POLAR DIFFUSION: DECOUPLING RADIAL AND TANGENTIAL DENOISING IN DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models typically perform denoising in Euclidean space using a single isotropic prediction objective, implicitly assigning equal importance to errors along all geometric directions. High-dimensional data, however, often concentrate near lower-dimensional structures, where radial and tangential variations can play fundamentally different roles in generation. We introduce Polar Diffusion, a geometric reformulation that factorizes the diffusion process over a product space of magnitude (radial) and normalized direction (tangential). We derive the corresponding product-space score-matching formulation and show that, when mapped back to Euclidean coordinates, the resulting dynamics recover the same marginal distributions as standard variance-preserving diffusion. This factorization exposes a degree of freedom hidden by the Euclidean formulation: radial and tangential denoising objectives can be controlled and weighted independently. Polar Diffusion thus provides a simple geometric extension of conventional diffusion models that enables explicit control over the relative contributions of magnitude and structural errors during denoising. Ablation studies show that moderately down-weighting the radial objective consistently improves generation quality, suggesting that radial and tangential errors need not contribute equally to effective denoising, with greater emphasis on the tangential component providing closer accounting for the distribution of structural information in the data.

1 INTRODUCTION

Diffusion, as a fundamental paradigm for generative modeling, has achieved remarkable success Ho et al. (2020). Its numerous extensions, including score-based generative modeling Song et al. (2020) and flow-matching approaches Liu et al. (2022); Lipman et al. (2022), have further revealed key structural and dynamical properties underlying the approach effectiveness, contributing to steadily improving performance and widespread adoption across generative tasks in vision Peebles & Xie (2023), audio Guo et al. (2024), and video Shaul et al. (2026). Diffusion modeling consists of a Markovian forward process that progressively perturbs the data distribution with noise, eventually transforming it into an approximately Gaussian distribution, together with a learned reverse process that progressively removes this noise. The reverse dynamics are approximated by a deep neural network trained to estimate the injected noise, the score, or an equivalent denoising quantity at each diffusion time step, thereby enabling samples to be generated by evolving Gaussian noise back toward the data distribution. The scalability of diffusion models, together with their increasingly strong generative performance, have rapidly attracted substantial attention across a diverse range of tasks, and positioned them as compelling alternatives to generative adversarial networks (GANs) Dhariwal & Nichol (2021); Stypulkowski et al. (2024). Score-based generative modeling further generalized this framework to continuous time through stochastic differential equations (SDEs), where a neural network is trained to estimate the time-dependent score function—the gradient of the log-density of the progressively perturbed data distribution. The learned score determines the reverse-time dynamics, enabling samples from the terminal noise distribution to be transported back toward the data distribution. Flow matching, by contrast, formulates generation through a deterministic probability flow, learning a time-dependent vector field associated with a prescribed probability path that continuously transports samples from a simple prior, typically Gaussian, to the target data distribution.

Mumford's work on the intrinsic geometry of natural images Lee et al. (2003); Carlsson et al. (2008) showed that local image structures admit a low-dimensional organization with Klein-bottle-like

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.

topology, and subsequent estimates confirm that natural images concentrate near low-dimensional manifolds in the ambient space Pope et al. (2021). This invites a reconsideration of diffusion processes defined purely in ambient Euclidean space: if the data are concentrated near a structured nonlinear manifold, treating all ambient directions as equivalent expends modeling capacity along directions that are only weakly relevant to the intrinsic geometry, whereas an adapted diffusion can let noise injection and denoising follow geometrically meaningful directions. Recent work has begun to exploit this structure more directly; for example, Li & He (2026) leverages the underlying data geometry to facilitate direct prediction of clean data rather than the injected noise, further illustrating the potential benefits of geometry-aware formulations of diffusion. The work in Yue et al. (2026) constrains the latent representation to a hypersphere to facilitate efficient sampling. In Kumar & Patel (2026) it is observed that DINOv2 embeddings Oquab et al. (2023) exhibit hyperspherical geometry and accordingly construct flow-matching paths along geodesics on the hypersphere, improving convergence and achieving state-of-the-art sample quality. Riemannian score-based models De Bortoli et al. (2022) extend Euclidean diffusion models to manifold-valued data by defining the diffusion and its reverse process directly on a particular Riemannian manifold.

Taken together, these geometric insights, along with the recognition that conventional diffusion models operating in Euclidean space implicitly treat variations in magnitude and intrinsic data structure on an essentially uniform footing, suggest an opportunity to further improve upon their already remarkable success. Riemannian score-based models De Bortoli et al. (2022) explicitly incorporate geometric structure into the generative process, but typically require prior analytical knowledge of the underlying manifold and have thus far been demonstrated primarily on relatively low-dimensional data. These requirements limit their applicability to image generation, where the intrinsic data manifold remains high-dimensional, complex, and generally unavailable in an explicitly analytical form.

This raises a natural question: can one exploit meaningful geometric structure in high-dimensional data without first identifying or parameterizing an unknown intrinsic manifold? A useful starting point is already present in the Euclidean representation itself. Every nonzero data vector possesses two geometrically distinct attributes: its magnitude and its normalized direction. Conventional Euclidean diffusion does not distinguish these modes of variation, coupling radial and directional changes within a single isotropic representation and denoising objective. Yet the two are distinct functions of the underlying data and may differ substantially in distribution, variability, sensitivity to perturbation, and perceptual significance. In high dimensions there is therefore no compelling reason to expect their contributions to be optimally treated at the same scale, and uniform weighting has indeed been found suboptimal Liu et al. (2026). This motivates separating the two components explicitly and controlling their respective weights in the denoising objective.

We introduce *Polar Diffusion*, a radial–angular reformulation of diffusion in Euclidean data space. Each nonzero vector is represented by its magnitude $r > 0$ and normalized direction $x \in \mathbb{S}^{n-1}$, yielding the product space $\mathbb{R}_+ \times \mathbb{S}^{n-1}$, where $\mathbb{S}^{n-1}$ is the unit hypersphere embedded in $\mathbb{R}^n$. Equivalently, introducing the log-radius $\rho = \log r$ gives the representation $\mathbb{R} \times \mathbb{S}^{n-1}$. This factorization provides a known geometric structure without requiring explicit knowledge of the intrinsic data manifold: the radial coordinate retains magnitude information, while normalization places directional structure on the hypersphere, which serves as an ambient geometric space capable of accommodating substantially lower-dimensional and topologically nontrivial intrinsic structure[1]. In this sense, the formulation is compatible with the geometric organization suggested by natural-image studies such as those of Lee et al. (2003), without assuming that the intrinsic image manifold itself is spherical or that its topology is known.

Formulating diffusion in these coordinates separates radial denoising, associated with changes in magnitude, from tangential denoising, associated with changes in normalized directional structure. We establish an explicit relationship between Polar Diffusion and conventional Euclidean diffusion, showing how the radial and tangential dynamics jointly recover the corresponding Euclidean process. The principal advantage therefore lies not in targeting a different distribution, but in exposing two geometrically and statistically distinct components that are otherwise implicitly coupled within a single isotropic objective. Their relative contributions can then be set explicitly rather than fixed by the weighting induced by the ambient Euclidean metric.

[1]The concentration of natural images near a lower-dimensional manifold may also be viewed from a statistical-physics perspective, with interacting pixel or feature variables defining an energy functional whose low-energy configurations correspond to plausible images.

This relationship motivates a weighted training objective that balances radial and tangential prediction errors independently. Synthetic experiments illustrate the concentration of information near lower-dimensional submanifolds and clarify the distinct roles of the two denoising components. Experiments on image-generation benchmarks further demonstrate that appropriately balancing these components consistently improves generation quality over comparable Euclidean diffusion baselines.

2 Background and Problem Statement

2.1 Euclidean Diffusion Models

We adopt the widely accepted notation for an image data sample $\boldsymbol{y}_0 \sim q_{\text{data}}$ where q is the associated probability distribution. In continuous time, the variance-preserving (VP) diffusion process Song et al. (2020) is defined by the SDE

$$d\boldsymbol{y}_t = -\frac{1}{2}\beta_t \boldsymbol{y}_t \, dt + \sqrt{\beta_t}\, d\boldsymbol{W}_t^{\mathbb{R}^n}, \qquad t \in [0,1], \tag{1}$$

where $\beta_t > 0$ denotes the noise schedule and $\boldsymbol{W}_t^{\mathbb{R}^n}$ is a Euclidean Brownian motion in $\mathbb{R}^n$ (See Appendix A.1 for details regarding the Brownian motion). By careful design of β_t, as t increases, the forward process progressively perturbs the data distribution toward an isotropic standard normal distribution.

Equation 1's conditional transition distribution admits a closed-form Gaussian expression Song et al. (2020)

$$p_t(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{y}_t; \alpha_t \boldsymbol{y}_0, \sigma_t^2 \boldsymbol{I}\right), \tag{2}$$

where $\alpha_t = \exp\left(-\frac{1}{2}\int_0^t \beta_s \, ds\right)$, and $\sigma_t = \sqrt{1-\alpha_t^2}$. Equivalently, a sample from the forward process at any time t can be generated directly as

$$\boldsymbol{y}_t = \alpha_t \boldsymbol{y}_0 + \sigma_t \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{\mathbb{R}^n}). \tag{3}$$

The corresponding reverse-time diffusion is determined by the score of the time-dependent marginal distribution. Following Song et al. (2020), the reverse-time SDE is

$$d\boldsymbol{y}_t = \left[-\frac{1}{2}\beta_t \boldsymbol{y}_t - \beta_t \nabla_{\boldsymbol{y}_t} \log p_t(\boldsymbol{y}_t)\right] dt + \sqrt{\beta_t}\, d\bar{\boldsymbol{W}}_t^{\mathbb{R}^n}, \qquad t: 1 \to 0, \tag{4}$$

where $\bar{\boldsymbol{W}}_t^{\mathbb{R}^n}$ denotes the Euclidean Brownian motion backward in time from 1 to 0, $p_t(\boldsymbol{y}_t)$ denotes the marginal distribution of the forward process at time t, and $\nabla_{\boldsymbol{y}_t} \log p_t(\boldsymbol{y}_t)$ is its score function. Therefore, sampling from the data distribution can be performed by estimating the time-dependent score and numerically integrating the reverse-time SDE from the prior distribution at $t = 1$ to the data distribution at $t = 0$.

In practice, following the noise-prediction parameterization of DDPM Ho et al. (2020), a neural network $\boldsymbol{\epsilon}_\theta(\cdot, \cdot)$ can be trained to predict the Gaussian noise in Equation 3 using

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t, \boldsymbol{y}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, t) - \boldsymbol{\epsilon}\right\|_2^2\right]. \tag{5}$$

The noise-prediction parameterization is directly related to score estimation. From Equation 2, the conditional score is $\nabla_{\boldsymbol{y}_t} \log p_t(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = -\frac{\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0}{\sigma_t^2} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$. At the optimum of the denoising objective, $\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, t)$ can be used to estimate the marginal score $\nabla_{\boldsymbol{y}_t} \log p_t(\boldsymbol{y}_t)$, which can then be used in the reverse-time SDE in Equation 4.

2.2 Hypersphere – Induced geometric properties

Our polar representation decomposes each nonzero data point into a radial magnitude and a normalized directional component. While the radial variable lies in a one-dimensional Euclidean space, the directional component is constrained to have unit norm and therefore evolves on the hypersphere $\mathbb{S}^{n-1}$. Consequently, modeling the directional part of the diffusion requires respecting the geometry

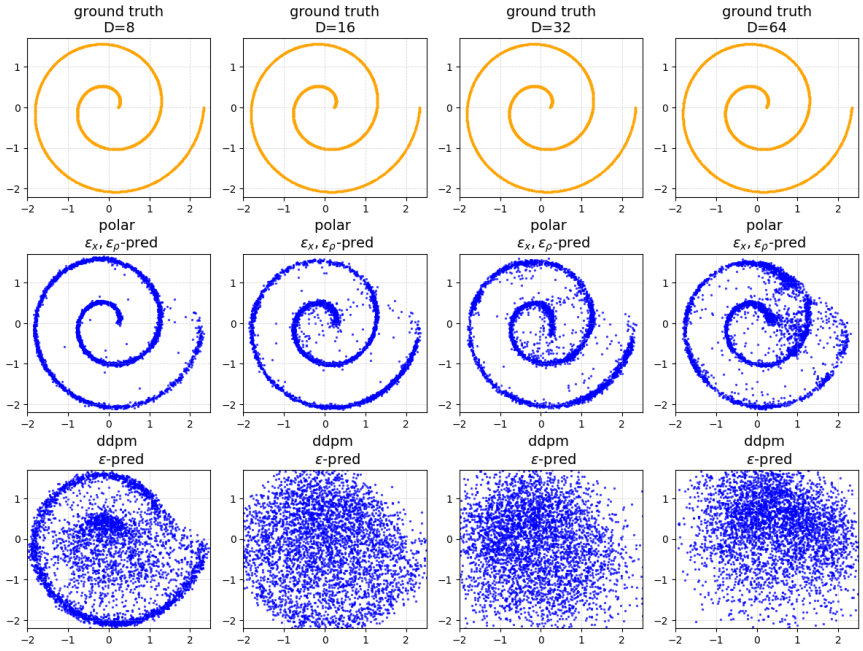


Figure 1: Toy experiment on a two-dimensional spiral embedded into $\mathbb{R}^D$ using a random linear projection. Both Polar Diffusion and DDPM are trained in the D-dimensional ambient space, and generated samples are visualized after mapping back to two dimensions. Each column corresponds to a different ambient dimension $D \in \{8, 16, 32, 64\}$. From top to bottom, the rows show the ground-truth data, samples from Polar Diffusion, and samples from the DDPM baseline.

of the hypersphere rather than treating it as an unconstrained Euclidean variable. We therefore introduce the basic geometric objects used throughout our polar diffusion formulation, including the tangent space, tangent and normal projections, and the Riemannian exponential map.

Let the ambient data space be $\mathbb{R}^n$, where a unit $(n-1)$-sphere embedded in $\mathbb{R}^n$ is defined as $\mathbb{S}^{n-1} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 = 1\} \subset \mathbb{R}^n$. The tangent space at $\boldsymbol{x} \in \mathbb{S}^{n-1}$ of manifold $\mathbb{S}^{n-1}$ is defined as $T_{\boldsymbol{x}}\mathbb{S}^{n-1} := \{\boldsymbol{v} \in \mathbb{R}^n : \boldsymbol{x}^\top \boldsymbol{v} = 0\}$. The projection matrix $P^\perp(\boldsymbol{x}) = \boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^T \in \mathbb{R}^{n\times n}$ projects vectors onto $T_{\boldsymbol{x}}\mathbb{S}^{n-1}$ together with the normal projection matrix $P(\boldsymbol{x}) = \boldsymbol{I} - P^\perp(\boldsymbol{x})$.

For a Riemannian manifold $\mathcal{M}$ and a point $\boldsymbol{x} \in \mathcal{M}$, the Riemannian exponential map (Exp) is a mapping, $\mathrm{Exp}_{\boldsymbol{x}} : T_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$. Here, we simplify this definition and refer the reader to Lee (2019) for more details. On the unit hypersphere $\mathbb{S}^{n-1}$, the exponential map admits the closed-form expression $\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{v}) = \cos(||\boldsymbol{v}||_2)\,\boldsymbol{x} + \sin(||\boldsymbol{v}||_2)\,\frac{\boldsymbol{v}}{||\boldsymbol{v}||_2}$, where $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathbb{S}^{n-1}$.

3 Polar Diffusion

3.1 Polar Decomposition and Diffusion in Polarized Space

The intrinsic geometry of the data manifold is generally unknown, particularly for high-dimensional images, making it difficult to construct a diffusion process that directly respects its underlying structure. Nevertheless, without explicit knowledge of this geometry, every nonzero data point admits a natural decomposition into two geometrically distinct components: its magnitude and normalized direction. Instead of an isotropic Euclidean diffusion unnecessarily dispersing probability mass across geometrically unequivalent ambient directions, we propose a decomposition of the dynamics into a radial–angular product representation. This partially resolves this mismatch by isolating scale variation in a one-dimensional radial coordinate while constraining structural perturbations to the tangent bundle of the hypersphere. The resulting diffusion thereby preserving more of the geometric organization of the data during corruption, yields naturally separated radial and tangential score fields, ensuring an independent control of their contributions to denoising.

Our representation of a non-zero image $\boldsymbol{y}_0 \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, proceeds in polar coordinates as $(\rho_0, \boldsymbol{x}_0)$ where $\rho_0 = \log||\boldsymbol{y}_0||_2 \in \mathbb{R}$ and $\boldsymbol{x}_0 = \boldsymbol{y}_0/\|\boldsymbol{y}_0\|_2 \in \mathbb{S}^{n-1}$. Equivalently, $\boldsymbol{y}_0 = e^{\rho_0}\boldsymbol{x}_0$ where the pair $(\rho_0, \boldsymbol{x}_0)$ serves as the initial condition for the polarized diffusion process herein.

Definition 1 (Polarized Diffusion)**.**

$$\begin{cases} d\rho_t = \left[-\frac{1}{2}\beta_t + \frac{n-2}{2}\beta_t e^{-2\rho_t}\right] dt + \sqrt{\beta_t} e^{-\rho_t} dW_t^{\mathbb{R}}, \\ d\boldsymbol{x}_t = \sqrt{\beta_t} e^{-\rho_t} \circ d\boldsymbol{W}_t^{\mathbb{S}}, \end{cases} \tag{6}$$

where $W_t^{\mathbb{R}}$ is a standard one-dimensional Brownian motion, while $W_t^{\mathbb{S}}$ denotes Brownian motion on the hypersphere (See Appendix A.2 for details). The symbol $\circ$ indicates that the spherical SDE is interpreted in the Stratonovich[2] *sense.*

Note that the diffusion dynamics of an image process $\boldsymbol{y}_t$ evolving in the ambient Euclidean space can be equivalently decomposed into two coupled stochastic processes: a radial process ρ_t and an angular process $\boldsymbol{x}_t$ such that $\rho_t \in \mathbb{R}$, $\boldsymbol{x}_t \in \mathbb{S}^{n-1}$, $\forall\, t \in [0,1]$. Equivalently, defining $r_t = e^{\rho_t}$ guarantees $r_t \in \mathbb{R}_+$, while the spherical dynamics preserve $\|\boldsymbol{x}_t\|_2 = 1$. Thus, the diffusion explicitly separates variation in magnitude, governed by ρ_t, from changes in normalized direction, governed by $\boldsymbol{x}_t$. Although $e^{-\rho_t}$ can become large when ρ_t is strongly negative, we did not find it necessary to clamp this term, as no numerical instabilities were observed in our experiments.

Proposition 1 (Marginal consistency)**.** *Let $(\rho_t, \boldsymbol{x}_t)$ follow the polarized VP-SDE in Equation 6, and define $\boldsymbol{y}_t = e^{\rho_t}\boldsymbol{x}_t$. Then the marginal distribution of $\boldsymbol{y}_t$ is* ***identical*** *to that of the standard VP-SDE in Equation 1.*

Proof: see Appendix D. We construct stochastic dynamics for $(\rho_t, \boldsymbol{x}_t)$ such that, after mapping back to Euclidean space through $\boldsymbol{y}_t = e^{\rho_t}\boldsymbol{x}_t$, the resulting $\boldsymbol{y}_t$ follows the same marginal distribution as the VP-SDE in Equation 1. In this sense, the polar SDE is not arbitrarily chosen, but is designed to remain distributionally consistent with the original Euclidean diffusion process.

The forward process induces a reverse-time diffusion on the polarized space. In particular, applying the reverse-time SDE formulation Song et al. (2020); Anderson (1982) gives the following corollary.

Corollary 1 (Reverse-time polarized diffusion)**.** *The reverse-time process corresponding to the polarized VP-SDE in Equation 6 is given by*

$$\begin{cases} d\rho_t = \left[-\frac{1}{2}\beta_t + \frac{n+2}{2}\beta_t e^{-2\rho_t} - \beta_t e^{-2\rho_t}\nabla_{\rho_t} \log p_t(\rho_t, \boldsymbol{x}_t)\right] dt + \sqrt{\beta_t} e^{-\rho_t} d\bar{W}_t^{\mathbb{R}}, \\ d\boldsymbol{x}_t = -\beta_t e^{-2\rho_t}\nabla_{\boldsymbol{x}_t}^{\mathbb{S}} \log p_t(\rho_t, \boldsymbol{x}_t)\, dt + \sqrt{\beta_t} e^{-\rho_t} \circ d\bar{\boldsymbol{W}}_t^{\mathbb{S}}, \end{cases} \tag{7}$$

where $d\bar{W}_t^{\mathbb{R}}$ denotes the Brownian motion for time flowing backwards from 1 to 0, $d\bar{\boldsymbol{W}}_t^{\mathbb{S}}$ denotes Brownian motion on the hypersphere for time flowing backwards from 1 to 0, and $\nabla_{\boldsymbol{x}_t}^{\mathbb{S}}$ denotes the Riemannian gradient [3] *on $\mathbb{S}^{n-1}$.*

Proof: See Appendix E. Because the polar process is induced by the same VP-SDE transition kernel, its conditional score can be obtained directly from the Gaussian transition density in Euclidean space.

Proposition 2 (Score for polarized diffusion)**.** *Following polarized SDE in Equation 6, the radial and angular components of the conditional score are*

$$\begin{cases} \nabla_{\rho_t} \log q_t(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = n - \dfrac{e^{2\rho_t} - \alpha_t e^{\rho_t + \rho_0}\boldsymbol{x}_t^\top \boldsymbol{x}_0}{\sigma_t^2} \in \mathbb{R}^1, \\ \nabla_{\boldsymbol{x}_t}^{\mathbb{S}} \log q_t(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = \dfrac{\alpha_t e^{\rho_t + \rho_0}}{\sigma_t^2}\left[\boldsymbol{x}_0 - (\boldsymbol{x}_t^\top \boldsymbol{x}_0)\boldsymbol{x}_t\right] \in T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}. \end{cases} \tag{8}$$

where α_t and σ_t are defined in Equation 2.

Proof: see Appendix F.

[2] The radial SDE is written in the Itô sense, while the spherical SDE is written in the Stratonovich sense to preserve the intrinsic geometry of the hypersphere. The corresponding Itô form of the spherical SDE includes a curvature-induced drift term. See Appendix B for the derivation of the corresponding Itô process.

[3] For a hypersphere with the Riemannian metric inherited from the ambient Euclidean space, the Riemannian gradient is obtained by orthogonally projecting the ambient Euclidean gradient onto the tangent space, $\nabla_{\boldsymbol{x}}^{\mathbb{S}} f(\boldsymbol{x}) = P^{\perp}(\boldsymbol{x})\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$

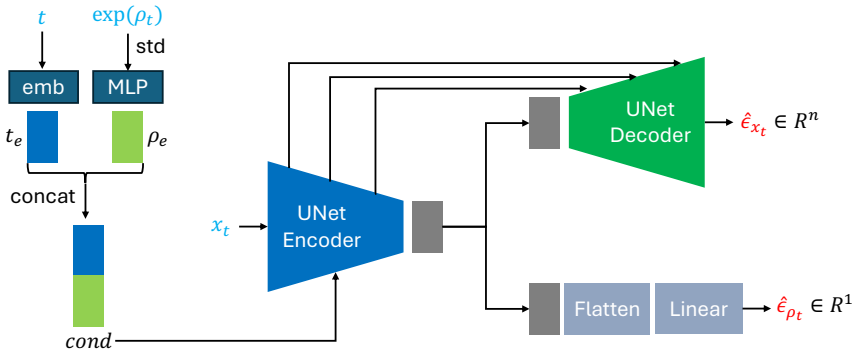


Figure 2: Unconditional polar diffusion UNet model structure. The inputs are highlighted in blue, while the outputs are highlighted in red.

3.2 Noise-Prediction Parameterization

Directly learning the conditional score can be numerically challenging because its magnitude scaling factor, proportional to σ_t^{-1}, can grow large at small noise levels. Inspired by the noise-prediction parameterization in DDPM Ho et al. (2020), we instead parameterize the polar score through the Gaussian noise injected by the underlying VP-SDE.

Corollary 2 (Polar score–noise parameterization)**.** *Given a Euclidean noisy sample* $\boldsymbol{y}_t = \alpha_t \boldsymbol{y}_0 + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{\mathbb{R}^n})$, *we decompose* $\boldsymbol{\epsilon}$ *as,*

$$\begin{aligned} \epsilon_{\rho_t} &= \boldsymbol{x}_t^\top \boldsymbol{\epsilon} \in \mathbb{R}, \\ \boldsymbol{\epsilon}_{\boldsymbol{x}_t} &= P^\perp(\boldsymbol{x}_t)\boldsymbol{\epsilon} = \boldsymbol{\epsilon} - \epsilon_{\rho_t}\boldsymbol{x}_t \in T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}. \end{aligned} \tag{9}$$

Let $\rho_t = \log \|\boldsymbol{y}_t\|$ *and* $\boldsymbol{x}_t = \boldsymbol{y}_t / \|\boldsymbol{y}_t\|$. *The conditional polar scores are directly related to these two noise components:*

$$\boxed{\begin{aligned} \nabla_{\rho_t} \log q_t(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) &= n - \frac{e^{\rho_t}}{\sigma_t}\epsilon_{\rho_t}, \\ \nabla^{\mathbb{S}}_{\boldsymbol{x}_t} \log q_t(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) &= -\frac{e^{\rho_t}}{\sigma_t}\boldsymbol{\epsilon}_{\boldsymbol{x}_t}. \end{aligned}} \tag{10}$$

Proof: See Appendix G. Here, ϵ_{ρ_t} represents the noise component along the radial direction, whereas $\boldsymbol{\epsilon}_{\boldsymbol{x}_t}$ represents the angular component which is tangent to the hypersphere. We can see that predicting the radial and tangential components of the injected noise is equivalent, up to known time-dependent transformations, to estimating the corresponding components of the conditional score. We thus parameterize the noise-prediction model as $\boldsymbol{\epsilon}_\theta\left(e^{\rho_t}, \boldsymbol{x}_t, t\right) = \left(\hat{\boldsymbol{\epsilon}}_{\theta,\boldsymbol{x}_t}, \hat{\epsilon}_{\theta,\rho_t}\right)$, where $\left(e^{\rho_t}, \boldsymbol{x}_t, t\right)$ is the model input. $\hat{\epsilon}_{\theta,\rho_t} \in \mathbb{R}$ predicts the radial component and $\hat{\boldsymbol{\epsilon}}_{\theta,\boldsymbol{x}_t} \in \mathbb{R}^n$ the tangential component. This decomposition allows the two geometrically distinct prediction errors to be flexibly and separately weighted. To that end, we define a training objective to seek optimal denoising:

$$\mathcal{L}_{\text{polar}} = \mathbb{E}_{\boldsymbol{y}_0,t,\boldsymbol{\epsilon}}\left[w_{\boldsymbol{x}} \left\|\boldsymbol{\epsilon}_{\theta,\boldsymbol{x}}(e^{\rho_t}, \boldsymbol{x}_t, t) - \boldsymbol{\epsilon}_{\boldsymbol{x}_t}\right\|_2^2 + w_\rho \left(\epsilon_{\theta,\rho}(e^{\rho_t}, \boldsymbol{x}_t, t) - \epsilon_{\rho_t}\right)^2\right], \tag{11}$$

where $w_{\boldsymbol{x}}$ and w_ρ control the relative importance of the tangential and radial denoising objectives, respectively. It is interesting to note that the standard DDPM objective may be similarly re-expressed as an analogously orthogonal decomposition as follows.

Corollary 3 (DDPM loss decomposition)**.** *The Euclidean noise-prediction loss in Equation 5 can be written as*

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\boldsymbol{y}_0,t,\boldsymbol{\epsilon}}\left[\left\|P^\perp(\boldsymbol{x}_t)\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, t) - \boldsymbol{\epsilon}_{\boldsymbol{x}_t}\right\|_2^2 + \left(\boldsymbol{x}_t^\top \boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, t) - \epsilon_{\rho_t}\right)^2\right]. \tag{12}$$

Details are provided in Appendix H. The perspective of the radial–tangential decomposition does shed some light on the fact that standard DDPM, as noted earlier, implicitly assigns equal weight to the two components: $w_{\boldsymbol{x}} = w_\rho = 1$.

In addition to providing the flexibility of selectively accounting for noise along different geometric directions, our formulation explicitly distinguishes the two most consequential components of the perturbation: radial errors, which primarily affect the overall magnitude or scale of a sample, and tangential errors, which alter its directional and structural content. By associating these components with adapted weights w_ρ and $w_{\boldsymbol{x}}$, respectively, the formulation enables their contributions to the denoising objective to be controlled independently. For image data, these two error modes play markedly different roles in determining perceptual generation quality, thereby motivating their separate treatment during denoising. This relative weighting is further examined empirically in Section 5.3. In addition, predicting the radial and tangential components through separate output heads allows the network to specialize its estimates to these two geometrically distinct quantities.

The complete training and sampling procedures are summarized in Algorithms 1 and 2, respectively. During training, we first sample $\boldsymbol{y}_t$ using the standard VP-SDE marginal and then transform it into its polar representation $(\rho_t, \boldsymbol{x}_t)$. The injected Euclidean noise is subsequently decomposed into radial and tangential components, which is the target of the two output heads of the model.

During sampling, we numerically integrate the reverse polar dynamics using the Euler–Maruyama discretization. Since the ground-truth tangential noise satisfies $\boldsymbol{\epsilon}_{\boldsymbol{x}_t} \in T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}$, the predicted tangential component should ideally lie in the same tangent space. In practice, however, the neural network output may contain a small radial component due to approximation error. We therefore explicitly project the predicted tangential score onto the tangent space using $P^{\perp}(\boldsymbol{x}_t)$. The angular state is then updated through the Riemannian exponential map, which ensures that the resulting sample remains on $\mathbb{S}^{n-1}$ throughout the reverse process.

4 Geometry-Aligned Polar-UNet

We adapt the UNet architecture of Ho et al. (2020) and Nichol & Dhariwal (2021) to construct our Polar-UNet, modifying its prediction structure to align with the geometry of the proposed radial–angular diffusion for image generation task, with details included in Figure 2. Our code and implementation are available at https://anonymous.4open.science/r/polar-diffusion-E28A.

4.1 Standardization

Since $\exp(\rho_t) = \|\boldsymbol{y}_t\|$ and $\boldsymbol{x}_t = \boldsymbol{y}_t/\|\boldsymbol{y}_t\|$ have different numerical scales, we standardize the radial magnitude using statistics computed from the training data: $\widetilde{r}_t = \frac{\exp(\rho_t)-\mu_r}{\sigma_r}$, where $\mu_r = \frac{1}{N}\sum_{i=1}^{N}\exp(\rho_i)$, and $\sigma_r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\exp(\rho_i)-\mu_r\right)^2}$, where N denotes the total number of data. Here, μ_r and σ_r denote the empirical mean and standard deviation of the radial magnitudes in the training set.

4.2 Condition

We adopt the standard time-conditioning mechanism used in diffusion models, where the diffusion timestep t is first encoded using a sinusoidal embedding and then passed through an MLP to obtain a time embedding $t_e \in \mathbb{R}^d$. Since the standardized radial magnitude $\widetilde{r}_t$ is a scalar, we additionally incorporate it into the conditioning branch of the UNet. Specifically, $\widetilde{r}_t$ is passed through a separate MLP to obtain a radial embedding $\rho_e \in \mathbb{R}^d$. We then concatenate the time and radial embeddings to form the final conditioning vector: $cond = [t_e; \rho_e] \in \mathbb{R}^{2d}$. This conditioning vector is injected into each layer of the UNet. We also experimented with element-wise addition of t_e and ρ_e; however, we found that concatenation consistently yielded better empirical performance.

4.3 Main Branch

Instead of directly using $\boldsymbol{y}_t$ as the input to the UNet encoder, we use $\boldsymbol{x}_t$. The standard UNet encoder–decoder branch then predicts $\hat{\epsilon}_{\boldsymbol{x}_t} \in T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}$.

To additionally predict the radial noise component, we introduce a scalar prediction branch from the latent representation produced by the UNet encoder. Let the encoder output be a latent tensor

Table 1: Image generation results. **Input** lists the quantities fed to the denoising network, the timestep t is also an input in every case and is omitted for brevity. *no-std-e^{ρ_t}* ablates the standardization of e^{ρ_t} before it enters the network.

Dataset	Name	Input	w_ρ	IS↑	FID↓
CIFAR-10	DDPM Ho et al. (2020)	$\boldsymbol{y}_t$	1.0	**9.46**	3.17
	Polar Diffusion (Ours)	$(e^{\rho_t}, \boldsymbol{x}_t)$	0.05	9.45	**3.09**
	Polar Diffusion (no-std-e^{ρ_t})	$(e^{\rho_t}, \boldsymbol{x}_t)$	0.05	9.24	3.26
	Polar Diffusion	$(\rho_t, \boldsymbol{x}_t)$	0.05	9.40	**3.09**
	Polar Diffusion	$(e^{\rho_t}, \boldsymbol{y}_t)$	0.05	9.40	**3.15**
	Polar Diffusion	$(\rho_t, \boldsymbol{y}_t)$	0.05	9.38	3.18
ImageNet 64x64	Improved DDPM Nichol & Dhariwal (2021)	$\boldsymbol{y}_t$	1.0	-	27.0
	Polar Diffusion	$(e^{\rho_t}, \boldsymbol{x}_t)$	0.05	-	**23.7**
	Polar Diffusion (no-std-e^{ρ_t})	$(e^{\rho_t}, \boldsymbol{x}_t)$	0.05	-	29.1

$z \in \mathbb{R}^{C\times H\times W}$. We first flatten z into a vector in $\mathbb{R}^{CHW}$ and then apply a linear layer to obtain the scalar radial prediction: $\hat{\epsilon}_{\rho_t} = \mathrm{Linear}\left(\mathrm{Flatten}(z)\right) \in \mathbb{R}$.

5 Experiments

5.1 Toy Experiment

Following the low-dimensional manifold setting considered in Li & He (2026), we first evaluate whether polar diffusion better preserves the structure of data embedded in a higher-dimensional ambient space. We construct a two-dimensional spiral dataset and embed it into $\mathbb{R}^D$ using a fixed random linear projection. Both polar diffusion and the DDPM baseline are trained directly in the resulting D-dimensional space.

For both methods, we use the same 5-layer ReLU MLP hidden architecture for noise prediction. The DDPM baseline takes $(\boldsymbol{y}_t, t)$ as input and predicts the Euclidean noise $\boldsymbol{\epsilon} \in \mathbb{R}^D$. In contrast, Polar Diffusion takes $(\boldsymbol{x}_t, e^{\rho_t}, t)$ as input and predicting a radial component $\epsilon_\rho \in \mathbb{R}$ together with a tangential component $\boldsymbol{\epsilon}_{\boldsymbol{x}} \in T_{\boldsymbol{x}_t}\mathbb{S}^{D-1} \subset \mathbb{R}^D$. Thus, the Polar Diffusion model produces $D+1$ scalar outputs in total. To evaluate the learned distribution, the generated D-dimensional samples are mapped back to the original two-dimensional space using the corresponding projection operator.

Figure 1 visually compares the generated samples for different ambient dimensions D. When D is small, both methods are able to approximate the underlying spiral distribution. However, as the ambient dimension increases, the samples produced by the DDPM baseline progressively deviate from the low-dimensional data structure. In contrast, Polar Diffusion more faithfully preserves the geometry of the spiral across the tested dimensions. These results suggest that explicitly separating radial and tangential denoising can be beneficial when the data distribution is concentrated near a low-dimensional manifold embedded in a higher-dimensional Euclidean space.

5.2 Image Generation

We evaluate Polar Diffusion on both CIFAR-10 Brock et al. (2018) and ImageNet 64×64 Razavi et al. (2019) datasets. For CIFAR-10, we use DDPM Ho et al. (2020) as the baseline, while for ImageNet 64×64, we use improved DDPM in Nichol & Dhariwal (2021). Both baselines adopt a UNet-based architecture. To ensure a fair comparison, we use the same model configurations and training hyperparameters as the corresponding baseline models. The quantitative results are reported in Table 1, while the qualitative results can be found in Appendix J. We test different model inputs and found $(e^{\rho_t}, \boldsymbol{x}_t, t)$ yields the best result. Also, applying standardization provides better results.

For the ImageNet 64×64 experiment, we use a UNet with channel widths $[C, 2C, 3C, 4C]$, where $C = 128$. We adopt the L_{simple} objective, a cosine noise schedule, and $T = 4000$ diffusion steps, following Nichol & Dhariwal (2021). For evaluation, we generate $10{,}000$ samples, consistent with the evaluation protocol used in Table 1 of Nichol & Dhariwal (2021). We compute the Inception Score (IS) and Fréchet Inception Distance (FID) using the evaluation pipeline provided by Torch-Fidelity[4] Obukhov et al. (2020).

[4]https://github.com/toshas/torch-fidelity

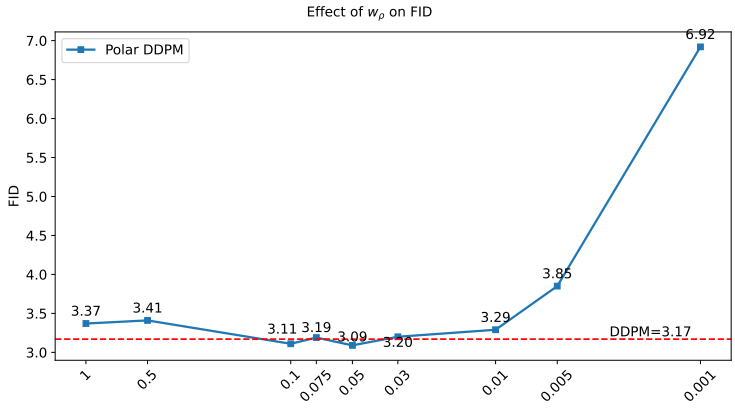


Figure 3: Effect of the w_ρ on CIFAR-10 generation performance. $w_\rho = 0.0001$ is not displayed since it does not converge.

As shown in Table 1, Polar Diffusion achieves a lower FID score than the corresponding baseline models, indicating improved sample quality under similar training and architectural settings.

5.3 Ablation Study on w_ρ

Prior work Liu et al. (2026) has shown that, under a variance-exploding diffusion formulation, the normal component induced by the standard Euclidean denoising objective (Equation 5) can dominate the tangential component in magnitude when the noise scale is small. This observation motivates us to investigate whether the radial and tangential prediction errors should be weighted equally in our polar formulation.

A key flexibility of Polar Diffusion is that the radial and tangential objectives can be controlled independently through w_ρ and $w_{\boldsymbol{x}}$ in Equation 11. In all experiments, we fix $w_{\boldsymbol{x}} = 1$ and vary w_ρ. We consider $w_\rho \in \{1.0, 0.5, 0.1, 0.075, 0.05, 0.03, 0.01, 0.005, 0.001, 0.0001\}$. Here, $w_\rho = 1$ corresponds to assigning equal weight to the radial and tangential prediction errors, consistent with the decomposition of the standard Euclidean DDPM objective. We primarily evaluate values $w_\rho \leq 1$ to study the effect of reducing the contribution of the radial term relative to the tangential term.

The results are shown in Figure 3. We observe that moderate down-weighting of the radial loss improves generation quality, with $w_\rho \in \{0.1, 0.05\}$ outperforming the equal-weight setting. Among the tested values, $w_\rho = 0.05$ achieves the best performance in FID score. In contrast, excessively small values of w_ρ lead to a degradation in performance, indicating that radial information remains important for accurate generation.

These results suggest that the standard equal weighting of radial and tangential errors is not necessarily optimal for image generation, and that there exists an intermediate regime in which reducing the radial contribution is beneficial. We leave the study of time-dependent weighting schemes and a more complete theoretical characterization of the optimal weighting strategy to future work.

6 Conclusion

In this work, we introduced Polar Diffusion, a geometric reformulation of Euclidean diffusion models that decomposes each sample into a radial component and an angular component on the hypersphere. We showed that the resulting polar dynamics can be constructed to induce the same marginal distribution as the standard VP-SDE after mapping back to Euclidean space. This formulation separates the denoising problem into radial and tangential components and allows their prediction errors to be weighted independently during training.

Our experiments demonstrate the benefit of this decomposition across both synthetic and image-generation tasks. In particular, our ablation study shows that assigning equal weight to the radial and tangential losses is not necessarily optimal: moderate down-weighting of the radial objective consistently improves generation quality. Our results and model structure suggest that explicitly distinguishing radial and tangential denoising provides a simple and effective way to incorporate geometric structure into conventional diffusion models without changing their underlying forward noising distribution. Deriving principled or time-dependent weighting strategies for the radial and tangential objectives is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with checking mathematical derivations and implementing parts of the experimental code. We did not use generative AI tools to generate synthetic datasets, formulate the central research hypothesis, design the core methodology, or interpret the experimental results; the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to improve the readability and grammatical clarity of portions of the manuscript, and assist with formatting and presentation.

All AI-assisted mathematical content was independently verified by the authors. AI-assisted code was manually reviewed and tested through the experiments reported in this paper, and all AI-assisted writing was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76:1–12, 2008.

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.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Zilu Guo, Qing Wang, Jun Du, Jia Pan, Qing-Feng Liu, and Chin-Hui Lee. A variance-preserving interpolation approach for diffusion models with applications to single channel speech enhancement and recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32: 3025–3038, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Ann B. Lee, Kim S. Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54:83–103, 2003.

J.M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019. ISBN 9783319917542. URL `https://books.google.com/books?id=UIPltQEACAAJ`.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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.

Zichen Liu, Wei Zhang, and Tiejun Li. Improving the euclidean diffusion generation of manifold data by mitigating score function singularity. *Advances in Neural Information Processing Systems*, 38:110885–110917, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PmLR, 2021.

Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. URL https://github.com/toshas/torch-fidelity. Version: 0.4.0, DOI: 10.5281/zenodo.3786539.

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications*, pp. 65–84. Springer, 2003.

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

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

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

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.

Neta Shaul, Chao Liu, Arash Vahdat, and Julius Berner. Parallel decoding distillation for fast image and video generation. *arXiv preprint arXiv:2607.26004*, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Michal Stypulkowski, Konstantinos Vougioukas, Sen He, Maciej Zieba, Stavros Petridis, and Maja Pantic. Diffused Heads: Diffusion Models Beat GANs on Talking-Face Generation . In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5089–5098, Los Alamitos, CA, USA, January 2024. IEEE Computer Society. doi: 10.1109/WACV57701.2024.00502. URL https://doi.ieeecomputersociety.org/10.1109/WACV57701.2024.00502.

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

A Brownian motion

A.1 Properties of Euclidean Brownian Motion

Let $\boldsymbol{W}_t = (W_t^1, \dots, W_t^n)^\top$ denote a standard Brownian motion in $\mathbb{R}^n$. Its components are independent one-dimensional Brownian motions satisfying

$$\mathbb{E}[d\boldsymbol{W}_t] = \boldsymbol{0}, \qquad \mathbb{E}[d\boldsymbol{W}_t d\boldsymbol{W}_t^\top] = \boldsymbol{I}\, dt. \tag{13}$$

In Itô differential notation, this is equivalently written as

$$dW_t^i\, dW_t^j = \delta_{ij}\, dt, \tag{14}$$

where δ_{ij} is the Kronecker delta. Consequently,

$$d\boldsymbol{W}_t^\top d\boldsymbol{W}_t = n\, dt. \tag{15}$$

More generally, for vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$,

$$(\boldsymbol{a}^\top d\boldsymbol{W}_t)(\boldsymbol{b}^\top d\boldsymbol{W}_t) = \boldsymbol{a}^\top \boldsymbol{b}\, dt. \tag{16}$$

In particular, if $\|\boldsymbol{a}\|_2 = 1$, then

$$\boldsymbol{a}^\top d\boldsymbol{W}_t \tag{17}$$

has variance dt and therefore represents an infinitesimal one-dimensional Brownian increment.

A.2 Properties of Brownian Motion on the Hypersphere

Brownian motion on the unit hypersphere can be constructed by continuously projecting Euclidean Brownian noise onto the tangent space. In Stratonovich form, the spherical Brownian motion $\boldsymbol{x}_t \in \mathbb{S}^{n-1}$ satisfies

$$d\boldsymbol{x}_t = P^{\perp}(\boldsymbol{x}_t) \circ d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{18}$$

The Stratonovich formulation is particularly convenient on manifolds because it obeys the ordinary chain rule and preserves the geometric constraint.

Indeed, applying the Stratonovich chain rule to $\|\boldsymbol{x}_t\|_2^2$ gives

$$d\|\boldsymbol{x}_t\|_2^2 = 2\boldsymbol{x}_t^{\top} \circ d\boldsymbol{x}_t \tag{19}$$

$$= 2\underbrace{\boldsymbol{x}_t^{\top} P^{\perp}(\boldsymbol{x}_t)}_{=0} \circ d\boldsymbol{W}_t^{\mathbb{R}^n} = 0. \tag{20}$$

Therefore, if $\|\boldsymbol{x}_0\|_2 = 1$, then

$$\|\boldsymbol{x}_t\|_2 = 1 \qquad \text{for all } t, \tag{21}$$

so the process remains on $\mathbb{S}^{n-1}$.

B From Stratonovich to Itô on the Hypersphere

Proposition 3. *For stratonovich SDE written as*

$$d\boldsymbol{x}_t = g_t \circ d\boldsymbol{W}_t^{\mathbb{S}}, \tag{22}$$

where $g_t \in \mathbb{R}$, the corresponding Itô's SDE is

$$d\boldsymbol{x}_t = -\frac{n-1}{2}\, g_t^2\, \boldsymbol{x}_t\, dt + g_t\, P^{\perp}(\boldsymbol{x}_t)\, d\boldsymbol{W}_t^{\mathbb{R}^n}, \tag{23}$$

Proof. By Equation 18, we have $dW_t^{\mathbb{S}} = P^{\perp}(\boldsymbol{x}_t) \circ d\boldsymbol{W}_t^{\mathbb{R}^n}$ with $P^{\perp}(\boldsymbol{x}) = \boldsymbol{I}_n - \boldsymbol{x}\boldsymbol{x}^{\top}$. Since g_t is a scalar and the Stratonovich integral is linear in its integrand, Equation 22 can be written as

$$d\boldsymbol{x}_t = g_t\, P^{\perp}(\boldsymbol{x}) \circ d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{24}$$

According to Øksendal (2003), this will induce the Itô's formula:

$$d\boldsymbol{x}_t = -\frac{n-1}{2}\, g_t^2\, \boldsymbol{x}_t\, dt + g_t\, P^{\perp}(\boldsymbol{x}_t)\, d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{25}$$

□

C Multidimensional Itô's formula

Consider an Itô process $\boldsymbol{y}_t \in \mathbb{R}^n$ satisfying

$$d\boldsymbol{y}_t = \boldsymbol{b}(\boldsymbol{y}_t, t)\, dt + \boldsymbol{\sigma}(\boldsymbol{y}_t, t)\, d\boldsymbol{W}_t^{\mathbb{R}^n}, \tag{26}$$

where $\boldsymbol{W}_t^{\mathbb{R}^d}$ is a d-dimensional Brownian motion in $\mathbb{R}^n$ and $\boldsymbol{\sigma}(\boldsymbol{y}_t, t) \in \mathbb{R}^{n\times d}$. Define

$$\boldsymbol{a}(\boldsymbol{y}_t, t) = \boldsymbol{\sigma}(\boldsymbol{y}_t, t)\boldsymbol{\sigma}(\boldsymbol{y}_t, t)^{\top}. \tag{27}$$

Let

$$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m, \qquad \boldsymbol{f} = (f_1, \ldots, f_m)^{\top}, \tag{28}$$

be twice continuously differentiable. Applying Itô's formula component-wise gives

$$d\boldsymbol{f}(\boldsymbol{y}_t) = J_{\boldsymbol{f}}(\boldsymbol{y}_t)\, d\boldsymbol{y}_t + \frac{1}{2}\begin{bmatrix} \mathrm{Tr}\left[\boldsymbol{a}_t \nabla^2 f_1(\boldsymbol{y}_t)\right] \\ \vdots \\ \mathrm{Tr}\left[\boldsymbol{a}_t \nabla^2 f_m(\boldsymbol{y}_t)\right] \end{bmatrix} dt, \tag{29}$$

where $J_{\boldsymbol{f}}(\boldsymbol{y}_t) \in \mathbb{R}^{m\times n}$ is the Jacobian of $\boldsymbol{f}$, and $\nabla^2 f$ denotes the Hessian of function f.

Note that under isotropic covariance $\boldsymbol{a}_t = c_t \boldsymbol{I}$ where $c_t \in \mathbb{R}_+$, Equation 29 reduces to

$$d\boldsymbol{f}(\boldsymbol{y}_t) = J_{\boldsymbol{f}}(\boldsymbol{y}_t)\, d\boldsymbol{y}_t + \frac{c_t}{2}\Delta \boldsymbol{f}(\boldsymbol{y}_t)\, dt. \tag{30}$$

where Δ is the Laplace operator.

D PROOF OF PROPOSITION 1

Proof. We now prove that the Euclidean diffusion in Equation 1 induces the polar diffusion in Equation 6, using multidimensional Itô's formula (see appendix C).

For VP-SDE in Equation 1, we use its polar representation and define

$$r_t = \|\boldsymbol{y}_t\|, \qquad \rho_t = \log r_t, \qquad \boldsymbol{x}_t = \frac{\boldsymbol{y}_t}{\|\boldsymbol{y}_t\|} \in \mathbb{S}^{n-1}. \tag{31}$$

Normal dynamics ρ_t. Let

$$f(\boldsymbol{y}) = \|\boldsymbol{y}\|. \tag{32}$$

It is obviously a smooth function. Its gradient, hessian, and laplacian are

$$\nabla f(\boldsymbol{y}) = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} = \boldsymbol{x}, \tag{33}$$

$$\nabla^2 f(\boldsymbol{y}) = \frac{1}{r}\left(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top\right). \tag{34}$$

$$\Delta f(\boldsymbol{y}) = \mathrm{Tr}(\nabla^2 f(\boldsymbol{y})) = \frac{1}{r}(n-1) \tag{35}$$

Since the diffusion covariance of Equation 1 is

$$\boldsymbol{a}_t = \beta_t \boldsymbol{I}, \tag{36}$$

applying Equation 30 gives

$$dr_t = \boldsymbol{x}_t^\top d\boldsymbol{y}_t + \frac{1}{2}\beta_t \frac{1}{r_t}(n-1)dt \tag{37}$$

$$= \boldsymbol{x}_t^\top \left(-\frac{1}{2}\beta_t \boldsymbol{y}_t\, dt + \sqrt{\beta_t}\, d\boldsymbol{W}_t^{\mathbb{R}^n}\right) + \frac{\beta_t}{2r_t}(n-1)dt. \tag{38}$$

$$= -\frac{1}{2}\beta_t \boldsymbol{x}_t^\top \boldsymbol{y}_t dt + \sqrt{\beta_t}\boldsymbol{x}_t^\top d\boldsymbol{W}_t^{\mathbb{R}^n} + \frac{\beta_t}{2r_t}(n-1)dt \tag{39}$$

Since $\boldsymbol{y}_t = r_t \boldsymbol{x}_t$, and $\|\boldsymbol{x}_t\| = 1$, we have

$$\boldsymbol{x}_t^\top \boldsymbol{y}_t = r_t. \tag{40}$$

Therefore,

$$dr_t = \left[-\frac{1}{2}\beta_t r_t + \frac{n-1}{2r_t}\beta_t\right] dt + \sqrt{\beta_t}\, dW_t^{\mathbb{R}}, \tag{41}$$

where

$$dW_t^{\mathbb{R}} = \boldsymbol{x}_t^\top d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{42}$$

Since $\|\boldsymbol{x}_t\| = 1$, its quadratic variation satisfies

$$(dW_t^{\mathbb{R}})^2 = \boldsymbol{x}_t^\top \boldsymbol{x}_t\, dt = dt, \tag{43}$$

so $W_t^{\mathbb{R}}$ is a one-dimensional Brownian motion.

Now let

$$\rho_t = \log r_t. \tag{44}$$

Applying the one-dimensional Itô's formula to Equation 41 yields

$$d\rho_t = \frac{1}{r_t} dr_t - \frac{\beta_t}{2r_t^2} dt \tag{45}$$

$$= \frac{1}{r_t}\left[-\frac{1}{2}\beta_t r_t + \frac{n-1}{2r_t}\beta_t\right] dt + \frac{\sqrt{\beta_t}}{r_t} dW_t^{\mathbb{R}} - \frac{\beta_t}{2r_t^2} dt \tag{46}$$

$$= \left[-\frac{1}{2}\beta_t + \frac{n-2}{2r_t^2}\beta_t\right] dt + \frac{\sqrt{\beta_t}}{r_t} dW_t^{\mathbb{R}}. \tag{47}$$

Using $r_t = e^{\rho_t}$, we obtain

$$\boxed{d\rho_t = \left[-\frac{1}{2}\beta_t + \frac{n-2}{2}\beta_t e^{-2\rho_t}\right] dt + \sqrt{\beta_t} e^{-\rho_t} dW_t^{\mathbb{R}}}. \tag{48}$$

Tangential dynamics $\boldsymbol{x}_t$. We next derive the dynamics of

$$\boldsymbol{x}_t = \frac{\boldsymbol{y}_t}{\|\boldsymbol{y}_t\|}. \tag{49}$$

Consider the smooth map

$$\boldsymbol{f} : \mathbb{R}^n \setminus \{\mathbf{0}\} \to \mathbb{S}^{n-1} \subset \mathbb{R}^n, \qquad \boldsymbol{f}(\boldsymbol{y}) = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}. \tag{50}$$

Its Jacobian is

$$J_{\boldsymbol{f}}(\boldsymbol{y}) = \frac{1}{r}\left(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top\right) \tag{51}$$

$$= \frac{1}{r}P^\perp(\boldsymbol{x}). \tag{52}$$

For each component

$$f_i(\boldsymbol{y}) = \frac{y_i}{r}, \tag{53}$$

the laplacian is

$$\Delta f_i(\boldsymbol{y}) = -\frac{n-1}{r^2}f_i(\boldsymbol{y}) = -\frac{n-1}{r^2}x_i. \tag{54}$$

Hence, in vector form,

$$\Delta \boldsymbol{f}(\boldsymbol{y}) = -\frac{n-1}{r^2}\boldsymbol{x}. \tag{55}$$

Applying Itô's formula in Equation 30 gives

$$d\boldsymbol{x}_t = \frac{1}{r}P^\perp(\boldsymbol{x}_t)d\boldsymbol{y}_t + \frac{\beta_t}{2}\Delta\boldsymbol{f}(\boldsymbol{y}_t)dt \tag{56}$$

$$= \frac{1}{r}P^\perp(\boldsymbol{x}_t)(-\frac{1}{2}\beta_t\boldsymbol{y}_t dt + \sqrt{\beta_t}d\boldsymbol{W}_t^{\mathbb{R}^n}) + \frac{\beta_t}{2}\Delta\boldsymbol{f}(\boldsymbol{y}_t)dt \tag{57}$$

$$= -\frac{\beta_t}{2r}\underbrace{P^\perp(\boldsymbol{x}_t)\boldsymbol{y}_t}_{=0}\,dt + \frac{\sqrt{\beta_t}}{r}P^\perp(\boldsymbol{x}_t)d\boldsymbol{W}_t^{\mathbb{R}^n} + \frac{\beta_t}{2}\Delta\boldsymbol{f}(\boldsymbol{y}_t)dt. \tag{58}$$

$$= \frac{\sqrt{\beta_t}}{r_t}P^\perp(\boldsymbol{x}_t)\,d\boldsymbol{W}_t^{\mathbb{R}^n} - \frac{n-1}{2}\frac{\beta_t}{r_t^2}\boldsymbol{x}_t\,dt. \tag{59}$$

Using $r_t = e^{\rho_t}$,

$$d\boldsymbol{x}_t = -\frac{n-1}{2}\beta_t e^{-2\rho_t}\boldsymbol{x}_t\,dt + \sqrt{\beta_t}e^{-\rho_t}P^\perp(\boldsymbol{x}_t)\,d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{60}$$

Equation 60 is the extrinsic Itô representation of a Brownian diffusion on the sphere, with instantaneous diffusion rate $\beta_t e^{-2\rho_t}$. The first term is the Itô curvature correction required to keep $\boldsymbol{x}_t$ on $\mathbb{S}^{n-1}$. Equivalently, in Stratonovich form, this curvature correction is absorbed into the stochastic integral (Proposition 3), yielding

$$d\boldsymbol{x}_t = \sqrt{\beta_t}e^{-\rho_t}P^\perp(\boldsymbol{x}_t) \circ d\boldsymbol{W}_t^{\mathbb{R}^n}. \tag{61}$$

Since $P^\perp(\boldsymbol{x}_t)$ projects the ambient Brownian increment onto $T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}$, we may equivalently denote the resulting intrinsic spherical Brownian increment by $d\boldsymbol{W}_t^{\mathbb{S}}$ and write

$$\boxed{d\boldsymbol{x}_t = \sqrt{\beta_t}e^{-\rho_t} \circ d\boldsymbol{W}_t^{\mathbb{S}}}. \tag{62}$$

Combining Equation 48 and Equation 62, the Euclidean VP-SDE induces the polar dynamics

$$\begin{cases} d\rho_t = \left[-\frac{1}{2}\beta_t + \frac{n-2}{2}\beta_t e^{-2\rho_t}\right]dt + \sqrt{\beta_t}e^{-\rho_t}dW_t^{\mathbb{R}}, \\ d\boldsymbol{x}_t = \sqrt{\beta_t}e^{-\rho_t} \circ d\boldsymbol{W}_t^{\mathbb{S}}. \end{cases} \tag{63}$$

□

E PROOF OF COROLLARY 1

Proof. We recall the time-reversal theorem in the general form we need (Anderson, 1982; Song et al., 2020).

Theorem 1 (Reverse-time SDE, general diffusion coefficient)**.** *Let $\boldsymbol{z}_t \in \mathbb{R}^m$ solve the Itô SDE*

$$d\boldsymbol{z}_t = f(\boldsymbol{z}_t, t)\, dt + G(\boldsymbol{z}_t, t)\, d\boldsymbol{W}_t^{\mathbb{R}^n}, \qquad D(\boldsymbol{z}, t) := G(\boldsymbol{z}, t)G(\boldsymbol{z}, t)^\top, \tag{64}$$

and suppose $\boldsymbol{z}_t$ admits a density p_t w.r.t. Lebesgue measure. Then the time-reversed process solves the backward Itô SDE

$$d\boldsymbol{z}_t = \Big[f(\boldsymbol{z}_t, t) - \nabla\cdot D(\boldsymbol{z}_t, t) - D(\boldsymbol{z}_t, t)\nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z}_t)\Big]dt + G(\boldsymbol{z}_t, t)\, d\bar{\boldsymbol{W}}_t^{\mathbb{R}^n}, \tag{65}$$

where $(\nabla\cdot D)_i := \sum_j \partial_{z_j} D_{ij}$. When $G = g(t)I$ for $g(t) \in \mathbb{R}$, $\nabla\cdot D = 0$.

Throughout, $p_t(\rho, \boldsymbol{x})$ denotes the joint density of $(\rho_t, \boldsymbol{x}_t)$ with respect to the product measure $\sigma_{\mathbb{S}^{n-1}}(d\boldsymbol{x}) \otimes d\rho$, i.e. the polar density $p_t(\rho, \boldsymbol{x}) = e^{n\rho}\, \tilde{p}_t(e^\rho \boldsymbol{x})$ where $\tilde{p}_t$ is the ambient density on $\mathbb{R}^n$.

Reverse SDE for ρ_t. The process ρ_t is autonomous and one-dimensional, but its diffusion coefficient $\sqrt{\beta_t}e^{-\rho_t}$ is *state dependent*, so Theorem 1 must be applied with the divergence term retained. With

$$f(\rho_t, t) = -\tfrac{1}{2}\beta_t + \tfrac{n-2}{2}\beta_t e^{-2\rho_t}, \qquad D(\rho_t, t) = \beta_t e^{-2\rho_t}, \qquad \partial_\rho D = -2\beta_t e^{-2\rho_t}, \tag{66}$$

applying Theorem 1 yields

$$d\rho_t = \Big[-\tfrac{1}{2}\beta_t + \tfrac{n+2}{2}\beta_t e^{-2\rho_t} - \beta_t e^{-2\rho_t}\, \nabla_\rho \log p_t(\rho_t, \boldsymbol{x}_t)\Big]dt + \sqrt{\beta_t}\, e^{-\rho_t}\, d\bar{W}_t^{\mathbb{R}}. \tag{67}$$

Reverse SDE for $\boldsymbol{x}_t$. We reduce to Theorem 1 in a local orthonormal frame, where the spherical noise is a standard Brownian motion in $\mathbb{R}^{n-1}$ and a Lebesgue density exists. Fix a chart $\theta \in \mathbb{R}^{n-1}$ of $\mathbb{S}^{n-1}$ with metric g_{ij}, and set $\gamma_t^2 := \beta_t e^{-2\rho_t}$. Since $\gamma_t \circ d\boldsymbol{W}_t^{\mathbb{S}}$ has generator $\frac{\gamma_t^2}{2}\Delta_g$, its Itô form in these coordinates is

$$d\theta^i = f^i\, dt + G^i{}_k\, dB^k, \qquad D^{ij} := (GG^\top)^{ij} = \gamma_t^2 g^{ij}, \qquad f^i = \tfrac{\gamma_t^2}{2}\Big[\partial_j g^{ij} + g^{ij}\partial_j \log\sqrt{|g|}\Big], \tag{68}$$

with $\boldsymbol{B}_t$ a standard Brownian motion on $\mathbb{R}^{n-1}$. Note that f is exactly the Itô–Stratonovich correction associated with G, since equation 68 is driftless in Stratonovich form.

Let p_t denote the density with respect to the Riemannian volume measure and $\bar{p}_t = p_t\sqrt{|g|}$ the Lebesgue density in the chart. The diffusion matrix in equation 68 is nondegenerate, so Theorem 1 applies directly, giving the reverse drift $b^i = f^i - \partial_j D^{ij} - D^{ij}\partial_j \log \bar{p}_t$. Substituting $\partial_j D^{ij} = \gamma_t^2 \partial_j g^{ij}$ and $\partial_j \log \bar{p}_t = \partial_j \log p_t + \partial_j \log\sqrt{|g|}$,

$$b^i = \underbrace{-\tfrac{\gamma_t^2}{2}\partial_j g^{ij} - \tfrac{\gamma_t^2}{2} g^{ij}\partial_j \log\sqrt{|g|}}_{=-f^i} - \gamma_t^2 g^{ij}\partial_j \log p_t = -f^i - \gamma_t^2\big(\operatorname{grad}_g \log p_t\big)^i. \tag{69}$$

Because the reverse SDE is a backward Itô SDE, its Stratonovich correction carries the opposite sign to the forward one, so $-f^i\, dt + G^i{}_k\, d\bar{B}^k = \gamma_t \circ d\bar{\boldsymbol{W}}_t^{\mathbb{S}}$ and

$$d\boldsymbol{x}_t = -\beta_t e^{-2\rho_t}\, \nabla^{\mathbb{S}}_{\boldsymbol{x}_t} \log p_t(\rho_t, \boldsymbol{x}_t)\, dt + \sqrt{\beta_t}\, e^{-\rho_t} \circ d\bar{\boldsymbol{W}}_t^{\mathbb{S}}. \tag{70}$$

Both sides of equation 70 are chart-independent, so the identity holds globally. The joint diffusion $(\rho_t, \boldsymbol{x}_t)$ on $\mathbb{R} \times \mathbb{S}^{n-1}$ is block diagonal and γ_t is independent of θ, so the angular computation is unaffected by the radial component. □

F PROOF FOR PROPOSITION 2

The DDPM-style forward process in Equation 2 has conditional density in closed form

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \frac{1}{(2\pi\sigma_t^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_t^2} \left\|\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0\right\|^2\right). \tag{71}$$

Using

$$\boldsymbol{y}_t = r_t \boldsymbol{x}_t = e^{\rho_t} \boldsymbol{x}_t, \qquad \boldsymbol{y}_0 = r_0 \boldsymbol{x}_0 = e^{\rho_0} \boldsymbol{x}_0, \tag{72}$$

we have

$$\left\|\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0\right\|^2 = \left\|r_t \boldsymbol{x}_t - \alpha_t r_0 \boldsymbol{x}_0\right\|^2 \tag{73}$$

$$= r_t^2 + \alpha_t^2 r_0^2 - 2\alpha_t r_t r_0 \boldsymbol{x}_t^\top \boldsymbol{x}_0. \tag{74}$$

Hence

$$\log q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = -\frac{1}{2\sigma_t^2} \left\|\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0\right\|^2 + \text{const} \tag{75}$$

$$= -\frac{1}{2\sigma_t^2} \left\|r_t \boldsymbol{x}_t - \alpha_t r_0 \boldsymbol{x}_0\right\|^2 + \text{const} \tag{76}$$

$$= -\frac{r_t^2 + \alpha_t^2 r_0^2 - 2\alpha_t r_t r_0 \boldsymbol{x}_t^\top \boldsymbol{x}_0}{2\sigma_t^2} + \text{const} \tag{77}$$

Let

$$c_t = \boldsymbol{x}_t^\top \boldsymbol{x}_0. \tag{78}$$

Then

$$\left\|\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0\right\|^2 = r_t^2 + \alpha_t^2 r_0^2 - 2\alpha_t r_t r_0 c_t. \tag{79}$$

The Euclidean volume element satisfies

$$d\boldsymbol{y}_t = r_t^{n-1} dr_t \, d\Omega(\boldsymbol{x}_t), \tag{80}$$

where $d\Omega(\boldsymbol{x}_t)$ denotes the Riemannian volume element (surface measure) on the unit hypersphere $\mathbb{S}^{n-1}$. Since $r_t = e^{\rho_t}$ and $dr_t = r_t d\rho_t$, we have

$$d\boldsymbol{y}_t = r_t^n d\rho_t d\Omega(\boldsymbol{x}_t) = e^{n\rho_t} d\rho_t d\Omega(\boldsymbol{x}_t). \tag{81}$$

Hence,

$$q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = q(e^{\rho_t}\boldsymbol{x}_t \mid e^{\rho_0}\boldsymbol{x}_0) e^{n\rho_t}. \tag{82}$$

Taking logarithms gives

$$\log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = -\frac{r_t^2 + \alpha_t^2 r_0^2 - 2\alpha_t r_t r_0 c_t}{2\sigma_t^2} + n\rho_t + \text{const}. \tag{83}$$

Equivalently,

$$\log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = -\frac{e^{2\rho_t} + \alpha_t^2 e^{2\rho_0} - 2\alpha_t e^{\rho_t + \rho_0} \boldsymbol{x}_t^\top \boldsymbol{x}_0}{2\sigma_t^2} + n\rho_t + \text{const}. \tag{84}$$

The **radial** ρ_t score is

$$\nabla_{\rho_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = \partial_{\rho_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) \tag{85}$$

$$= -\frac{1}{2\sigma_t^2} \left(2e^{2\rho_t} - 2\alpha_t e^{\rho_t + \rho_0} \boldsymbol{x}_t^\top \boldsymbol{x}_0\right) + n \tag{86}$$

$$= \boxed{n - \frac{e^{2\rho_t} - \alpha_t e^{\rho_t + \rho_0} \boldsymbol{x}_t^\top \boldsymbol{x}_0}{\sigma_t^2}}. \tag{87}$$

The Euclidean gradient for $\boldsymbol{x}_t$ is

$$\nabla_{\boldsymbol{x}_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = \nabla_{\boldsymbol{x}_t} \left(\frac{\alpha_t e^{\rho_t + \rho_0}}{\sigma_t^2} \boldsymbol{x}_t^\top \boldsymbol{x}_0 \right) \tag{88}$$

$$= \frac{\alpha_t e^{\rho_t + \rho_0}}{\sigma_t^2} \boldsymbol{x}_0. \tag{89}$$

To obtain the **tangential** score for $\boldsymbol{x}_t$, which is the Riemannian gradient on the sphere, we project the Euclidean gradient onto the tangent space $T_{\boldsymbol{x}_t}\mathbb{S}^{n-1}$:

$$\nabla^{\mathbb{S}}_{\boldsymbol{x}_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) = P^{\perp}(\boldsymbol{x}) \nabla_{\boldsymbol{x}_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) \tag{90}$$

$$= \boxed{\frac{\alpha_t e^{\rho_t + \rho_0}}{\sigma_t^2} \left(\boldsymbol{x}_0 - (\boldsymbol{x}_t^\top \boldsymbol{x}_0) \boldsymbol{x}_t \right)}. \tag{91}$$

G Proof for corollary 2

From Euclidean $\boldsymbol{y}_t = \alpha_t \boldsymbol{y}_0 + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I})$, we can derive

$$\boldsymbol{\epsilon} = \frac{\boldsymbol{y}_t - \alpha_t \boldsymbol{y}_0}{\sigma_t} \tag{92}$$

$$= \frac{r_t \boldsymbol{x}_t - \alpha_t r_0 \boldsymbol{x}_0}{\sigma_t}. \tag{93}$$

Using decomposition in Equation 9, we get

$$\boldsymbol{\epsilon}_\rho = \boldsymbol{x}_t^T \boldsymbol{\epsilon} = \frac{r_t - \alpha_t r_0 \boldsymbol{x}_t^T \boldsymbol{x}_0}{\sigma_t}, \tag{94}$$

and

$$\boldsymbol{\epsilon}_x = -\frac{\alpha_t r_0}{\sigma_t} \left[\boldsymbol{x_0} - (\boldsymbol{x}_t^T \boldsymbol{x}_0) \boldsymbol{x}_t \right], \tag{95}$$

Substitute in Equation 87 and Equation 91,

$$\boxed{\begin{aligned} \nabla_{\rho_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) &= n - \frac{r_t}{\sigma_t} \epsilon_\rho \\ \nabla^{\mathbb{S}}_{\boldsymbol{x}_t} \log q(\rho_t, \boldsymbol{x}_t \mid \rho_0, \boldsymbol{x}_0) &= -\frac{r_t}{\sigma_t} \boldsymbol{\epsilon_x} \end{aligned}}. \tag{96}$$

H Proof for corollary 3

The DDPM loss in Equation 5 can be written as

$$\mathcal{L} = \mathbb{E}\left[|| \left(P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta + P(\boldsymbol{x}_t) \boldsymbol{\epsilon_\theta} \right) - \left(P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon} + P(\boldsymbol{x}_t) \boldsymbol{\epsilon} \right) ||^2 \right] \tag{97}$$

$$= \mathbb{E}\left[|| \left(P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta - P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon} \right) + \left(P(\boldsymbol{x}_t) \boldsymbol{\epsilon_\theta} - P(\boldsymbol{x}_t) \boldsymbol{\epsilon} \right) ||^2 \right] \tag{98}$$

$$= \mathbb{E}\left[||P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta - P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}||^2 + ||P(\boldsymbol{x}_t) \boldsymbol{\epsilon_\theta} - P(\boldsymbol{x}_t) \boldsymbol{\epsilon}||^2 \right] \tag{99}$$

$$+ \mathbb{E}\left[2(\boldsymbol{\epsilon_\theta} - \boldsymbol{\epsilon})^T \underbrace{(P^{\perp}(\boldsymbol{x}_t))^T P(\boldsymbol{x}_t)}_{=\mathbf{0}} (\boldsymbol{\epsilon_\theta} - \boldsymbol{\epsilon}) \right] \tag{100}$$

$$= \mathbb{E}\left[||P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta - \boldsymbol{\epsilon}_{\boldsymbol{x}_t}||^2 + ||\boldsymbol{x}_t \boldsymbol{x}_t^T (\boldsymbol{\epsilon_\theta} - \boldsymbol{\epsilon})||^2 \right] \tag{101}$$

$$= \mathbb{E}\left[||P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta - \boldsymbol{\epsilon}_{\boldsymbol{x}_t}||^2 + \underbrace{||\boldsymbol{x}_t||^2}_{=1} \cdot (\boldsymbol{x}_t^T (\boldsymbol{\epsilon_\theta} - \boldsymbol{\epsilon}))^2 \right] \tag{102}$$

$$= \mathbb{E}\left[||P^{\perp}(\boldsymbol{x}_t) \boldsymbol{\epsilon}_\theta - \boldsymbol{\epsilon}_{\boldsymbol{x}_t}||^2 + (\boldsymbol{x}_t^T \boldsymbol{\epsilon_\theta} - \epsilon_{\rho_t})^2 \right] \tag{103}$$

I ALGORITHMS

Algorithm 1 Polar Diffusion Training

while not converged **do**
  Sample $\boldsymbol{y}_0 \sim q_{data}$
  Sample $k \sim \text{Uniform}\{1, \dots, T\}$
  Sample $\boldsymbol{\epsilon_y} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{y}_k \leftarrow \alpha_k \boldsymbol{y}_0 + \sigma_k \boldsymbol{\epsilon_y}$
  $\boldsymbol{x}_k \leftarrow \dfrac{\boldsymbol{y}_k}{\|\boldsymbol{y}_k\|}, \quad \rho_k \leftarrow \log \|\boldsymbol{y}_k\|$
  $\epsilon_\rho \leftarrow \boldsymbol{x}_k^\top \boldsymbol{\epsilon_y}$
  $\boldsymbol{\epsilon_x} \leftarrow \boldsymbol{\epsilon_y} - \epsilon_\rho \boldsymbol{x}_k$
  $\tilde{r}_k = \text{std}(\exp(\rho_k))$ ▷ Section 4.1
  $\mathcal{L} = w_\rho \left\| \epsilon_{\theta,\rho}(\tilde{r}_k, \boldsymbol{x}_k, k) - \epsilon_\rho \right\|_2^2 + w_x \left\| \boldsymbol{\epsilon}_{\theta,\boldsymbol{x}}(\tilde{r}_k, \boldsymbol{x}_k, k) - \boldsymbol{\epsilon_x} \right\|_2^2$
  Optimize $\mathcal{L}$
end while

Algorithm 2 Polar Diffusion Sampling

Sample $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$
$\boldsymbol{x}_T \leftarrow \dfrac{\boldsymbol{y}_T}{\|\boldsymbol{y}_T\|}, \quad \rho_T \leftarrow \log \|\boldsymbol{y}_T\|$
$\Delta \leftarrow 1/T$ ▷ step size
for $k = T, T-1, \dots, 1$ **do**
  $\tilde{r}_k = \text{std}(\exp(\rho_k))$ ▷ Section 4.1
  $(\hat{\epsilon}_\rho, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{x}}) \leftarrow \boldsymbol{\epsilon}_\theta(\tilde{r}_k, \boldsymbol{x}_k, k)$
  $s_{\rho_k} \leftarrow n - \dfrac{e^{\rho_k}}{\sigma_k} \hat{\epsilon}_\rho$ ▷ score for radial diffusion
  $\boldsymbol{s}_{\boldsymbol{x}_k} \leftarrow -\dfrac{e^{\rho_k}}{\sigma_k} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{x}}$ ▷ score for angular diffusion
  $\boldsymbol{s}_{\boldsymbol{x}_k} \leftarrow P^\perp(\boldsymbol{x}_k) \boldsymbol{s}_{\boldsymbol{x}_k}$ ▷ ensures $\boldsymbol{s}_{\boldsymbol{x}_k} \in T_{\boldsymbol{x}_k} \mathbb{S}^{n-1}$
  $\rho_{k-1} \leftarrow \rho_k - \left[-\frac{1}{2}\beta_k + \frac{n+2}{2}\beta_k e^{-2\rho_k} - \beta_k e^{-2\rho_k} \cdot s_{\rho_k} \right] \Delta + \sqrt{\beta_k \Delta} e^{-\rho_k} \epsilon_\rho$, where $\epsilon_\rho \sim \mathcal{N}(0,1)$
  $\boldsymbol{x}_{k-1} \leftarrow \text{Exp}_{\boldsymbol{x}_k}(\beta_k e^{-2\rho_k} \cdot \boldsymbol{s}_{\boldsymbol{x}_k} \Delta + \sqrt{\beta_k \Delta} e^{-\rho_k} P^\perp(\boldsymbol{x}_k) \boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$
end for
$\boldsymbol{y}_0 \leftarrow e^{\rho_0} \boldsymbol{x}_0$
return $\boldsymbol{y}_0$

J QUALITATIVE RESULTS

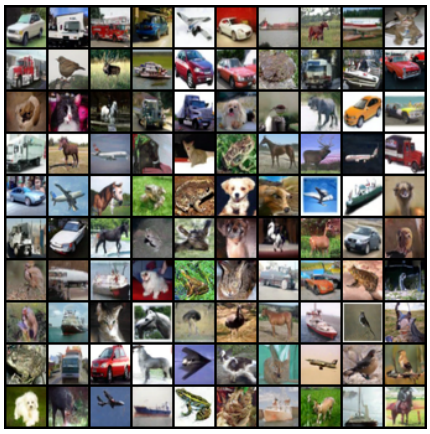


Figure 4: Unconditional qualitative results on CIFAR-10 dataset.

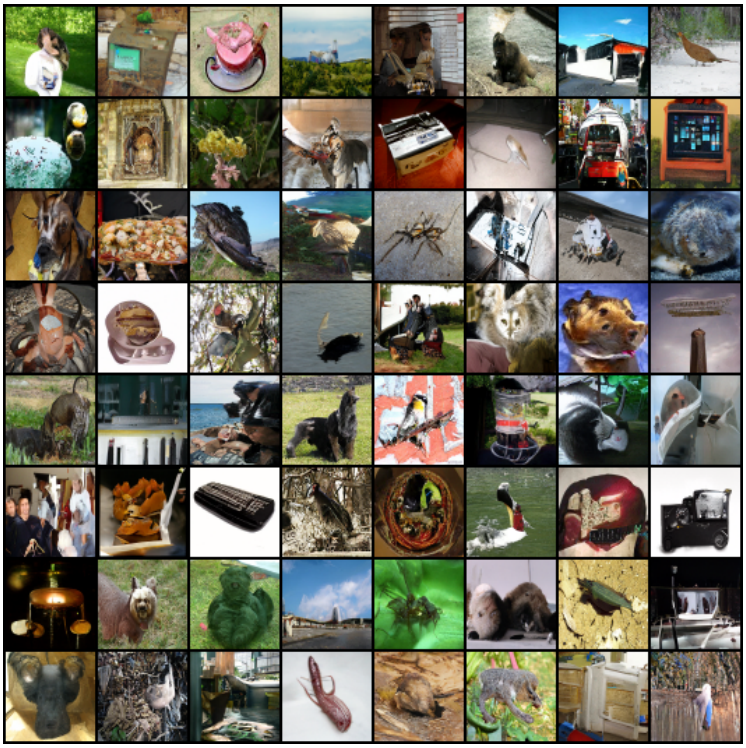


Figure 5: Unconditional qualitative results on ImageNet 64x64 dataset.