Efficient Diffusion Models for Symmetric Manifolds

Anonymous authors
Paper under double-blind review

Abstract

We present a framework for designing efficient diffusion models on symmetric Riemannian manifolds, which include the torus, sphere, special orthogonal group, and unitary group. While diffusion models on symmetric manifolds have gained significant attention, existing approaches often rely on the manifolds' heat kernels, which lack closed-form expressions and result in exponential-in-dimension per-iteration runtimes during training. We introduce a new diffusion model for symmetric-space manifolds, leveraging a projection of Euclidean Brownian motion to bypass explicit heat kernel computations. Our training algorithm minimizes a novel objective function derived via Ito's Lemma, with efficiently computable gradients, allowing each iteration to run in polynomial time for symmetric manifolds. Additionally, the symmetries of the manifold ensure the diffusion satisfies an "average-case" Lipschitz condition, enabling accurate and efficient sample generation. These improvements enhance both the training runtime and sample accuracy for key cases of symmetric manifolds, helping to bridge the gap between diffusion models on symmetric manifolds and Euclidean space.

1 Introduction

In recent years, diffusion-based methods have achieved significant success in generating synthetic data, including highly realistic images and videos (see OpenAI (2023)). Given a dataset $D \subseteq \mathbb{R}^d$ in a d-dimensional Euclidean space sampled from some unknown probability distribution π, the goal of a diffusion model (or any generative model) is to learn a distribution ν which approximates the distribution π and to generate new samples from ν. While most existing diffusion models generate samples from a probability distribution in Euclidean space $\mathbb{R}^d$ Ho et al. (2020); Rombach et al. (2022), many applications require data constrained to a d-dimensional non-Euclidean manifold $\mathcal{M}$, as seen in fields such as robotics Feiten et al. (2013); Urain et al. (2023); Shi et al. (2023); Selig (2013) and molecular drug discovery Shapovalov and Dunbrack (2011); Maji et al. (2019); Cheng et al. (2021); Leach et al. (2022); Watson et al. (2023), where configurations are often represented on symmetric-space manifolds like the torus, sphere, special orthogonal group $\mathbb{SO}(n)$, or unitary group $\mathbb{U}(n)$ where $n = \sqrt{d}$. It is possible to enforce manifold constraints by mapping samples from Euclidean space $\mathbb{R}^d$ to the manifold $\mathcal{M}$. However, this often leads to low-quality samples due to geometric distortions caused by the mapping Leach et al. (2022); Watson et al. (2023) For example, consider generating points from a distribution μ on the d-dimensional torus $\mathbb{T}_d = \mathbb{S}_1 \times \cdots \times \mathbb{S}_1$. A naive approach would map the dataset D to Euclidean space via the map ψ converting points on the torus to angles in $[0, 2\pi)^d \subseteq \mathbb{R}^d$. One can then train a Euclidean diffusion model on the dataset $\psi(D)$. However, this can severely distort the geometry of μ, leading to a multimodal distribution that is harder for a diffusion model to learn compared to the original unimodal distribution on the torus (see Appendix C).

To address this, several works have developed diffusion models directly constrained to non-Euclidean Riemannian manifolds De Bortoli et al. (2022); Lou et al. (2024); Huang et al. (2022); Zhu et al. (2024); Yim et al. (2023). However, a significant gap remains between the runtime and sampling guarantees of Euclidean and manifold-based diffusion models. For instance, while Euclidean models have a per-iteration runtime of $O(d)$ arithmetic operations and $O(1)$ gradient evaluations Ho et al. (2020); Rombach et al. (2022), the objectives of manifold diffusion models often require exponential runtime in the dimension De Bortoli et al. (2022); Lou et al. (2024). Reducing this runtime gap, particularly for symmetric manifolds, remains an open challenge.

To understand the technical difficulty, first consider the Euclidean case. At a high level, a diffusion model consists of two components: a forward noising process that adds noise over time $T > 0$ until the data is (nearly) indistinguishable from a Gaussian distribution, and a reverse denoising process that starts from a sample of this Gaussian distribution and gradually removes the noise to generate samples approximating the original distribution π Ho et al. (2020); Rombach et al. (2022). A latent variable model is used to approximate the reverse diffusion, where the latent variables

$z(t_1), z(t_2), \ldots, z(T)$ model random updates over discrete time intervals, approximating these updates as Gaussian distributions whose mean (and sometimes covariance) is modeled by a neural network. In the manifold case, the forward diffusion is standard Brownian motion on the manifold, and the reverse diffusion is the time-reversal of this process De Bortoli et al. (2022); Lou et al. (2024); Huang et al. (2022). However, because Brownian motion on a manifold involves adding *infinitesimal* Gaussian noise in the tangent space at each point, it is unclear how to model the reverse diffusion as a Gaussian latent variable model.

To overcome this, De Bortoli et al. (2022); Huang et al. (2022) move to continuous time, where the updates of the reverse diffusion Y_t converge to Gaussian distributions on the tangent space. The reverse diffusion is governed by a stochastic differential equation (SDE) involving the manifold's heat kernel. The heat kernel $p_{\tau|b}(\cdot|b)$ represents the density of Brownian motion at time τ, initialized at a point b. Training the reverse diffusion model thus involves minimizing an objective function that depends on the heat kernel De Bortoli et al. (2022); Huang et al. (2022); Lou et al. (2024). Even in the Euclidean case, the training objective is nonconvex, and there are generally no guarantees of a polynomial-in-dimension runtime for the overall training process. However, in Euclidean space, the heat kernel has a closed-form expression that can be computed in time linear in d, allowing each iteration of the training algorithm to run in polynomial time. For non-Euclidean manifolds, the lack of a closed-form heat kernel creates significant challenges, making the heat kernel computation a bottleneck during each iteration De Bortoli et al. (2022). On symmetric manifolds like the orthogonal group, the heat kernel can only be computed via inefficient series expansions which require a runtime that grows exponentially with d. For this reason, inaccurate approximations are oftentimes used, degrading the quality of generated samples De Bortoli et al. (2022); Lou et al. (2024). Another issue is that, on manifolds with non-zero curvature, such as the sphere, orthogonal group, and unitary group, standard Brownian motion cannot be obtained as the projection of Brownian motion in $\mathbb{R}^d$. As a result, previous works rely on numerical SDE or ODE solvers to compute samples from the forward diffusion during each evaluation of the objective function De Bortoli et al. (2022); Lou et al. (2024). The use of these solvers introduces significant computational bottlenecks in training diffusion models.

Our contributions. We study the problem of designing efficient diffusion models when $\mathcal{M}$ is a symmetric-space manifold, such as the torus $\mathbb{T}_d$, sphere $\mathbb{S}_d$, special orthogonal group $\mathbb{SO}(n)$, and the unitary group $\mathbb{U}(n)$ where $n = \sqrt{d}$, as well as direct products of these manifolds such as the special Euclidean group $\mathrm{SE}(n)$ which is isomorphic to $\mathbb{R}^n \times \mathbb{SO}(n)$. We present a new training algorithm (Algorithm 1) for these manifolds, where each iteration can be computed in $O(d)$ arithmetic operations for $\mathbb{T}_d$ or $\mathbb{S}_d$, and $O(d^{\frac{\omega}{2}})$ arithmetic operations for $\mathbb{SO}(n)$ or $\mathbb{U}(n)$, and $O(1)$ evaluations of the gradient of a model for the drift and diffusion terms of the reverse diffusion. Here $\omega \approx 2.37$ is the matrix multiplication exponent. This significantly improves upon the per-iteration bounds of previous methods (see Table 1). For example, on $\mathbb{SO}(n)$ and $\mathbb{U}(n)$ our method achieves exponential improvements, bringing the per-iteration runtime closer to that of the Euclidean case. Subsequently, we provide a sampling algorithm (Algorithm 2) along with a guarantee on its accuracy and runtime. Given an ε-minimizer of our training objective, the algorithm achieves an $\varepsilon \times \mathrm{poly}(d)$ bound on the total variation distance accuracy and a $\mathrm{poly}(d)$ runtime (Theorem 2.2). This improves upon the sampling accuracy bounds of De Bortoli et al. (2022), which are not polynomial in the dimension. Theorem 2.2 holds for more general manifolds that satisfy an average-case Lipschitz condition (Assumption 2.1). Using tools from random matrix theory, we prove this condition holds for the manifolds of interest (Lemma B.4).

Our paper introduces several new ideas. For the training result, we define a novel forward diffusion on $\mathcal{M}$ obtained by projecting Brownian motion in $\mathbb{R}^d$ onto $\mathcal{M}$ via a given map $\varphi : \mathbb{R}^d \to \mathcal{M}$, which satisfies the average-case Lipschitz condition and can be efficiently computed via the singular value decomposition when $\mathcal{M}$ is the unitary or orthogonal group. This choice of forward diffusion ensures that we can efficiently sample from our forward diffusion process in a simulation free manner–without requiring the use of an SDE (or ODE) solver– by sampling from a Gaussian in $\mathbb{R}^d$ and projecting this point onto $\mathcal{M}$. We also introduce a new training objective that bypasses the need to compute the manifold's heat kernel. By applying Ito's Lemma from stochastic calculus, we project the SDE for a reverse diffusion in Euclidean space onto $\mathcal{M}$. The drift term of the resulting SDE is expressed as an expectation of the Euclidean heat kernel. Since the Euclidean heat kernel has a closed-form expression and the projection map φ can be computed efficiently, we compute the objective in time $O(d^{\frac{\omega}{2}})$.

For the sampling result, we demonstrate that the reverse SDE satisfies a Lipschitz condition provided the projection map satisfies the average-case Lipschitz condition (Lemma B.4). Since the projection introduces a non-constant term in the SDE on the manifold, Girsanov's theorem techniques from prior works cannot be used to bound the accuracy. To address this, we develop an optimal transport-based approach, leading to a novel probabilistic coupling argument that provides the desired accuracy and runtime bounds. This approach is entirely different from previous proofs in Euclidean space Chen et al. (2023b;a); Cheng et al. (2022); Benton et al. (2023) and manifold-based diffusion models De Bortoli et al. (2022), which rely on Girsanov's theorem.

Algorithm	Unitary or Orthogonal group	Sphere	Torus
Score-based Riemannian De Bortoli et al. (2022)	$2^d + \mathrm{poly}(d, \frac{1}{\delta})$	$2^d + \mathrm{poly}(d, \frac{1}{\delta})$	$2^d + \mathrm{poly}(d, \frac{1}{\delta})$
Scaling Riemannian Lou et al. (2024)	$2^d + \mathrm{poly}(d, \frac{1}{\delta})$	$\mathrm{poly}(d, \frac{1}{\delta})$	$d \log(\frac{1}{\delta})$
This paper	$d^{\frac{\omega}{2}} \log(\frac{1}{\delta})$	$d \log(\frac{1}{\delta})$	$d \log(\frac{1}{\delta})$

Table 1: Arithmetic operations to compute the objective function's gradient per-iteration of the training algorithm, when $\mathcal{M}$ is the unitary group, orthogonal group, sphere, or torus.

2 RESULTS

For a manifold $\mathcal{M}$, we are given a projection map $\varphi : \mathbb{R}^d \to \mathcal{M}$ and a restricted-inverse map $\psi : \mathcal{M} \to \mathbb{R}^d$ such that $\varphi(\psi(x)) = x$ for all $x \in \mathcal{M}$. Denote by $\mathcal{T}_x\mathcal{M}$ the tangent space of $\mathcal{M}$ at x. We assume access to an oracle that computes the exponential map $\exp(x, v)$ on $\mathcal{M}$ for any $x \in \mathcal{M}$ and $v \in \mathcal{T}_x\mathcal{M}$. This oracle is not needed for our training algorithm (Algorithm 1); it is only required for the sample generation algorithm (Algorithm 2), which uses the trained model. We are given a dataset $D \subseteq \mathcal{M}$ sampled from π with support on $\mathcal{M}$.

We set $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ and $\psi : \mathbb{R}^d \to \mathbb{R}^d$ as identity maps when $\mathcal{M} = \mathbb{R}^d$. For the torus $\mathbb{T}_d$, $\varphi(x)[i] = x[i] \mod 2\pi$ maps points to their angles, and ψ is its inverse on $[0, 2\pi)^d$. For the sphere $\mathbb{S}_d$, $\varphi(x) = \frac{x}{\|x\|}$, and ψ embeds the unit sphere into $\mathbb{R}^d$. For groups $\mathbb{SO}(n)$ and $\mathbb{U}(n)$, the map φ takes each upper triangular matrix $X \in \mathbb{R}^{n\times n}$ (or $X \in \mathbb{C}^{n\times n}$), computes the spectral decomposition $U^*\Lambda U$ of $X + X^*$, and outputs $\varphi(X) = U$. The map ψ takes each matrix $U \in \mathcal{M}$, computes $U^*\Lambda U$ where $\Lambda = \frac{1}{n}\mathrm{diag}(n, n-1, \ldots, 1)$, scales the diagonal by $\frac{1}{2}$, and outputs the upper triangular entries of the result. For all of the above maps, $\psi(\mathcal{M})$ is contained in a ball of radius $\mathrm{poly}(d)$. Our general results hold under this assumption on ψ. For manifolds $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, which are direct products of manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$, where one is given maps φ_1, ψ_1 for $\mathcal{M}_1$ and φ_2, ψ_2 for $\mathcal{M}_2$, one can use the concatenated maps $\varphi = (\varphi_1, \varphi_2)$ and $\psi = (\psi_1, \psi_2)$.

We give an algorithm (Algorithm 1) which trains our model by minimizing a nonconvex objective function via stochastic gradient descent. Our training algorithm outputs trained models $f(x, t)$ and $g(x, t)$ for the drift and covariance terms of our reverse diffusion, and passes these trained models as inputs to our sample generation algorithm (Algorithm 2).

Training. We show that the time per iteration of our training algorithm is dominated by the computation of the objective function gradient (Lines 13 and 15 in Algorithm 2), which requires calculating the gradient of the projection map $\nabla\varphi$ as well as the model gradients $\nabla_\theta f$ and $\nabla_\phi g$, where θ and ϕ are the model parameters of f and g. When $\mathcal{M}$ is one of the aforementioned symmetric manifolds, $\nabla\varphi$ can be computed at each iteration within error δ in $O(n^\omega \log(\frac{1}{\delta})) = O(d^{\omega/2} \log(\frac{1}{\delta}))$ arithmetic operations in the case of the special orthogonal group $\mathbb{SO}(n)$ or unitary group $\mathbb{U}(n)$, using the singular value decomposition of an $n \times n$ matrix, or in $O(d \log(\frac{1}{\delta}))$ operations for the sphere or torus. See Section 4 and Appendix A for details.

This significantly improves the per-iteration runtime of training diffusion models on symmetric manifolds (see Table 1). Specifically, it achieves an exponential improvement over the method in De Bortoli et al. (2022), which requires summing $\Omega(2^d)$ terms to compute the heat kernel on manifolds like the torus, sphere, orthogonal group, or unitary group. Similarly, it improves on Lou et al. (2024), where heat kernel computations for manifolds like the orthogonal or unitary group involve truncated expansions with approximately $\Omega(2^d)$ terms. Additionally, De Bortoli et al. (2022) and Huang et al. (2022) propose approximations to the training objective, but these are asymptotically biased and cannot be improved beyond a fixed error, regardless of computation time (see Theorem 4 of Huang et al. (2022)). Our approach further improves the accuracy dependence from polynomial-in-$\frac{1}{\delta}$ to logarithmic-in-$\frac{1}{\delta}$, as previous methods rely on numerical solvers for SDEs or ODEs, which require polynomial-in-$\frac{1}{\delta}$ iterations for high accuracy. In contrast, our forward diffusion is computed by adding a Gaussian vector and projecting onto the manifold, achieving any desired accuracy with only a logarithmic dependence on $\frac{1}{\delta}$.

Sample generation. Our training algorithm (Algorithm 1) outputs trained models $f(x, t)$ and $g(x, t)$ for the drift and covariance terms of our reverse diffusion. We then use these models to generate samples. First, we sample a point z from the stationary distribution of the Ornstein-Uhlenbeck process Z_t on $\mathbb{R}^d$, which is Gaussian distributed. Next, we project this point z onto the manifold to obtain a point $y = \varphi(z)$, and solve the SDE $\mathrm{d}Y_t = f(Y_t, t)\mathrm{d}t + g(Y_t, t)\mathrm{d}B_t$ given by our trained model for the reverse diffusion's drift and covariance over the time interval $[0, T]$, starting at the initial point y. To simulate this SDE we can use any off-the-shelf numerical SDE solver, which takes as input the trained model for f and g, and an oracle for computing the exponential map on $\mathcal{M}$. We give one such solver in Algorithm 2, and prove guarantees for the accuracy of the samples generated by this solver, and its runtime, in Theorem 2.2. Our guarantees assume that the trained models $f(x,t)$ and $g(x,t)$ we hand to this solver minimize our training objective within some error $\varepsilon > 0$.

Our theoretical guarantees hold when $\mathcal{M}$ satisfies a symmetry property and φ satisfies an "average-case" Lipschitz condition (Assumption 2.1). This symmetry property requires that each point $z \in \mathbb{R}^d$ can be parametrized as $z \equiv z(U, \Lambda)$ where $U = \varphi(z) \in \mathcal{M}$ and $\Lambda \equiv \Lambda(z) \in \mathcal{A}$ for some $\mathcal{A} \subset \mathbb{R}^{d-\dim(\mathcal{M})}$ is another parameter. For instance, on the sphere, $U = \frac{z}{\|z\|}$ is the projection onto the sphere, and $\Lambda = \|z\|$ is the distance to the origin. For $\mathbb{SO}(n)$ or $\mathbb{U}(n)$, the parametrization comes from the spectral decomposition $z = U\Lambda U^*$, where $U \in \mathcal{M}$ and Λ is a diagonal matrix. On the torus, $U = \varphi(x)$ is the projection onto the torus, and $\Lambda \in 2\pi\mathbb{Z}^d$. Z_t, $t \geq 0$, is the Ornstein-Uhlenbeck process on $\mathbb{R}^d$, $X_t = \varphi(Z_t)$, our forward diffusion process on $\mathcal{M}$, and $Y_t = X_{T-t}$ its time-reversal (see Section 3).

Assumption 2.1 (Average-case Lipschitz-ness). *$\forall t \in [0,T]$ there exists $\Omega_t \subseteq \mathbb{R}^d$, whose indicator function $\mathbb{1}_{\Omega_t}(x)$ depends only on $\Lambda \equiv \Lambda(x)$, for which $\mathbb{P}(Z_t \in \Omega_t \ \forall\ t \in [0,T]) \geq 1-\alpha$. For every $x \in \Omega_t$ we have $\|\nabla\varphi(x)\|_{2\to2} \leq L_1$, $\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(x)\|_{2\to2} \leq L_1$, $\|\nabla^2\varphi(x)\|_{2\to2} \leq L_2$, and $\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(x)\|_{2\to2} \leq L_2$. Moreover, $\|\frac{\mathrm{d}}{\mathrm{d}U}x\|_{2\to2} \leq \|x\|_2$.*

Roughly speaking, Assumption 2.1 states that the projection map $\varphi : \mathbb{R}^d \to \mathcal{M}$ satisfies a Lipschitz condition on a set of average-case points $\Omega_t \subseteq \mathbb{R}^d$, which contains the Euclidean-space forward diffusion Z_t with high probability. Additionally, Ω_t exhibits a symmetry property: the indicator function $1_{\Omega_t}(x)$ is independent of the projection $U = \varphi(x)$. We choose projection maps φ that satisfy this Assumption 2.1 with small Lipschitz constants. For example, for $\mathbb{T}_d$, $\varphi(x)[i] = x[i] \mathrm{mod} 2\pi$, $i \in [d]$ is 1-Lipschitz on all $\mathbb{R}^d$, trivially satisfying the assumption. For the sphere, $\varphi(x) = \frac{x}{\|x\|}$ is 2-Lipschitz outside a ball of radius $\frac{1}{2}$ around the origin, where the forward diffusion remains with high probability $(1 - O(2^{-d}))$. For $\mathbb{SO}(n)$ (or $\mathbb{U}(n)$), $\varphi(X)$, which computes the spectral decomposition $U^*\Lambda U$ of $X + X^*$, has derivatives with magnitude bounded by the inverse eigenvalue gaps $\frac{1}{\lambda_i - \lambda_j}$. While singularities occur at points with duplicate eigenvalues, random matrix theory shows that eigengaps are w.h.p. bounded below by $\frac{1}{\mathrm{poly}(d)}$, ensuring φ satisfies the average-case Lipschitz assumption. For the unitary group, we show that Assumption 2.1 holds for $L_1 = O(d^{1.5}\sqrt{T}\alpha^{-\frac{1}{3}})$ and $L_2 = O(d^2 T\alpha^{-\frac{2}{3}})$ (Lemma B.4). For the sphere, it holds for $L_1 = L_2 = O(\alpha^{-\frac{1}{d}})$. For the torus it holds for $L_1 = L_2 = 1$.

Theorem 2.2 (Accuracy and runtime of sampling algorithm). *Let $\varepsilon > 0$, and suppose that $\varphi : \mathcal{M} \to \mathbb{R}^d$ satisfies Assumption 2.1 for some $L_1, L_2 \leq \mathrm{poly(d)}$ and $\alpha \leq \varepsilon$, and $\psi(\mathcal{M})$ is bounded by a ball of radius $\mathrm{poly}(d)$. Suppose that $\hat{f}$ and $\hat{g}$ are outputs of Algorithm 2, and that $\hat{f}$ and $\hat{g}$ minimize our training objective for the target distribution π with objective function value $< \varepsilon$. Then Algorithm 2, with inputs $\hat{f}$ and $\hat{g}$, outputs a generated sample whose probability distribution ν satisfies $\|\nu - \pi\|_{\mathrm{TV}} < O(\varepsilon \times (d^3 L_1 + d^2 L_2)\log\left(\frac{d}{\varepsilon}\right)) = \tilde{O}(\varepsilon \times \mathrm{poly}(d))$. Moreover, Algorithm 2, takes $O((d^4 L_1 + d^2 L_2)\log\left(\frac{d}{\varepsilon}\right)) = \mathrm{poly}(d) \times \log\left(\frac{d}{\varepsilon}\right)$ iterations, where each iteration requires one evaluation of $\hat{f}$ and $\hat{g}$, one evaluation of an oracle for computing the exponential map on $\mathcal{M}$, plus O(d) arithmetic operations.*

Plugging in our bounds on the average-case Lipschitz constants in the case of the torus, sphere, special orthogonal group, and unitary group (Lemma B.4) into Theorem 2.2, we obtain the following guarantees for the accuracy and runtime of our sampling algorithm for these symmetric manifolds:

Corollary 2.3. *Suppose that $\mathcal{M}$ is $\mathbb{T}_d$, $\mathbb{S}_d$, $\mathbb{SO}(n)$, or $\mathbb{U}(n)$ with $n = \sqrt{d}$. Suppose that φ and ψ are chosen as specified above for these manifolds. Suppose that $\hat{f}$ and $\hat{g}$ are outputs of Algorithm 2, and that $\hat{f}$ and $\hat{g}$ minimize our training objective for the target distribution π with objective function value $< \varepsilon$. Then Algorithm 2, with inputs $\hat{f}$ and $\hat{g}$, outputs a generated sample whose probability distribution ν satisfies $\|\nu - \pi\|_{\mathrm{TV}} \leq O(\varepsilon \times d^6 \log\left(\frac{d}{\varepsilon}\right))$ for the torus and sphere (or $\|\nu - \pi\|_{\mathrm{TV}} < O(\varepsilon \times d^9 \log\left(\frac{d}{\varepsilon}\right))$ for $\mathbb{SO}(n)$ and $\mathbb{U}(n)$). Moreover, Algorithm 2, takes $O(d^4 \log\left(\frac{d}{\varepsilon}\right))$ iterations for the torus and sphere (or $O(d^{5.5} \log\left(\frac{d}{\varepsilon}\right))$ iterations for $\mathbb{SO}(n)$ and $\mathbb{U}(n)$), where each iteration requires one evaluation of $\hat{f}$ and $\hat{g}$, one evaluation of an oracle for computing the exponential map on $\mathcal{M}$, plus $O(d)$ arithmetic operations.*

An overview of the proof of Theorem 2.2 is given in Section 5; the full proof appears in Appendix B. Theorem 2.2 improves on the sampling accuracy guarantees of De Bortoli et al. (2022) in the special case when $\mathcal{M}$ is one of the aforementioned symmetric manifolds, since the accuracy bound in De Bortoli et al. (2022) is not polynomial in the dimension d (their "constant" term $C \equiv C(\mathcal{M}, d)$ has an unspecified dependence on the manifold and its dimension). Finally, we note that Lou et al. (2024); Huang et al. (2022) do not provide guarantees on the accuracy or runtime of their sampling algorithm. Improving the dependency on d in Theorem 2.2 remains an open problem.

3 Deriving the training and sampling algorithms

Given a standard Brownian motion W_t in $\mathbb{R}^d$, a $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $R : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, a stochastic process X_t satisfies the SDE $\mathrm{d}X_t = \mu(X_t)\mathrm{d}t + R(X_t)\mathrm{d}W_t$ with initial condition $x \in \mathbb{R}^d$ if $X_t = x + \int_0^t \mu(X_s)\mathrm{d}s + \int_0^t R(X_s)\mathrm{d}W_s$.

Lemma 3.1 (Ito's Lemma). *Let $\psi : \mathbb{R}^d \to \mathbb{R}^k$ be a second-order differentiable function, let B_t be a Brownian motion, and let $X(t) \in \mathbb{R}^d$ be an Ito diffusion process. Then*

$$\mathrm{d}\psi(X_t)[i] = (\nabla\psi(X_t)[i])^\top \mathrm{d}X_t + \tfrac{1}{2}(\mathrm{d}X_t)^\top(\nabla^2\psi(X_t)[i])\mathrm{d}X_t \qquad \forall t \geq 0,\ \ i \in \{1,\ldots,k\}. \tag{1}$$

The *transition kernel* $p_{t|\tau}(y|x)$ is the probability (density) that X will take the value y at time t conditional on X taking the value x at time τ. Given an initial distribution π, the probability density at time t is $p_t(x) = \int_{\mathcal{M}} p_{t|0}(x|z)\pi(z)\mathrm{d}z$. For any diffusion process X_t, $t \in [0,T]$, one can define its time-reversal Y_t to be the stochastic process such that $Y_t = X_{T-t}$ for $t \in [0,T]$. Y_t is also a diffusion, and its evolution is governed by an SDE. In the special case where X_t has identity covariance, $\mathrm{d}X_t = b(X_t)\mathrm{d}t + \mathrm{d}B_t$, the reverse diffusion satisfies Anderson (1982)

$$\mathrm{d}Y_t = -b(X_t)\mathrm{d}t + \nabla\log p_t(X_t)\mathrm{d}t + \mathrm{d}B_t. \tag{2}$$

One can also define diffusions on Riemannian manifolds, in which case $\mathrm{d}B_t$ corresponds to the derivative of Brownian motion on the tangent space (see Hsu (2002)). Below we show the key steps in the derivation of our diffusion model, training algorithm (Algorithm 1), and sampling algorithm (Algorithm 2).

Forward diffusion. Let $\{Z_t\}_{t\geq 0}$ be a diffusion on $\mathbb{R}^d$ with initial distribution $q_0 = \psi(\pi)$. We choose Z_t to be the Ornstein-Uhlenbeck process, defined by the SDE $\mathrm{d}Z_t = -\frac{1}{2}Z_t\mathrm{d}t + \mathrm{d}B_t$, which has a stationary distribution $N(0, I_d)$. The process is easy to sample from and has a closed-form Gaussian transition kernel:

$$q_{t|\tau}(y|x) = \frac{1}{\sqrt{2\pi(1-e^{-(t-\tau)})}}\exp\left(-\tfrac{1}{2}\|y-xe^{-\frac{1}{2}(t-\tau)}\|^2/(1-e^{-(t-\tau)})\right) \qquad \forall x,y \in \mathbb{R}^d, t > \tau > 0. \tag{3}$$

Let $X_t := \varphi(Z_t)$, the projection of Z_t onto $\mathcal{M}$. X_t is the forward diffusion of our model.

Reverse diffusion SDE. Let $Y_t := X_{T-t}$ denote the time-reversed diffusion of X_t. Y_t is a diffusion on $\mathcal{M}$, with its distribution at time T equal to the target distribution π. The reverse diffusion follows the SDE:

$$\mathrm{d}Y_t = f^\star(Y_t,t)\mathrm{d}t + g^\star(Y_t,t)\mathrm{d}B_t, \tag{4}$$

for some functions $f^\star(x,t) : \mathcal{M}\times[0,T] \to \mathcal{T}_x\mathcal{M}$ and $g^\star(x,t) : \mathcal{M} \to \mathcal{T}_x\mathcal{M}\times\mathcal{T}_x\mathcal{M}$. Here $\mathrm{d}B_t$ is the derivative of standard Brownian motion on $\mathcal{M}$'s tangent space. We write $\mathrm{d}B_t \equiv \mathrm{d}B_t^x$ when $x \in \mathcal{M}$ is clear from context.

We cannot directly apply (2) to obtain a tractable expression for the SDE for the reverse diffusion Y_t on $\mathcal{M}$ since we do not have a closed-form expression for the transition kernel of p_t of the forward diffusion X_t on $\mathcal{M}$. Instead, we first apply (2) to obtain an SDE for the reverse diffusion of Z_t in $\mathbb{R}^d$.

$$\mathrm{d}H_t = \left(\tfrac{1}{2}H_t + 2\nabla\log q_{T-t}(H_t)\right)\mathrm{d}t + \mathrm{d}B_t \tag{5}$$

We use Ito's Lemma to project this SDE onto $\mathcal{M}$, giving an SDE for the reverse diffusion on $\mathcal{M}$ (see Appendix B.1)

$$\mathrm{d}Y_t = \mathbb{E}[\nabla\varphi(H_t)^\top\mathrm{d}H_t + \tfrac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t \big| \varphi(H_t) = Y_t]. \tag{6}$$

Objective function of training algorithm. From (6), we show one can train a model f and g for $f^\star$, $g^\star$ by solving an optimization problem (Lemma B.2). Here, $f, g \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ be continuous functions from $\mathbb{R}^d$ to $\mathbb{R}^d$ and $t \sim \mathrm{Unif}[0,1]$.

$$\begin{aligned}\min\nolimits_f \mathbb{E}_t\mathbb{E}_{b\sim\pi}[\|(\nabla\varphi(Z_{T-t}))^\top\frac{Z_{T-t}-\psi(b)e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)}-1} + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(Z_{T-t})) - f(\varphi(Z_{T-t}),t)\|^2|Z_0 = \psi(b)],\\ \min\nolimits_g \mathbb{E}_t\mathbb{E}_{b\sim\pi}[\|((\nabla\varphi(Z_{T-t}))^\top\nabla\varphi(Z_{T-t}) - (g(\varphi(Z_{T-t}),t))^2\|_F^2|Z_0 = \psi(b)].\end{aligned} \tag{7}$$

Sampling algorithm. To (approximately) sample from π, one can approximate the drift and diffusion terms of the SDE for the reverse diffusion (4) using the trained models $\hat{f}$ and $\hat{g}$ obtained by solving (7) (in practice, we model these functions with neural networks $\hat{f}_\theta$ and $\hat{g}_\phi$ where θ and ϕ are the output of Algorithm 1). We initialize this SDE at $\varphi(N(0,I_d))$, the pushforward of $N(0,I_d)$ onto $\mathcal{M}$ with respect to the map φ.

$$\mathrm{d}\hat{Y}_t = \hat{f}(\hat{Y}_t,t)\mathrm{d}t + \hat{g}(\hat{Y}_t,t)\mathrm{d}B_t, \qquad \hat{Y}_0 \sim \varphi(N(0,I_d)). \tag{8}$$

Since (unlike the forward SDE) the solution $\hat{Y}_T$ at time T is not a Gaussian or other easy-to-sample distribution, to sample from $\hat{Y}_T$ one must instead numerically simulate the SDE (8). Towards this end, one can discretize the SDE in (8) with some small time-step size $\Delta > 0$:

$$\hat{y}_{i+1} = \exp(\hat{y}_i;\ \hat{f}(\hat{y}_i,t)\Delta + \hat{g}(\hat{y}_i,t)\sqrt{\Delta}\xi_i), \qquad i \in \{0,1,\ldots,T/\Delta\}, \tag{9}$$

with initial condition $\hat{y}_0 \sim \varphi(N(0,I_d))$.

Algorithm 1: Training algorithm

Input: An oracle for the "projection" map $\varphi : \mathbb{R}^d \to \mathcal{M}$, and for its gradient.
Input: An oracle for an "inverse" map $\psi : \mathcal{M} \to \mathbb{R}^d$ such that $\varphi(\psi(x)) = x$ for all $x \in \mathcal{M}$.
Input: Dataset $D = \{x_0^1, \ldots, x_0^m\} \subseteq \mathcal{M}$.
Input: $T > 0$.
Input: Model $f_{\hat{\theta}} : \mathcal{M} \times [0,T] \to \mathcal{TM}$ where $\hat{\theta} \in \mathbb{R}^{a_1}$ denote trainable parameters.
Input: Model $g_{\hat{\phi}} : \mathcal{M} \times [0,T] \to \mathcal{TM} \times \mathcal{TM}$ where $\hat{\theta} \in \mathbb{R}^{a_2}$ denote trainable parameters.
Input: Initial parameters $\theta_0 \in \mathbb{R}^{a_1}$, $\phi_0 \in \mathbb{R}^{a_2}$.
Input: Hyperparameters: Number of stochastic gradient descent iterations $r \in \mathbb{N}$. Step size $\eta > 0$, batch size $\mathfrak{b}$.

```
Define, for all θ̂ ∈ ℝ^{a_1} ẑ ∈ ℝ^d, b, x ∈ M, t̂ ∈ [0,T], the objective function
```

$$F(\hat{\theta}; b, \hat{z}, \hat{x}, \hat{t}) := \|(\nabla\varphi(\hat{z}))^\top \frac{\hat{z} - \psi(b)e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)} - 1} + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(\hat{z})) - f(\hat{x}, \hat{t})\|^2$$

Define for all $\hat{\theta} \in \mathbb{R}^{a_2}$ $\hat{z} \in \mathbb{R}^d$, $b, x \in \mathcal{M}$, $\hat{t} \in [0,T]$, the objective function

$$G(\hat{\phi}; b, \hat{z}, \hat{x}, \hat{t}) := \|(\nabla\varphi(\hat{z}))^\top \nabla\varphi(\hat{z}) - (g_{\hat{\phi}}(\hat{x}, \hat{t}))^2\|_F^2$$

Set $\theta \leftarrow \theta_0$
Set $\phi \leftarrow \phi_0$
for $i = 1, \ldots, r$ **do**
    Sample a random batch $S \subseteq [m]$ of size $\mathfrak{b}$
    Sample $t \sim \mathrm{Unif}([0,T])$
    for $j \in S$ **do**
        Sample $\xi \sim N(0, I_d)$
        Set $z_j \leftarrow \psi(x_0^j)e^{-\frac{1}{2}(T-t)} + \sqrt{1 - e^{-(T-t)}}\,\xi$
        Set $x_j \leftarrow \varphi(z_j)$
    end
    Compute $\Gamma \leftarrow \frac{1}{\mathfrak{b}} \sum_{j \in S} \nabla_\theta F(\theta; x_0^j, z_j, x_j, t)$
    $\theta \leftarrow \theta - \eta\Gamma$
    Compute $\Upsilon \leftarrow \frac{1}{\mathfrak{b}} \sum_{j \in S} \nabla_\phi G(\phi; x_0^j, z_j, x_j, t)$
    $\phi \leftarrow \phi - \eta\Upsilon$
end
Output: Trained parameters θ, ϕ for the models f_θ and g_ϕ

Algorithm 2: Sampling algorithm

Input: An oracle which returns the value of the exponential map $\exp(x, v)$ on some manifold $\mathcal{M}$, for any $x \in \mathcal{M}$, $v \in \mathcal{T}_x\mathcal{M}$.
Input: An oracle for the "projection" map $\varphi : \mathbb{R}^d \to \mathcal{M}$.
Input: Model $f_{\hat{\theta}} : \mathcal{M} \times [0,T] \to \mathcal{TM}$ where $\hat{\theta} \in \mathbb{R}^{a_1}$ denote trainable parameters.
Input: Model $g_{\hat{\phi}} : \mathcal{M} \times [0,T] \to \mathcal{TM} \times \mathcal{TM}$ where $\hat{\theta} \in \mathbb{R}^{a_2}$ denote trainable parameters.
Input: Trained parameters θ, ϕ (from output of Algorithm 1)
Input: $T > 0$, $N \in \mathbb{N}$
Input: Discretization size $\Delta > 0$ such that $\frac{T}{\Delta} \in N\mathbb{Z}$.

Sample $z_0 \sim N(0, I_d)$
Set $\hat{y}_0 \leftarrow \varphi(z_0)$
for $i = 0, 1, \ldots, \frac{T}{\Delta} - 1$ **do**
    Sample $\xi \sim N(0, I_d)$.
    Set $\hat{y}_{i+1} \leftarrow \exp(\hat{y}_i;\ \hat{f}(\hat{y}_i, i\Delta)\Delta + \hat{g}(\hat{y}_i, i\Delta)\sqrt{\Delta}\xi_i)$
end
Output: $\hat{y}_{\frac{T}{\Delta}}$

4 ILLUSTRATION OF OUR FRAMEWORK FOR THE SPHERE

Suppose we are given a dataset $D \subseteq \mathbb{S}_{d-1}$, which was sampled from an unknown distribution π with support on $\mathbb{S}_{d-1}$. The goal is to train a generative model which generates samples from a distribution ν which is close to the target distribution π. We construct the generative model using our general framework outlined in the previous sections. We first choose a projection map $\varphi : \mathbb{R}^d \to \mathbb{S}_{d-1}$ to be $\varphi(x) = \frac{x}{\|x\|}$ for $x \in \mathbb{S}_{d-1}$, and $\psi : \mathbb{S}_{d-1} \to \mathbb{R}^d$ to be the usual embedding of the unit sphere into $\mathbb{R}^d$.

Forward diffusion. Our model adds noise to the data by running a "forward" diffusion X_t constrained to the sphere initialized at the target distribution π. We define our forward diffusion to be the projection $X_t = \varphi(Z_t)$ of the Euclidean-space Ornstein-Uhlenbeck diffusion Z_t onto the manifold $\mathcal{M}$, where Z_t is initialized at the pushforward $\psi(\pi)$ of the target distribution π onto $\mathbb{R}^d$. Since the Ornstein-Uhlenbeck distribution Z_t is a Gaussian process, each sample from our forward diffusion to be computed by drawing a single sample from a Gaussian distribution, and computing the projection map φ once.

The forward and reverse diffusion of our model on the sphere are different than those of prior diffusion models on the sphere. The evolution of our forward diffusion X_t on the sphere is goverened by the SDE $\mathrm{d}X_t = \alpha(X_t, t)(-\frac{1}{2}X_t \mathrm{d}t + \mathrm{d}B_t)$ initialized at the target distribution π, where the coefficient $\alpha(t)$ is given by the coniditonal expectation $\alpha(X_t, t) := \mathbb{E}\left[\frac{1}{\|Z_t\|} | \varphi(Z_t) = X_t\right]$. Our forward (and reverse) diffusions has a (time-varying and) spatially-varying covariance term $\alpha(X_t, t)\mathrm{d}B_t$ not present in prior models De Bortoli et al. (2022) Lou et al. (2024). This covariance term, which accounts for the curvature of the sphere, allows our forward diffusion to be computed as a projection of Euclidean Brownian motion onto the sphere despite the sphere's non-zero curvature.

Training the model. The SDE for the reverse diffusion of our model has both a drift and covariance term. To train a model f for the drift term, we first sample a point b from the dataset D at a random time $t \in [0, T]$, and point $\hat{z}$ from the Ornstein-Uhlenbeck diffusion Z_t initialized at $\psi(b)$, which is Gaussian distributed. Next, we project this sample $\hat{z}$ to obtain a sample $\varphi(\hat{z})$ from our forward diffusion X_t on the manifold. Finally, we plug in the point $\varphi(\hat{z})$, and the datapoint b into the training objective function for the drift term f, which is given by the closed-form expression $\left\| \frac{1}{\|\hat{z}\|}(I - \frac{1}{\|\hat{z}\|^2}\hat{z}\hat{z}^\top)\frac{\hat{z} - \psi(b)e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)} - 1} - f(\varphi(\hat{z}), t) \right\|^2$. The model for the drift term f is trained by minimizing the expectation of this objective function over random samples of $b \sim D$ and $\hat{z} \sim Z_t$. To learn the SDE of the reverse diffusion, we must also train a model for the spatially-varying covariance term, which is given by a $d \times d$ covariance matrix. Learning a dense matrix model for this covariance term would require at least d^2 arithmetic operations. However, as a result of the symmetries of the sphere, the covariance matrix has additional structure: it is a multiple $\alpha(X_t, t)$ of the $d \times d$ identity matrix. Thus, to learn this covariance term, it is sufficient to train a model $\hat{\alpha}(X_t, t)$ for $\alpha(X_t, t)$. This can be accomplished by minimizing the objective function $(\hat{\alpha}(\varphi(\hat{z}), t) - \frac{1}{\|\hat{z}\|})^2$. Evaluating our objective functions for the drift term and covariance terms can thus be accomplished via a single evaluation of the projection map $\varphi(x) = \frac{x}{\|x\|}$, which requires $O(d \log \frac{1}{\delta})$ arithmetic operations to compute within accuracy $\delta > 0$, when generating the input to our training objective function, which is sublinear in the dimension d^2 of the covariance term.

In contrast, the forward diffusion used in prior diffusion models on the sphere De Bortoli et al. (2022) Lou et al. (2024), cannot be computed as the projection of a Euclidean Brownian motion and must instead be computed by solving an SDE (or probability flow ODE) on the sphere. This requires a number of arithmetic operations which is a higher-order polynomial in the dimension d and in the desired accuracy $\frac{1}{\delta}$ (the order of the polynomial depends on the specific SDE or ODE solver used). As their training objective function requires samples from the forward diffusion as input, the cost of computing their objective function is therefore at least a higher-order polynomial in d and $\frac{1}{\delta}$ (for De Bortoli et al. (2022) it is exponential in d, since their training objective relies on an inneficient expansion for the heat kernel which takes 2^d arithmetic operations to compute).

Sample generation. Once the models $f(x, t)$ and $g(x, t)$ for the drift and covariance terms of our reverse diffusion are trained, we use these models to generate samples. First, we sample a point z from the stationary distribution of the Ornstein-Uhlenbeck process Z_t on $\mathbb{R}^d$, which is Gaussian distributed. Next, we project this point z onto the manifold to obtain a point $y = \varphi(z)$, and solve the SDE $\mathrm{d}Y_t = f(Y_t, t)\mathrm{d}t + g(Y_t, t)\mathrm{d}B_t$ given by our trained model for the reverse diffusion's drift and covariance over the time interval $[0, T]$, starting at the initial point y. To simulate this SDE we can use any off-the-shelf numerical SDE solver. The point y_T computed by the numerical solver at time T is the output of our sample generation algorithm.

5 Proof Outline of Theorem 2.2

In the following, for any random variable X we denote its probability distribution by $\mathcal{L}_X$. As already mentioned, previous works use Girsanov's theorem to bound the accuracy of diffusion methods. However, Girsanov transformations do not exist for our diffusion as it has a non-constant covariance term which varies with the position x. Thus, we depart from previous works and instead use an optimal transport approach based on a carefully chosen optimal coupling between the "ideal diffusion" Y_t and the algorithm's process $\hat{y}_t$ Specifically, denoting by μ_t the distribution of Y_t and by ν_t the distribution of $\hat{Y}_t$, the goal is to bound the Wasserstein optimal transport distance $W_2(\mu_t, \nu_t) := \inf_{\kappa \in \mathcal{K}(\mu_t, \nu_t)} \mathbb{E}_{(Y_t, \hat{Y}_t)}[\rho^2(\hat{Y}_t, Y_2)]$ where $\mathcal{K}(\mu, \nu)$ is the collection of all couplings of the distributions μ and ν.Towards this end, we would like to find a coupling κ which (approximately) minimizes $\mathbb{E}_{(Y_t \sim \mu_t, \hat{Y}_t \sim \nu_t)}[\rho^2(\hat{Y}_t, Y_2)]$ at any given time t.

As a first attempt, we consider the simple coupling where we couple the "ideal" reverse diffusion Y_t,

$$\mathrm{d}Y_t = f^\star(Y_t, t)\mathrm{d}t + g^\star(Y_t, t)\mathrm{d}B_t, \tag{10}$$

and the reverse diffusion $\hat{Y}_t$ given by our trained model $\hat{f}, \hat{g}$,

$$\mathrm{d}\hat{Y}_t = \hat{f}(\hat{Y}_t, t)\mathrm{d}t + \hat{g}(Y_t, t)\mathrm{d}B_t. \tag{11}$$

To couple these two diffusions, we set their Brownian motion terms $\mathrm{d}B_t$ to be equal to each other at every time t. In a similar manner, we can also couple $\hat{Y}_t$ and the discrete-time algorithm $\hat{y}_i$ by setting the Gaussian term ξ_i in the stochastic finite difference equation Equation to be equal to $\xi_i = \frac{1}{\sqrt{\Delta}} \int_{\Delta i}^{\Delta(i+1)} \mathrm{d}B_t \mathrm{d}t$ for every i (9).

Step 1: Bounding the Wasserstein distance for everywhere-Lipschitz SDEs. To bound the Wasserstein distance $W_2(Y_t, \hat{y}_t) \leq W_2(Y_t, \hat{Y}_t) + W_2(\hat{Y}_t, \hat{y}_t)$, we first prove a generalization of Gronwall's inequality to Stochastic differential equations on manifolds (Lemma B.3). Gronwall's inequality Gronwall (1919) says that if $R : [0, T] \to \mathbb{R}$ satisfies the differential inequality $\frac{\mathrm{d}}{\mathrm{d}} R(t) \leq \beta(t) R(t)$ for all $t > 0$, where the coefficient $\beta(t) : [0, T] \to \mathbb{R}$ may also be a function of t, then the solution to this differential inequality satisfies $R(t) \leq R(0) e^{\int_0^t \beta(s) \mathrm{d}s}$.

Towards this end, we first couple Y_t and $\hat{Y}_t$ by setting their Brownian motion terms $\mathrm{d}B_t$ equal to each other and then derive an SDE for the squared geodesic distance $\rho^2(\hat{Y}_t, Y_t)$ using Ito's lemma. Taking the expectation of this SDE gives and ODE for $\mathbb{E}[\rho^2(\hat{X}_t, X_t)]$,

$$\begin{aligned} \mathrm{d}\mathbb{E}[\rho^2(\hat{X}_t, X_t)] &= \mathbb{E}\left[\nabla \rho^2(\hat{X}_t, X_t)^\top \begin{pmatrix} f^\star(X_t, t) \\ \hat{f}(\hat{X}_t, t) \end{pmatrix}\right] \mathrm{d}t \\ &+ \frac{1}{2} \mathbb{E}\left[\mathrm{Tr}\left[\begin{pmatrix} g^\star(X_t, t) & 0 \\ \hat{g}(X_t, t) & 0 \end{pmatrix}^\top [\nabla^2 \rho^2(\hat{X}_t, X_t)] \begin{pmatrix} g^\star(X_t, t) & 0 \\ \hat{g}(X_t, t) & 0 \end{pmatrix}\right]\right] \mathrm{d}t. \end{aligned} \tag{12}$$

To bound each term on the r.h.s., we first observe that, roughly speaking, due to the non-negative curvature of the manifold, by the Rauch comparison theorem, each derivative on the r.h.s. is at least no larger than in the Euclidean case $\mathcal{M} = \mathbb{R}^d$. In this case $\rho^2(\hat{X}_t, X_t) = \|\hat{X}_t - X_t\|_2^2$ and hence that

$$|\nabla \rho^2(\hat{X}_t, X_t)^\top \begin{pmatrix} f^\star(X_t, t) \\ \hat{f}(\hat{X}_t, t) \end{pmatrix}| \leq 2\|\hat{X}_t - X_t\| \times \|f^\star(X_t, t) - \hat{f}(\hat{X}_t, t)\| \leq 2\|\hat{X}_t - X_t\| \times (c\|\hat{X}_t - X_t\| + \varepsilon),$$

as long as we can show that $f^\star$ is c- Lipschitz for some $c > 0$ (see Step 2 below). Bounding the covariance term in a similar manner, and applying Gronwall's lemma to the differential inequality, we get that

$$W_2(\hat{Y}_t, Y_t) \leq \mathbb{E}[\rho^2(\hat{Y}_t, Y_t)] \leq (\rho^2(\hat{Y}_0, Y_0) + \varepsilon) e^{ct}. \tag{13}$$

Step 2: Showing that our diffusion satisfies an "average-case" Lipschitz condition. To apply (13), we must first show that the drift and diffusion terms $f^\star$ and $g^\star$ are Lipschitz on $\mathcal{M}$. Towards this end, we would ideally like to apply bounds on the derivatives of $\varphi : \mathbb{R}^d \to \mathcal{M}$ which defines our diffusion Y_t. Unfortunately, in general, φ may not be differentiable at every point. This is the case for the sphere, where the map $\varphi(z) = \frac{z}{\|z\|}$ has a singularity at $z = 0$. This issue also arises in the case of the unitary group and orthogonal group, since the derivative of the spectral decomposition $\varphi(z) = U^* \Lambda U$ has singularities at any matrix z which has an eigenvalue gap $\lambda_i - \lambda_{i+1} = 0$.

To tackle this challenge, we show that, for the aforementioned symmetric manifolds, the forward diffusion Z_t in $\mathbb{R}^d$ remains in some set $\Omega_t \subseteq \mathbb{R}^d$ with high probability $1-\alpha$, on which the map $\varphi(Z_t)$ has derivatives bounded by $\mathrm{poly}(d)$ (Assumption 2.1 and Lemma B.4). We then show how to "remove" the rare outcomes of our diffusion that do not fall inside Ω_t. As our forward diffusion X_t (and thus the reverse diffusion $Y_t = X_{T-t}$) remains at every t inside Ω_t with probability $\geq 1-\alpha$, removing these "bad" outcomes only adds a cost of α to the total variation error.

Showing that φ has $\mathrm{poly}(d)$ derivatives w.h.p. (showing that Assumption 2.1 holds). We first consider the sphere, which is the simplest case (aside from the trivial case of the torus, where the derivatives of φ are all $O(1)$ at every point). In the case when data is on the sphere, which we embed as a unit sphere in $\mathbb{R}^d$, one can easily observe that e.g. $\|\nabla\varphi(z)\| \leq O(1)$ for any z outside a ball of radius $r \geq \Omega(1)$ centered at the origin. As the volume of a ball of radius $r = \alpha$ is $\frac{1}{r^d}$, one can use standard Gaussian concentration inequalities to show that the Brownian motion X_t will remain outside this ball for time T with probability roughly $1 - O(\frac{1}{r^d T})$.

We next show that the Lipschitz property holds for the unitary group $\mathbb{U}(n)$. Similar techniques can be used for the case of the special orthogonal group, and we omit those details. We first recall results from random matrix theory which allow us to bound the eigenvalue caps of a matrix with Gaussian entries. Specifically, these results say that roughly speaking, if X_0 is any matrix and $X_t = X_0 + B(t)$, where $B(t)$ is a symmetric matrix with iid $N(0,t)$ entries undergoing Brownian motion, one has that the eigenvalues $\gamma_1(t) \geq \cdots \geq \gamma_n(t)$ of X_t satisfy (see e.g. Anderson et al. (2010); Mangoubi and Vishnoi (2023))

$$\mathbb{P}(\inf_{s\in[t_0,T]}(\gamma_{i+1}(t)-\gamma_i(t)) \leq s\frac{1}{\mathrm{poly}(n)\sqrt{t}}) \leq O(s^{\frac{1}{2}}) \qquad \forall s \geq 0. \tag{14}$$

Thus, if we define Ω_t to be the set of outcomes of such that $\gamma_{i+1}(t)-\gamma_i(t) \leq \alpha^2\frac{1}{\mathrm{poly}(n)\sqrt{t}}$, we have that $\mathbb{P}(X_t \in \Omega_t \;\; \forall t \in [t_0,T]) \geq 1-\alpha$.

Our high-probability bound on Ω_t allows us to show that φ satisfies a Lipschitz property at "most" points Ω_t. However, if we wish to apply (13), we need to show that drift term $f^\star$ and covariance term $g^\star$ in our diffusion satisfy a Lipschitz property at *every* point in $\mathbb{R}^d$. Towards this end, we first make a small modification to the objective function which allows us to exclude outcomes $\{X_t\}_{t\in[0,T]}$ of the forward diffusion such that $X_t \notin \Omega_t$ for some $t \in [0,T]$. Specifically, we multiply the objective function (7) by the indicator function $\mathbb{1}_{\Omega_t}(z)$. As determining whether a point $z \in \Omega_t$ requires only checking the eigenvalue gaps (when $\mathcal{M}$ is the unitary or orthogonal group), computing $\mathbb{1}_{\Omega_t}(z)$ can be done efficiently using the singular value decomposition.

Bounding the Lipschitz constant of $f^\star$ and $g^\star$. Recall that (when, e.g., $\mathcal{M}$ is one of the aforementioned symmetric manifolds) we may decompose any $z \in \mathbb{R}^d$ as $z \equiv z(U,\Lambda)$ where $U \in \mathcal{M}$. Note that $\mathbb{1}_{\Omega_t}(z)$ is *not* a continuous function of z. However, we will show that, as $\mathbb{1}_{\Omega_t}(z(U,\Lambda))$ depends only on Λ, multiplying our objective function by $\mathbb{1}_{\Omega_t}$ does not make $f^\star$ and $g^\star$ discontinuous (and thus does not prevent them from being Lipschitz). This is because $f^\star$ and $g^\star$ are given by conditional expectations conditioned on U, and can thus be decomposed as integrals over Λ. Towards this end we express $f^\star$ as an integral over the parameter Λ,

$$f^\star(U,t) =$$
$$c_U \int_{\Lambda\in\mathcal{A}} \left[\nabla\varphi(z(U,\Lambda))^\top \nabla \log q_{T-t|0}(z(U,\Lambda)) + \tfrac{1}{2}\mathrm{tr}\nabla^2\varphi(z(U,\Lambda))\right] q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda,$$

where $c_U = \left(\int_{\Lambda\in\mathcal{A}} q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda\right)^{-1}$ is a normalizing constant. Differentiating w.r.t. U,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}U} f^\star(U,t) =& \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}[\frac{\mathrm{d}}{\mathrm{d}U}((\nabla\varphi(z(U,\Lambda)))^\top \nabla_U \log q_{T-t|0}(z(U,\Lambda)) \\ &+ \tfrac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda))))\mathbb{1}_\Omega(\Lambda)|V=U] + \cdots,\end{aligned} \tag{15}$$

where "$\cdots$" includes three other similar terms that we omit due to space constraints. To bound the terms on the r.h.s. of (15), we apply Assumption 2.1 which says that the operator norms of $\nabla\varphi$, $\nabla^2\varphi$, $\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi$ and $\frac{\mathrm{d}}{\mathrm{d}U}\nabla^2\varphi$ are all bounded above by $\mathrm{poly}(d)$ whenever $z \in \Omega_t$. To bound the term $\nabla_U \log q_{T-t|0}(z(U,\Lambda))$ we note that $\nabla \log q_{T-t|0}(z(U,\Lambda))$ is the drift term of the reverse diffusion in Euclidean space. This term was previously shown to be dC^2-Lipschitz for all $t \geq \Omega(\frac{1}{d})$ when the support of the data distribution in $\mathbb{R}^d$ lies in a ball of radius C (see, e.g., Proposition 20 of Chen et al. (2023b)). Thus, plugging in the above bounds into (15) we have that $\|\frac{\mathrm{d}}{\mathrm{d}U} f^\star(U,t)\|_{2\to 2} \leq \mathrm{poly}(d)$. A similar calculation shows that $\|\frac{\mathrm{d}}{\mathrm{d}U} g^\star(U,t)\|_{2\to 2} \leq \mathrm{poly}(d)$. This immediately implies that $f^\star(U,t)$ and $g^\star(U,t)$ are $\mathrm{poly}(d)$-Lipschitz at *every* $U \in \mathcal{M}$.

Step 3: Improving the coupling to obtain polynomial-time bounds. Now that we have shown that $f^\star$ and $g^\star$ are $\mathrm{poly}(d)$-Lipschitz, we can apply (13) to bound the Wasserstein distance: $W_2(\hat{Y}_{t+\tau}, Y_{t+\tau}) \leq (\rho^2(\hat{Y}_t, Y_t) + \varepsilon)e^{c\tau} \quad \forall \tau \geq 0$, where $c \leq \mathrm{poly}(d)$.

Moreover, with slight abuse of notation, we may define $\hat{y}_{t+\tau}$ to be a continuous-time interpolation of the discrete process $\hat{y}$. Applying (13) to this process we get that, roughly, $W_2(\hat{Y}_{t+\tau}, \hat{y}_{t+\tau}) \leq (\rho^2(\hat{y}_t, Y_t) + \varepsilon + \Delta)e^{c\tau}$ for $\tau \geq 0$. Thus, we get a bound on the Wasserstein error,

$$W_2(Y_{t+\tau}, \hat{y}_{t+\tau}) \leq W_2(\hat{Y}_{t+\tau}, Y_{t+\tau}) + W_2(\hat{Y}_{t+\tau}, \hat{y}_{t+\tau}) \leq (\rho^2(\hat{y}_t, Y_t) + \varepsilon + \Delta)e^{c\tau} \qquad \tau \geq 0. \tag{16}$$

Unfortunately, after times $\tau > \frac{1}{c} = \frac{1}{\mathrm{poly(d)}}$, this bound grows exponentially with the dimension d.

To overcome this challenge, we define a new coupling between Y_t and $\hat{Y}_t$ which we "reset" after time intervals of length $\tau = \frac{1}{c}$ by converting our Wasserstein bound into a total variation bound after each time interval. Towards this end, we use the fact that if at any time t the total variation distance satisfies $\|\mathcal{L}_{Y_t} - \mathcal{L}_{\hat{y}_t}\|_{\mathrm{TV}} \leq \alpha$, then there exists a coupling such that $Y_t = \hat{Y}_t$ with probability at least $1 - \alpha$. In other words, w.p. $\geq 1 - \alpha$, we have $\rho(\hat{y}_{t+\tau}, Y_{t+\tau}) = 0$, and we can apply inequality (16) over the next time interval of τ without incurring an exponential growth in time. Repeating this process $\frac{T}{\tau}$ times, we get that $\|\mathcal{L}_{Y_T} - \mathcal{L}_{\hat{y}_T}\| \leq \alpha \times \frac{T}{\tau}$, where the TV error grows only *linearly* with T.

Converting Wasserstein bounds on the manifold to TV bounds. To complete the proof, we still need to show how to convert the Wasserstein bound into a TV bound (Lemma B.7). Towards this end, we begin by showing that the transition kernel $\tilde{p}_{t+\tau+\hat{\Delta}|t+\tau}(\cdot|H_{t+\tau})$ of the reverse diffusion H_t in $\mathbb{R}^d$ is close to a Gaussian in KL distance: $D_{\mathrm{KL}}(N(H_{t+\tau} + \hat{\Delta}\nabla\tilde{p}_{T-t-\tau}(H_{t+\tau}), \hat{\Delta}I_d) \,\|\, \tilde{p}_{t+\tau+\hat{\Delta}|t+\tau}(\cdot|H_{t+\tau})) \leq \frac{\alpha\tau}{T}$. One can do this via Girsanov's theorem, since, unlike the diffusion Y_t on the manifold, the reverse diffusion in Euclidean space H_t *does* have a constant diffusion term (see e.g. Theorem 9 of Chen et al. (2023b)).

Next, we use the fact that with probability at least $1-\alpha\frac{\tau}{T}$ the map φ in a ball of radius $\frac{1}{\mathrm{poly}(d)}$ about the point $H_{t+\tau}$ has c-Lipschitz Jacobian where $c = \mathrm{poly}(d)$, and that the inverse of the exponential map $\exp(\cdot)$ has $O(1)$-Lipschitz Jacobian, to show that the transition kernel p_t of $Y_t = \varphi(H_t)$ satisfies $D_{\mathrm{KL}}(\nu_1 \,\|\, p_{t+\tau+\hat{\Delta}|t+\tau}(\cdot|Y_{t+\tau})) \leq (1 + \hat{\Delta}c)^d \frac{\alpha\tau}{T} \leq 2\frac{\alpha\tau}{T}$ if we choose $\hat{\Delta} \leq O(\frac{1}{cd})$, where $\nu_1 := \exp_{Y_{t+\tau}}(N(Y_{t+\tau} + \hat{\Delta}f^\star(Y_{t+\tau}, t+\tau),\ \hat{\Delta}g^{\star 2}(Y_{t+\tau}, t+\tau)I_d))$.

Next, we plug in our Wasserstein bound $W(Y_{t+\tau}, \hat{y}_{t+\tau}) \leq O(\varepsilon)$ into the formula for the KL divergence between two Gaussians to bound $\|\mathcal{L}_{Y_{t+\tau+\hat{\Delta}}} - \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}\|_{\mathrm{TV}}$. Specifically, noting that $\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_t} = \exp_{\hat{y}_{t+\tau}}(N(\hat{y}_{t+\tau} + \hat{\Delta}f(\hat{y}_{t+\tau}, t+\tau), \hat{\Delta}g^2(\hat{y}_{t+\tau}, t+\tau)I_d))$, we have that

$$\begin{aligned} D_{\mathrm{KL}}(\nu_1, \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_{t+\tau}}) = {} & \big(\mathrm{Tr}(g^{\star 2}(Y_{t+\tau}, t+\tau))^{-1} g^2(\hat{y}_{t+\tau}, t+\tau)\big) \\ & - d + \log \tfrac{\det g^{\star 2}(Y_{t+\tau}, t+\tau)}{\det g^2(\hat{y}_{t+\tau}, t+\tau)} + w^\top(\hat{\Delta}g^{\star 2}(Y_{t+\tau}, t))^{-1} w, \end{aligned}$$

where $w := Y_{t+\tau} - \hat{y}_{t+\tau} + \hat{\Delta}(f^\star(Y_{t+\tau}, t+\tau) - f(\hat{y}_{t+\tau}, t+\tau))$. Since with probability $\geq 1 - \alpha\frac{\tau}{T}$ we have $g^\star(Y_{t+\tau}) \succeq \mathrm{poly}(d)$, plugging in the error bounds $\|f^\star(Y_{t+\tau}, t) - f(Y_{t+\tau}, t)\| \leq \varepsilon$ and $\|g^\star(Y_{t+\tau}, t) - g(Y_{t+\tau}, t)\|_F \leq \varepsilon$ and the c-Lipschitz bounds on $f^\star$ and $g^\star$, where $c = \mathrm{poly}(d)$, (Assumption 2.1), we get that $D_{\mathrm{KL}}(\nu_1, \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}) \leq O(\varepsilon^2 c^2)$. Thus, by Pinsker's inequality, we have

$$\begin{aligned} & \|\mathcal{L}_{Y_{t+\tau+\hat{\Delta}}} - \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}\|_{\mathrm{TV}} - \|\mathcal{L}_{Y_t} - \mathcal{L}_{\hat{y}_t}\|_{\mathrm{TV}} \\ & \leq \sqrt{D_{\mathrm{KL}}(\nu_1 \,\|\, p_{t+\tau+\hat{\Delta}|t+\tau}(\cdot|Y_{t+\tau}))} + \sqrt{D_{\mathrm{KL}}(\nu_1 \| \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_t})} \leq O(\varepsilon c). \end{aligned} \tag{17}$$

Step 4: Bounding the accuracy. Recall that q_t is the distribution of the forward diffusion Z_t in Euclidean space after time t, which is an Ornstein-Uhlenbeck process. Standard mixing bounds for Ornstein-Uhlenbeck process imply that, $\|q_t - N(0, I_d)\|_{TV} \leq O(Ce^{-t})$ for all $t > 0$ (see e.g. Bakry et al. (2014)), where $C \leq \mathrm{poly}(d)$ is the diameter of the support of $\psi(\pi)$. Thus, it is sufficient to choose $T = \log(\frac{C}{\varepsilon})$ to ensure $\|\mathcal{L}_{Y_T} - \pi\|_{\mathrm{TV}} = \|q_T - N(0, I_d)\|_{\mathrm{TV}} \leq O(\varepsilon)$.

As (17) holds for all $t \in \tau\mathbb{N}$, the distribution $\nu = \mathcal{L}_{\hat{y}_T}$ of our sampling algorithm's output satisfies, since $\tau = \frac{1}{c}$,

$$\|\pi - \nu\|_{\mathrm{TV}} = \|\mathcal{L}_{Y_T} - \pi\|_{\mathrm{TV}} + \|\mathcal{L}_{Y_T} - \nu\|_{\mathrm{TV}} \leq O(\varepsilon + \varepsilon c\tfrac{T}{\tau}) = O(\varepsilon c^2 \log(\tfrac{dC}{\varepsilon})) = \tilde{O}(\varepsilon \times \mathrm{poly}(d)).$$

Step 5: Bounding the runtime. Since our accuracy bound requires $T = \log(\frac{dC}{\varepsilon})$, and requires a time-step size of $\Delta = cd \leq \frac{1}{\mathrm{poly}(d)}$, the number of iterations is bounded by $\frac{T}{\Delta} = cdT \leq O\left(\mathrm{poly}(d) \times \log\left(\frac{dC}{\varepsilon}\right)\right)$.

REFERENCES

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

Greg W Anderson, Alice Guionnet, and Ofer Zeitouni. *An introduction to random matrices*. Number 118. Cambridge university press, 2010.

Dominique Bakry, Ivan Gentil, Michel Ledoux, et al. *Analysis and geometry of Markov diffusion operators*, volume 103. Springer, 2014.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763. PMLR, 2023a.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023b. URL `https://openreview.net/forum?id=zyLVMgsZ0U_`.

Xiang Cheng, Jingzhao Zhang, and Suvrit Sra. Theory and algorithms for diffusion processes on riemannian manifolds. *arXiv preprint arXiv:2204.13665*, 2022.

Yu Cheng, Yongshun Gong, Yuansheng Liu, Bosheng Song, and Quan Zou. Molecular design in drug discovery: a comprehensive review of deep generative models. *Briefings in bioinformatics*, 22(6):bbab344, 2021.

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.

Wendelin Feiten, Muriel Lang, and Sandra Hirche. Rigid motion estimation using mixtures of projected gaussians. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1465–1472. IEEE, 2013.

Thomas Hakon Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, pages 292–296, 1919.

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

Elton P Hsu. *Stochastic analysis on manifolds*. Number 38. American Mathematical Soc., 2002.

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.

Adam Leach, Sebastian M Schmon, Matteo T Degiacomi, and Chris G Willcocks. Denoising diffusion probabilistic models on so (3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

Aaron Lou, Minkai Xu, Adam Farris, and Stefano Ermon. Scaling riemannian diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Debanjana Maji, Alan Grossfield, and Clara L Kielkopf. Structures of sf3b1 reveal a dynamic achilles heel of spliceosome assembly: Implications for cancer-associated abnormalities and drug discovery. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1862(11-12):194440, 2019.

Oren Mangoubi and Nisheeth K Vishnoi. Private covariance approximation and eigenvalue-gap bounds for complex gaussian perturbations. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1522–1587. PMLR, 2023.

OpenAI. Video generation models as world simulators, 2023. URL `https://openai.com/index/video-generation-models-as-world-simulators/`.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 8(82):1–9, 2013.

Jon M Selig. *Geometrical methods in robotics*. Springer Science & Business Media, 2013.

Maxim V Shapovalov and Roland L Dunbrack. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6):844–858, 2011.

Di Shi, Long Li, Yixin Shao, Wuxiang Zhang, and Xilun Ding. Multi-mode control strategy for robotic rehabilitation on special orthogonal group so (3). *IEEE Transactions on Industrial Electronics*, 2023.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pages 40001–40039. PMLR, 2023.

Yuchen Zhu, Tianrong Chen, Lingkai Kong, Evangelos A Theodorou, and Molei Tao. Trivialized momentum facilitates diffusion generative modeling on lie groups. *arXiv preprint arXiv:2405.16381*, 2024.

CONTENTS

A Illustration of our framework for the Euclidean space, torus, special orthogonal group, and unitary group

1. **Euclidean space $\mathbb{R}^d$.** In Euclidean case, our algorithm (with the above choice of φ, ψ) recovers the algorithms of diffusion models on $\mathbb{R}^d$ from prior works . The forward diffusion is the Ornstein-Uhlenbeck process with SDE $\mathrm{d}Z_t = -\frac{1}{2}Z_t\mathrm{d}t + \mathrm{d}B_t$ initialized at the target distribution π, where B_t is the standard Brownian motion.

   The training objective for the drift term $f(z,t)$ of the reverse diffusion is given by $\|(\hat{z}^\top \frac{\hat{z}-be^{-\frac{1}{2}(T-t)}}{e^{-(T-t)}-1} - f(\hat{z},t)\|^2$ where b is a point sampled from the dataset and $\hat{z}$ is a point sampled from $Z_{T-t}|\{Z_0 = b\}$ which is Gaussian distributed as $N(be^{-\frac{1}{2}(T-t)}, \sqrt{1-e^{-(T-t)}}I_d)$ (see Section 3). The number of arithmetic operations to compute the training objective is therefore the same as for previous diffusion models in Euclidean space.

2. **Torus $\mathbb{T}_d$.** For the torus, the forward and reverse diffusion of our model are the same as the models used in previous diffusion models on the torus De Bortoli et al. (2022) Lou et al. (2024). The Forward diffusion is given by the SDE $\mathrm{d}X_t = -\frac{1}{2}X_t\mathrm{d}t + \mathrm{d}B_t$ on the torus, initialized at the target distribution π.

   The only difference is in the training objective function. To obtain our objective function we observe that X_t is the projection of the $X_t = \varphi(Z_t)$ of the Ornstein-Uhlenbeck diffusion on $\mathbb{R}^d$ via our choice of projection map φ for the torus. The drift term f for the reverse diffusion can be trained by minimizing the objective function $\|\hat{z}^\top \frac{\hat{z}-\psi(b)e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)}-1} - f(\varphi(\hat{z}),t)\|^2$, where $\hat{z} \sim N(be^{-\frac{1}{2}(T-t)}, \sqrt{1-e^{-(T-t)}}I_d)$. Our objective function can be computed in $O(d)$ arithmetic operations, improving by an exponential factor on the per-iteration training runtime of De Bortoli et al. (2022) which relies on an inefficient expansion of the heat kernel which requires and exponential-in-d number of arithmetic operations to compute, and matching the per-iteration training runtime of Lou et al. (2024) who derive a more efficient expansion for the heat kernel in the special case of the torus.

3. **Special Orthogonal group $\mathbb{SO}(n)$ and Unitary group $\mathbb{U}(n)$.**

   For the Special Orthogonal group $\mathbb{SO}(n)$ and Unitary group $\mathbb{U}(n)$, the forward and reverse diffusion of our model are also different from those of previous works, as our model's diffusions have a spatially-varying covariance term to account for the non-zero curvature of these manifolds. As a result of this covariance term, our forward diffusion can be computed as a projection φ of the Ornstein-Uhlenbeck process in $\mathbb{R}^d \equiv \mathbb{R}^{n\times n}$ (or $\mathbb{C}^{n\times n}$) onto the manifold $\mathbb{SO}(n)$ ($\mathbb{U}(n)$). This projection can be computed via a single evaluation of the singular value decomposition of a $n \times n$ matrix, which requires at most $O(n^\omega) = O(d^{\frac{\omega}{2}})$ arithmetic operations, where $\omega \approx 2.37$ is the matrix multiplication exponent and $d = n^2$ is the manifold dimension.

   The forward diffusion $U(t) \in \mathbb{SO}(n)$ (or $U(t) \in \mathbb{U}(n)$) of our model is given by the system of stochastic differential equations

$$\mathrm{d}u_i(t) = \sum_{j\in[n], j\neq i} \alpha_{ij}(t)\mathrm{d}B_{ij}u_j(t) - \frac{1}{2}\sum_{j\in[d], j\neq i}\beta_{ij}(t)u_i(t)\mathrm{d}t, \tag{18}$$

   where $\alpha_{ij}(t) := \mathbb{E}\left[\frac{1}{\lambda_i-\lambda_j}|\varphi(Z_t) = U(t)\right]$ and $\beta_{ij}(t) := \mathbb{E}\left[\frac{1}{(\lambda_i-\lambda_j)^2}|\varphi(Z_t) = U(t)\right]$ for every $i, j \in [d]$.

   A model for the drift term f for the reverse diffusion can be trained by minimizing the objective function $\|R - \frac{1}{2}DU - f(\varphi(\hat{z}),t)\|_F^2$ where R is the matrix with i'th column $R_i = \frac{e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)}-1}U(\lambda_i I - \Lambda)^+ U^*\psi(b)u_i$ for each $i \in [n]$, and D is the diagonal matrix with i'th diagonal entry $D_{ii} = \sum_{j\in[n], j\neq i}\frac{1}{\lambda_i-\lambda_j}$ for each $i \in [n]$, where $\hat{z} = be^{-\frac{1}{2}(T-t)} + \sqrt{1-e^{-(T-t)}}G$ where G is a Gaussian random matrix with iid $N(0,1)$ entries and and $U\Lambda U^*$ denotes the spectral decomposition of $\hat{z} + \hat{z}^*$.

   To learn the SDE of the reverse diffusion, we must also train a model for the covariance term, which is given by a $d \times d = n^2 \times n^2$ covariance matrix. To train a model for this covariance term with runtime sublinear in the number of matrix entries n^4, we observe that as a result of the symmetries of the orthogonal (or unitary) group, the covariance term in (18) is fully determined by the n^2 scalar terms $\alpha_{ij}(t)$ for $i, j \in [n]$ and the $n \times n$ matrix U. Thus, to learn the covariance term, it is sufficient to train a model $\mathcal{A}(U,t) \in \mathbb{R}^{n\times n}$ for these n^2 terms, which can be done by minimizing the objective function $\|\mathcal{A}(U,t) - A\|_F^2$, where A is the $n \times n$ matrix with (i,j)'th entry $A_{ij} = \frac{1}{\lambda_i-\lambda_j}$ for $i, j \in [n]$, and λ_i denotes the i'th diagonal entry of Λ.

   The training objective function for both the drift and covariance term can thus be computed via a singular value decomposition of an $n \times n$ matrix (and matrix multiplications of $n \times n$ matrices), which requires at most

$O(n^\omega) = O(d^{\frac{\omega}{2}})$ arithmetic operations, where $\omega \approx 2.37$ is the matrix multiplication exponent and $d = n^2$ is the manifold dimension.

In contrast, the training objectives in prior works including De Bortoli et al. (2022) Lou et al. (2024) require an exponential in dimension number of arithmetic operations to compute as they rely on the heat kernel of the manifold, which lacks an efficient closed-form expression. Instead, their training algorithm requires computing an expansion for the heat kernel of these manifolds which is given as a sum of terms over the d-dimensional lattice, and one requires computing roughly 2^d of these terms to compute the heat kernel within an accuracy of $O(1)$.

B Proof of Theorem 2.2

In the following, we denote by $\rho(x, y)$ the geodesic distance between $x, y \in \mathcal{M}$, and by $\Gamma_{x\to y}(v)$ the parallel transport of a vector $v \in \mathcal{T}_x$ from x to y.

For convenience, we denote $\varphi_i(\cdot) := \varphi(\cdot)[i]$.

Recall that we have assumed that $\psi(\mathcal{M})$ is contained in a ball of radius $C = \mathrm{poly}(d)$. We will prove our results under the more general assumption (Assumption B.1(ψ, π, C)), which is satisfied whenever $\psi(\mathcal{M}) \leq C$.

Assumption B.1 (Bounded Support (ψ, π, C)). *The pushforward of $\psi(\pi)$ of π with respect to the map $\psi : \mathcal{M} \to \mathbb{R}^d$ has support on a ball of radius C centered at 0.*

B.1 Correctness of the training objective functions

Lemma B.2. *$f^\star$ and $g^\star$ are solutions to the following optimization problems:*

$$\min_{f\in\mathcal{C}(\mathbb{R}^d,\mathbb{R}^d)} \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\left[\left\|(\nabla\varphi(Z_{T-t}))^\top \frac{Z_{T-t} - \psi(b)e^{-\frac{1}{2}(T-t)}}{e^{-(T-t)} - 1} + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(Z_{T-t})) - f(\varphi(Z_{T-t}), t)\right\|^2 \middle| Z_0 = \psi(b)\right], \tag{19}$$

$$\min_{g\in\mathcal{C}(\mathbb{R}^d,\mathbb{R}^{d\times d})} \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\left[\left\|((\nabla\varphi(Z_{T-t}))^\top\nabla\varphi(Z_{T-t}) - (g(\varphi(Z_{T-t}), t))^2\right\|_F^2 \middle| Z_0 = \psi(b)\right].$$

Proof. **Step 1: Obtaining an expression for the reverse diffusion SDE in $\mathbb{R}^d$:**

We cannot in general directly apply (2) to obtain a tractable expression for the SDE for the reverse diffusion Y_t in $\mathcal{M}$, since we do not have a tractable formula for the transition kernel of p_t of the forward diffusion X_t on $\mathcal{M}$. Instead, we will first obtain an SDE for the reverse diffusion of Z_t in $\mathbb{R}^d$, and then "project" this SDE onto $\mathcal{M}$. Let $H_t := Z_{T-t}$ denote the time-reversed diffusion of Z_t. H_t is a diffusion in $\mathbb{R}^d$. From (2), we have that the SDE for the reverse diffusion H_t on $\mathbb{R}^d$ is given by the following formula:

$$\mathrm{d}H_t = \left(\frac{1}{2}H_t + 2\nabla\log q_{T-t}(H_t)\right)\mathrm{d}t + \mathrm{d}W_t \tag{20}$$

Equation (20) can be re-written as

$$\mathrm{d}H_t = \left(\frac{1}{2}H_t + 2\mathbb{E}_{b\sim q_{0|t}(\cdot|H_t)}[\nabla\log q_{T-t|0}(H_t|b)]\right)\mathrm{d}t + \mathrm{d}W_t \tag{21}$$

The r.h.s. of (21) is tractable since we have a tractable expression for the transition kernel $q_{T-t|0}$ (it is just a time re-scaling of the Gaussian Kernel, the trasition kernel of Brownian motion).

Step 2: Obtaining an expression for the reverse diffusion SDE in $\mathcal{M}$:

Note that there exists a coupling between Z_t and H_t such that $H_t = Z_{T-t}$ and that $Y_t = X_{T-t}$ for all $t \in [0, T]$. Thus, under this choice of coupling, we have that $Y_t = X_{T-t} = \varphi(Z_{T-t}) = \varphi(H_t)$ for all $t \in [0, T]$. In the special case when there is only one datapoint x_0, the SDE for the reverse diffusion Y_t on $\mathcal{M}$ can be obtained by applying Ito's lemma (Lemma 3.1) to $Y_t = \varphi(H_t)$:

$$\mathrm{d}Y_t[i] = \nabla\varphi_i(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi_i(H_t))\mathrm{d}H_t \qquad \forall i \in [d]. \tag{22}$$

In the following, to simplify notation, we drop the "i" index from the notation φ_i and $\mathrm{d}Y_t[i]$. Unfortunately, the r.h.s. of (22) is not a (deterministic) function of $Y_t = \varphi(H_t)$, since φ is not an invertible map. To solve this problem, we can take the conditional expectation of (22) with respect to $Y_t = \varphi(H_t)$:

$$\mathrm{d}Y_t = E[\mathrm{d}Y_t|Y_t] = E[\mathrm{d}Y_t|\varphi(H_t)] = E[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t|\varphi(H_t)]. \tag{23}$$

The drift term on the r.h.s. of (23) is a deterministic function of Y_t. Denote this function by $f^\star : \mathcal{M} \times [0, T] \to \mathfrak{T}\mathcal{M}$ for any input $x \in \mathcal{M}$ and output in the tangent space $\mathfrak{T}_x\mathcal{M}$ at of $\mathcal{M}$ at x.

Moreover, by (2), the diffusion term on the r.h.s. of (23) must be the same as the diffusion term for the forward diffusion Y_t on $\mathcal{M}$. This diffusion term can be obtained from the diffusion term $\mathrm{d}W_t$ on $\mathbb{R}^d$, via Ito's lemma, which implies that the diffusion term is $\mathbb{E}[\nabla\varphi(H_t)^\top \mathrm{d}W_t|\varphi(H_t)]$. The diffusion term is also a deterministic function $g^\star$ of Y_t, where $g^\star(Y_t)$ is a symmetric $k \times k$ matrix,

$$E[\nabla\varphi(H_t)^\top \mathrm{d}W_t|\varphi(H_t)] = g^\star(Y_t, t)\mathrm{d}\tilde{W}_t, \tag{24}$$

where $\tilde{W}_t$ is a standard Brownian motion on $\mathcal{M}$.

Since $\mathrm{d}W_t$ is the derivative of a standard Brownian motion in $\mathbb{R}^d$, and $\mathrm{d}\tilde{W}_t$ is the derivative of a standard Brownian motion on the tangent space of $\mathcal{M}$, we have that

$$E[(\nabla\varphi(H_t))^\top \nabla\varphi(H_t)|\varphi(H_t)] = (g^\star(Y_t, t))^2. \tag{25}$$

Thus, (23) can be expressed as:

$$\mathrm{d}Y_t = E[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t|\varphi(H_t)] = f^\star(Y_t, t)\mathrm{d}t + g^\star(Y_t, t)\mathrm{d}\tilde{W}_t. \tag{26}$$

In the more general setting when there is more than one datapoint, (26) generalizes to:

$$\mathrm{d}Y_t = \mathbb{E}_{b\sim\pi}E[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t|\varphi(H_t), H_T = b]] \tag{27}$$

$$= f^\star(Y_t, t)\mathrm{d}t + g^\star(Y_t, t)\mathrm{d}\tilde{W}_t. \tag{28}$$

Since $Y_t = \varphi(H_t)$, we can bring $f^\star(Y_t, t)\mathrm{d}t$ and $g^\star(Y_t, t)\mathrm{d}\tilde{W}_t$ inside the conditional expectation:

$$\mathbb{E}_{b\sim\pi}E[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t - f^\star(Y_t, t)\mathrm{d}t|\varphi(H_t), H_T = b]] = g^\star(Y_t, t)\mathrm{d}\tilde{W}_t.$$

We can re-write this as

$$\mathbb{E}_{b\sim\pi}E_{\varphi(H_t)}[E_{H_t|\varphi(H_t)}[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t - f^\star(Y_t, t)\mathrm{d}t|H_t, H_T = b]]]$$
$$= g^\star(Y_t, t)\mathrm{d}\tilde{W}_t.$$

This simplifies to

$$\mathbb{E}_{b\sim\pi}\left[\nabla\varphi(H_t)^\top \mathrm{d}H_t + \frac{1}{2}(\mathrm{d}H_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}H_t - f^\star(Y_t, t)\mathrm{d}t\middle| H_T = b\right] = g^\star(Y_t, t)\mathrm{d}\tilde{W}_t. \tag{29}$$

where the expectation is taken over the outcomes of H_t. Plugging in (21) into (29), and separating the drift and the diffusion terms on both sides of the equation (and noting that the higher-order differentials $(\mathrm{d}t)^2$ and $\mathrm{d}W_t\mathrm{d}t$ vanish), we get that the drift terms satisfy

$$\begin{aligned}\mathbb{E}_{b\sim\pi}\Big[&(\nabla\varphi(H_t))^\top \left(H_t + 2\nabla \log q_{T-t|0}(H_t|b)\right)\mathrm{d}t \\ &+ \frac{1}{2}(\mathrm{d}W_t)^\top(\nabla^2\varphi(H_t))\mathrm{d}W_t - f^\star(Y_t, t)\mathrm{d}t\Big| H_T = b\Big] = 0.\end{aligned} \tag{30}$$

Noting that $(\mathrm{d}W_t[i])^2 = \mathrm{d}t$ and $\mathrm{d}W_t[i]\mathrm{d}W_t[j] = 0$ for all $i \neq j$, we get

$$\mathbb{E}_{b\sim\pi}\Big[(\nabla\varphi(H_t))^\top \left(H_t + 2\nabla \log q_{T-t|0}(H_t|b)\right)\mathrm{d}t \\ + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(H_t))\mathrm{d}t - f^\star(Y_t,t)\mathrm{d}t\Big| H_T = b\Big] = 0. \tag{31}$$

Dividing both sides by $\mathrm{d}t$, we get an expression for the drift term $f^\star$

$$\mathbb{E}_{b\sim\pi}\Big[(\nabla\varphi(H_t))^\top \left(H_t + 2\nabla \log q_{T-t|0}(H_t|b)\right) \\ + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(H_t)) - f^\star(Y_t,t)\Big| H_T = b\Big] = 0. \tag{32}$$

Finally, from (25), we have that diffusion term $g^\star$ satisfies

$$\mathbb{E}_{b\sim\pi}\left[E\left[(\nabla\varphi(H_t))^\top\nabla\varphi(H_t) - (g^\star(Y_t,t))^2\Big|\varphi(H_t)\right] \ \Big| H_T = b\right] = 0. \tag{33}$$

Step 3: Training the drift term.

From (32), we have that function $f^\star$ is the solution to the following optimization problem:

$$\min_f \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\Big[\Big\|(\nabla\varphi(H_t))^\top\left(\frac{1}{2}H_t + 2\nabla\log q_{T-t|0}(H_t|b)\right) \\ + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(H_t)) - f(Y_t,t)\Big\|^2\Big| H_T = b\Big]. \tag{34}$$

where the inner expectation is taken over $b \sim \pi$ and over the outcomes of H_t at time t conditioned on $H_T = b$ (Note that $Y_t = \varphi(H_t)$ is a deterministic function of H_t).

Now, $H_t|\{H_T = b\}$ has the same probability distribution as $Z_{T-t}|\{Z_0 = b\}$ (and that $Y_t|\{H_T = b\}$ has the same probability distribution as $X_{T-t}|\{Z_0 = b\}$). Thus, we can re-write (34) as

$$\min_f \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\Big[\Big\|(\nabla\varphi(Z_{T-t}))^\top\left(Z_{T-t} + 2\nabla\log q_{T-t|0}(Z_{T-t}|b)\right) \\ + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(Z_{T-t})) - f(X_{T-t},t)\Big\|^2\Big| Z_0 = b\Big], \tag{35}$$

Step 4: Training the diffusion term.

From (33) we have that $g^\star$ is the solution to the following optimization problem:

$$\min_g \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\left[\left\|(\nabla\varphi(H_t))^\top\nabla\varphi(H_t) - (g(Y_t,t))^2\right\|_F^2 \Big| H_T = b\right],$$

where $\|\cdot\|_F$ is the Frobenius norm. Since $H_t|\{H_T = b\}$ has the same probability distribution as $Z_{T-t}|\{Z_0 = b\}$ (and that $Y_t|\{H_T = b\}$ has the same probability distribution as $X_{T-t}|\{Z_0 = b\}$), we can re-write (34) as

$$\min_g \mathbb{E}_{t\sim\mathrm{Unif}([0,1])}\mathbb{E}_{b\sim\pi}\left[\left\|((\nabla\varphi(Z_{T-t}))^\top\nabla\varphi(Z_{T-t}) - (g(X_{T-t},t))^2\right\|_F^2 \Big| Z_0 = b\right].$$

$\square$

B.2 Proof of Lemma B.3

In the proof of Theorem 2.2 we will use the following lemma.

Lemma B.3 (Gronwall-like inequality for SDEs on a manifold of non-negative curvature). *Suppose that $\mathcal{M}$ is a Riemannian manifold with non-negative curvature, and let $\rho(x, y)$ denote the geodesic distance between any $x, y \in \mathcal{M}$. Suppose also that X_t and $\hat{X}_t$ are two diffusions on $\mathcal{M}$ such that*

$$\mathrm{d}X_t = b(X_t, t) + \sigma(X_t, t)\mathrm{d}W_t,$$

and

$$\mathrm{d}\hat{X}_t = \hat{b}(\hat{X}_t, t) + \hat{\sigma}(X_t, t)\mathrm{d}W_t,$$

where b is $C_1(t)$-Lipschitz and σ is $C_2(t)$-Lipschitz at every time $t \in [0, T]$. Moreover, assume that

$$\|b(x, t) - \hat{b}(x, t)\| \le \varepsilon$$

and

$$\|\sigma(x, t) - \hat{\sigma}(x, t)\|_F^2 \le \varepsilon$$

for all $x \in \mathcal{M}$. Then there exists a coupling between X_t and $\hat{X}_t$ such that, for all $t \ge 0$,

$$\mathbb{E}[\rho^2(\hat{X}_t, X_t)] \le \left(\mathbb{E}[\rho^2(\hat{X}_0, X_0)] + \inf_{s\in[0,t]} \frac{5\varepsilon^2}{2C_1(s) + 3C_2(s)^2 + 2}\right) e^{\int_0^t (2C_1(s)+3C_2(s)^2+2\mathrm{d}s}.$$

Proof of Lemma B.3. We first couple X_t and $\hat{X}_t$ by setting their underlying Brownian motion terms $\mathrm{d}W_t$ to be equal to each other.

Next, we compute the distance $\rho^2(\hat{X}_t, X_t)$ using Ito's Lemma.

Letting $h(x, y) := \rho^2(x, y)$, we have

By Ito's Lemma, we have

$$\begin{aligned}
\mathrm{d}\rho^2(\hat{X}_t, X_t) &= \mathrm{d}h(\hat{X}_t, X_t) \\
&= \nabla h(\hat{X}_t, X_t)^\top \begin{pmatrix} b(X_t, t) \\ \hat{b}(\hat{X}_t, t) \end{pmatrix} \mathrm{d}t \\
&+ \frac{1}{2}\mathrm{Tr}\left[\begin{pmatrix} \sigma(X_t, t) & 0 \\ \hat{\sigma}(X_t, t) & 0 \end{pmatrix}^\top [\nabla^2 h(\hat{X}_t, X_t)] \begin{pmatrix} \sigma(X_t, t) & 0 \\ \hat{\sigma}(X_t, t) & 0 \end{pmatrix}\right] \mathrm{d}t \\
&+ \nabla h(\hat{X}_t, X_t)^\top \begin{pmatrix} \sigma(X_t, t) & 0 \\ \hat{\sigma}(X_t, t) & 0 \end{pmatrix} \mathrm{d}\begin{pmatrix} W_t \\ \hat{W}_t \end{pmatrix}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{d}\mathbb{E}[\rho^2(\hat{X}_t, X_t)] &= \mathbb{E}\left[\nabla h(\hat{X}_t, X_t)^\top \begin{pmatrix} b(X_t, t) \\ \hat{b}(\hat{X}_t, t) \end{pmatrix}\right] \mathrm{d}t \\
&+ \frac{1}{2}\mathbb{E}\left[\mathrm{Tr}\left[\begin{pmatrix} \sigma(X_t, t) & 0 \\ \hat{\sigma}(X_t, t) & 0 \end{pmatrix}^\top [\nabla^2 h(\hat{X}_t, X_t)] \begin{pmatrix} \sigma(X_t, t) & 0 \\ \hat{\sigma}(X_t, t) & 0 \end{pmatrix}\right]\right] \mathrm{d}t \\
&+ 0.
\end{aligned} \tag{36}$$

Now, since $\mathcal{M}$ has non-negative curvature, by the Rauch comparison theorem we have

$$\begin{aligned}
\left|\nabla h(\hat{X}_t, X_t)^\top \begin{pmatrix} b(X_t, t) \\ \hat{b}(\hat{X}_t, t) \end{pmatrix}\right| &\le 2\rho(\hat{X}_t, X_t) \times \|\hat{b}(\hat{X}_t, t) - \Gamma_{X_t \to \hat{X}_t}(b(X_t, t))\| \\
&\le 2\rho(\hat{X}_t, X_t) \times \left(\|b(\hat{X}_t, t) - \Gamma_{X_t \to \hat{X}_t}(b(X_t, t))\| + \|b(\hat{X}_t, t) - \hat{b}(\hat{X}_t, t)\|\right) \\
&\le 2\rho(\hat{X}_t, X_t) \times (C_1(t)\rho(\hat{X}_t, X_t) + \varepsilon)
\end{aligned} \tag{37}$$

where the last inequality holds since b is $C_1(t)$-Lipschitz.

Moreover, since $\mathcal{M}$ has non-negative curvature, by the Rauch comparison theorem we also have that

$$\begin{aligned}
&\frac{1}{2}\mathrm{Tr}\left[\begin{pmatrix}\sigma(X_t,t) & 0\\ \hat{\sigma}(X_t,t) & 0\end{pmatrix}^\top [\nabla^2 h(\hat{X}_t,X_t)]\begin{pmatrix}\sigma(X_t,t) & 0\\ \hat{\sigma}(X_t,t) & 0\end{pmatrix}\right]\\
&\qquad\leq \left\|\hat{\sigma}(\hat{X}_t,t)-\Gamma_{X_t\to\hat{X}_t}(\sigma(X_t,t))\right\|_F^2\\
&\qquad\leq \left(\left\|\sigma(\hat{X}_t,t)-\Gamma_{X_t\to\hat{X}_t}(\sigma(X_t,t))\right\|_F+\left\|\hat{\sigma}(\hat{X}_t,t)-\sigma(\hat{X}_t,t)\right\|_F\right)^2\\
&\leq 3\left\|\sigma(\hat{X}_t,t)-\Gamma_{X_t\to\hat{X}_t}(\sigma(X_t,t))\right\|_F^2+3\left\|\hat{\sigma}(\hat{X}_t,t)-\sigma(\hat{X}_t,t)\right\|_F^2\\
&\leq 3C_2(t)^2\rho^2(\hat{X}_t,X_t)+3\varepsilon^2
\end{aligned} \tag{38}$$

Plugging (37) and (38) into (36), we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\rho^2(\hat{X}_t,X_t)]\leq 2\mathbb{E}[C_1(t)\rho^2(\hat{X}_t,X_t)+\varepsilon\rho(\hat{X}_t,X_t)]+3C_2(t)^2\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+3\varepsilon^2 \qquad \forall t\geq 0. \tag{39}$$

Hence,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\rho^2(\hat{X}_t,X_t)] &\leq 2\mathbb{E}[C_1(t)\rho^2(\hat{X}_t,X_t)+\rho^2(\hat{X}_t,X_t)]+3C_2(t)^2\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+5\varepsilon^2\\
&= 2\mathbb{E}[C_1(t)\rho^2(\hat{X}_t,X_t)+\rho^2(\hat{X}_t,X_t)]+3C_2(t)^2\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+5\varepsilon^2\\
&= (2C_1(t)+3C_2(t)^2+2)\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+5\varepsilon^2
\end{aligned}$$

Let $\tau\in[0,T]$ be some number, and define $R(t):=\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+\inf_{s\in[0,\tau]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}$ for all $t\in[0,\tau]$.

Then we have,

$$\frac{\mathrm{d}}{\mathrm{d}t}R(t)\leq(2C_1(t)+3C_2(t)^2+2)R(t) \qquad \forall t\geq 0 \tag{40}$$

Thus, plugging (40) into Gronwall's lemma, we have, for all $t\geq 0$,

$$\begin{aligned}
R(t)&\leq R(0)e^{\int_0^t(2C_1(s)+3C_2(s)^2+2\mathrm{d}s}\\
&=\left(\mathbb{E}[\rho^2(\hat{X}_0,X_0)]+\inf_{s\in[0,\tau]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}\right)e^{\int_0^t 2C_1(s)+3C_2(s)^2+2\mathrm{d}s}
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbb{E}[\rho^2(\hat{X}_t,X_t)]+\inf_{s\in[0,\tau]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}\\
&\leq\left(\mathbb{E}[\rho^2(\hat{X}_0,X_0)]+\inf_{s\in[0,T]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}\right)e^{\int_0^t 2C_1(s)+3C_2(s)^2+2\mathrm{d}s}
\end{aligned}$$

Hence, for all $t\geq 0$,

$$\mathbb{E}[\rho^2(\hat{X}_t,X_t)]\leq\left(\mathbb{E}[\rho^2(\hat{X}_0,X_0)]+\inf_{s\in[0,\tau]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}\right)e^{\int_0^t 2C_1(s)+3C_2(s)^2+2\mathrm{d}s}.$$

Plugging in $\tau=t$ in the above equation, we have, for all $t\geq 0$,

$$\mathbb{E}[\rho^2(\hat{X}_t,X_t)]\leq\left(\mathbb{E}[\rho^2(\hat{X}_0,X_0)]+\inf_{s\in[0,t]}\frac{5\varepsilon^2}{2C_1(s)+3C_2(s)^2+2}\right)e^{\int_0^t 2C_1(s)+3C_2(s)^2+2\mathrm{d}s}.$$

$\square$

B.3 PROOF THAT AVERAGE-CASE LIPSCHITZNESS HOLDS ON SYMMETRIC MANIFOLDS OF INTEREST (LEMMA B.4)

Lemma B.4 (Average-case Lipschitzness)**.** *For the Unitary group, we have that Assumption($\varphi, L_1, L_2, \alpha$) 2.1 holds for $L_1 = O(d^{1.5}\sqrt{T}\alpha^{-\frac{1}{3}})$ and $L_2 = O(d^2 T\alpha^{-\frac{2}{3}})$. For the sphere, it holds for $L_1 = L_2 = O(\alpha^{-\frac{1}{d}})$. For the Torus it holds for $L_1 = L_2 = 1$.*

Proof. For the torus, the map $\varphi(x)$ has $\nabla\varphi(x) = I_d$ at every $x \in \mathbb{R}^d$, which implies that Assumption 2.1 is satisfied for $L_1 = L_2 = 1$.

Sphere. In the case of the sphere, which we embed via the map ψ as a unit sphere in $\mathbb{R}^d$, one can easily observe that e.g. $\|\nabla\varphi(z)\| \leq O(1)$ for any z outside a ball of radius $r \geq \Omega(1)$ centered at the origin. As the volume of a ball of radius $r = \alpha$ is $\frac{1}{r^d}$ times the volume of the unit ball, one can use standard Gaussian concentration inequalities to show that the Brownian motion X_t will remain outside this ball for time T with probability at least $1 - 4\frac{1}{r^d T}$.

Moreover, by standard Gaussian concentration inequalities Rudelson and Vershynin (2013), we have that $\|X_t\| \leq 2\sqrt{Td}\log(\frac{1}{\alpha})$ with probability at least $1 - 2\alpha$ for all $t \in [0, T]$.

This motivates defining the set $\Omega_t := \{z \in \mathbb{R}^d : (4\frac{1}{\alpha T})^{\frac{1}{d}} \leq \|z\| \leq 2\sqrt{Td}\log(\frac{1}{\alpha})\}$, as we then have

$$\mathbb{P}(X_t \in \Omega_t \ \forall\ t \in [0, T]) \geq 1 - \alpha.$$

Since $\|z\| \geq (4\frac{1}{\alpha T})^{\frac{1}{d}}$ for any $z \in \Omega_t$ and any $t \in [0, T]$, we must have that

$$\|\nabla\varphi(z(U, \Lambda))\|_{2\to 2} \leq 3(4\frac{1}{\alpha T})^{\frac{2}{d}} = L_1,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(z(U, \Lambda))\|_{2\to 2} \leq 3(4\frac{1}{\alpha T})^{\frac{2}{d}} = L_1,$$

$$\|\nabla^2\varphi(z(U, \Lambda))\|_{2\to 2} \leq 3(4\frac{1}{\alpha T})^{\frac{3}{d}} = L_2,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(z(U, \Lambda))\|_{2\to 2} \leq 3(4\frac{1}{\alpha T})^{\frac{3}{d}} = L_2,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}(z(U, \Lambda))\|_{2\to 2} \leq \|x\|,$$

Unitary group. We next show that the Lipschitz property holds for the unitary group $\mathbb{U}(n)$. Similar techniques can be used for the case of the special orthogonal group, and we omit those details. We first recall results from random matrix theory which allow us to bound the eigenvalue caps of a matrix with Gaussian entries. Specifically, these results say that, roughly speaking, if X_0 is any matrix and $X_t = X_0 + B(t)$, where $B(t)$ is a symmetric matrix with iid $N(0, t)$ entries undergoing Brownian motion, one has that the eigenvalues $\gamma_1(t) \geq \cdots \geq \gamma_n(t)$ of X_t satisfy (see e.g. Anderson et al. (2010); Mangoubi and Vishnoi (2023))

$$\mathbb{P}(\inf_{s\in[t_0,T]}(\gamma_{i+1}(t) - \gamma_i(t)) \leq s\frac{1}{\mathrm{poly}(d)\sqrt{t}}) \leq O(s^{\frac{1}{2}}) \qquad \forall s \geq 0. \tag{41}$$

Thus, if we define Ω_t to be the set of outcomes of such that $\gamma_{i+1}(t) - \gamma_i(t) \leq \alpha^2\frac{1}{\mathrm{poly}(n)\sqrt{t}}$, we have that $\mathbb{P}(X_t \in \Omega_t \ \ \forall t \in [t_0, T]) \geq 1 - \alpha$.

From the Matrix calculus formulas for $\nabla\varphi(U^\top \Lambda U)$, $\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(U^\top \Lambda U)$, $\nabla\varphi(U^\top \Lambda U)$, and $\frac{\mathrm{d}}{\mathrm{d}U}\nabla^2\varphi(U^\top \Lambda U)$, we have that, for all $z(U, \Lambda) = U\Lambda U^\top \in \Omega$,

$$\|\nabla\varphi(z(U, \Lambda))\|_{2\to 2} \leq \sum_{i=1}^{d}\frac{1}{\lambda_{i+1} - \lambda_i} \leq d^{1.5}\sqrt{t}\alpha^{-\frac{1}{3}} = L_1,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(z(U,\Lambda))\|_{2\to 2} \leq \|\Lambda\|_{2\to 2}\sum_{i=1}^{d}\frac{1}{\lambda_{i+1}-\lambda_i}$$

$$\leq (C+\sqrt{T}d\log(\frac{1}{\alpha}))\times\sum_{i=1}^{d}\frac{1}{\lambda_{i+1}-\lambda_i} \leq d^{1.5}\sqrt{t}\alpha^{-\frac{1}{3}} = L_1,$$

$$\|\nabla^2\varphi(z(U,\Lambda))\|_{2\to 2} \leq \sum_{i=1}^{d}\frac{1}{(\lambda_{i+1}-\lambda_i)^2} \leq d^2 t\alpha^{-\frac{2}{3}} = L_2,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}\nabla\varphi(z(U,\Lambda))\|_{2\to 2} \leq \|\Lambda\|_{2\to 2}\sum_{i=1}^{d}\frac{1}{(\lambda_{i+1}-\lambda_i)^2}$$

$$(C+\sqrt{T}d\log(\frac{1}{\alpha}))\times\sum_{i=1}^{d}\frac{1}{(\lambda_{i+1}-\lambda_i)^2} \leq d^2 t\alpha^{-\frac{2}{3}} = L_2,$$

$$\|\frac{\mathrm{d}}{\mathrm{d}U}(z(U,\Lambda))\|_{2\to 2} \leq \|\Lambda\|_{2\to 2}$$

since $\lambda_{i+1}-\lambda_i \leq \alpha^{\frac{1}{3}}\frac{1}{\sqrt{d}\sqrt{t}}$ for all $i\in[d]$ and $\|\Lambda\|_{2\to 2} \leq 2\sqrt{Td}\log(\frac{1}{\alpha})$ whenever $z(U,\Lambda)\in\Omega_t$

$\square$

B.4 Proof of Lipschitzness of $f^\star$ and $g^\star$ on all of $\mathcal{M}$(Lemma B.6)

We will use the following Proposition of Chen et al. (2023b):

Proposition B.5 (Proposition 20 of Chen et al. (2023b))**.** *Suppose that $\psi(\pi)$ has support on a ball of radius $C>0$.*

For any $\alpha>0$, define the "early stopping time" $t_0 := \min(\frac{\alpha}{C},\frac{\alpha^2}{d})$.

Then the drift term $\nabla\log q_t(\cdot)$ of the reverse diffusion SDE in Euclidean space is $O(\frac{1}{\alpha^2}dC^2(\min(C,\sqrt{d})^2))$-Lipschitz at every time $t>t_0$.

Moreover, $W_2(q_{t_0},\pi)\leq\alpha$.

Denote by $\Gamma_{x\to y}(v)$ the parallel transport of a vector v from x to y.

Lemma B.6. *Suppose that Assumption 2.1($\varphi, L_1, L_2, \alpha$) and Assumption B.1($\psi,\pi,C$) both hold. Then for every $t\in[t_0,T]$,*

$$\|f^\star(x,t)-\Gamma_{x\to y}(f^\star(x,t))\| \leq \mathcal{C}\times\rho(x,y), \qquad \forall x,y\in\mathcal{M} \tag{42}$$

and

$$\|g^\star(y,t)-\Gamma_{x\to y}(g^\star(x,t))\|_F \leq \mathcal{C}\times\rho(x,y) \qquad \forall x,y\in\mathcal{M} \tag{43}$$

where $\mathcal{C} := (C+\sqrt{T}d\log(\frac{1}{\alpha}))^4\times L_3^2\times L_1 + (C+\sqrt{T}d\log(\frac{1}{\alpha}))^2\times L_3\times L_2$ and $t_0 := \min(\frac{\alpha}{C},\frac{\alpha^2}{d})$, and $L_3 = O(\frac{1}{\alpha^2}dC^2(\min(C,\sqrt{d})^2))$.

Proof. Recall that (when, e.g., $\mathcal{M}$ is one of the aformentioned symmetric manifolds) we may decompose any $z\in\mathbb{R}^d$ as $z\equiv z(U,\Lambda)$ where $U\in\mathcal{M}$.

We have the following expression for $f^\star(U,t)$

$$f^\star(U,t) = c_U\int_{\Lambda\in\mathcal{A}}\left[(\nabla\varphi(z(U,\Lambda)))^\top\nabla\log q_{T-t|0}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \times q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda,$$

where $c_U = \left(\int_{\Lambda\in\mathcal{A}} q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda\right)^{-1}$ is a normalizing constant.

Then

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}U}f^{\star}(U,t) \\
&= c_U \times \frac{\mathrm{d}}{\mathrm{d}U}\int_{\Lambda\in\mathcal{A}}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&+ (\frac{\mathrm{d}}{\mathrm{d}U}c_U) \times \int_{\Lambda\in\mathcal{A}}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda
\end{aligned} \tag{44}
$$

For the first term on the r.h.s. of (44) we have,

$$
\begin{aligned}
&c_U \times \frac{\mathrm{d}}{\mathrm{d}U}\int_{\Lambda\in\mathcal{A}}\left[(\nabla_U\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&= c_U \times \int_{\Lambda\in\mathcal{A}}\left(\frac{\mathrm{d}}{\mathrm{d}U}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right]\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&+ c_U \times \int_{\Lambda\in\mathcal{A}}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times \frac{\mathrm{d}}{\mathrm{d}U}q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&= c_U \times \int_{\Lambda\in\mathcal{A}}\left(\frac{\mathrm{d}}{\mathrm{d}U}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right]\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&+ c_U \times \int_{\Lambda\in\mathcal{A}}\left[(\nabla\varphi(z(U,\Lambda)))^{\top}\nabla\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad \times \nabla_U\log q_{T-t}(z(U,\Lambda)) \times q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\mathrm{d}\Lambda, \\
&= \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\left[\frac{\mathrm{d}}{\mathrm{d}U}\left((\nabla\varphi(z(U,\Lambda)))^{\top}\nabla_U\log q_{T-t|0}(z(U,\Lambda))\right.\right. \\
&\qquad\qquad\qquad\qquad\left.\left. + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right)\mathbb{1}_{\Omega}(\Lambda)\middle| V=U\right], \\
&+ \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\left[\left((\nabla\varphi(z(U,\Lambda)))^{\top}\nabla_U\log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right)\right. \\
&\qquad\qquad \left.\times \nabla_U\log q_{T-t}(z(U,\Lambda))\mathbb{1}_{\Omega}(\Lambda)\middle| V=U\right],
\end{aligned}
$$

For the second term on the r.h.s. of (44) we have,

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}U}c_U &= c_U^2 \int_{\Lambda\in\mathcal{A}} \frac{\mathrm{d}}{\mathrm{d}U}(q_{T-t}(z(U,\Lambda)))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda \\
&= c_U^2 \int_{\Lambda\in\mathcal{A}} \frac{\mathrm{d}}{\mathrm{d}U}(e^{\log q_{T-t}(z(U,\Lambda))})\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda \\
&= c_U^2 \int_{\Lambda\in\mathcal{A}} \nabla_U \log q_{T-t}(z(U,\Lambda))(e^{\log q_{T-t}(z(U,\Lambda))})\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda \\
&= c_U^2 \int_{\Lambda\in\mathcal{A}} \nabla_U \log q_{T-t}(z(U,\Lambda)) \times q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda \\
&= c_U \times \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\left[\nabla_U \log q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda) \,\middle|\, V=U\right]
\end{aligned}
$$

and hence,

$$
\begin{aligned}
&(\frac{\mathrm{d}}{\mathrm{d}U}c_U) \times \int_{\Lambda\in\mathcal{A}} \left[(\nabla\varphi(z(U,\Lambda)))^\top \nabla \log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \times q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\mathrm{d}\Lambda \\
&= \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\left[\nabla_U \log q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda) \,\middle|\, V=U\right] \\
&\quad \times \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\Bigg[\Big((\nabla\varphi(z(U,\Lambda)))^\top \nabla \log q_{T-t}(z(U,\Lambda)) \\
&+ \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\Big)\mathbb{1}_\Omega(\Lambda) \,\Bigg|\, V=U\Bigg]
\end{aligned}
$$

Thus

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}U}f^\star(U,t) \qquad (45) \\
&= \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\Bigg[\frac{\mathrm{d}}{\mathrm{d}U}\Big((\nabla\varphi(z(U,\Lambda)))^\top \nabla_U \log q_{T-t|0}(z(U,\Lambda)) \\
&\qquad\qquad + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\Big)\mathbb{1}_\Omega(\Lambda)\Bigg|V=U\Bigg], \\
&+ \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\Bigg[\Big((\nabla\varphi(z(U,\Lambda)))^\top \nabla_U \log q_{T-t}(z(U,\Lambda)) + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\Big) \\
&\qquad\qquad\qquad\qquad \times \nabla_U \log q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda)\Bigg|V=U\Bigg] \\
&+ \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\left[\nabla_U \log q_{T-t}(z(U,\Lambda))\mathbb{1}_\Omega(\Lambda) \,\middle|\, V=U\right] \\
&\quad \times \mathbb{E}_{z(U,\Lambda)\sim q_{T-t}}\Bigg[\Big((\nabla\varphi(z(U,\Lambda)))^\top \nabla \log q_{T-t}(z(U,\Lambda)) \qquad (46) \\
&\qquad\qquad + \frac{1}{2}\mathrm{tr}(\nabla^2\varphi(z(U,\Lambda)))\Big)\mathbb{1}_\Omega(\Lambda) \,\Bigg|\, V=U\Bigg] \qquad (47)
\end{aligned}
$$

Moreover, by standard Gaussian Concentration inequalities we have that $\|z(U,\Lambda)\|_F \leq C + \sqrt{T}d\log(\frac{1}{\alpha})$. From Proposition B.5 we have that $\nabla \log p_{T-t|0}(z(U,\Lambda))$ is L_3-Lipschitz where $L_3 := O(\frac{1}{\alpha^2}dC^2(\min(C,\sqrt{d})^2))$ and hence that

$$
\begin{aligned}
\|\nabla_U \log p_{T-t|0}(z(U,\Lambda))\|_{2\to 2} &\leq \|\frac{\mathrm{d}}{\mathrm{d}U}(z(U,\Lambda))\|_{2\to 2} \times \|\nabla \log p_{T-t|0}(z(U,\Lambda))\|_{2\to 2} \\
&\leq \|\frac{\mathrm{d}}{\mathrm{d}U}(z(U,\Lambda))\|_{2\to 2} \times L_3 \times \|z(U,\Lambda)\|_F \leq L_3 \times (C + \sqrt{T}d\log(\frac{1}{\alpha})) \\
&\leq (C + \sqrt{T}d\log(\frac{1}{\alpha}))^2 \times L_3, \qquad (48)
\end{aligned}
$$

where the last inequality holds by Assumption B.1 and standard Gaussian concentration inequalities.

Thus, plugging Assumption 2.1 and (48) into (45), we have that

$$
\left\|\frac{\mathrm{d}}{\mathrm{d}U} f^\star(U,t)\right\|_{2\to 2} \leq (C + \sqrt{T}d\log(\frac{1}{\alpha}))^4 \times L_3^2 \times L_1 + (C + \sqrt{T}d\log(\frac{1}{\alpha}))^2 \times L_3 \times L_2 \qquad (49)
$$

Replacing σ with μ in the above calculation, we also get that

$$
\left\|\frac{\mathrm{d}}{\mathrm{d}U} g^\star(U,t)\right\|_{2\to 2} \leq (C + \sqrt{T}d\log(\frac{1}{\alpha}))^4 \times L_3^2 \times L_1 + (C + \sqrt{T}d\log(\frac{1}{\alpha}))^2 \times L_3 \times L_2 \qquad (50)
$$

Thus, (49) and (50) imply that

$$
\|f^\star(y,t) - \Gamma_{x\to y}(f^\star(x,t))\| \leq \mathcal{C} \times \rho(x,y), \qquad \forall x,y \in \mathcal{M} \qquad (51)
$$

and

$$
\|g^\star(y,t) - \Gamma_{x\to y}(g^\star(x,t))\|_F \leq \mathcal{C} \times \rho(y,x) \qquad \forall x \in \mathcal{M}, \qquad (52)
$$

where $\mathcal{C} := (C + \sqrt{T}d\log(\frac{1}{\alpha}))^4 \times L_3^2 \times L_1 + (C + \sqrt{T}d\log(\frac{1}{\alpha}))^2 \times L_3 \times L_2$.

□

B.5 WASSERSTEIN TO TV CONVERSION ON THE MANIFOLD (LEMMA B.7)

Lemma B.7 (Wasserstein to TV conversion on the manifold). *There is a number $c \leq \mathrm{poly(d)}$ such that for every $t \in [t_0, T]$ and any $\tau \leq \frac{1}{c}$ we have*

$$
\begin{aligned}
&\|\mathcal{L}_{Y_{t+\tau+\hat{\Delta}}} - \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}\|_{\mathrm{TV}} - \|\mathcal{L}_{Y_t} - \mathcal{L}_{\hat{y}_t}\|_{\mathrm{TV}} \\
&\leq \sqrt{D_{\mathrm{KL}}(\nu_1 \,\|\, p_{t+\tau+\hat{\Delta}|t+\tau}(\,\cdot\,|Y_{t+\tau}))} + \sqrt{D_{\mathrm{KL}}(\nu_1 \| \mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_t})} \leq O(\varepsilon c). \qquad (53)
\end{aligned}
$$

Proof of Lemma B.7. Now that we have shown that $f^\star$ and $g^\star$ are $\mathrm{poly}(d)$-Lipschitz (by Lemmas B.4 and B.6), we can apply Lemma B.3 to bound the Wasserstein distance: $W_2(\hat{Y}_{t+\tau}, Y_{t+\tau}) \leq (\rho^2(\hat{Y}_t, Y_t) + \varepsilon)e^{c\tau} \quad \forall \tau \geq 0$, where $c \leq \mathrm{poly}(d)$.

Moreover, with slight abuse of notation, we may define $\hat{y}_{t+\tau}$ to be a continuous-time interpolation of the discrete process $\hat{y}$. Applying (13) to this process we get that, roughly, $W_2(\hat{Y}_{t+\tau}, \hat{y}_{t+\tau}) \leq (\rho^2(\hat{y}_t, Y_t) + \varepsilon + \Delta)e^{c\tau}$ for $\tau \geq 0$. Thus, we get a bound on the Wasserstein error,

$$
W_2(Y_{t+\tau}, \hat{y}_{t+\tau}) \leq W_2(\hat{Y}_{t+\tau}, Y_{t+\tau}) + W_2(\hat{Y}_{t+\tau}, \hat{y}_{t+\tau}) \leq (\rho^2(\hat{y}_t, Y_t) + \varepsilon + \Delta)e^{c\tau} \qquad \tau \geq 0 \qquad (54)
$$

Unfortunately, after times $\tau > \frac{1}{c} = \frac{1}{\mathrm{poly(d)}}$, this bound grows exponentially with the dimension d.

To overcome this challenge, we define a new coupling between Y_t and $\hat{Y}_t$ which we "reset" after time intervals of length $\tau = \frac{1}{c}$ by converting our Wasserstein bound into a total variation bound after each time interval. Towards this end, we use the fact that if at any time t the total variation distance satisfies $\|\mathcal{L}_{Y_t} - \mathcal{L}_{\hat{y}_t}\|_{\mathrm{TV}} \leq \alpha$, then there exists a coupling such that $Y_t = \hat{Y}_t$ with probability at least $1-\alpha$. In other words, w.p. $\geq 1-\alpha$, we have $\rho(\hat{y}_{t+\tau}, Y_{t+\tau}) = 0$, and we can apply inequality (54) over the next time interval of τ without incurring an exponential growth in time. Repeating this process $\frac{T}{\tau}$ times, we get that $\|\mathcal{L}_{Y_T} - \mathcal{L}_{\hat{y}_T}\| \leq \alpha \times \frac{T}{\tau}$, where the TV error grows only *linearly* with T.

Converting Wasserstein bounds on the manifold to TV bounds. To complete the proof, we still need to show how to convert the Wasserstein bound into a TV bound (Lemma B.7). Towards this end, we begin by showing that the transition kernel $\tilde{p}_{t+\tau+\hat{\Delta}|t+\tau}(\cdot\,|H_{t+\tau})$ of the reverse diffusion H_t in $\mathbb{R}^d$ is close to a Gaussian in KL distance:

$$D_{\mathrm{KL}}(N(H_{t+\tau}+\hat{\Delta}\nabla\tilde{p}_{T-t-\tau}(H_{t+\tau}),\hat{\Delta}I_d)\,\|\,\tilde{p}_{t+\tau+\hat{\Delta}|t+\tau}(\cdot\,|H_{t+\tau}))\leq\frac{\alpha\tau}{T}$$

. One can do this using Girsanov's theorem, since, unlike the diffusion Y_t on the manifold, the reverse diffusion in Euclidean space H_t *does* have a constant diffusion term (see e.g. Theorem 9 of Chen et al. (2023b)).

Next, we use the fact that with probability at least $1-\alpha\frac{\tau}{T}$ the map φ in a ball of radius $\frac{1}{\mathrm{poly}(d)}$ about the point $H_{t+\tau}$ has c-Lipschitz Jacobian where $c=\mathrm{poly}(d)$, and that the inverse of the exponential map $\exp(\cdot)$ has $O(1)$-Lipschitz Jacobian, to show that the transition kernel p_t of $Y_t=\varphi(H_t)$ satisfies

$$D_{\mathrm{KL}}(\nu_1\,\|\,p_{t+\tau+\hat{\Delta}|t+\tau}(\cdot\,|Y_{t+\tau}))\leq(1+\hat{\Delta}c)^d\frac{\alpha\tau}{T}\leq 2\frac{\alpha\tau}{T}$$

if we choose $\hat{\Delta}\leq O(\frac{1}{cd})$, where $\nu_1:=\exp_{Y_{t+\tau}}(N(Y_{t+\tau}+\hat{\Delta}f^\star(Y_{t+\tau},t+\tau),\ \hat{\Delta}g^{\star 2}(Y_{t+\tau},t+\tau)I_d))$.

Next, we plug in our Wasserstein bound $W(Y_{t+\tau},\hat{y}_{t+\tau})\leq O(\varepsilon)$ into the formula for the KL divergence between two Gaussians to bound $\|\mathcal{L}_{Y_{t+\tau+\hat{\Delta}}}-\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}\|_{\mathrm{TV}}$. Specifically, noting that $\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_t}=\exp_{\hat{y}_{t+\tau}}(N(\hat{y}_{t+\tau}+\hat{\Delta}f(\hat{y}_{t+\tau},t+\tau),\hat{\Delta}g^2(\hat{y}_{t+\tau},t+\tau)I_d))$, we have that

$$\begin{aligned}D_{\mathrm{KL}}(\nu_1,\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_{t+\tau}})&=\big(\mathrm{Tr}(g^{\star 2}(Y_{t+\tau},t+\tau))^{-1}g^2(\hat{y}_{t+\tau},t+\tau)\big)\\&\quad-d+\log\tfrac{\det g^{\star 2}(Y_{t+\tau},t+\tau)}{\det g^2(\hat{y}_{t+\tau},t+\tau)}+w^\top(\hat{\Delta}g^{\star 2}(Y_{t+\tau},t))^{-1}w,\end{aligned}$$

where $w:=Y_{t+\tau}-\hat{y}_{t+\tau}+\hat{\Delta}(f^\star(Y_{t+\tau},t+\tau)-f(\hat{y}_{t+\tau},t+\tau))$. Since with probability $\geq 1-\alpha\frac{\tau}{T}$ we have $g^\star(Y_{t+\tau})\succeq\mathrm{poly}(d)$, plugging in the error bounds $\|f^\star(Y_{t+\tau},t)-f(Y_{t+\tau},t)\|\leq\varepsilon$ and $\|g^\star(Y_{t+\tau},t)-g(Y_{t+\tau},t)\|_F\leq\varepsilon$ and the c-Lipschitz bounds on $f^\star$ and $g^\star$, where $c=\mathrm{poly}(d)$, (Assumption 2.1), we get that $D_{\mathrm{KL}}(\nu_1,\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}})\leq O(\varepsilon^2c^2)$. Thus, by Pinsker's inequality, we have

$$\begin{aligned}&\|\mathcal{L}_{Y_{t+\tau+\hat{\Delta}}}-\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}}\|_{\mathrm{TV}}-\|\mathcal{L}_{Y_t}-\mathcal{L}_{\hat{y}_t}\|_{\mathrm{TV}}\\&\leq\sqrt{D_{\mathrm{KL}}(\nu_1\,\|\,p_{t+\tau+\hat{\Delta}|t+\tau}(\cdot\,|Y_{t+\tau}))}+\sqrt{D_{\mathrm{KL}}(\nu_1\|\mathcal{L}_{\hat{y}_{t+\tau+\hat{\Delta}}|\hat{y}_t})}\leq O(\varepsilon c).\end{aligned}\tag{55}$$

□

B.6 Completing the proof of Theorem 2.2

Bounding the accuracy. Recall that q_t is the distribution of the forward diffusion Z_t in Euclidean space after time t, which is an Ornstein-Uhlenbeck process. Standard mixing bounds for Ornstein-Uhlenbeck process imply that

$$\|q_t-N(0,I_d)\|_{TV}\leq O(Ce^{-t})$$

for all $t>0$ (see e.g. Bakry et al. (2014)). Thus, it is sufficient to choose $T=\log(\frac{1}{C\varepsilon})$ to ensure that

$$\|\mathcal{L}_{Y_T}-\pi\|_{\mathrm{TV}}=\|q_T-N(0,I_d)\|_{\mathrm{TV}}\leq O(\varepsilon)$$

.

As Lemma B.7 holds for all $t\in\tau\mathbb{N}$, the distribution $\nu=\mathcal{L}_{\hat{y}_T}$ of our sampling algorithm's output satisfies

$\|\pi-\nu\|_{\mathrm{TV}}=\|\mathcal{L}_{Y_T}-\pi\|_{\mathrm{TV}}+\|\mathcal{L}_{Y_T}-\nu\|_{\mathrm{TV}}\leq O(\varepsilon)+O(\varepsilon c\times\frac{T}{\tau})=O(\varepsilon\times\mathrm{poly}(d))$.

Bounding the runtime of the sampling algorithm. Since our accuracy bound requires $T=\log(\frac{d}{\varepsilon C})$, and requires a time-step size of $\Delta\leq\frac{1}{\mathrm{poly}(d)}$, the number of iterations is bounded by

$$\frac{T}{\Delta}\leq O\left(\mathrm{poly}(d)\times\log\left(\frac{d}{\varepsilon C}\right)\right).$$

C CHALLENGES ENCOUNTERED WHEN APPLYING EUCLIDEAN DIFFUSION FOR GENERATING POINTS CONSTRAINED TO NON-EUCLIDEAN SYMMETRIC MANIFOLDS

The following examples illustrate why using Euclidean diffusion models to enforce symmetric manifold constraints may be insufficient.

Example 1: Consider the problem of generating points from a distribution μ on the d-dimensional torus $\mathbb{T}_d = \mathbb{S}_1 \times \cdots \times \mathbb{S}_1$, given a dataset D sampled from μ. A naive approach is to map the dataset D from the torus to Euclidean space via the map ψ which maps each point on the torus to its angles in $[0, 2\pi)^d \subseteq \mathbb{R}^d$. One can then train a Euclidean diffusion model on the dataset $\psi(D)$.

However, the map ψ can greatly distort the geometry of μ. To see why, let μ be a unimodal distribution on $\mathbb{T}_d$ with mode cenetered near $(0, \ldots, 0)$. The pushforward of μ under ψ consists of a distribution with 2^d modes, each near the 2^d corners of the d-cube $[0, 2\pi)^d$ (see Figure 1). Thus, a Euclidean diffusion model needs to learn a multimodal distribution, which may be much harder than learning a unimodal distribution.

Example 2: Another example is the problem of generating samples from a distribution on the manifold $\mathbb{SO}(3)$ of rotation matrices. There is a natural map ψ from $\mathbb{SO}(3)$ to $\mathbb{R}^3$ which maps any $M \in \mathbb{SO}(3)$ to its three Euler angles $(a, b, c) \in [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\pi, \pi] \subseteq \mathbb{R}^3$. However, ψ has a singularity at $b = \frac{\pi}{2}$, which may make it harder to learn distributions with a region of high probability density passing through this singularity, as ψ may separate this region into multiple disconnected regions.

Additionally, it has been observed empirically that applying Euclidean diffusion models to generate Euler angles in $\mathbb{R}^3$ leads to samples of lower quality than those generated by diffusion models on the manifold $\mathbb{SO}(3)$; see e.g. Leach et al. (2022), and Watson et al. (2023).

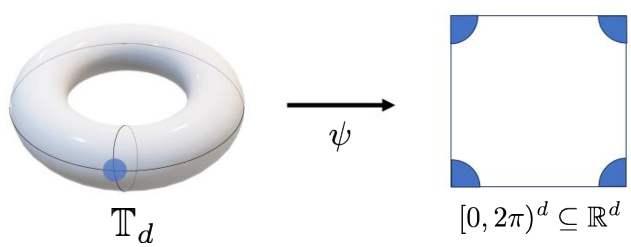

Figure 1: A probability density μ with one mode (blue) on the torus. The map ψ, which maps points in the d-dimensional torus $\mathbb{T}_d$ to Euclidean space $\mathbb{R}^d$, may break up the single mode on the torus into up to 2^d separated modes in $\mathbb{R}^d$. This can make the task of learning the pushforward of the target distribution on $\mathbb{R}^d$ much more challenging than the task of learning the original target distribution on the torus, as the distribution in $\mathbb{R}^d$ may have exponentially-in-d more modes.