z}}_k + \mathbf{t}_k) = \mathbf{t}_k, \quad \text{and} \quad \|\mathbf{e}_k\|_2 \leq \|\mathbf{P}_k J_k\|_2 \|\mathbf{e}_{<k}\|_2 + \|\mathbf{P}_k \boldsymbol{\eta}_k\|_2. \quad (13)$$

Thus radial (scale) errors are annihilated before refeeding and cannot cascade along the AR chain; only directional (tangential) errors propagate.

Scope. Statements are local (first-order) at points on $\mathbb{S}_R^{d-1}$ and assume g is C^1 near $\bar{\mathbf{z}}_{<k}$.

B Gaussian Posterior with Post-hoc Normalization: A Looser Bound

We compare the ‘‘Gaussian+norm’’ objective

$$\mathcal{L}_G(\phi, \psi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\psi(\mathbf{x} | N_R(\mathbf{z}))] - D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})), \quad N_R(\mathbf{z}) = \frac{R\mathbf{z}}{\|\mathbf{z}\|_2},$$

with the spherical ELBO

$$\mathcal{L}_{\text{S-VAE}}(\phi, \psi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{u}|\mathbf{x})} [\log p_\psi(\mathbf{x} | R\mathbf{u})] - D_{\text{KL}}(q_\phi(\mathbf{u} | \mathbf{x}) \| \text{Unif}(\mathbb{S}^{d-1})), \quad \mathbf{u} \in \mathbb{S}^{d-1}.$$

Write the polar decomposition $\mathbf{z} = (r, \mathbf{u})$ with $r = \|\mathbf{z}\|_2 \in \mathbb{R}_{\geq 0}$ and $\mathbf{u} = \mathbf{z}/\|\mathbf{z}\|_2 \in \mathbb{S}^{d-1}$. Let $q_\phi(\mathbf{u} | \mathbf{x})$ be the pushforward of $q_\phi(\mathbf{z} | \mathbf{x})$ under $\mathbf{z} \mapsto \mathbf{u}$. Since $N_R(\mathbf{z}) = R\mathbf{u}$ depends only on direction, the reconstruction terms coincide:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\psi(\mathbf{x} | N_R(\mathbf{z}))] = \mathbb{E}_{q_\phi(\mathbf{u}|\mathbf{x})} [\log p_\psi(\mathbf{x} | R\mathbf{u})].$$

⁴For numerical robustness we implement $N_R(\mathbf{z}) = R\mathbf{z}/\max(\|\mathbf{z}\|_2, \varepsilon)$ with $\varepsilon = 10^{-7}$.

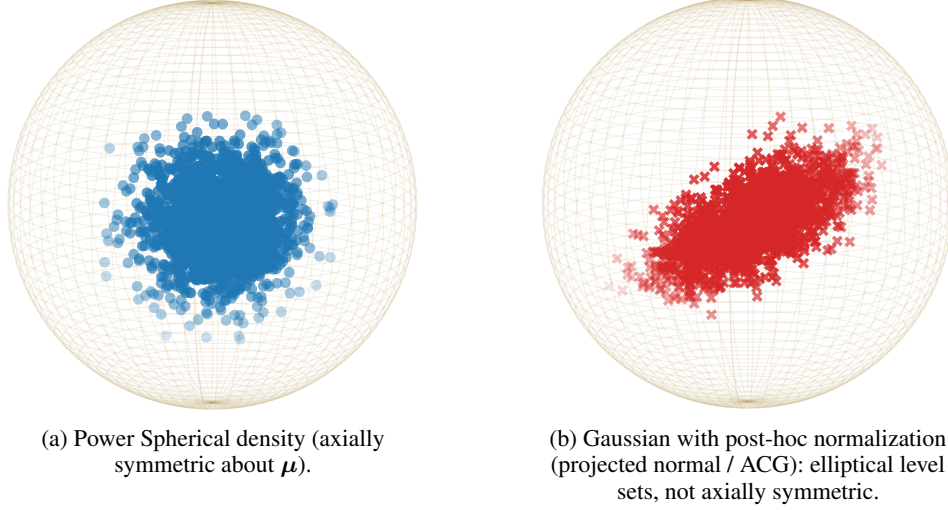


Figure 5: Directional distributions on the sphere. **Left:** Power Spherical respects purely directional geometry—the density depends only on $\mu^\top \mathbf{u}$ and is axially symmetric, with a single concentration parameter κ . **Right:** Gaussian+norm induces a projected-normal (ACG) law whose level sets follow $\mathbf{u}^\top \Sigma^{-1} \mathbf{u}$; symmetry axes are determined by Σ (and the Gaussian mean), so the density is typically elliptical rather than axially symmetric.

For the isotropic Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, one has $p(\mathbf{z}) = p(r) \text{Unif}(\mathbf{u})$ (with $p(r)$ the χ -law in $\mathbb{R}^d$). The KL chain rule (disintegration over $\mathbf{u}$) gives

$$D_{\text{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})) = D_{\text{KL}}(q_\phi(\mathbf{u} \mid \mathbf{x}) \parallel \text{Unif}(\mathbb{S}^{d-1})) + \mathbb{E}_{q_\phi(\mathbf{u} \mid \mathbf{x})} \left[D_{\text{KL}}(q_\phi(r \mid \mathbf{u}, \mathbf{x}) \parallel p(r)) \right].$$

Combining the two displays yields

$$\mathcal{L}_G(\phi, \psi; \mathbf{x}) = \mathcal{L}_{\text{S-VAE}}(\phi, \psi; \mathbf{x}) - \mathbb{E}_{q_\phi(\mathbf{u} \mid \mathbf{x})} \left[D_{\text{KL}}(q_\phi(r \mid \mathbf{u}, \mathbf{x}) \parallel p(r)) \right] \leq \mathcal{L}_{\text{S-VAE}}(\phi, \psi; \mathbf{x}),$$

with equality only if $q_\phi(r \mid \mathbf{u}, \mathbf{x}) = p(r)$ almost surely (i.e., the posterior’s radial law exactly matches the prior and is independent of $\mathbf{x}, \mathbf{u}$). Thus, Gaussian posterior with post-hoc normalization pays an extra nonnegative *radial* KL penalty while the decoder discards radius; a spherical posterior avoids this mismatch and aligns the bound with the constant-norm constraint.

Remark (axial symmetry vs. projected normal on $\mathbb{S}^{d-1}$). The Power Spherical (PS) density on the unit sphere has the form $f_{\text{PS}}(\mathbf{u}) = Z_d(\kappa) (1 + \mu^\top \mathbf{u})^\kappa$ with $\|\mu\|_2 = 1$ and $\kappa \geq 0$. It is *axially rotationally symmetric* about μ : for any rotation Q with $Q\mu = \mu$, one has $f_{\text{PS}}(Q\mathbf{u}) = f_{\text{PS}}(\mathbf{u})$. The single scalar κ monotonically controls geodesic concentration (with $\kappa = 0$ yielding the uniform law).

By contrast, *Gaussian+norm*—take $\mathbf{z} \sim \mathcal{N}(\mu_g, \Sigma)$ in $\mathbb{R}^d$ and map to the sphere via $\mathbf{u} = \mathbf{z} / \|\mathbf{z}\|_2$ —induces a projected-normal (Angular Central Gaussian, ACG) directional law. For the zero-mean case ($\mu_g = \mathbf{0}$), its density is $f_{\text{ACG}}(\mathbf{u}) \propto (\mathbf{u}^\top \Sigma^{-1} \mathbf{u})^{-d/2}$: level sets follow the quadratic form $\mathbf{u}^\top \Sigma^{-1} \mathbf{u}$ and are generally *elliptical*, not axially symmetric; axial symmetry holds only if $\Sigma \propto \mathbf{I}$ (then the law is uniform). With nonzero mean $\mu_g \neq \mathbf{0}$, the density also depends on $\mu_g^\top \Sigma^{-1} \mathbf{u}$, producing skewed, non-axially symmetric level sets (and, for anisotropic Σ , possible antipodal bimodality when $\mu_g = \mathbf{0}$). Therefore the Power Spherical family matches the intended purely directional geometry on $\mathbb{S}^{d-1}$, whereas Gaussian+norm inherits Euclidean anisotropy from (μ_g, Σ) and does not.

C Comparison with MAR’s VAE

We perform an ablation that swaps our S-VAE for MAR’s VAE [12] while keeping the AR backbone, training schedule, and evaluation protocol identical (see Sec. 4.3). As summarized in Table 3, S-VAE yields a large FID gain ($4.54 \rightarrow 2.52$) and higher IS ($241.6 \rightarrow 258.4$). These results indicate that constant-norm, directional latents from S-VAE materially strengthen continuous-token AR generation.

Table 3: Ablation: swapping our S-VAE for MAR’s VAE *without retraining* (same AR backbone and training/evaluation settings; only the VAE changes).

Type	FID↓	IS↑	Pre.↑	Rec.↑
AR + MAR’s VAE	4.54	241.6	0.84	0.45
AR + S-VAE (ours)	2.52	258.4	0.82	0.56

Table 4: Comparison of VAE backbones. Training speed measured in iterations per second on 8 A100 GPUs, with batch size 256.

Type	#Params	Training Speed↑	LPIPS↓	FID↓	IS↑	Pre.↑	Rec.↑
CNN	70M	2.25	0.167	2.63	262.6	0.82	0.55
ViT	170M	7.19	0.178	2.81	251.3	0.82	0.55
Hybrid (ours)	75M	5.81	0.166	2.52	258.4	0.82	0.56

D VAE Architecture — CNN vs. ViT vs. Hybrid

Most latent VAEs for image generation adopt a VQGAN-style encoder-decoder [3], i.e., a convolutional (CNN) backbone with downsampling/upsampling blocks. This design is parameter-efficient ($\sim 70\text{M}$) but throughput is often limited by activation memory and bandwidth on large feature maps, leading to slow training and inference. ViT-VQGAN [4] replaces the CNN backbone with a Vision Transformer (ViT) to improve efficiency; however, as shown in Table 4, a pure ViT backbone yields weaker generative metrics than a CNN.

To balance efficiency and quality, we adopt a *hybrid* VAE architecture. The encoder first uses a lightweight CNN stem (with downsampling blocks) for *patchification* and early spatial mixing, imparting CNN inductive bias while reducing spatial resolution. The resulting patch tokens are then processed by a bidirectional Transformer (ViT blocks) to model long-range dependencies. The decoder mirrors this design: a ViT stack refines the latent tokens, followed by a lightweight CNN (with upsampling blocks) for *unpatchification* and pixel-level reconstruction. This hybrid preserves the CNN’s strong local bias while leveraging the ViT’s global receptive field at token resolution, yielding a favorable speed-quality tradeoff.

In our implementation, to match the parameter scale of a VQGAN-style CNN encoder-decoder [3], our S-VAE uses 6 ViT blocks in the encoder and 12 in the decoder, each with hidden size 512. The encoder’s CNN stem performs patchification via 4 downsampling stages (overall $16\times$ reduction) with channel widths [64, 64, 128, 256, 512]; the decoder mirrors this with 4 upsampling stages and an extra residual block per stage. In total, the S-VAE has $\sim 75\text{M}$ parameters.

We compare three encoder-decoder backbones under the same training setup and losses: (i) a VQGAN-style CNN [3]; (ii) a pure ViT [4] (12 Transformer blocks in both encoder and decoder, hidden size 768; $\sim 170\text{M}$ params); and (iii) our *Hybrid* design. To reduce compute, the training follows the setting in Sec. 4.3. We report training throughput (iterations/s) of VAE, perceptual distortion (LPIPS with VGG-16), and downstream ImageNet generative metrics. Results in Table 4 show: *ViT* is fastest (7.19 it/s) but slightly worse on LPIPS/FID/IS; the *CNN* is slowest (2.25 it/s) yet competitive in IS; our *Hybrid* retains the best reconstruction (LPIPS 0.166) and the best FID (2.52) while running at 5.81 it/s—about $2.6\times$ faster than the CNN and at 81% of ViT throughput (5.81 vs 7.19 it/s). Overall, the hybrid backbone offers the most favorable speed-quality trade-off.

E Model Generated Examples

We show the uncured 256×256 samples generated by our 479M *SphereAR-L*, from Fig. 6 to Fig. 17.



Figure 6: Uncurated 256×256 *SphereAR-L* samples. Class label: "frilled lizard" (43).



Figure 7: Uncurated 256×256 *SphereAR-L* samples. Class label: "sulphur-crested cockatoo" (89).



Figure 8: Uncurated 256×256 *SphereAR-L* samples. Class label: "golden retriever" (207).



Figure 9: Uncurated 256×256 *SphereAR-L* samples. Class label: "Border collie" (232).



Figure 10: Uncurated 256×256 *SphereAR-L* samples. Class label: "tabby cat" (281).



Figure 11: Uncurated 256×256 *SphereAR-L* samples. Class label: "ladybug" (301).



Figure 12: Uncurated 256×256 *SphereAR-L* samples. Class label: "chimpanzee" (367).



Figure 13: Uncurated 256×256 *SphereAR-L* samples. Class label: "beacon" (437).



Figure 14: Uncurated 256×256 *SphereAR-L* samples. Class label: "castle" (483).



Figure 15: Uncurated 256×256 *SphereAR-L* samples. Class label: "icecream" (928).



Figure 16: Uncurated 256×256 *SphereAR-L* samples. Class label: "cliff" (972).



Figure 17: Uncurated 256×256 *SphereAR-L* samples. Class label: "coral reef" (973).