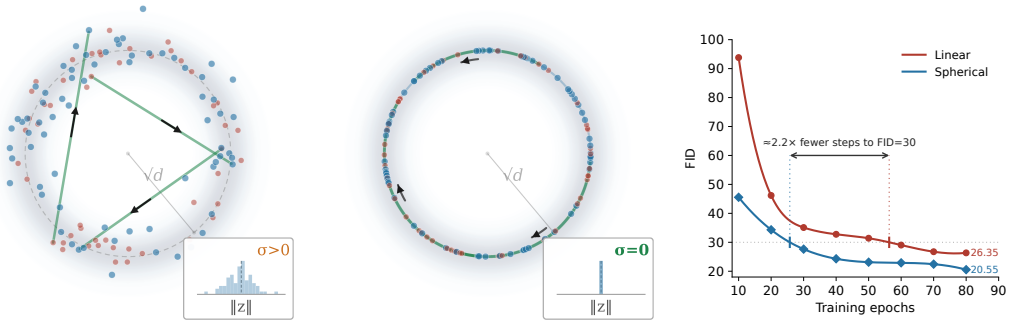
ALIGNING LATENT GEOMETRY FOR SPHERICAL FLOW MATCHING IN IMAGE GENERATION

Anonymous authors
Paper under double-blind review



(a) Linear (b) Spherical (ours) (c) FID, SiT-B/2, FLUX.2 VAE

Figure 1: **From VAE latent shells to spherical flow matching.** (a) Linear flow matching connects Gaussian noise and VAE tokens along straight paths that change both radius and direction. Each point z represents one latent token; blue and orange denote noise and data tokens, respectively, and green denotes the interpolation path. (b) We project both endpoints onto a shared fixed-radius sphere, lightly finetune the pretrained decoder with the encoder frozen, and train the flow along spherical arcs using slerp. (c) On FLUX.2 SiT-B/2, spherical flow matching reaches FID$=30$ in approximately $2.2\times$ fewer flow-training steps than vanilla-linear.

ABSTRACT

Latent flow matching typically treats pretrained variational autoencoder (VAE) representations as fixed, although accurate reconstruction does not ensure that their geometry is well suited to downstream generative modeling. We study whether adapting the latent geometry improves image generation while retaining the encoder and diffusion architecture. Across three tokenizer families, component-swap probes show that preserving token direction retains substantially more decoded perceptual and semantic similarity than preserving radius. Based on this observation, we project each latent token onto a fixed-radius sphere, with the encoder frozen, and lightly finetune the pretrained decoder to reconstruct from the projected representation. We then train spherical flow matching on the induced spheres. On class-conditional ImageNet-256 with FLUX.2 SiT-B/2, the complete method reduces FID from 26.35 to 20.55 under a matched training recipe and a fixed 50-update sampling protocol, and reaches FID$=30$ in approximately $2.2\times$ fewer flow-training steps. Matched decoder controls support the combined latent construction and adaptation. Recipe-transfer comparisons yield lower FID across three tested tokenizer families and SiT-B/XL backbones, and the gain remains under a matched representation-alignment objective.

1 INTRODUCTION

Diffusion (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2022) often generate high-fidelity images by modeling the latent representations of pretrained variational autoencoders (VAEs) (Rombach et al., 2022; Esser et al., 2024). The generator learns image content through the distribution produced by the tokenizer.

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

Reconstruction quality alone is a weak predictor of downstream generation quality (Xu et al., 2026; Yao et al., 2025). Standard latent flow matching typically keeps this representation fixed and learns to transport noise to it. This raises a question: can adapting the geometry of a pretrained latent space make flow matching more effective without replacing the encoder or changing the diffusion architecture?

Pretrained VAE token radii are tightly concentrated around a characteristic scale, but they are not fixed, and the decoder was trained on this variable-radius latent distribution. Reusing such a tokenizer for spherical transport therefore requires more than changing the interpolation path: the chosen latent support must also remain useful for reconstruction. Prior work studies hyperspherical VAEs and fixed-radius latents for generative modeling (Davidson et al., 2018; Ke & Xue, 2025). Riemannian flow matching defines transport on these spaces (Chen & Lipman, 2023); most closely, Kumar & Patel (2026) use spherical transport with Jacobi regularization on representation-encoder features. We study how to construct fixed-radius tokens from a pretrained VAE encoder and adapt its existing decoder before spherical transport, measuring how removal of radial variation affects reconstruction.

Across FLUX.2, VA-VAE, and REPA-E FLUX.1, we show that token radii concentrate near a spherical shell and that preserving token direction retains more decoded perceptual and semantic similarity than preserving radius. By evaluating changes in decoded images, our component-swap probes connect the latent geometry to the information preserved by the tokenizer. We also show that linear interpolation between comparable endpoint shells contracts token radii and introduces radial motion into the training target. The endpoints can therefore lie near the same shell while the path between them passes through its interior. Together, these observations motivate constructing fixed-radius latents and restricting the flow to angular motion.

We therefore project each token onto a fixed-radius sphere and lightly finetune the existing decoder to reconstruct from the projected latents, keeping the encoder frozen. This adapts the decoder to the inputs it will receive during generation, rather than expecting the original decoder to handle a changed latent distribution. Spherical flow matching then transports a uniform spherical prior along paths that remain on this support. Noise, intermediate states, and generated tokens thus share the same fixed-radius geometry. Unlike approaches that replace the VAE with a representation encoder, our construction retains the pretrained VAE encoder and leaves the diffusion architecture unchanged. It requires no auxiliary representation encoder and remains compatible with representation alignment.

On class-conditional ImageNet-256, controlled FLUX.2 experiments show that adapting the latent support and decoder improves generation quality, with further gains from spherical transport. Decoder controls distinguish this combined intervention from simply finetuning on the original latents or changing transport while leaving the decoder unadapted. The spherical pipeline also reaches a fixed FID threshold in fewer training steps than the vanilla-linear baseline. Broader evaluations show lower FID (Heusel et al., 2017) across the three studied tokenizer families and both SiT-B and SiT-XL backbones at matched guidance, and the gains persist when combined with REPA (Yu et al., 2024). These results motivate treating latent geometry as a design choice rather than a fixed property of the pretrained tokenizer.

Our contributions are: (i) an empirical study of the radial and angular structure of pretrained VAE latents, connecting component substitutions to decoded perceptual and semantic similarity; (ii) a post-hoc spherical latent construction that reuses the pretrained encoder and adapts its decoder for established spherical flow matching; and (iii) controlled evaluations of the combined construction, together with cross-tokenizer and cross-backbone experiments demonstrating improved generation and compatibility with representation alignment.

2 Related work

Hyperspherical latent spaces. L_2-normalization is standard in discriminative representation learning (Wang et al., 2017; Deng et al., 2019). Davidson et al. (2018) introduce a VAE with a von Mises-Fisher posterior and a uniform spherical prior; Xu & Durrett (2018) use fixed-concentration von Mises-Fisher posteriors to mitigate collapse in text VAEs. SphereAR (Ke & Xue, 2025) combines a hyperspherical VAE with continuous-token autoregressive generation (Li et al., 2024) and token-level diffusion, using fixed-radius projection to stabilize classifier-free guidance. It also evaluates

post-hoc normalization of Gaussian VAE latents. Yue et al. (2026) train an encoder to distribute images uniformly on a sphere and a decoder to generate directly from random sphere points.

Riemannian and manifold flow matching. Manifold generative models extend continuous normalizing flows (Chen et al., 2018; Mathieu & Nickel, 2020; Rozen et al., 2021) and score-based diffusion (De Bortoli et al., 2022; Huang et al., 2022). Riemannian flow matching (Chen & Lipman, 2023) extends simulation-free flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2022) to manifolds, with closed-form conditional geodesics on spheres. Fisher-Flow (Davis et al., 2024) maps categorical probabilities to the sphere's positive orthant through a square-root transformation for geodesic training. Riemannian Gaussian variational flow matching (Zaghen et al., 2025) relates endpoint and velocity prediction through curvature-dependent Jacobi terms. Most closely, RJF (Kumar & Patel, 2026) combines projected Gaussian noise, slerp, tangent targets and model outputs, and exponential-map sampling for normalized representation features, with additional Jacobi loss reweighting. Their evaluated pipeline replaces the VAE with a representation encoder and an RAE decoder. In contrast, we retain the pretrained VAE encoder and adapt its existing decoder to projected tokens, avoiding tokenizer replacement. Radius and direction substitution probes motivate this construction: radius substitution better preserves decoded image similarity.

Representation-space diffusion and spherical guidance. Representation autoencoders (Zheng et al., 2025) pair frozen DINOv2 (Oquab et al., 2023), SigLIP2, or MAE encoders with decoders trained for their features, then learn flow matching in that space using linear paths. Alignment methods retain the VAE: VA-VAE aligns its latents with frozen vision-model features during tokenizer training (Yao et al., 2025), REPA aligns diffusion-model hidden states with frozen clean-image representations (Yu et al., 2024), and REPA-E propagates this alignment into the VAE encoder during joint training (Leng et al., 2025). The projection applies to both representation-aligned and non-aligned pretrained tokenizers (Yao et al., 2025; Rombach et al., 2022; Black Forest Labs, 2025). Separately, DSG (Yang et al., 2024) and DiffRGD (Liao et al., 2026) apply spherical constraints to training-free sampling guidance, leaving the learned latent representation and training paths unchanged.

Latent-space structure versus reconstruction fidelity. For image tokenizers (Esser et al., 2021; Rombach et al., 2022), reconstruction quality alone poorly predicts downstream generation (Yao et al., 2025). Xu et al. (2026) quantify this decoupling across autoencoders and propose interpolated FID as a more predictive metric, while Qiu et al. (2025b;a) examine tokenizer robustness to sampling errors. Structural interventions include decoder scale-equivariance motivated by spectral analysis (Skorokhodov et al., 2025), latent equivariance regularization (Kouzelis et al., 2025), joint training (Leng et al., 2025), semantic regularization of discrete tokenizers (Xiong et al., 2025), and non-variational tokenizers with discriminative latents (Chen et al., 2025). Our projection constrains latent support rather than imposing spectral or semantic targets.

3 Geometry of pretrained VAE latents

A pretrained autoencoder maps an image x to a latent $z_1 = \mathcal{E}(x) \in \mathbb{R}^{d\times h\times w}$ with one token in $\mathbb{R}^d$ per spatial position. The decoder $\mathcal{D}$ reconstructs the image from this representation. We define z in the flow-training coordinates, after any fixed tokenizer preprocessing such as scale, shift, packing, or channel standardization.

3.1 Measuring radial concentration

The baseline flow transports Gaussian noise to encoded data. We therefore use the Gaussian token-radius distribution as a reference when measuring the radial scale and relative variation of VAE latents.

For a Gaussian noise token $z_0 \sim \mathcal{N}(0, I_d)$ and a threshold $a > 0$, concentration of measure gives

$$\mathbb{P}\Big(\big|\, \|z_0\| - \sqrt{d}\,\big| > a\Big) \le 2\exp(-c\,a^2), \tag{1}$$

where $c > 0$ is an absolute constant (Vershynin, 2018, Theorem 3.1.1). The radius fluctuates on an $\mathcal{O}(1)$ scale around $\sqrt{d}$, so the relative shell width decreases as $\mathcal{O}(1/\sqrt{d})$. The root-mean-square radius is $\sqrt{d}$. The mean radius is slightly smaller: $\mathbb{E}\|z_0\|_2 = \sqrt{2}\Gamma((d+1)/2)/\Gamma(d/2)$, where Γ is Euler's Gamma function. The Gaussian entries in Tab. 1 use this exact mean, derived in Sec. A.1.

Table 1: Per-token norm statistics. Raw $\bar{r}$ is the mean radius of encoder tokens; processed $\bar{r}$ is computed after each tokenizer's flow-training preprocessing. Gaussian rows give the analytical mean radius $\mathbb{E}\|z\|_2$ and coefficient of variation for $z \sim \mathcal{N}(0, I_d)$; see Sec. A.1. VAE statistics are measured on 2048 ImageNet-256 images.

Source	d	Preproc.	Raw $\bar{r}$	Raw CV	Proc. $\bar{r}$	Proc. CV
Gaussian Noise	16	–	3.94	0.18	3.94	0.18
Gaussian Noise	32	–	5.61	0.13	5.61	0.13
FLUX.2 VAE	32	pack+std.	9.43	0.19	5.33	0.20
REPA-E FLUX.1 VAE	16	affine/stat.	9.13	0.22	1.38	0.23
VA-VAE	32	channel std.	19.85	0.16	5.38	0.16

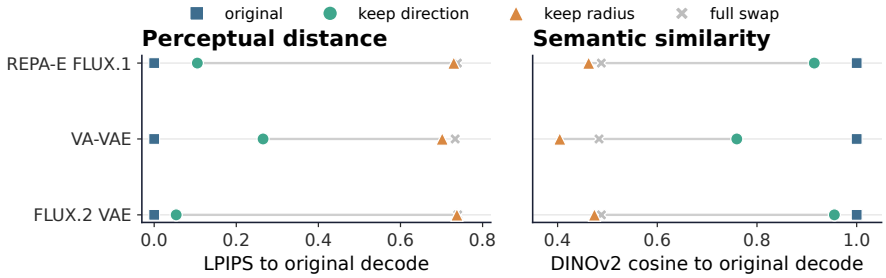


Figure 2: Radius and direction swaps between same-class images. LPIPS (left) and DINOv2 cosine similarity (right) are measured relative to the anchor decode. Markers show averages over 1024 anchor-neighbor pairs; their shapes identify the swap condition.

For the pretrained VAEs (Kingma & Welling, 2013), we measure the mean token radius $\bar{r}$ and coefficient of variation (CV), the radius standard deviation divided by its mean. Table 1 compares these statistics with the Gaussian reference, both before and after each tokenizer's fixed flow-training preprocessing.

All three tokenizers exhibit radial concentration (CV ≤ 0.23), and raw and processed CV agree within 0.02. After preprocessing, FLUX.2 and VA-VAE sit close to the Gaussian concentration radius ($\bar{r}/\sqrt{d} = 0.94$ and 0.95), while REPA-E FLUX.1 has a smaller mean radius ($\bar{r}/\sqrt{d} = 0.35$).

3.2 Probing radius and direction

We swap radius and direction between latents from images of the same class to measure their effect on the decoded image. At each spatial position, write the anchor and neighbor tokens as $z_a = r_a \hat{z}_a$ and $z_n = r_n \hat{z}_n$, with radius r and unit direction $\hat{z}$. The hybrid grids use $r_n \hat{z}_a$ (keep direction) and $r_a \hat{z}_n$ (keep radius). We decode both hybrids and the original anchor and neighbor grids.

Figure 2 reports LPIPS (Zhang et al., 2018) and DINOv2 cosine similarity relative to the anchor decode. Keeping direction gives lower LPIPS and higher DINOv2 similarity than keeping radius for all three tokenizers. Replacing direction gives results close to the full-neighbor swap on both metrics.

The same-class swaps exchange radii that vary across tokens. To test a constant radius, the population-mean control in Fig. A2 substitutes the dataset-wide mean radius while keeping each token's direction. Keeping direction again gives lower LPIPS and higher DINOv2 similarity than keeping radius across all three tokenizers. This supports preserving token directions while replacing their variable radii with a constant.

3.3 Radial motion in linear flow matching

Our baseline uses linear-path latent flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2022) with a Scalable Interpolant Transformer (SiT) (Ma et al., 2024) backbone. We train a velocity field $v_\theta(z_t, t, y)$, conditioned on the ImageNet class label y, along linear interpolations

between independently sampled Gaussian noise $z_0 \sim \mathcal{N}(0, I)$ and encoded data z_1:

$$z_t = (1-t)\, z_0 + t\, z_1, \quad t \in [0,1], \tag{2}$$

with conditional velocity $u_t = z_1 - z_0$ and objective

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\, z_0,\, z_1,\, y}\left[\|v_\theta(z_t, t, y) - u_t\|^2\right]. \tag{3}$$

For token endpoints $z_0 = r_0\hat{z}_0$ and $z_1 = r_1\hat{z}_1$, the squared norm along the chord is

$$\|(1-t)z_0 + tz_1\|^2 = (1-t)^2 r_0^2 + t^2 r_1^2 + 2t(1-t)r_0 r_1 \langle \hat{z}_0, \hat{z}_1 \rangle. \tag{4}$$

Since the Gaussian noise direction is isotropic and independent of the data token, $\mathbb{E}\langle \hat{z}_0, \hat{z}_1 \rangle = 0$. Equal endpoint radii $r_0 = r_1 = R$ give a root-mean-square midpoint radius of $R/\sqrt{2}$. FLUX.2 and VA-VAE have comparable endpoint radii, and their measured norm profiles show this inward motion. The REPA-E FLUX.1 path also connects different noise and data shell radii (Fig. 3a).

Figure 3 compares linear and shell paths in the original flow-training coordinates with spherical-slerp on projected latents. The shell path interpolates directions along a spherical arc and radii linearly, as defined in Sec. 4.2. Linear and shell paths retain the original noise and data endpoint radii. Spherical-slerp instead uses projected data tokens and noise on the same sphere of radius $\sqrt{d}$.

Let μ_e and σ_e denote the mean and standard deviation of the token radius at each vanilla endpoint $e \in \{0, 1\}$. For linear and shell paths, the standardized distance of a path radius r_t from the nearer endpoint shell is $\Delta(r_t) = \min_e |r_t - \mu_e|/\sigma_e$. The maximum deviations along the linear paths are 1.4σ for FLUX.2, 1.8σ for VA-VAE, and 2.5σ for REPA-E FLUX.1 (Fig. 3). These values are in units of the nearer endpoint's token-radius standard deviation. The spherical-slerp curve is referenced to its fixed-radius support, on which its radial deviation is zero.

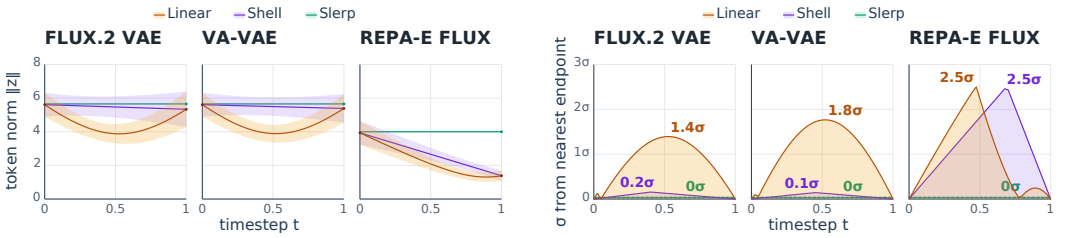
For each token, the radial and tangential velocity targets are $u_t^{\mathrm{rad}} = \langle u_t, \hat{z}_t \rangle \hat{z}_t$ and $u_t^{\tan} = u_t - u_t^{\mathrm{rad}}$, where $\hat{z}_t = z_t/\|z_t\|$. We measure the share of each token's squared velocity target that changes its radius by $\|u_t^{\mathrm{rad}}\|^2/\|u_t\|^2$, computed in each tokenizer's flow-training coordinates (Fig. A1). Along linear paths, this share is about 50% at both endpoints for FLUX.2 and VA-VAE, and about 90% at the noise endpoint for REPA-E FLUX.1, whose processed mean token radius is about $0.35\sqrt{d}$ (Tab. 1).

We therefore preserve token directions (Sec. 3.2) and fix their radius by projecting encoded tokens onto $\mathcal{S}^{d-1}(\sqrt{d})$. The noise comes from the same sphere. Spherical interpolation between these endpoints keeps $\|z_t\| = \sqrt{d}$ throughout the path and yields an entirely tangential velocity target. These observations motivate fixing token radii while preserving their directions.

4 From Latent Shells to Spherical Flow Matching

4.1 Fixing the Radius and Adapting the Decoder

We insert a token-wise L_2 projection after the pretrained tokenizer's fixed flow-training preprocessing. For an encoder $\mathcal{E}$ with token dimension d and spatial resolution $h \times w$, define $\pi : \mathbb{R}^d \setminus \{0\} \to$



(a) Per-token norm along linear, shell, and slerp paths for representative tokenizers. Lines average 2048 pairs, with bands showing $\pm 1\sigma$. The horizontal reference is the fixed spherical radius.

(b) Standardized radial distance from the nearer vanilla endpoint shell for linear and shell paths. Spherical-slerp has zero radial deviation from its own fixed-radius support.

Figure 3: Linear paths follow Euclidean chords. Shell paths interpolate directions and radii separately. Both use vanilla endpoints; spherical-slerp connects fixed-radius noise and projected data tokens.

$\mathcal{S}^{d-1}(\sqrt{d})$ and apply it independently at each spatial position:

$$z_{i,j} \leftarrow \pi(z_{i,j}) = \sqrt{d} \cdot \frac{z_{i,j}}{\|z_{i,j}\|}, \quad (i,j) \in \{1,\dots,h\} \times \{1,\dots,w\}. \tag{5}$$

Each token retains its direction and satisfies $\|z_{i,j}\| = \sqrt{d}$. The radius $\sqrt{d}$ matches the concentration radius of a standard Gaussian in d dimensions and sets a common radius for projected data and the spherical noise prior. The full latent tensor lies on a product of $h \cdot w$ copies of $\mathcal{S}^{d-1}(\sqrt{d})$, one for each spatial position.

The pretrained decoder was trained on variable-radius latents. To adapt it, we encode a real image and project its tokens, then train the decoder to reconstruct that image from the projected tokens. With the encoder frozen, we finetune the decoder and discriminator for five epochs on ImageNet. We use FLUX.2 (Black Forest Labs, 2025) (f=8), VA-VAE (Yao et al., 2025) (f=16, DINOv2-aligned), and REPA-E FLUX.1 (Leng et al., 2025) (f=8), where f denotes the spatial downsampling factor.

The reconstruction objective retains the original pixel L_1, LPIPS, and patch-level adversarial losses (Isola et al., 2017; Esser et al., 2021). We omit the KL term because the encoder is frozen and the term is independent of the trainable decoder parameters. We also omit the VA-VAE DINOv2 alignment terms, whose gradients are zero once the encoder is frozen. The alignment learned during pretraining remains in the encoder weights.

After decoder adaptation, we train the flow model to transport spherical noise to the projected encoder latents. At generation time, the learned flow produces a latent grid, which the adapted decoder maps to an image.

We report rFID for reconstructions of real images and FID for generated images. The validation study compares original decoders with vanilla and spherical decoders finetuned under matched budgets (Tab. A7); projection changes rFID relative to finetuned vanilla by $+0.586$, -0.052, and $+0.065$ for FLUX.2, VA-VAE, and REPA-E FLUX.1, respectively. Separately, the 50K FLUX.2 decode-only control holds projected real latents fixed and uses the generation training reference: adaptation reduces rFID from 17.37 to 0.882 (Tab. A2). These protocols are not directly comparable. Generation controls are consolidated in Tab. 2, with the full decoder swaps in Tab. A6.

The projected latents retain their $d \times h \times w$ grid, and SiT's architecture is unchanged. The base construction uses no auxiliary encoder; optional REPA experiments add frozen DINOv2 (Oquab et al., 2023) supervision during flow training (Tab. 4).

4.2 Learning and sampling on the sphere

The projected data tokens from Sec. 4.1 lie on $\mathcal{S}^{d-1}(\sqrt{d})$. To obtain noise on the same sphere, we independently sample $\epsilon \sim \mathcal{N}(0, I_d)$ at each spatial position and set

$$z_0 = \sqrt{d}\,\frac{\epsilon}{\|\epsilon\|_2}. \tag{6}$$

By rotational invariance of the isotropic Gaussian, $\epsilon/\|\epsilon\|_2$ is uniform on $\mathcal{S}^{d-1}$, so $z_0 \sim \mathrm{Uniform}(\mathcal{S}^{d-1}(\sqrt{d}))$. Normalization preserves the Gaussian direction while fixing its radius. Independent draws across spatial positions give a product of uniform spherical priors, matching the token-wise geometry of the projected data; Sec. A.2 gives the derivation. This is the uniform prior used in Riemannian flow matching and directional statistics (Chen & Lipman, 2023; Mardia & Jupp, 2009).

Slerp path. We interpolate the noise and data tokens at corresponding spatial positions using spherical linear interpolation (slerp) (Shoemake, 1985). Write $\hat{z} = z/\|z\|$, so that $z = \sqrt{d}\,\hat{z}$, and let $\omega = \arccos\langle \hat{z}_0, \hat{z}_1 \rangle$ be the endpoint angle. For $0 < \omega < \pi$, the path is

$$z_t = \sqrt{d} \cdot \mathrm{slerp}(\hat{z}_0, \hat{z}_1;\, t) = \sqrt{d}\left(\frac{\sin((1-t)\,\omega)}{\sin\omega}\,\hat{z}_0 + \frac{\sin(t\,\omega)}{\sin\omega}\,\hat{z}_1\right). \tag{7}$$

This is the unique shortest geodesic between the endpoints (do Carmo Valero, 1992). It moves at constant angular speed and keeps each token at radius $\sqrt{d}$ throughout the path. The treatment of nearly coincident and antipodal endpoints is described in Sec. B.1.

Velocity target and loss. Following Chen & Lipman (2023), we obtain the conditional velocity target by differentiating the path and projecting onto the tangent space $T_{z_t}\mathcal{S}^{d-1}(\sqrt{d})$:

$$u_t = \frac{d}{dt}\, z_t, \qquad u_t \leftarrow u_t - \frac{\langle u_t, z_t\rangle}{\|z_t\|^2}\, z_t. \tag{8}$$

The slerp derivative is already tangent in exact arithmetic; the explicit projection safeguards against radial roundoff. Because tangent vectors are orthogonal to the current token z_t, the target has zero radial component and is purely angular (Fig. A1).

SiT predicts an ambient velocity $v_\theta(z_t, t, y)$. We apply the same token-wise tangent projection to its output before comparing it with the projected target (Chen & Lipman, 2023; Rozen et al., 2021). The training loss is

$$\mathcal{L}_{\text{slerp}} = \mathbb{E}_{t,\, z_0,\, z_1,\, y}\left[\left\|\Pi_{z_t}\, v_\theta(z_t, t, y) - \Pi_{z_t}\, u_t\right\|^2\right], \qquad \Pi_{z_t}\, v = v - \frac{\langle v, z_t\rangle}{\|z_t\|^2}\, z_t. \tag{9}$$

Here Π_{z_t} removes the component parallel to z_t, so the loss trains the tangent component of the prediction. The endpoint pair determines the training path and target. SiT predicts the velocity from the current latent grid z_t, time t, and class label y. Section B.2 specifies the time-sampling and optimization settings used in the experiments.

Linear path on spherical latents. We also apply the Euclidean interpolation $z_t = (1-t)\, z_0 + t\, z_1$ from Sec. 3.3 to projected data and the uniform spherical prior. For distinct endpoints, the intermediate states lie inside the sphere. This variant uses the Euclidean loss in Eq. 3, with unprojected velocity targets and predictions, and the Linear solver. By retaining the spherical data representation, prior, and adapted decoder, this Spherical-Linear control tests the spherical construction under linear flow matching (Tab. 2 and Sec. 5).

Shell path. We interpolate direction and radius separately for vanilla data latents and a Gaussian prior, retaining their original endpoint radii (Fig. 4). This control uses explicit Euler (Tab. 2). Writing $z = r\,\hat{z}$, the path is

$$\begin{aligned} \hat{z}_t &= \text{slerp}(\hat{z}_0, \hat{z}_1;\, t), \\ r_t &= (1-t)\, r_0 + t\, r_1, \qquad z_t = r_t\, \hat{z}_t. \end{aligned} \tag{10}$$

The target is the path derivative, with radial velocity $\dot{r}_t = r_1 - r_0$. The radius avoids the inward chord dip (Fig. 3a), but unequal endpoint radii retain radial motion. If $r_0 = r_1 = \sqrt{d}$, the path reduces to spherical-slerp with a purely tangential target.

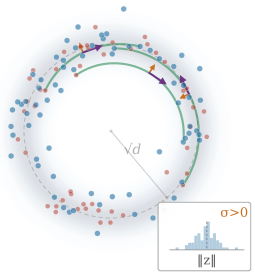


Figure 4: Shell path: separate radius and direction interpolation.

Sampling. Starting from the spherical prior, generation uses first-order exponential-map Euler to integrate the tangent field from $t = 0$ to $t = 1$. Its token-wise update at step size Δt is

$$z_{t+\Delta t} = \exp_{z_t}(\Pi_{z_t}\, v_\theta(z_t, t, y) \cdot \Delta t)\,. \tag{11}$$

This update preserves token radius (Sec. A.3). Vanilla-linear, vanilla-shell, and spherical-linear generation use explicit Euler, $z_{t+\Delta t} = z_t + \Delta t\, v_\theta(z_t, t, y)$, without radial projection. The separate inversion study uses two-stage Heun (Sec. A.4).

For the known conditional slerp velocity u_t, the exponential-map step reaches the corresponding point $z_{t+\Delta t}$ exactly along that slerp curve. With the same conditional velocity, an ambient linear step followed by radial projection stays on the same great circle and advances less far by $\sqrt{d}\,\big(\Delta t\,\omega - \arctan(\Delta t\,\omega)\big)$ in arc length (Sec. A.3). Table A8 compares solver configurations at equal sampling-step budgets.

5 Experiments

We compare spherical-slerp with vanilla-linear in controlled FLUX.2 ablations and evaluate transfer across tokenizer families and backbone scales.

Experimental setup. Within each vanilla-linear and spherical-slerp pair, we keep the architecture, data, training budget, and evaluator fixed. We use ImageNet-256 class-conditional generation (Russakovsky et al., 2015) with the SiT family (Ma et al., 2024): B/2 and XL/2 on FLUX.2, B/1 and XL/1 on VA-VAE, and B/2 on REPA-E FLUX.1. We adjust patch size to each tokenizer's downsampling factor so that all settings yield 256 tokens per 256×256 image. Models train from scratch for 80 epochs at batch size 256 in bfloat16 with AdamW (Loshchilov & Hutter, 2019). The spherical variant applies the token-wise projection after each tokenizer's fixed preprocessing, before decoder finetuning and flow training. The encoder and spherical projection remain frozen throughout diffusion training. The appendix reports further ablations.

All generation comparisons, including the sampling sweeps and recipe ablations, use first-order Euler: explicit for Linear paths and exponential-map Euler for the default Spherical-slerp method. Only the inversion study uses two-stage Heun (Sec. A.4). Generation FID-50K uses the ImageNet-256 training reference. Headline, REPA, and 200-epoch results use 250 updates; the transport ablation, training curve, recipe and decoder controls use 50 updates (Sec. B.3). The approximately $2.2\times$ improvement measures flow-training steps to FID$= 30$ at fixed sampling settings.

Table 2: **Latent construction, decoder adaptation, and transport controls.** ImageNet-256 FID-50K on FLUX.2 SiT-B/2 after 80 epochs, at CFG$= 1.0$ with 50 first-order sampling updates. Linear and shell paths use explicit Euler; spherical-slerp uses tangent projection and exponential-map Euler. Vanilla and spherical decoder finetunes use the same adaptation budget.

Latents	Prior	Transport	Decoder	FID↓
Vanilla	Gaussian	Linear	Original	26.35
Vanilla	Gaussian	Linear	Finetuned on vanilla latents	26.95
Vanilla	Gaussian	Shell	Original	27.26
Projected	Spherical uniform	Linear	Adapted to projected latents	22.85
Projected	Spherical uniform	Slerp	Original	43.17
Projected	Spherical uniform	Slerp	Adapted to projected latents	**20.55**

Reconstruction and generation quality. Relative to the matched finetuned-vanilla control, projection increases validation rFID for FLUX.2 and REPA-E FLUX.1 and slightly lowers it for VA-VAE (Tab. A7). The separate decode-only control holds projected real latents fixed while comparing original and adapted decoders (Tab. A2); it measures decoder adaptation without flow sampling.

Latent construction and transport controls. Finetuning on vanilla latents gives FID 26.95, compared with 26.35 for the original decoder. For spherical flow on projected latents, decoder adaptation improves FID from 43.17 to 20.55. With an adapted decoder, the projected representation and spherical prior already improve FID to 22.85 under linear transport. The complete spherical-slerp pipeline reaches 20.55 (Tab. 2). These controls support the combined latent construction and decoder adaptation, with a further improvement from the spherical transport configuration. Shell transport on vanilla latents gives 27.26, above the baseline (Sec. C.1).

Generation across tokenizers and scales. The controlled FLUX.2 B/2 comparison uses matched time sampling, timestep shift, optimizer, and training budget. Spherical-slerp achieves FID 20.55 versus 26.35 for vanilla-linear under this matched recipe (Tab. A5). The remaining comparisons evaluate transfer of the complete spherical recipe, using logit-normal time sampling for Spherical and uniform time sampling for the canonical Linear baselines. The transferred recipe obtains lower FID across the evaluated tokenizer and backbone settings at matched CFG (Tab. 3), and across all sampling-step budgets in the B-scale sweep (Tab. A8).

Scaling. At XL scale, the transferred spherical recipe outperforms the linear baseline at both guidance settings on FLUX.2 and VA-VAE (Tab. 3), with unchanged diffusion architectures and no auxiliary encoder.

Extended training. At 200 epochs on FLUX.2 SiT-B/2, Spherical achieves FID 8.35 versus Linear's 9.15 at CFG$= 1.0$, and 2.91 versus 3.22 at CFG$= 1.5$ (Tab. A3). The spherical advantage persists beyond the 80-epoch evaluation budget.

Representation alignment. We compare both methods with a matched REPA objective (Yu et al., 2024) on FLUX.2 SiT-B/2. Architecture, data, seed, and REPA configuration are shared within each pair. REPA improves both pipelines, and Spherical retains lower FID at both guidance settings (Tab. 4). With REPA, Spherical reduces FID from 21.91 to 18.47 at CFG$= 1.0$ and from 7.24 to 6.03

Table 3: ImageNet-256 FID-50K and IS across tokenizers and scales at 80 epochs and 250 sampling updates, at matched CFG. Training-recipe parity is established for the controlled FLUX.2 B/2 setting; the remaining tokenizer and scale comparisons evaluate complete recipes (Sec. B.2).

Setting			FID↓			IS↑	
Tokenizer	Scale	CFG	Linear	Spherical	Rel. Δ	Linear	Spherical
FLUX.2	B/2	1.0	26.14	20.21	−22.7%	52.5	66.5
		1.5	9.99	6.29	−37.0%	141.8	169.7
	XL/2	1.0	13.00	10.71	−17.6%	85.4	102.3
		1.5	4.32	3.95	−8.6%	208.1	238.9
VA-VAE	B/1	1.0	26.99	21.96	−18.6%	49.0	58.2
		1.5	10.86	7.81	−28.1%	120.7	143.4
		2.0	6.84	5.46	−20.2%	199.6	231.6
	XL/1	1.0	12.16	10.36	−14.8%	93.0	107.3
		1.5	4.23	3.88	−8.2%	227.9	263.2
REPA-E FLUX.1	B/2	1.0	38.00	26.07	−31.4%	42.9	63.1
		1.5	13.83	6.88	−50.2%	118.3	172.1
		2.0	7.58	5.43	−28.4%	204.2	274.0

at CFG$= 1.5$. Thus latent-support adaptation remains useful alongside representation alignment, rather than requiring that alignment supervision be removed or replaced. The appendix gives the alignment settings.

Table 4: FLUX.2 SiT-B/2 REPA comparison: FID-50K at 250 sampling updates; architecture, data, seed, and REPA configuration are shared within pairs.

		No REPA		+REPA	
Metric	CFG	Linear	Spherical	Linear	Spherical
FID↓	1.0	26.14	20.21	21.91	**18.47**
FID↓	1.5	9.99	6.29	7.24	**6.03**

6 Conclusion

Component-swap probes show that retaining token direction preserves more decoded similarity than retaining radius. We use this observation to construct fixed-radius latents from pretrained VAEs, adapting the decoder while keeping the encoder frozen. Projected Gaussian directions provide a uniform spherical prior. Spherical interpolation keeps the training path on the same support as its endpoints, and exponential-map Euler updates preserve that support during sampling. The base construction retains the diffusion architecture and requires no auxiliary representation encoder.

The controlled FLUX.2 experiments support the combined latent construction and decoder adaptation. Under linear transport, the projected representation, spherical prior, and adapted decoder improve FID from 26.35 to 22.85; spherical-slerp further reduces it to 20.55 under a matched training recipe. The decoder controls complement this comparison: finetuning on vanilla latents yields 26.95, while spherical flow with the original decoder yields 43.17. Adapting the decoder to the projected support is therefore an integral part of the construction. These comparisons connect the geometric intervention to generation quality while retaining the pretrained encoder and the same diffusion backbone.

FLUX.2 gains persist with REPA and extended training. At 200 epochs and CFG$= 1.5$, Spherical reaches FID 2.91, compared with 3.22 for Linear. SiT-B/2 reaches FID$= 30$ in approximately $2.2\times$ fewer flow-training steps under the fixed evaluator. The experiments demonstrate the effectiveness of post-hoc latent-support adaptation for FLUX.2, VA-VAE, and REPA-E FLUX.1, including transfer across backbone scales and compatibility with representation alignment. Appendix C.8 characterizes tokenizer-dependent behavior.

AI USE STATEMENT

Generative-AI tools assisted with language editing. The authors checked the resulting prose and mathematics. They remain responsible for all content and conclusions.

ETHICS STATEMENT

We use the publicly available ImageNet-256 dataset and pretrained tokenizers under their respective licenses. The study involves no crowdsourcing, human-subject research, or deployment, and we collected no additional personal data. The class-conditional generators share the potential societal impacts and misuse risks of the underlying image-generation technology.

REPRODUCIBILITY STATEMENT

Sections 4.1 and 4.2 describe latent projection, decoder adaptation, and spherical flow training and sampling. The appendix provides the solver updates used for numerical inversion (Sec. A.4), numerical safeguards, training settings, evaluation protocols, compute measurements, and additional controls. We provide training and evaluation scripts and decoder-adaptation configurations for reproducing the experiments from scratch (Sec. B.5). The reported SiT-B and SiT-XL flow-matching checkpoints will be released after publication.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek J Bose. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Manfredo Perdigao do Carmo Valero. *Riemannian geometry*. 1992.

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

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

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

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

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*, pp. 1125–1134, 2017.

Guolin Ke and Hui Xue. Hyperspherical latents improve continuous-token autoregressive generation. *arXiv preprint arXiv:2509.24335*, 2025.

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

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. EQ-VAE: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025.

Amandeep Kumar and Vishal M Patel. Learning on the manifold: Unlocking standard diffusion transformers with representation encoders. *arXiv preprint arXiv:2602.10099*, 2026.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

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.

Jia-Wei Liao, Li-Xuan Peng, Mei-Heng Yueh, Min Sun, Cheng-Fu Chou, and Jun-Cheng Chen. Diffrgd: An inference-time diffusion guidance through riemannian gradient descent. *arXiv preprint arXiv:2606.28417*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 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*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

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

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in neural information processing systems*, 33:2503–2515, 2020.

M. Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, H. Jégou, J. Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2023. . URL `https://doi.org/10.48550/arxiv.2304.07193`.

Kai Qiu, Xiang Li, Hao Chen, Jason Kuen, Xiaohao Xu, Jiuxiang Gu, Yinyi Luo, Bhiksha Raj, Zhe Lin, and Marios Savvides. Image tokenizer needs post-training. *arXiv preprint arXiv:2509.12474*, 2025a.

Kai Qiu, Xiang Li, Jason Kuen, Hao Chen, Xiaohao Xu, Jiuxiang Gu, Yinyi Luo, Bhiksha Raj, Zhe Lin, and Marios Savvides. Robust latent matters: Boosting image generation with sampling error synthesis. *arXiv preprint arXiv:2503.08354*, 2025b.

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*, pp. 10684–10695, 2022.

Noam Rozen, Aditya Grover, Maximilian Nickel, and Yaron Lipman. Moser flow: Divergence-based generative modeling on manifolds. *Advances in neural information processing systems*, 34: 17669–17680, 2021.

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

Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 245–254, 1985.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

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

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

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

Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4503–4513, 2018.

Tongda Xu, Mingwei He, Shady Abu-Hussein, Jose Miguel Hernandez-Lobato, Haotian Zhang, Kai Zhao, Chao Zhou, Ya-Qin Zhang, and Yan Wang. Making reconstruction fid predictive of diffusion generation fid. *arXiv preprint arXiv:2603.05630*, 2026.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. *arXiv preprint arXiv:2402.03201*, 2024.

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

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Kaiyu Yue, Menglin Jia, Ji Hou, and Tom Goldstein. Image generation with a sphere encoder. *arXiv preprint arXiv:2602.15030*, 2026.

Olga Zaghen, Floor Eijkelboom, Alison Pouplin, Cong Liu, Max Welling, Jan-Willem van de Meent, and Erik J Bekkers. Riemannian variational flow matching for material and protein design. *arXiv preprint arXiv:2502.12981*, 2025.

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*, pp. 586–595, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Supplementary Material

A Analytical Derivations

A.1 Analytical Gaussian Norm Statistics

For the standard Gaussian reference in Sec. 3.1, the token norm follows a chi distribution. We derive the mean radius and coefficient of variation used in Tab. 1, distinguishing the mean from the root-mean-square radius.

Let

$$z \sim \mathcal{N}(0, I_d), \qquad R = \|z\|_2.$$

Because the entries of z are independent standard normal variables, we have

$$R^2 = \sum_{i=1}^{d} z_i^2 \sim \chi_d^2.$$

Therefore, the norm $R = \|z\|_2$ follows a chi distribution with d degrees of freedom.

The exact mean of a chi random variable with d degrees of freedom is

$$\mathbb{E}[R] = \sqrt{2}\,\frac{\Gamma((d+1)/2)}{\Gamma(d/2)}.$$

This is the value we use as the analytical Gaussian mean radius in Tab. 1.

The second moment is

$$\mathbb{E}[R^2] = \mathbb{E}\|z\|_2^2 = \sum_{i=1}^{d} \mathbb{E}[z_i^2] = d.$$

Thus, the variance of the Gaussian norm is

$$\mathrm{Var}(R) = \mathbb{E}[R^2] - \mathbb{E}[R]^2 = d - \left(\sqrt{2}\,\frac{\Gamma((d+1)/2)}{\Gamma(d/2)}\right)^2.$$

The coefficient of variation is therefore

$$\mathrm{CV}(R) = \frac{\sqrt{\mathrm{Var}(R)}}{\mathbb{E}[R]}.$$

These two quantities, $\mathbb{E}[R]$ and $\mathrm{CV}(R)$, give the Gaussian entries reported in Tab. 1.

Relation to the $\sqrt{d}$ Gaussian shell. The Gaussian shell is often described as being located near radius $\sqrt{d}$. This comes from the root-mean-square norm:

$$\sqrt{\mathbb{E}\|z\|_2^2} = \sqrt{d}.$$

However, the mean norm is slightly smaller:

$$\mathbb{E}\|z\|_2 < \sqrt{\mathbb{E}\|z\|_2^2} = \sqrt{d},$$

where the inequality follows from Jensen's inequality.

For large d, the exact mean has the expansion

$$\mathbb{E}\|z\|_2 = \sqrt{d}\left(1 - \frac{1}{4d} + O(d^{-2})\right).$$

Equivalently,

$$\mathbb{E}\|z\|_2 = \sqrt{d} - \frac{1}{4\sqrt{d}} + O(d^{-3/2}).$$

Thus, the difference between $\mathbb{E}\|z\|_2$ and $\sqrt{d}$ becomes negligible in high dimensions.

Squaring this expansion gives

$$(\mathbb{E}\|z\|_2)^2 = d - \tfrac{1}{2} + O(d^{-1}),$$

which motivates the closed-form approximation

$$\mathbb{E}\|z\|_2 \approx \sqrt{d - \tfrac{1}{2}},$$

accurate to $O(d^{-3/2})$. This approximation is accurate at the dimensions used in our experiments. For example,

$$d = 16: \qquad \sqrt{d - \frac{1}{2}} = 3.937, \qquad \mathbb{E}\|z\|_2 = 3.938,$$

and

$$d = 32: \qquad \sqrt{d - \frac{1}{2}} = 5.612, \qquad \mathbb{E}\|z\|_2 = 5.613.$$

The Gaussian rows in Tab. 1 therefore use the exact mean norm, rather than the root-mean-square radius $\sqrt{d}$.

For $z \sim \mathcal{N}(0, I_d)$ and $t > 0$, the concentration bound used in the main paper is

$$\mathbb{P}\left(\left|\|z\|_2 - \sqrt{d}\right| > t\right) \le 2\exp(-ct^2),$$

for an absolute constant $c > 0$ (Vershynin, 2018, Theorem 3.1.1). At any fixed confidence level, this bound gives a shell of $\mathcal{O}(1)$ width around radius $\sqrt{d}$, so its relative thickness decreases as $\mathcal{O}(1/\sqrt{d})$. This analytical bound assumes a standard Gaussian token; concentration of the pretrained VAE token radii is measured empirically, not inferred from the Gaussian bound.

A.2 Projected Gaussian Noise is Uniform on the Sphere

The spherical flow uses a noise endpoint on the fixed-radius sphere $\mathcal{S}^{d-1}(\sqrt{d})$. This prior is the radial projection of the same isotropic Gaussian noise used by the Euclidean baseline.

Let $\epsilon \sim \mathcal{N}(0, I_d)$ and define

$$\pi_R(\epsilon) = R\,\frac{\epsilon}{\|\epsilon\|_2}, \qquad R > 0.$$

Since $\mathbb{P}(\epsilon = 0) = 0$, this is well-defined almost surely. We show that

$$\pi_R(\epsilon) \sim \mathrm{Uniform}(\mathcal{S}^{d-1}(R)).$$

The isotropic Gaussian law is invariant under every orthogonal transformation, and radial projection commutes with these transformations. Its projected law is therefore rotationally invariant, which characterizes the uniform probability measure on the sphere. The polar-coordinate calculation below also establishes independence of direction and radius.

In polar coordinates, write $\epsilon = ru$ with $r = \|\epsilon\|_2 \in [0, \infty)$ and $u \in \mathcal{S}^{d-1}$. The Lebesgue measure factorizes as $d\epsilon = r^{d-1}\,dr\,d\sigma(u)$, where $d\sigma$ is the surface measure on $\mathcal{S}^{d-1}$; after normalization, $d\sigma/\sigma(\mathcal{S}^{d-1})$ is the uniform probability measure. The Gaussian density therefore satisfies

$$p(\epsilon)\,d\epsilon = (2\pi)^{-d/2}\exp\left(-\frac{r^2}{2}\right) r^{d-1}\,dr \cdot d\sigma(u),$$

which factorizes into a radial term and a part constant in u. Hence $u = \epsilon/\|\epsilon\|_2$ is uniform on $\mathcal{S}^{d-1}$ and independent of r, and

$$\pi_R(\epsilon) = Ru \sim \mathrm{Uniform}(\mathcal{S}^{d-1}(R)) \qquad \text{for every } R > 0.$$

Setting $R = \sqrt{d}$ gives the spherical noise endpoint used in the main paper.

For a latent tensor with $N = h \cdot w$ spatial positions, let $\{\epsilon_{i,j}\}$ be *independent* with $\epsilon_{i,j} \sim \mathcal{N}(0, I_d)$. The token-wise projection is a measurable coordinatewise map, so independence is preserved and the full noise tensor

$$z_0 = \left\{\sqrt{d}\,\frac{\epsilon_{i,j}}{\|\epsilon_{i,j}\|_2}\right\}_{i,j}$$

is sampled from the product measure $\bigotimes_{i,j} \mathrm{Uniform}(\mathcal{S}^{d-1}(\sqrt{d}))$. The spherical prior keeps the Gaussian direction and discards only its radius.

A.3 Exponential-Map Integration for Slerp Targets

We derive the one-step identity in Sec. 4.2 for the known conditional velocity along a slerp path. An exponential-map step reaches the corresponding point on that path exactly; an ambient linear step followed by radial projection stays on the same great circle but advances less far. This comparison concerns the conditional path, not exact integration of the learned velocity field.

We write R for the sphere radius, θ_0 for the endpoint angle, and $h \in [0,\ 1-t]$ for the step size. These correspond to $\sqrt{d}$, ω, and Δt in Sec. 4.2.

Setup. Let $z_0, z_1 \in \mathcal{S}^{d-1}(R)$ with $\langle z_0, z_1\rangle = R^2\cos\theta_0$ and $\theta_0 \in (0,\pi)$. The slerp path is

$$z_t \;=\; \frac{\sin((1-t)\theta_0)}{\sin\theta_0}\, z_0 \;+\; \frac{\sin(t\theta_0)}{\sin\theta_0}\, z_1, \qquad t \in [0,1],$$

which traces the great circle from z_0 to z_1 at constant angular speed θ_0. Differentiating in t gives the velocity

$$\dot z_t \;=\; \frac{\theta_0}{\sin\theta_0}\Big(-\cos((1-t)\theta_0)\, z_0 \;+\; \cos(t\theta_0)\, z_1\Big),$$

which is tangent to the sphere at z_t and has constant norm $\|\dot z_t\| = R\,\theta_0$.

Spherical update. At a point $p \in \mathcal{S}^{d-1}(R)$, the exponential map in the direction of a tangent vector $v \in T_p\mathcal{S}^{d-1}(R)$ is

$$\exp_p(v) \;=\; \cos\Big(\tfrac{\|v\|}{R}\Big)\, p \;+\; R\sin\Big(\tfrac{\|v\|}{R}\Big)\,\frac{v}{\|v\|}.$$

At $v = 0$, the expression is defined by continuity and returns p. Stepping from z_t along the true velocity with step size h uses $\|h\dot z_t\| = hR\theta_0$, so

$$\exp_{z_t}(h\dot z_t) \;=\; \cos(h\theta_0)\, z_t \;+\; \sin(h\theta_0)\,\frac{\dot z_t}{\theta_0}.$$

Substituting the slerp expressions for z_t and $\dot z_t$ and applying the angle-addition identities for $\sin$ collapses the right-hand side to

$$\exp_{z_t}(h\dot z_t) \;=\; \frac{\sin((1-t-h)\theta_0)}{\sin\theta_0}\, z_0 \;+\; \frac{\sin((t+h)\theta_0)}{\sin\theta_0}\, z_1 \;=\; z_{t+h},$$

the slerp value at time $t+h$. This identity assumes the velocity of the specified endpoint-conditioned slerp curve. The learned field depends on the current state, time, and class rather than the endpoint pair, and its integral curves need not be constant-speed great circles. Even an exact fit to the flow-matching regression field therefore does not make a finite exponential-map step an exact learned-ODE solution.

Linear update followed by radial projection. The Linear update produces $\tilde z = z_t + h\,\dot z_t$, which lies in the plane spanned by $\{z_t,\ \dot z_t/\theta_0\}$, the same plane that contains the slerp great circle. Since $z_t \perp \dot z_t$ and $\|\dot z_t\| = R\theta_0$, its squared norm is

$$\|\tilde z\|^2 \;=\; R^2 + h^2R^2\theta_0^2 \;=\; R^2\big(1+h^2\theta_0^2\big).$$

Radial projection $\Pi(\tilde z) = R\,\tilde z/\|\tilde z\|$ then gives

$$\Pi(\tilde z) \;=\; \frac{1}{\sqrt{1+h^2\theta_0^2}}\, z_t \;+\; \frac{h\theta_0}{\sqrt{1+h^2\theta_0^2}}\,\frac{\dot z_t}{\theta_0}.$$

Matching this against the exponential-map form $\cos\phi\, z_t + \sin\phi\,\dot z_t/\theta_0$ identifies the projected Linear update as a rotation of z_t on the same great circle, but at the smaller angular displacement $\arctan(h\theta_0)$ rather than $h\theta_0$.

One-step deficit. Both updates stay on the great circle through z_t and z_{t+h}. The Spherical update advances by arc length $R\,h\theta_0$, while the projected Linear update advances by $R\,\arctan(h\theta_0)$. The one-step arc-length deficit is therefore

$$R\big(h\theta_0 - \arctan(h\theta_0)\big) \;=\; R\,\frac{(h\theta_0)^3}{3} \;+\; O\big((h\theta_0)^5\big),$$

which reduces to the expression quoted in Sec. 4.2 after substituting $R = \sqrt{d}$, $\theta_0 = \omega$, $h = \Delta t$. The expansion is for $h\theta_0 \to 0$ at fixed R. The stated range $0 \le h \le 1 - t$ ensures that $0 \le h\theta_0 < \pi$, so the comparison is along the specified shortest slerp arc, without wraparound.

For reverse steps with $-t \le h < 0$, the exponential-map identity still holds, with $\|h\dot{z}_t\| = |h|R\theta_0$. The quantity $R(h\theta_0 - \arctan(h\theta_0))$ is then a signed displacement difference, not a positive arc-length deficit; the deficit in distance advanced is $R(|h|\theta_0 - \arctan(|h|\theta_0))$. At coincident endpoints, the continuous limit is a constant path with zero velocity, and both updates leave the state unchanged. Antipodal endpoints are excluded from this derivation because their shortest geodesic is not unique; the endpoint handling used in training is described in Sec. B.1.

A.4 Heun Solver for Numerical Inversion

Unlike first-order generation sampling (Secs. B.3 and 4.2), the inversion study uses a two-stage, sphere-aware Heun update. From state z_n at time t_n with signed step h, define

$$k_1 = \Pi_{z_n} v_\theta(z_n, t_n, y), \qquad \widetilde{z} = \mathrm{Exp}_{z_n}(hk_1), \tag{12}$$

$$k_2 = \Pi_{\widetilde{z}} v_\theta(\widetilde{z}, t_n + h, y), \qquad z_{n+1} = \mathrm{Exp}_{z_n}\left(\frac{h}{2}\left[k_1 + \mathcal{P}_{\widetilde{z}\to z_n}(k_2)\right]\right). \tag{13}$$

The Linear round-trip control uses the corresponding Euclidean Heun update, with $\ell_1 = v_\theta(x_n, t_n, y)$, $\widetilde{x} = x_n + h\ell_1$, $\ell_2 = v_\theta(\widetilde{x}, t_n + h, y)$, and $x_{n+1} = x_n + \frac{h}{2}(\ell_1 + \ell_2)$. Both methods evaluate their respective velocity fields twice per Heun step; the 50- and 250-step round trips in Tab. A12 therefore do not use the first-order generation solvers. Here Π_z is the tangent projection defined in Sec. 4.2, and Exp is the exponential map above, including its zero-vector limit. Parallel transport maps the second-stage velocity k_2 from the tangent space at the predicted point $\widetilde{z}$ back to the tangent space at z_n, where it is averaged with k_1. For unit directions $p = \widetilde{z}/R$ and $q = z_n/R$, let $c = \langle p, q\rangle$, $s = \sqrt{1 - c^2}$, and $w = (q - cp)/s$. For distinct, non-antipodal points, transport along their unique shortest geodesic is

$$\mathcal{P}_{\widetilde{z}\to z_n}(v) = v + \langle v, w\rangle\big[(c-1)w - sp\big]. \tag{14}$$

Near-coincident points use the identity map as a small-angle approximation to the coincident-point limit. The displayed formula is undefined for antipodal points, where a shortest geodesic is not unique.

For a time-continuous tangent field that is Lipschitz in the state, the continuous ODE on the sphere has unique trajectories and can be inverted by integrating backward in time. Numerical inversion uses the same two-stage update with $h < 0$, but reversing the step sign does not in general invert a finite forward update exactly. The matched round-trip measurements in Tab. A12 quantify the solver's numerical inversion error on latent trajectories.

B Implementation Details

B.1 Slerp Numerical Handling

Per-token slerp treats each spatial position's d-dimensional vector as a point on $\mathcal{S}^{d-1}(\sqrt{d})$. For numerical evaluation, we clamp the endpoint cosine to $[-1 + 10^{-6}, 1 - 10^{-6}]$ before applying $\arccos$ and use a token-norm floor of 10^{-8}. This regularizes the interpolation angle and keeps the $\sin\omega$ denominator nonzero. A near-antipodal safeguard uses the cutoff $\omega > \pi - 0.1$. In a census of $10{,}289{,}152$ training pairs, the maximum angle was 2.460 radians, so no pair reached this cutoff.

B.2 Training Hyperparameters

We select the spherical recipe on SiT-B/2 with FLUX.2 and transfer it to the cross-tokenizer and cross-scale spherical comparisons, adapting only the timestep shift by tokenizer. The controlled FLUX.2

B/2 comparison matches time sampling, timestep shift, optimizer, and training budget (Tab. A5). The remaining comparisons evaluate transfer of the complete spherical recipe, using logit-normal time sampling for Spherical and uniform time sampling for the canonical Linear baselines. Table A1 lists the shared controlled setting and the transferred spherical recipe.

Table A1: Shared hyperparameters in the controlled FLUX.2 B/2 comparison and the transferred spherical recipe.

Setting	Value
Optimizer	AdamW (Loshchilov & Hutter, 2019), β_1=0.9, β_2=0.999
Weight decay	0
EMA decay	0.9999, averaged from step 0, no warmup
Mixed precision	bf16 (parameters and EMA in fp32; forward/backward autocast)
Global batch size	256
Training duration	80 epochs
Learning rate	4×10^{-4}
LR schedule	constant, no warmup
Gradient clip	1.0
Class-label dropout	0.1 (classifier-free guidance training)
Time sampling	logit-normal
Timestep shift	4.63 (FLUX.2, REPA-E FLUX.1), 1.0 (VA-VAE)
Training-curve FID-50K	50K samples, CFG= 1.0, every 50K steps
Final FID-50K	50K samples, 250 updates, CFG values per Tab. 3

B.3 Evaluation Protocol

We report sampling steps, where one step denotes one numerical solver update. All generation experiments use first-order explicit Euler or exponential-map Euler. With classifier-free guidance, each sampling step computes conditional and unconditional network predictions, corresponding to two neural-network evaluations per generated sample. Accordingly, 250 sampling steps with CFG correspond to 500 neural-network evaluations. Within each comparison, Linear and Spherical use the same number of sampling steps and the same guidance scale.

The final evaluations (Tabs. 3 to 4) use 250 sampling steps, while the training-curve and FLUX.2 ablations (Fig. 1c and Tabs. 2 and A4 to A6) use 50 steps. The sweeps (Tabs. A8 and A9) also report sampling steps, with the guidance scales stated alongside the results.

Generation FID uses 50K generated samples and the ImageNet-256 training reference. Reconstruction rFID instead evaluates decoded real-image latents: Tab. A7 reports the ImageNet-256 validation reconstruction study with ImageNet reference images, whereas the separate 50K decode-only study in Tab. A2 uses the training reference used for generation FID.

B.4 Compute

The flow-training runs for the reported quality comparisons use NVIDIA H200 GPUs; the separate runtime benchmarks in Sec. C.9 use H100 GPUs. On H200, SiT-B runs use 2 GPUs each and SiT-XL runs use 4 GPUs each. Token counts are equalized across tokenizers by pairing the patch size with the VAE downsample factor: SiT-B/1 and XL/1 for VA-VAE (downsample factor 16), and SiT-B/2 and XL/2 for FLUX.2 and REPA-E FLUX.1 (downsample factor 8). These settings yield 256 tokens per 256×256 image.

A B-scale run completes 80 epochs in approximately 6.5 wall-clock hours on $2\times$H200 ($\approx$ 13 H200-hours); an XL-scale run completes in $\approx$ 16.8 hours on $4\times$H200 ($\approx$ 67 H200-hours).

B.5 Code and Checkpoint Release

We provide training and evaluation scripts and finetuning configurations for the FLUX.2, VA-VAE, and REPA-E FLUX.1 spherical decoders so the experiments can be reproduced from scratch. The trained SiT-B and SiT-XL flow-matching checkpoints reported in Tab. 3 will be released after publication.

C Additional Experiments and Ablations

C.1 Transport-Path Interpretation

On SiT-B/2 with the FLUX.2 VAE, the spherical construction—fixed-radius projection, a uniform spherical prior, and decoder adaptation—accounts for most of the measured FID reduction from vanilla-linear to spherical-slerp (Tab. 2). The Spherical-Linear row tests this construction under Euclidean transport; Spherical-Slerp additionally changes the path, tangent projection, and solver, and gives the lowest FID in this ablation. Shell-decomposed transport on vanilla latents (Fig. 4) does not improve the baseline: preserving the original endpoint radii retains a radial velocity target even when the path avoids the inward chord dip. Our construction instead fixes both endpoint radii and removes radial motion throughout the path. This design preserves the directional component favored by the component-swap probes (Fig. 2) and, with decoder adaptation and spherical transport, reaches the target FID in fewer training steps (Fig. 1c).

C.2 Additional Generation and Reconstruction Results

Decode-only reconstruction control. Table A2 evaluates each decode path on 50K real FLUX.2 latents from ImageNet-256, without diffusion sampling, against the ImageNet-256 training reference. Applying the fixed-radius projection to the original decoder raises rFID from 0.894 to 17.37, while adapting the decoder on projected latents brings it back to 0.882, close to the original decode path. This control demonstrates the importance of adapting the decoder to the projected representation. The validation reconstruction study in Tab. A7 separately compares the three tokenizers under its own reconstruction protocol.

Table A2: Decode-only reconstruction control: rFID for 50K real FLUX.2 latents against the same ImageNet-256 training reference used for generation FID.

Latent supplied to decoder	Decoder	rFID↓
z	Original	0.894
$\pi(z)$	Original	17.37
$\pi(z)$	Adapted spherical	**0.882**

Extended training. The 200-epoch comparison in Tab. A3 tests whether the observed ordering persists beyond the 80-epoch budget. At this longer horizon, Spherical has lower FID at both reported guidance settings.

Table A3: ImageNet-256 FID-50K at 200 epochs on SiT-B/2 with the FLUX.2 VAE; 250 sampling updates.

Method	CFG$=1.0$	CFG$=1.5$
Linear	9.15	3.22
Spherical	8.35	2.91

Compatibility with representation alignment. For the REPA comparison in Tab. 4, both models use frozen DINOv2-B patch-token targets and align block 4 of the 12-block SiT-B backbone. A three-layer projector ($768 \to 2048 \to 2048 \to 768$, with SiLU between linear layers) is trained with a negative-cosine alignment loss weighted by 0.5. Architecture, data, seed, and alignment settings are shared within each pair; the latent construction and transport follow the respective Linear and Spherical pipelines.

C.3 Core Recipe Checks

Recipe ablations use FLUX.2 as the canonical tokenizer; cross-tokenizer transfer at B-scale is reported in Tab. 3. We control the time-sampling recipe (Tab. A5), decoder-only finetuning, and transfer to an unadapted decoder (Tab. A6). On FLUX.2 SiT-B/2, the spherical pipeline retains lower FID after recipe matching and relative to decoder finetuning alone. The broader tokenizer and scale comparisons evaluate transfer of the spherical recipe (Sec. B.2).

Table A4: Selected recipe checks on SiT-B/2 with the FLUX.2 spherical VAE and slerp path (80 epochs, FID-50K, CFG$= 1.0$); each row changes one method-relevant factor.

Configuration	FID↓
Spherical-slerp default	20.55
Uniform time sampling	23.86
Unit-radius sphere	22.08
No tangent projection	25.79

C.4 Recipe Parity

On FLUX.2 SiT-B/2, matching time sampling, timestep shift, optimizer, and training budget improves the vanilla-linear baseline (Tab. A5). Spherical-slerp achieves lower FID under this matched recipe. The Spherical pipeline combines latent projection, a spherical prior, decoder adaptation, spherical interpolation, tangent projection, and exponential-map integration.

Table A5: Recipe parity on SiT-B/2 with the FLUX.2 VAE (80 epochs, FID-50K, CFG$= 1.0$).

Configuration	Training recipe	FID↓
Vanilla-linear	Baseline	34.30
Vanilla-linear	Matched	26.35
Spherical-slerp	Matched	20.55

C.5 Decoder-Flow Coupling

On FLUX.2 B/2, decoder finetuning *without* projection gives a FID close to the vanilla baseline, whereas the complete spherical pipeline has lower FID (Tab. A6). Decoder finetuning alone therefore does not account for the observed improvement. Swapping decoders between vanilla and spherical flows increases FID for both pairings, indicating a mismatch between the generated latent distribution and the decoder's training distribution. The decoder and flow are trained separately: decoder adaptation reconstructs real images from projected encoder latents, and flow training subsequently models those latents with the encoder frozen.

Table A6: Decoder-flow coupling on FLUX.2 SiT-B/2 (CFG=1.0).

Latent	Path	Decoder	Finetune	Prior	FID↓
Vanilla	Linear	Vanilla	No	Gaussian	26.35
Vanilla	Linear	Vanilla	Matched	Gaussian	26.95
Spherical	Slerp	Spherical	Matched	Sph. Uniform	20.55
Vanilla	Linear	Spherical	Matched	Gaussian	112.17
Spherical	Slerp	Vanilla	No	Sph. Uniform	43.17

C.6 Reconstruction across tokenizers

The matched five-epoch decoder study in Tab. A7 evaluates ImageNet-256 validation reconstruction. Relative to finetuned vanilla, Spherical changes rFID by $+0.586$, -0.052, and $+0.065$ for FLUX.2, VA-VAE, and REPA-E FLUX.1, respectively. Reconstruction outcomes therefore vary by tokenizer, while the spherical recipe gives lower generation FID across all three families (Tab. 3). The separate 50K reconstruction control in Tab. A2 uses the training reference rather than this validation protocol.

Table A7: ImageNet-256 validation rFID. Spherical uses the fixed-radius projection with decoder/discriminator finetuning; finetuned vanilla (FT-vanilla) is the no-projection control.

Tokenizer			rFID↓		
VAE	d	f	Original	FT-vanilla	Spherical
FLUX.2	32	8	0.186	0.486	1.072
VA-VAE	32	16	0.298	1.116	1.064
REPA-E FLUX.1	16	8	0.209	0.759	0.824

C.7 SAMPLING-STEP SWEEP

Table A8 compares Linear and Spherical at equal sampling-step budgets and CFG$= 1.5$ for every B-scale pair. The 250-step column reproduces the corresponding B-scale results from Tab. 3. Spherical has lower FID at all three budgets for all three tokenizers, extending the generation improvement across the evaluated sampling budgets. The tokenizer-specific training recipes are described in Sec. B.2.

Table A8: FID-50K at CFG 1.5 with 50, 100, and 250 sampling steps for both methods, corresponding to 100, 200, and 500 neural-network evaluations, respectively.

		Sampling steps		
Tokenizer	Solver	50	100	250
FLUX.2 B/2	Linear	10.23	10.10	9.99
	Spherical	8.06	7.00	6.29
VA-VAE B/1	Linear	12.77	11.49	10.86
	Spherical	7.97	7.84	7.81
REPA-E FLUX.1 B/2	Linear	14.11	14.00	13.83
	Spherical	9.52	7.99	6.88

VA-VAE sampling-step sweep. Table A9 reports a complementary VA-VAE B/1 sweep at unguided sampling, whereas Tab. A8 compares all three tokenizers with guidance. Spherical has lower FID at every sampling-step budget in both sweeps.

Table A9: Sampling-step sweep on ImageNet-256 with VA-VAE B/1 at CFG$= 1.0$.

Sampling steps	Linear	Spherical	Gap
20	70.44	50.60	-19.84
50	31.99	23.88	-8.11
100	28.25	22.04	-6.21
500	27.26	21.96	-5.30

C.8 TOKENIZER-DEPENDENT BEHAVIOR

On the $d = 4$ SD-VAE, the linear and spherical pipelines obtain FIDs of 43.14 and 52.63, respectively, under matched SiT-B/2 architecture, data, training budget, seed, and evaluation at CFG$= 1.0$ with 250 sampling steps. This tokenizer has broader radial variation than the three tokenizer families evaluated in the main experiments. We encode 10,000 ImageNet-256 training images with the posterior mean and measure one channel vector per spatial position. Its token radius has CV 0.407, close to the broad χ_4 reference, and 8.77% of tokens lie below half the mean radius (Tab. A10). The Linear midpoint is 0.86σ from the endpoint radial distribution, a smaller standardized deviation than for the other tested tokenizers.

Table A10: Radius concentration across tokenizers. CV is $\mathrm{Std}[R]/\mathbb{E}[R]$; midpoint deviation is measured in endpoint-radius standard deviations.

Tokenizer	d	Radius CV	χ_d CV	Midpoint	$R < \frac{1}{2}\mathbb{E}[R]$
SD-VAE ft-ema	4	0.407	0.363	0.86σ	8.77%
REPA-E FLUX.1	16	0.234	0.178	1.31σ	0.20%
FLUX.2	32	0.201	0.126	1.50σ	0.014%
VA-VAE	32	0.166	0.126	1.81σ	0.0004%

C.9 Measured Runtime

The H100 measurements below are separate runtime benchmarks; the flow-training runs reported in Sec. B.4 use H200 GPUs. On 2×H100 GPUs with global batch size 256 and bf16, excluding warm-up, Linear processes 14.76 training steps/s and Spherical 14.29 steps/s. Extrapolating these rates to 400K steps gives approximately 7.5 and 7.8 hours, a 3% overhead. Combining the measured step-rate ratio with the $2.2\times$ fewer training steps to reach FID$=30$ under the fixed evaluator gives an estimated $2.1\times$ flow-training-time advantage on this hardware. This estimate excludes the preceding decoder adaptation and is neither a measured end-to-end time to threshold nor an end-to-end acceleration claim. On one H100 at batch size 50 and CFG 1.0, CUDA-synchronized timing after 10 warm-up batches gives the first-order sampling latency in Tab. A11; each entry averages 30 batches and per-batch standard deviation is below 0.5 ms.

Table A11: Per-image inference latency on one H100 at matched sampling budgets; decoder timing shown separately.

Measurement	Linear	Spherical
50 updates	62.4 ms	62.7 ms
250 updates	317.3 ms	318.9 ms
Decode	4.9 ms	4.9 ms

C.10 Optimal-Transport Pairing

For vanilla-linear transport, we tested exact Hungarian assignment in every minibatch with squared Euclidean distance over the full raw latent tensors, $c_{\mathrm{lin}}(X, Y) = \|X - Y\|_F^2$. At 80 epochs, this OT coupling gives FID 26.44, compared with 26.14 for independent coupling. Independent coupling performs slightly better in this comparison.

In a separate experiment, minibatch optimal-transport pairing is applied to spherical endpoints. We form an exact Hungarian assignment within each minibatch using the token-averaged cosine cost

$$c_{\mathrm{sph}}(X, Y) = 1 - \frac{1}{N}\sum_{p=1}^{N}\langle \widehat{X}_p, \widehat{Y}_p\rangle. \tag{15}$$

In a separate paired Spherical comparison, a schedule that applies this assignment to 25% of minibatches reaches 24.06 FID, compared with 20.40 for independent coupling. The assignment changes endpoint pairing while preserving spherical support. Independent coupling again gives lower FID, supporting its use in our default recipe.

C.11 Deterministic Inversion

The learned tangent ODE defines an invertible flow under the usual smoothness conditions. We numerically invert a data latent by integrating from $t{=}1$ to $t{=}0$ with the two-stage sphere-aware Heun update and a negative step, requiring neither retraining nor an auxiliary model (Sec. A.4). The Linear control uses Euclidean Heun with the same step counts. Reversing the step sign is not an exact inverse of either finite-step numerical map. On 512 real FLUX.2 latents, the methods have comparable round-trip error at 50 and 250 Heun steps; Spherical maintains fixed-radius support to numerical precision at the evaluated forward and backward integration states (Tab. A12). Whether this property improves downstream editing remains untested.

Table A12: Deterministic data→noise→data round trips with sphere-aware versus Euclidean Heun at equal step counts (two velocity evaluations per step). Error is mean relative latent L_2; radius deviation is the maximum of $|\|z\|_2 - \sqrt{d}|$ over samples, tokens, and integration times.

Measurement	Spherical	Linear
Round-trip error, 250 updates	4.5×10^{-5}	5.8×10^{-5}
Round-trip error, 50 updates	3.2×10^{-3}	3.3×10^{-3}
Max. absolute radius deviation	1.4×10^{-6}	7.3

C.12 Direction-vs-Radius Component Ablation

Figure 2 substitutes a same-class partner's direction or radius for the anchor's. The population-mean control in Fig. A2 instead uses the mean direction and radius computed across the whole ImageNet validation split. In particular, replacing radius by a dataset-wide constant tests how much decoded similarity is retained without token-specific radial variation. Each tokenizer reports LPIPS and DINOv2 cosine similarity to the original decode for four conditions: original, keep direction (own direction with substitute radius), keep radius (substitute direction with own radius), and full substitute. Lower LPIPS indicates greater perceptual similarity, while higher DINOv2 cosine indicates greater similarity in the representation used as a semantic proxy.

For FLUX.2 and REPA-E FLUX.1 with the population-mean substitute, keep-direction reaches LPIPS $\approx$ 0.04–0.08 and DINOv2 cosine $\approx$ 0.94–0.97, retaining high similarity to the original decode on both metrics. VA-VAE has a larger change (LPIPS $\approx$ 0.18, DINOv2 $\approx$ 0.86). Keep-radius lies at LPIPS $\approx$ 0.88–0.93 and DINOv2 $\approx$ 0.01–0.02 across all three tokenizers, close to the full mean substitute. Thus retaining the anchor's radius after replacing its direction provides little improvement in these measured similarities. The same-class substitute (Fig. 2) yields the same ordering: keep-direction remains closer to the original, while keep-radius nearly matches the full-neighbor swap.

Figures A3 and A4 plot the per-sample distributions underlying these averages, with one point per image. The x-axis is LPIPS for keep direction, and the y-axis is LPIPS for keep radius. Every plotted point lies above $y{=}x$ for all three tokenizers in both substitute regimes. The VA-VAE points extend farther along the x-axis than those of FLUX.2 and REPA-E FLUX.1, indicating a larger decoded change under the tested radius substitutions.

Across the tested component substitutions, retaining direction preserves more decoded similarity than retaining radius, with a smaller, tokenizer-dependent effect from radius replacement. Linear flow paths nevertheless allocate substantial squared target velocity to radial motion (Fig. A1). Fixed-radius projection retains token directions, decoder adaptation accommodates the changed support, and spherical interpolation removes the radial component of the training target. The generation controls in Tab. 2 evaluate this combined construction and the additional benefit of spherical transport.

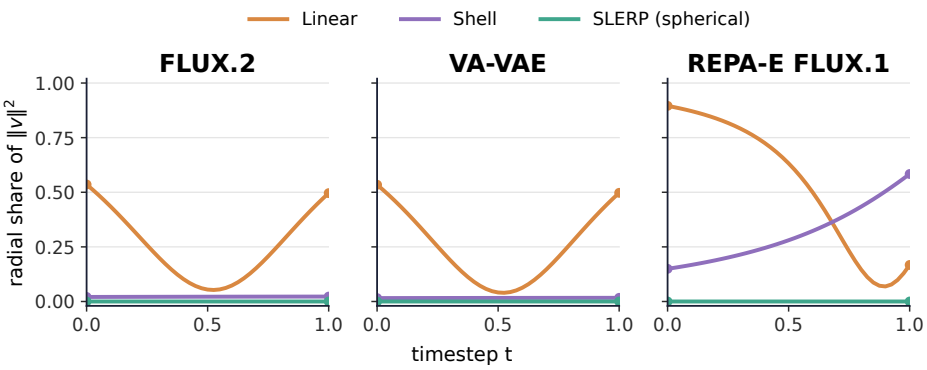


Figure A1: Radial share of the flow-matching velocity target, computed in each tokenizer's flow-training coordinates. Linear paths allocate a substantial fraction of the squared target velocity to radial motion; slerp on the sphere has zero radial velocity by construction. For endpoints on comparable shells, the Linear chord is closest to the origin near mid-path, where its radial velocity changes sign and the radial energy share is smallest, producing the U shape. REPA-E connects unequal radial scales: its processed data radius is about $0.35\sqrt{d}$, below the prior radius, and its measured radial share is about 90% near the noise endpoint and 17% near the data endpoint.

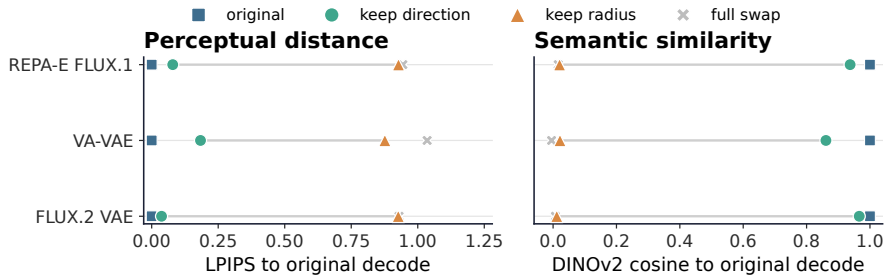


Figure A2: Decoded similarity under direction/radius component substitutions using the population mean. For each tokenizer, markers report mean LPIPS (left) and DINOv2 cosine similarity (right) to the original decode under four conditions: original, keep direction, keep radius, full mean substitute; 1024 images per condition.

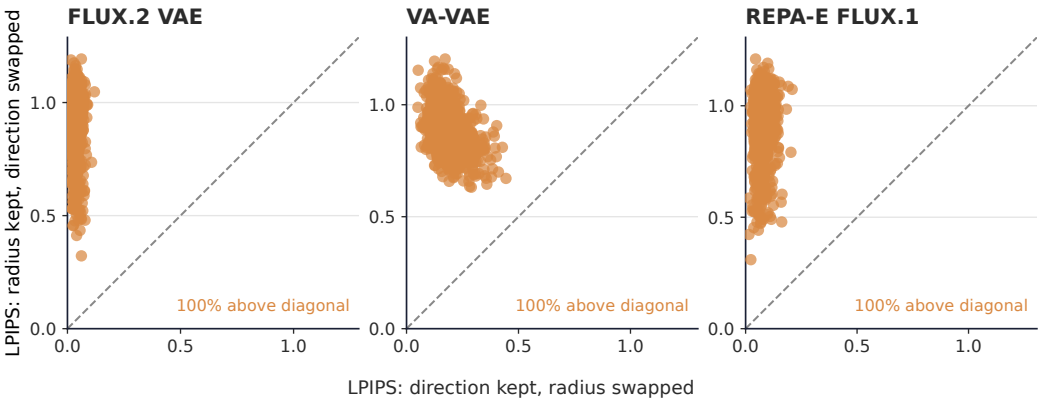


Figure A3: Per-sample decoded similarity under direction/radius component substitutions using the population mean. Each point is one image; x-axis: LPIPS after replacing radius with the population mean (direction kept); y-axis: LPIPS after replacing direction with the population mean (radius kept). Panels: FLUX.2, VA-VAE, REPA-E FLUX.1.

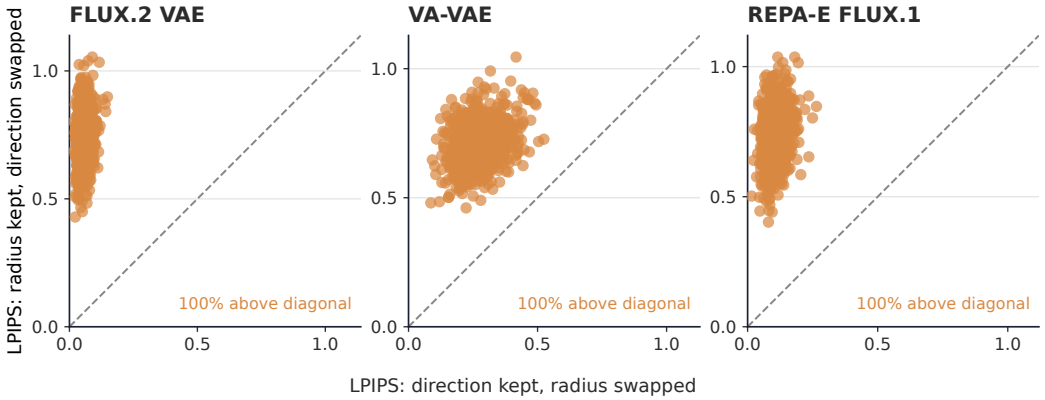


Figure A4: Per-sample decoded similarity under direction/radius component substitutions using a same-class partner. Axes as in Fig. A3; the substitute is a same-class partner's radius (x-axis) or direction (y-axis).

C.13 Scope of External Comparisons

Published representation-autoencoder results use different encoders, decoders, latent dimensions, diffusion architectures, and training budgets, so their FIDs are not controlled estimates of the effect of sphericalizing a pretrained VAE. They provide related tokenizer settings for comparison. The REPA experiment instead applies the same alignment objective to both FLUX.2 pipelines under a shared configuration (Tab. 4). RJF (Kumar & Patel, 2026) studies DINO-based representation autoencoders whose output normalization already yields spherical tokens. Here, the intervention begins with variable-radius pretrained VAE latents. Under the tested component substitutions, retaining direction preserves more decoded similarity than retaining radius. Projection with decoder adaptation constructs fixed-radius latents while retaining the frozen encoder.

Under the same shell-statistics protocol, RAE's DINOv2-B tokens (d=768) have radius CV 0.073 after channel standardization and no observed near-origin tokens; before that standardization, the measured LayerNorm outputs lie on the radius-$\sqrt{768}$ sphere to floating-point precision. RAE reports 1.51 unguided gFID at 800 epochs with a DiT-XL-scale model (Zheng et al., 2025). This result supplies context for generation with representation-encoder latents, but its tokenizer, architecture, and budget differ from those of the VAE intervention evaluated here.

D Qualitative Results

D.1 Class-Conditional Samples

Figure A5 shows class-conditional samples from the spherical-slerp SiT-XL/2 model with the FLUX.2 tokenizer. Each panel is a 4×4 grid of 16 samples for a single ImageNet-1k class.



(a) Chickadee (19)



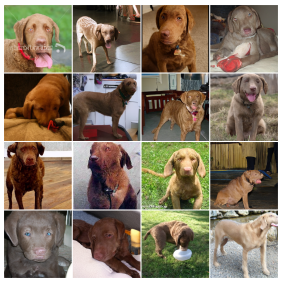
(b) Box turtle (37)



(c) Alligator lizard (44)



(d) Toy terrier (158)



(e) Chesapeake Bay retriever (209)



(f) Newfoundland (256)



(g) Cardigan (264)



(h) Cliff dwelling (500)



(i) Fountain (562)



(j) Moving van (675)



(k) Piggy bank (719)



(l) Rain barrel (756)



(m) Stone wall (825)



(n) Ice cream (928)



(o) Mushroom (947)

Figure A5: Class-conditional samples from spherical-slerp SiT-XL/2 with the FLUX.2 tokenizer. Each panel: 16 samples (4×4) for one ImageNet-1k class.

D.2 Same-Class Component Swaps

Figure 2 reports LPIPS and DINOv2 cosine similarity relative to the anchor decode, averaged over 1024 anchor-neighbor pairs. Lower LPIPS and higher DINOv2 similarity indicate closer agreement with the anchor. Figures A6 to A8 show four example pairs for each tokenizer. Columns are *Original 1* (anchor), *Angular 1 + Radius 2* (anchor direction with neighbor radius, the keep-direction hybrid), *Angular 2 + Radius 1* (anchor radius with neighbor direction, the keep-radius hybrid), and *Original 2* (neighbor). Components are exchanged at corresponding spatial positions, as in Sec. 3.2. In these examples, the keep-direction hybrids retain the anchor's object layout, whereas the keep-radius hybrids follow the neighbor's layout. Radius swaps can still alter appearance: the VA-VAE examples show changes in color and texture despite retaining the anchor directions.



Figure A6: FLUX.2 same-class component swaps. Columns: anchor (Original 1), keep-direction hybrid (Angular 1 + Radius 2), keep-radius hybrid (Angular 2 + Radius 1), neighbor (Original 2). Each row is one (anchor, neighbor) pair.

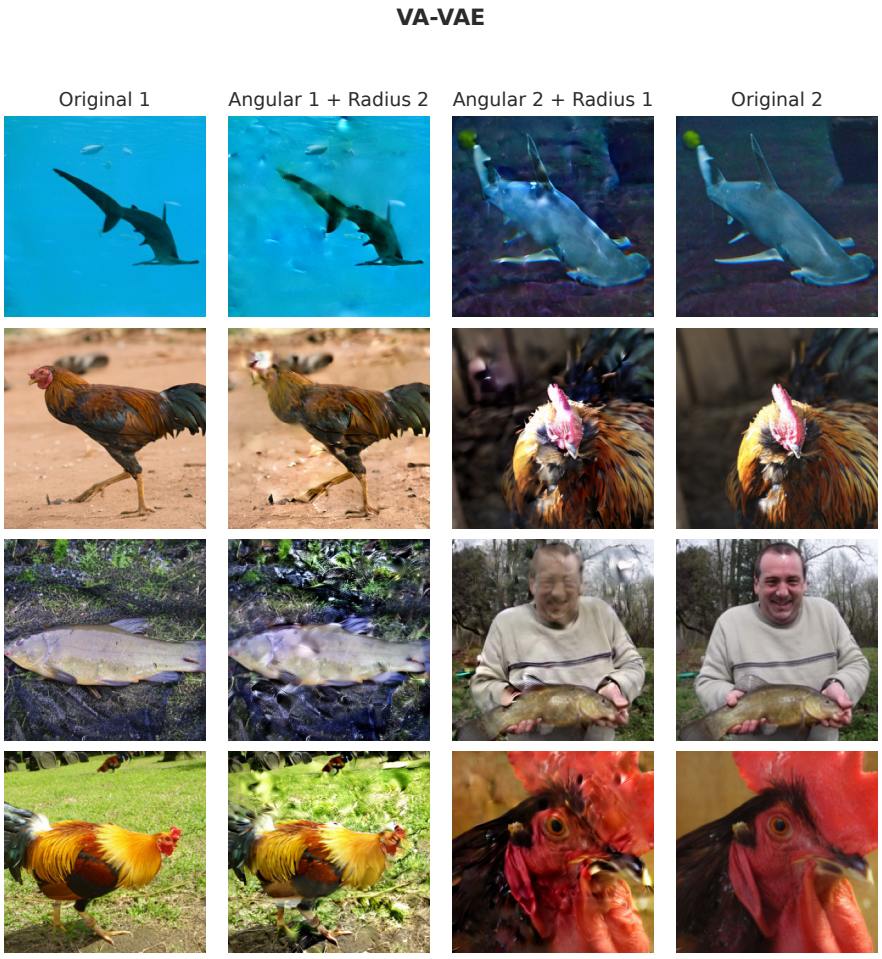


Figure A7: VA-VAE same-class component swaps; columns as in Fig. A6.



Figure A8: REPA-E FLUX.1 same-class component swaps; columns as in Fig. A6.

D.3 RECONSTRUCTIONS

Figures A9 to A11 compare baseline reconstructions, the matched-compute vanilla decoder finetune (Vanilla FT), and the spherical decoder finetune (Spherical FT) for eight ImageNet images, one from each listed class. Vanilla FT uses unprojected latents, whereas Spherical FT uses projected latents; the encoder remains frozen in both conditions. For these 256×256 images, the reconstructions preserve the overall scene composition, with differences in fine detail across conditions. These examples complement the validation rFID comparison in Tab. A7: rFID measures a distribution-level discrepancy, not reconstruction error for an individual image.

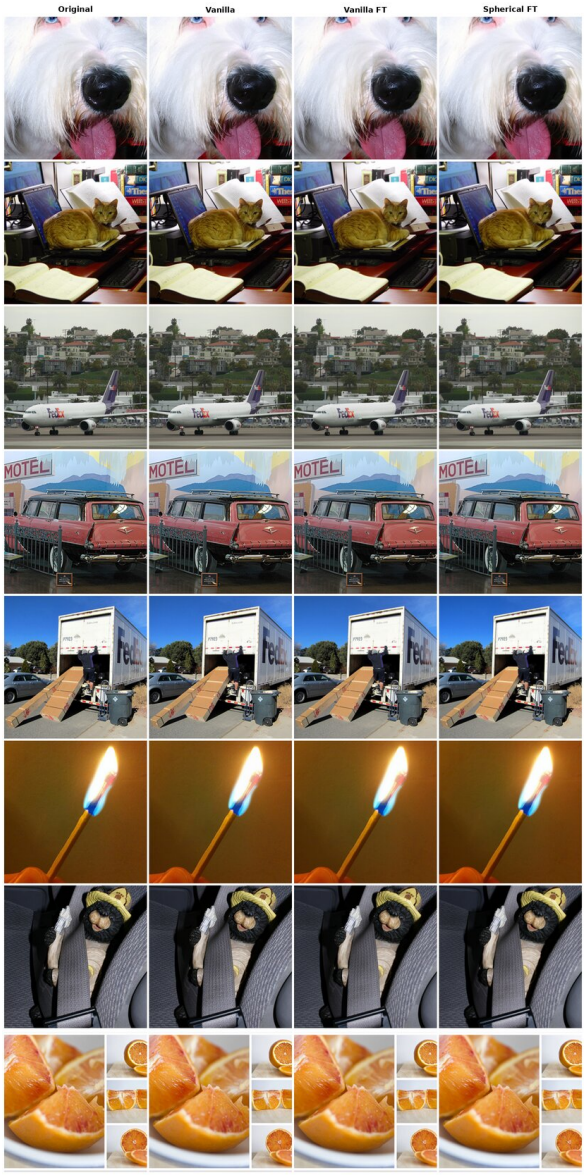


Figure A9: FLUX.2 reconstructions. Columns: original image, original (Vanilla) decoder, matched-compute vanilla decoder finetune (Vanilla FT), and spherical decoder finetune (Spherical FT). Rows: sheepdog, tabby, airliner, beach wagon, crate, matchstick, seat belt, orange.

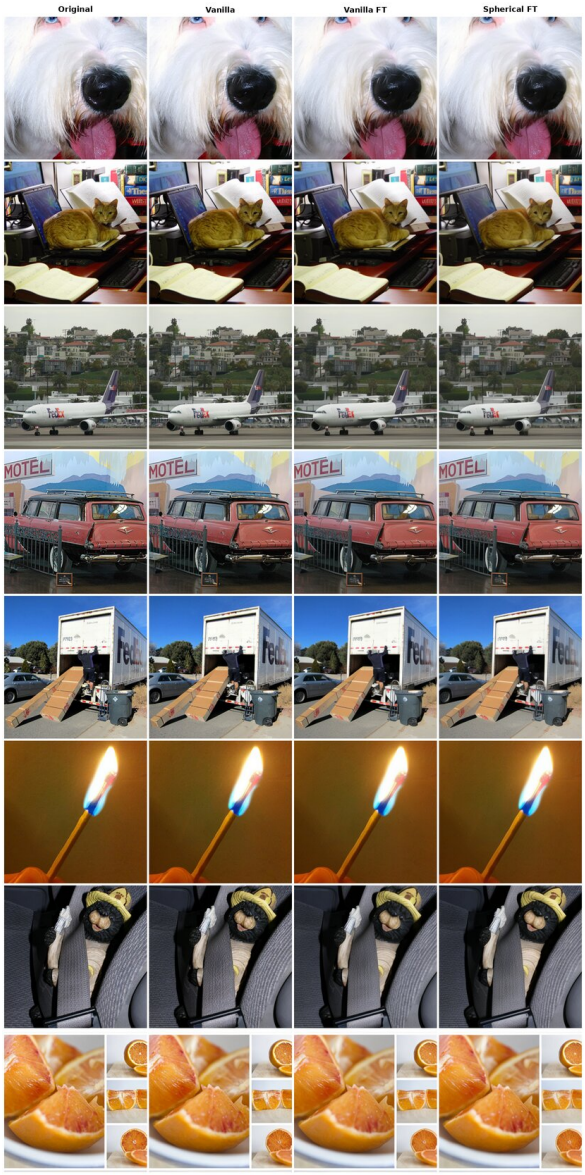


Figure A10: VA-VAE reconstructions; columns and rows as in Fig. A9.

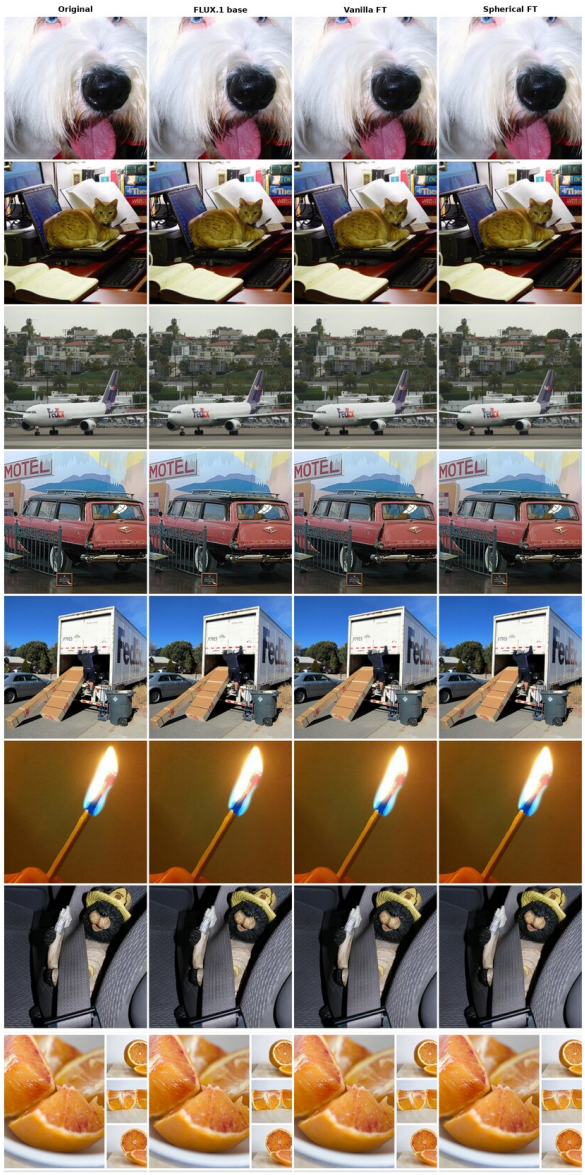


Figure A11: REPA-E FLUX.1 reconstructions. Columns: original image, baseline reconstruction (labeled FLUX.1 base), Vanilla FT, and Spherical FT. Rows as in Fig. A9.

E SCOPE AND FUTURE DIRECTIONS

Our experiments establish generation improvements on class-conditional ImageNet-256 across three pretrained tokenizer families with token dimensions $d = 16$ and $d = 32$. The broad-radius $d = 4$ SD-VAE case (Sec. C.8 and Tab. A10) motivates adaptive-radius extensions for representations with appreciable radial variation.

Extending post-hoc latent-support adaptation to text-conditioned generation, higher resolutions, non-square aspect ratios, and jointly trained tokenizers is a natural next step. The token-wise spherical construction provides a common formulation for studying these settings, with reconstruction and generation evaluated together.