Kernel-Gradient Drifting Models

Anonymous Authors[1]

Abstract

We propose kernel-gradient drifting, a one-step generative modeling framework that replaces the fixed Euclidean displacement direction in drifting models with directions induced by the kernel itself. Standard drifting is attractive because it enables fast, high-quality generation without distilling a large pretrained diffusion model, but its theory is currently understood mainly for Gaussian kernels, where the drift coincides with smoothed score matching and is identifiable. Our gradient-based reformulation exposes this score-based structure for general kernels: the resulting drift is the score difference between kernel-smoothed data and model distributions, yielding identifiability for characteristic kernels and a smoothed-KL descent interpretation of the drifting dynamics. Since kernel gradients are intrinsic tangent vectors, the same construction extends naturally to Riemannian manifolds and to discrete data via the Fisher-Rao geometry of the probability simplex. Across spherical geospatial data, promoter DNA and molecule generation, kernel-gradient drifting enables state-of-the-art one-step generation beyond the Euclidean setting without distillation.

1. Introduction

Many generative models rely on transport-based methods that map a simple prior to a complex data distribution (Lipman et al., 2023; Ho et al., 2020; Song et al., 2021). These methods are empirically strong and well understood, but sampling typically requires solving an SDE or ODE through many sequential updates. This has motivated a growing line of work on one-step or few-step generation, either by distilling a pretrained model (Salimans & Ho, 2022; Hu et al., 2023; Yin et al., 2024) or by training such generators directly through flow-map, consistency-style, or velocity-matching objectives (Boffi et al., 2025a;b; Davis et al., 2025; Roos et al., 2026; Geng et al., 2026; 2025; Zhou et al., 2026). Drifting models (Deng et al., 2026) instead train a one-step generator directly by moving model samples along a kernel-induced attraction and repulsion field. The kernel measures similarity between samples and determines the strength of their pairwise interactions, making it an important design choice for the model. In this way, drifting shifts the refinement that would normally occur at sampling time into the training procedure.

Current theory for drifting is best understood when working with Gaussian kernels. However, other non-Gaussian kernels, such as Laplace, seem to perform better empirically, making it important to understand drifting beyond the Gaussian case. In the Gaussian case, the original kernel-weighted displacement field coincides with a smoothed score-matching direction, which gives a clean *identifiability* story: at the population level, vanishing drift implies that the model distribution matches the data distribution (Lai et al., 2026; Turan & Ovsjanikov, 2026), an essential property for the model to be sound. At the same time, this result also exposes a point of tension in the original formulation. Drifting combines two choices: the kernel $k(x, y)$, which determines how samples are weighted, and the Euclidean displacement $y - x$, which determines the direction of motion. For Gaussian kernels this displacement direction is natural, as it points in the direction of steepest increase in similarity. For other kernels, however, the same displacement direction doesn't necessarily point in the direction of highest increase in similarity and, in general, doesn't ensure identifiability. Consequently, it is unclear when the model converges to the data distribution, what objective the drifting dynamics are optimizing, or how the method should be formulated on non-Euclidean domains.

Contributions. We address this by introducing *kernel-gradient drifting*. Instead of using the kernel only to weight Euclidean displacements, we let the kernel itself define the direction of motion through its gradient. This recovers ordinary Gaussian drifting as a special case, while providing a more general and coherent formulation of drifting beyond Euclidean Gaussian kernels. The key consequence is that drifting is no longer tied to a particular displacement rule or ambient Euclidean geometry, but becomes a kernel-defined gradient flow. In particular, this re-formulation extends the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

score-based interpretation to general kernels: the resulting drift is exactly the difference between the scores of kernel-smoothed data and model distributions. This gives identifiability whenever the smoothing kernel is characteristic, and yields a variational interpretation of the drifting dynamics as descent of a smoothed Kullback–Leibler divergence. Because kernel gradients are intrinsic tangent vectors, the same construction also extends naturally to Riemannian manifolds, and allows for discrete data generation through the Fisher–Rao geometry of the probability simplex.

We evaluate our ideas across three complementary settings [1]. Controlled synthetic experiments test whether the gradient drift direction improves performance. The Earth event modeling on $\mathbb{S}^2$ tests how different kernel choices behave on real spherical data, and promoter DNA generation and molecule generation (through SMILES strings) tests whether the simplex construction gives a practical route to one-step discrete generation. Across these settings, kernel-gradient drifting provides a principled extension of drifting beyond the Gaussian Euclidean case, while remaining teacher-free and competitive with one-step flow and distillation-based baselines.

Figure 1. Riemannian kernel-gradient drifting on $\mathbb{S}^2$. Samples are transported along tangent vectors and mapped back to the sphere with the exponential map, allowing drifting to respect the geometry of the data support.

2. Drifting and the Gaussian limitation

Drifting models. Drifting models (Deng et al., 2026) train one-step generators by moving the iterative refinement usually performed at sampling time into the training procedure. Let p be a target distribution on $\mathbb{R}^d$, and let $f_\theta : \mathbb{R}^k \to \mathbb{R}^d$ be a generator with pushforward distribution $q_\theta = (f_\theta)_\# \mathcal{N}(0, I)$. Given a positive normalized kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{>0}$, drifting defines a discrepancy field

$$V_{p,q_\theta}(x) = V_p^+(x) - V_{q_\theta}^-(x),$$

where

$$V_p^+(x) = \underbrace{\frac{\mathbb{E}_{y\sim p}[k(x,y)(y-x)]}{\mathbb{E}_{y\sim p}[k(x,y)]}}_{\text{attraction to data}}, \qquad V_{q_\theta}^-(x) = \underbrace{\frac{\mathbb{E}_{x'\sim q_\theta}[k(x,x')(x'-x)]}{\mathbb{E}_{x'\sim q_\theta}[k(x,x')]}}_{\text{self-repulsion}}. \tag{1}$$

[1]Code available at https://anonymous.4open.science/r/kernel-grad-drift-B4D5.

Thus, V_{p,q_θ} has an attractive–repulsive structure: it *attracts* samples towards regions where the data distribution is locally better represented than the current model distribution, while *repelling* model samples from each other to ensure coverage. During training, a noise sample $\epsilon \sim \mathcal{N}(0, I)$ is mapped to $x = f_\theta(\epsilon)$, and transported to

$$\widetilde{x} = x + \eta V_{p,q_\theta}(x), \tag{2}$$

where $\eta \in \mathbb{R}$ is a step-size. The generator f_θ is trained via a stop-gradient loss, whose *value* is

$$\begin{aligned}\mathcal{L}_{\text{drift}}(\theta) &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \left\| f_\theta(\epsilon) - \text{sg}(\widetilde{x}) \right\|_2^2 \\ &= \eta^2 \mathbb{E}_{x\sim q_\theta} \left\| V_{p,q_\theta}(x) \right\|_2^2 .\end{aligned} \tag{3}$$

The transported samples are therefore treated as a frozen target, and the next generator trains on this transported sample cloud. (Turan & Ovsjanikov, 2026) shows that applying stop-gradient is algorithmically necessary for ensuring convergence of the method.

Identifiability. For drifting to define a meaningful population objective, vanishing drift should imply that the model distribution has matched the data distribution. Otherwise, the generator could reach a fixed point even when $q_\theta \neq p$. We refer to this property as *identifiability*: the drift operator V_{p,q_θ} is identifiable if

$$V_{p,q_\theta}(x) = 0 \ \text{ for all } x \qquad \Longrightarrow \qquad p = q_\theta. \tag{4}$$

The Gaussian case. Concurrent work has shown that drifting has a clean score-based interpretation for Gaussian kernels (Lai et al., 2026; Turan & Ovsjanikov, 2026). Consider

$$k_\tau(x, y) = \frac{1}{(2\pi\tau^2)^{d/2}} \exp\left(-\frac{\|x-y\|_2^2}{2\tau^2} \right).$$

for a temperature value $\tau > 0$. For this kernel,

$$\nabla_x k_\tau(x, y) = \frac{1}{\tau^2} k_\tau(x, y)(y - x).$$

Hence the displacement direction used in (1) is proportional to the gradient of the kernel itself. If we define the Gaussian convolved density

$$\widehat{p}_\tau(x) = \mathbb{E}_{y\sim p}[k_\tau(x, y)], \qquad \widehat{q}_{\theta,\tau}(x) = \mathbb{E}_{x'\sim q_\theta}[k_\tau(x, x')],$$

then

$$V_p^+(x) = \tau^2 \nabla_x \log \widehat{p}_\tau(x) \qquad V_{q_\theta}^-(x) = \tau^2 \nabla_x \log \widehat{q}_{\theta,\tau}(x).$$

Consequently, Gaussian drifting satisfies

$$V_{p,q_\theta}(x) = \tau^2 \left(\nabla_x \log \widehat{p}_\tau(x) - \nabla_x \log \widehat{q}_{\theta,\tau}(x) \right)$$

and the training objective reduces to one-step smoothed score matching in reverse-Fisher form. Vanishing drift implies equality of the smoothed scores, and injectivity of the Gaussian kernel yields $p = q_\theta$. This gives Gaussian drifting an identifiable population fixed point. This formulation also reduces the drifting objective to the *score-difference* setting described in (Weber, 2023), connecting kernel drifting methods to previous generative models literature.

3. Kernel-gradient drifting

The previous section shows that Gaussian drifting is special because the kernel-weighted displacement direction coincides with a smoothed score direction. This suggests that the natural object is not the Euclidean displacement $y - x$ itself, but the direction induced by the kernel. In this section, we use this observation to define *kernel-gradient drifting*.

3.1. What goes wrong in the original formulation?

The Gaussian result also exposes the limitation of the original formulation. Equation (1) combines two separate design choices: the kernel $k(x, y)$, which determines how nearby samples are weighted, and the Euclidean displacement $y - x$, which determines the direction of motion. For Gaussian kernels these two choices are compatible, because the displacement-weighted update is proportional to a kernel-gradient direction. As a consequence, the direction of the drift field is exactly the mean-shift direction of the smoothed density ratio $\hat{p}_\tau / \hat{q}_{\theta,\tau}$ and, thus, the model moves samples toward regions where the smoothed data distribution is locally under-represented relative to the smoothed model distribution. For a general kernel, however,

$$k(x, y)(y - x) \not\propto \nabla_x k(x, y)$$

and the induced direction no longer coincides with the mean-shift direction. Instead, it is biased toward distant neighbors, whose influence is amplified by the distance-dependent directional term (see Figure 2 and Appendix C for details). This perturbation becomes more pronounced at larger temperatures, since farther points contribute more strongly, and on manifolds with high curvature. It is precisely this perturbation that breaks the identifiability property for non-Gaussian kernels. This is especially relevant because non-Gaussian kernels, such as Laplace kernels, often perform well in practice. Turan & Ovsjanikov (2026) attribute this behavior to the spectral properties of the Laplace kernel which make its Fourier modes decay polynomially rather than exponentially, unlike in the Gaussian case, which helps preserve finer details of the distribution.

3.2. The kernel-gradient drift field

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{>0}$ be a positive normalized kernel differentiable in its first argument. Define the *kernel-gradient drift*

$$G_p^+(x) = \frac{\mathbb{E}_{y\sim p}[\nabla_x k(x, y)]}{\mathbb{E}_{y\sim p}[k(x, y)]},$$
$$G_{q_\theta}^-(x) = \frac{\mathbb{E}_{x'\sim q_\theta}[\nabla_x k(x, x')]}{\mathbb{E}_{x'\sim q_\theta}[k(x, x')]}. \tag{5}$$

We then define the kernel-gradient drift as

$$V_{p,q_\theta}^\nabla(x) = G_p^+(x) - G_{q_\theta}^-(x) = \frac{\mathbb{E}_{y\sim p}[\nabla_x k(x, y)]}{\mathbb{E}_{y\sim p}[k(x, y)]} - \frac{\mathbb{E}_{x'\sim q_\theta}[\nabla_x k(x, x')]}{\mathbb{E}_{x'\sim q_\theta}[k(x, x')]}. \tag{6}$$

When the kernel is Gaussian, this recovers the original drifting field up to the constant temperature factor τ^2. Training proceeds exactly as described in Section 2.

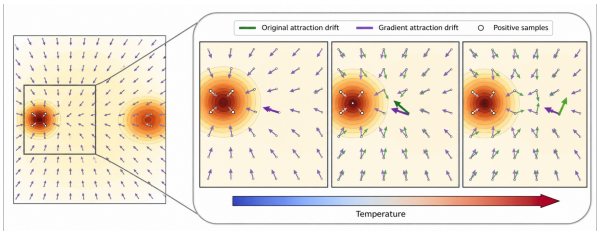

Figure 2. Attraction drift direction induced by the Laplacian kernel. Under the original formulation, shown by the green arrows, the drift does not point towards the region of highest density ratio between p and q_θ. Instead, its direction is biased toward distant samples. This phenomenon becomes more pronounced as the temperature increases, since distant samples receive greater weight. At small temperatures, the two directions are approximately aligned.

3.3. Smoothed score-ratio interpretation

The main advantage of the gradient formulation is that it admits a score-based interpretation. Define the kernel-smoothed density

$$\widehat{p}_k(x) = \int k(x, y)\, p(y)\, dy.$$

Note that when working in $\mathbb{R}^d$, this corresponds to applying a convolution operator to p and q_θ. This is not true on a curved manifold in general, as most manifolds don't have a group structure. Then, under standard regularity assumptions allowing differentiation under the integral sign (see Appendix D for details), we have that $G_p^+(x) = \nabla_x \log \widehat{p}_k(x)$. The same identity applied to negative samples coming from q_θ leads to

$$V_{p,q_\theta}^\nabla(x) = \nabla_x \log \widehat{p}_k(x) - \nabla_x \log \widehat{q}_{\theta,k}(x) = \nabla_x \log \frac{\widehat{p}_k(x)}{\widehat{q}_{\theta,k}(x)}.$$

This result is formalized in Proposition 3.1. See Appendix D for the formal statement, assumptions, and proof.

Proposition 3.1 (Kernel-gradient drifting is smoothed score-ratio matching)**.** *Let k be a positive, normalized kernel differentiable in its first argument, and assume that differentiation may be exchanged with integration. Then the kernel-gradient drift defined in (6) satisfies*

$$V^{\nabla}_{p,q_\theta}(x) = \nabla_x \log \frac{\widehat{p}_k(x)}{\widehat{q}_{\theta,k}(x)}.$$

Consequently,

$$\mathcal{L}^{\nabla}_{\text{drift}}(\theta) = \eta^2 \mathbb{E}_{x\sim q_\theta} \left\| \nabla_x \log \widehat{p}_k(x) - \nabla_x \log \widehat{q}_{\theta,k}(x) \right\|_2^2.$$

This connects kernel-gradient drifting models to the score-difference and mean-shift framework discussed in Section 2, for any choice of kernel satisfying assumptions of Proposition 3.1.

3.4. Identifiability with characteristic kernels

The score-ratio identity gives a direct route to identifiability. If the drift vanishes, then the smoothed data and model distributions have the same score. As we are working with normalized kernels, this implies that the smoothed densities are equal. The remaining question is whether equality of smoothed distributions implies equality of the original distributions. This is exactly the property of *characteristic* kernels.

Definition 3.2 (Characteristic kernel (Fukumizu et al., 2008))**.** A kernel k is characteristic on a class of probability distributions if the smoothing map

$$\pi \mapsto \widehat{\pi}_k, \qquad \widehat{\pi}_k(x) = \int k(x,y)\,\pi(y)\,dy,$$

is injective. Equivalently,

$$\widehat{p}_k(x) = \widehat{q}_k(x) \quad \text{for all } x \quad \Longrightarrow \quad p = q.$$

Proposition 3.3 (Identifiability)**.** *Assume that the kernel k is characteristic and satisfies the assumptions from Proposition 3.1. Then*

$$V^{\nabla}_{p,q_\theta}(x) = 0 \quad \textit{for all } x \qquad \Longrightarrow \qquad p = q_\theta.$$

Refer to Appendix D for the details on the proof. This result clarifies why an *anti-symmetric* drift is important. If one considers an imbalanced field of the form $c\,G^+_p(x) - d\,G^-_q(x)$ for $c \neq d$, then vanishing drift would imply $c\,\nabla \log \widehat{p}_k(x) = d\,\nabla \log \widehat{q}_k(x)$, which does not imply $\widehat{p}_k = \widehat{q}_k$ when $c \neq d$. Thus, the balanced difference in (6) is not merely a convention, but a necessary condition for distributional matching. This provides further insights into why Deng et al. (2026) observed a catastrophic behavior when not working with anti-symmetric kernels.

3.5. Smoothed-KL descent

The score-ratio identity also gives an optimization interpretation of the drifting dynamics.

Proposition 3.4 (Smoothed-KL descent)**.** *The smoothened gradient drift direction gives the steepest infinitesimal decrease of the KL-divergence between the smoothed distributions $\hat{p}(x)$ and $\hat{q}_\theta(x)$. This corresponds to the Wasserstein gradient flow of $D_{\mathrm{KL}}(\hat{q}_{\theta,\tau} \,\|\, \hat{p}_\tau)$.*

The proof follows from (Weber, 2023) and is included in Appendix D for completeness. This result should be read together with Proposition 3.3. The gradient-drifting monotonically descends a smoothed distributional objective, and characteristic kernels ensure that matching the smoothed distributions identifies the original distributions.

4. Kernels, geometry, and discrete data

The previous section shows that kernel-gradient drifting is identifiable when the smoothing kernel is characteristic. This turns kernel choice into a central design decision. In Euclidean space, many familiar kernels are characteristic. On curved manifolds, however, kernel construction is more delicate, and it has been approached in many different ways depending on the space on which the kernel is defined: in the ambient space (Ozakin & Gray, 2009; Bae & Polonik, 2026), in the Reproducing Kernel Hilbert Space induced by the manifold heat kernel (Caseiro et al., 2012), in the tangent space and then projected back to the manifold with the exponential map (Kim & Park, 2013; Wu & Wu, 2021), or in the manifold itself, either through geodesic distances (Pelletier, 2005) or spectral kernels (Cleanthous et al., 2020). This section focuses on spectral kernels and uses our gradient formulation to extend drifting to Riemannian manifolds and categorical data.

4.1. Which kernels are valid?

In Euclidean space, translation-invariant kernels with non-vanishing Fourier transform are characteristic. On manifolds, the situation is more subtle. A tempting construction is the family of exponential radial geodesic kernels

$$k_g(x,y) = \exp(-\tau d_g(x,y)), \qquad \tau > 0.$$

Although this resembles the Euclidean Gaussian kernel, it is not positive definite in general on curved manifolds. In fact, these kernels are positive definite for all temperatures if and only if the space is flat (Feragen et al., 2015; Jayasumana et al., 2014; Steinert et al., 2025). Thus, geodesic radial kernels can be useful in practice, especially when the distance is known in closed form and cheap to compute, but they do

not provide the identifiability guarantees of Proposition 3.3. This further motivates the gradient reformulation of the drifting field: on general geometries, the original formulation is not identifiable over all temperature values for any geodesic radial kernel.

4.2. Spectral and Matérn kernels

Let $(, g)$ be a compact Riemannian manifold, and let $\{(\lambda_n, \phi_n)\}_{n=0}^{\infty}$ denote the eigenpairs of the Laplace–Beltrami operator. Spectral kernels are defined by

$$k(x, x') = \sum_{n=0}^{\infty} f(\lambda_n)\, \phi_n(x)\phi_n(x'), \tag{7}$$

where $f : [0, \infty) \to \mathbb{R}_+$ is a spectral density. If $f(\lambda_n) > 0$ for every eigenvalue, then the kernel is characteristic on the corresponding function class. A particularly useful family is given by Matérn kernels on manifolds (Borovitskiy et al., 2020). For parameters $\sigma^2, \tau, \nu > 0$, one can define

$$k_\nu(x, x') = \sigma^2 C_\nu \sum_{n=0}^{\infty} \left(\frac{2\nu}{\tau^2} + \lambda_n\right)^{-\left(\nu + \frac{d}{2}\right)} \phi_n(x)\phi_n(x'), \tag{8}$$

$$k_\infty(x, x') = \sigma^2 C_\infty \sum_{n=0}^{\infty} \exp\left(-\frac{\tau^2}{2}\lambda_n\right) \phi_n(x)\phi_n(x'). \tag{9}$$

The constants C_ν and C_∞ are normalization constants. The limit k_∞ corresponds to the heat kernel. Since all spectral coefficients are strictly positive, these kernels are characteristic under the assumptions above, and therefore yield identifiable kernel-gradient drifts. In practice, spectral kernels can be approximated by truncated expansion. On simple manifolds such as spheres, the eigenfunctions are available in closed form through spherical harmonics. On meshes, graphs, and more general geometries, one can use numerical eigenfunctions or finite-dimensional feature approximations, as implemented in packages for geometric kernels such as (Mostowsky et al., 2025). This provides a practical route to kernel-gradient drifting without relying on geodesic computations.

4.3. Intrinsic drifting on Riemannian manifolds

The gradient formulation extends naturally to Riemannian manifolds. Let $(, g)$ be a Riemannian manifold, and let $f_\theta : \mathbb{R}^k \to$ be a generator whose pushforward distribution is $q_\theta = (f_\theta)_{\#}\mathcal{N}(0, I)$. Given a kernel $k : \times \to \mathbb{R}_{>0}$, define

$$G_p^+(x) = \frac{\mathbb{E}_{y\sim p}[\nabla_x k(x, y)]}{\mathbb{E}_{y\sim p}[k(x, y)]}, \qquad G_{q_\theta}^-(x) = \frac{\mathbb{E}_{x'\sim q_\theta}[\nabla_x k(x, x')]}{\mathbb{E}_{x'\sim q_\theta}[k(x, x')]}. \tag{10}$$

where ∇_x is the Riemannian gradient. The *Riemannian kernel-gradient drift* is then

$$V_{p,q_\theta}^{\nabla}(x) = G_p^+(x) - G_{q_\theta}^-(x), \qquad V_{p,q_\theta}^{\nabla}(x) \in T_x. \tag{11}$$

Generated samples are transported using the exponential map, $\widetilde{x} =_x \left(\eta V_{p,q_\theta}^{\nabla}(x)\right)$. The corresponding *value* of the stop-gradient objective is then

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[d_g\big(f_\theta(\epsilon), \mathrm{sg}\big[_{f_\theta(\epsilon)}\big(\eta V_{p,q_\theta}^{\nabla}(f_\theta(\epsilon))\big)\big]\big)^2\right] \\ &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left\|_{f_\theta(\epsilon)}\big(\mathrm{sg}\big[_{f_\theta(\epsilon)}\big(\eta V_{p,q_\theta}^{\nabla}(f_\theta(\epsilon))\big)\big]\big)\right\|_g^2 \\ &= \eta^2 \mathbb{E}_{x\sim q_\theta}\left\|V_{p,q_\theta}^{\nabla}(x)\right\|_g^2,\end{aligned} \tag{12}$$

where we assume $\eta\|V_{p,q_\theta}^{\nabla}(x)\|_g < \mathrm{inj}(x)$, with inj denoting the injectivity radius at point x. Note that the Euclidean formulation is recovered by taking $= \mathbb{R}^d$ with the standard Euclidean metric, for which $_x(v) = x + v$ and $_x(y) = y - x$.

4.4. Discrete data via Fisher–Rao simplex geometry

The Riemannian formulation also gives a route to discrete generation. Following the framework from Davis et al. (2024), consider the d-dimensional probability simplex

$$\Delta^d = \left\{x \in \mathbb{R}^{d+1} \,\middle|\, \mathbf{1}^\top x = 1,\ x \geq 0\right\}.$$

Then, a categorical distribution $p(x)$ over $K = d + 1$ categories can be represented in Δ^d by placing a Dirac mass δ_i with weight p_i at each vertex $i \in \{0, \ldots, d\}$. Denote $\mathring{\Delta}^d := \left\{x \in \Delta^d \mid x > 0\right\}$ for the relative interior of the simplex. We can endow this manifold with the Fisher-Rao metric to obtain a *statistical manifold*, such that there exists an isometric map $\phi : \mathring{\Delta}^d \to S_+^d$, from the simplex to the positive orthant of the hypersphere. Thus, to generate discrete data, one can just implement the Riemannian drifting model on the positive orthant of the sphere. For more details on this construction, refer to Appendix B.

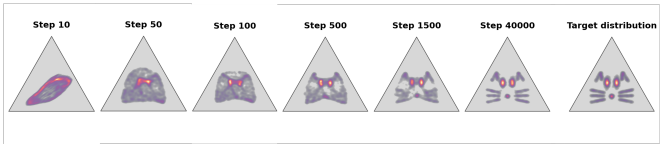

Figure 3. Evolution of the 'cat' distribution on the probabiblity simplex during training.

5. Experiments

Goal of the experiments. Our experiments evaluate three aspects of kernel-gradient drifting. First, we test whether the gradient direction improves synthetic data generation and what is the effect of the Matérn smoothness parameter ν. Second, we assess the method on non-Euclidean real-world data under different kernel and geometry choices.

Finally, we assess whether the Fisher–Rao geometry on the probability simplex provides a practical route to one-step discrete generation.

Synthetic experiments. We use controlled synthetic experiments to isolate the effect of drift direction and geometry. *Base* versus *Gradient* compares displacement-based and kernel-gradient drift, while *Euclidean* versus *Manifold* compares ambient updates with intrinsic manifold updates. We evaluate checkerboard and swissroll targets on $\mathbb{S}^2$ and $\mathbb{H}^2$, projecting samples to the corresponding 2D charts for metric computation. Note that for the Euclidean formulations we project the samples to the manifold before evaluation.

Table 1 shows that the gradient formulation improves over the base formulation in almost all controlled comparisons. The manifold formulation gives further gains in some cases, especially on checkerboard targets, though less consistently than the gradient correction.

Table 1. Drift quality on checkerboard and swissroll targets across spherical and hyperbolic 2D geometries using a Laplace kernel. Results reported as mean ± standard deviation over 3 seeds. *SW* denotes sliced Wasserstein-2 distance, and *Tile* is black-square accuracy on checkerboard targets.

Dataset	Geom.	Metric	Base	Grad.
Checkerboard	Sphere	Euc. SW ↓	0.031±0.009	0.012±0.000
		Man. SW ↓	0.032±0.008	**0.011±0.002**
		Euc. Tile ↑	0.76±0.03	0.86±0.01
		Man. Tile ↑	0.80±0.04	**0.87±0.01**
	Hyper.	Euc. SW ↓	0.026±0.003	**0.025±0.001**
		Man. SW ↓	0.063±0.022	0.037±0.013
		Euc. Tile ↑	0.79±0.01	0.85±0.02
		Man. Tile ↑	0.80±0.02	**0.90±0.01**
Swissroll	Sphere	Euc. SW ↓	0.018±0.002	**0.014±0.002**
		Man. SW ↓	0.026±0.000	**0.014±0.002**
	Hyper.	Euc. SW ↓	0.041±0.001	**0.030±0.003**
		Man. SW ↓	0.042±0.007	0.044±0.015

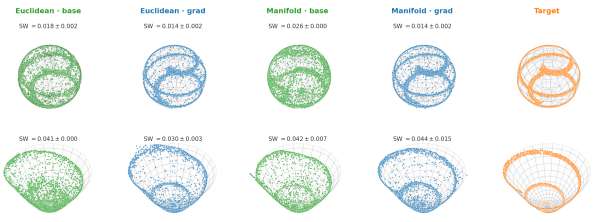

Figure 4. Generated distributions for the swissroll dataset on spherical and hyperboloid manifolds.

To assess the effect of kernel choice, we evaluate Euclidean swissroll generation with Matérn kernels of varying smoothness ν. Results are shown in Figure 5. The sweep confirms that performance of the two methods is equivalent for the Gaussian kernel, but that the gradient direction matters for the non-Gaussian ones, such as Laplace ($\nu = 0.5$).

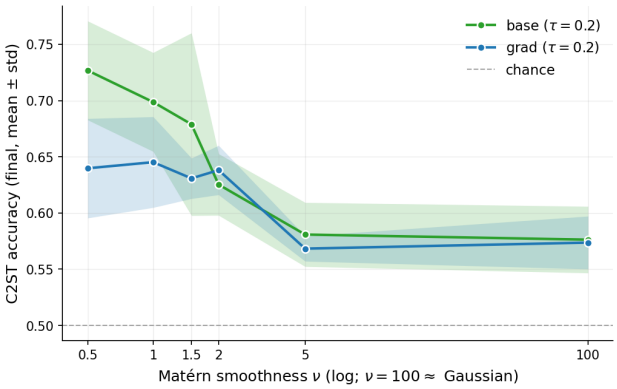

Figure 5. Classifier Two-Sample Test (C2ST) accuracy on the Euclidean Swiss-roll generation task for different Matérn smoothness values ν. Lower is better, with chance performance indicated at 0.5.

Geospatial data on the sphere. We next evaluate on the geospatial Earth events benchmark on $\mathbb{S}^2$ (Mathieu & Nickel, 2020). This benchmark contains real-world distributions supported on the sphere, and tests whether the geometric formulation is useful beyond controlled synthetic examples.

Table 2 compares the Euclidean Laplace drifting baseline with our gradient, spherical, and spectral variants. Since Gaussian displacement-based drifting is equivalent to its kernel-gradient form, we report only the Gaussian gradient variants. Here, *Gradient* denotes the kernel-gradient formulation, while *Spherical* uses the intrinsic geometry of $\mathbb{S}^2$ for distances and updates. The results show that our kernel, direction, and geometry choices improve performance on curved real-world data, obtain the best Maximum Mean Discrepancy (MMD) on all four datasets. Additional metrics are reported in Appendix E.

Table 2. MMD results on the four Earth datasets. Lower is better. The first row shows the vanilla Euclidean Laplace Drift baseline from (Deng et al., 2026); the remaining rows show our gradient, spherical, and spectral extensions. Best, second-best, and third-best results are highlighted per dataset.

Method	Vol. ↓	Eq. ↓	Fire ↓	Flood ↓
Euclidean Laplace (Deng et al., 2026)	0.146	0.044	0.036	0.064
Euclidean Laplace Gradient (ours)	0.128	0.038	0.030	0.067
Euclidean Gaussian Gradient (ours)	0.143	0.047	0.048	0.072
Spherical Laplace (ours)	0.113	0.043	0.049	0.064
Spherical Laplace Gradient (ours)	0.113	0.056	0.047	**0.053**
Spherical Gaussian Gradient (ours)	**0.112**	0.158	0.039	0.058
Spectral ($\nu = 2.5$) (ours)	0.126	**0.037**	**0.029**	0.070

Discrete data generation. Finally, we evaluate discrete generation on promoter DNA and QM9. For DNA, we generate sequences over the 4 nucleotides and report 6-mer correlation. For QM9, we follow Park et al. (2026), generating SMILES strings and reporting validity and uniqueness. In

both settings we compare against one-step flow-distilled baselines. See Appendix F for setup details. The results show that kernel-gradient drifting is a viable teacher-free approach to one-step categorical generation. On promoter DNA (Table 3), the kernel gradient improves over vanilla spherical drifting and reduces the gap with flow-distilled baselines. On QM9 (Table 4), our spherical gradient formulation substantially outperforms the Euclidean baseline and approaches the strongest distilled baselines, despite being teacher-free.

Table 3. Promoter DNA generation results using 6-mer correlation (↑). We report only the 1-NFE results from previous flow-based methods.

Method	6-mer Corr. ↑
Flow distillation	
E-RMF + v-pred (Woo et al., 2026)	**0.96**
E-RMF + x-pred (Woo et al., 2026)	0.96
S-RMF + v-pred (Woo et al., 2026)	0.93
S-RMF + x-pred (Woo et al., 2026)	0.84
L-RMF + v-pred (Woo et al., 2026)	0.85
L-RMF + x-pred (Woo et al., 2026)	0.88
Drifting	
Spherical Laplace	0.88
Spherical Laplace Gradient	0.89
Spherical Gaussian Gradient	**0.90**

Table 4. QM9 molecule generation results for one-step methods. Higher is better. Baseline percentages derived from $N = 1{,}024$ batch samples.

Method	Valid ↑	Unique ↑
Flow distillation		
UDLM (+DCD) (Park et al., 2026)	31.5	31.3
PairFlow (+DCD) (Park et al., 2026)	**44.3**	**44.1**
UDLM (+ReDi) (Park et al., 2026)	5.8	5.8
PairFlow (+ReDi) (Park et al., 2026)	35.3	35.1
Drifting		
Euclidean Laplace (Deng et al., 2026)	22.0	40.0
Spherical Laplace Gradient (ours)	**38.9**	**44.1**

Concurrent works on gradient-based drifting models. The limitations of Gaussian-kernel drifting have also motivated concurrent theoretical work. Franz et al. (2026) analyze drifting from an optimization perspective and show that the original (Deng et al., 2026) style drift is conservative only for the Gaussian kernel. Therefore, it cannot be written as the gradient of any scalar potential, in general. Cao et al. (2026) derive a kernel density estimation framework that unifies drifting models as gradient flows of divergence functionals. Our work differs in both emphasis and scope. Rather than developing a divergence-level theoretical unification, we focus on aligning drift with the mean-shift direction and evaluate our approach across a range of real-world experiments, whereas their work has not been studied in an experimental setting beyond 2D toy examples.

6. Conclusion

We introduced *kernel-gradient drifting*, a generalization of the original drifting field in which the kernel defines the direction of sample motion through its gradient. This perspective gives a more coherent picture of drifting models: it recovers Gaussian drifting as a special case, extends the score-based interpretation to general kernels, and connects the drifting dynamics to descent of a smoothed Kullback–Leibler divergence. This construction naturally generalizes to data on manifolds and to discrete data generation. Empirically, experiments on synthetic data, spherical Earth event modeling, promoter DNA and molecule generation show that kernel-gradient drifting provides a flexible, teacher-free route to one-step generation. However, there are also some limitations to our work. For instance, our work only focuses on kernel methods, specifically on Matérn kernels, and we found the method to be sensitive to hyperparameter choice. An interesting avenue to explore is to extend the analysis to a broader kernel class and to perform a more systematic study of training stability, which may help close the gap between our method and state-of-the-art.

References

Avdeyev, P., Shi, C., Tan, Y., Dudnyk, K., and Zhou, J. Dirichlet diffusion score model for biological sequence generation. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

Bae, E. and Polonik, W. Kernel smoothing on manifolds. *arXiv preprint arXiv:2601.16777*, 2026.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025a.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. How to build a consistency model: Learning flow maps via self-distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. Matérn gaussian processes on riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Cao, J., Wei, Z., and Liu, Y. Gradient flow drifting: Generative modeling via wasserstein gradient flows of kde-approximated divergences. *arXiv preprint arXiv:2603.10592*, 2026.

Caseiro, R., Henriques, J., Martins, P., and Batista, J. Semi-intrinsic mean shift on riemannian manifolds. In *Computer Vision, ECCV*, volume 7572, pp. 342–355, 10 2012.

Cleanthous, G., Georgiadis, A. G., Kerkyacharian, G., Petrushev, P., and Picard, D. Kernel and wavelet density estimators on manifolds and more general metric spaces. *Bernoulli Society for Mathematical Statistics and Probability*, 26, 2020.

Davis, O., Kessler, S., Petrache, M., Ceylan, I. I., Bronstein, M. M., and Bose, J. Fisher flow matching for generative modeling over discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Davis, O., Albergo, M. S., Boffi, N. M., Bronstein, M. M., and Bose, A. J. Generalised flow maps for few-step generative modelling on riemannian manifolds. *arXiv preprint arXiv:2510.21608*, 2025.

Deng, M., Li, H., Li, T., Du, Y., and He, K. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Feragen, A., Lauze, F., and Hauberg, S. Geodesic exponential kernels: When curvature and linearity conflict. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3032–3042, 2015.

Franz, L., Hoffmann, S., and Martius, G. Drifting fields are not conservative. *arXiv preprint arXiv:2604.06333*, 2026.

Fukumizu, K., Gretton, A., Schölkopf, B., and Sriperumbudur, B. K. Characteristic kernels on groups and semigroups. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Geng, Z., Lu, Y., Wu, Z., Shechtman, E., Kolter, J., and He, K. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.0201*, 2025.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Gorham, J. and Mackey, L. Measuring sample quality with stein's method. In *Advances in Neural Information Processing Systems*, 2015.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Hu, T., Li, Z., Luo, W., Sun, J., Zhang, Z., and Zhang, S. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *arXiv preprint arXiv:2305.18455*, 2023.

Jayasumana, S., Hartley, R., Salzmann, M., li, H., and Harandi, M. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 2014.

Kim, Y. T. and Park, H. S. Geometric structures arising from kernel density estimation on riemannian manifolds. *Journal of Multivariate Analysis*, 114:112–126, 02 2013. doi: 10.1016/j.jmva.2012.07.006.

Lai, C.-H., Nguyen, B., Murata, N., Takida, Y., Uesaka, T., Mitsufuji, Y., Ermon, S., and Tao, M. A unified view of drifting and score-based models, 2026.

Lee, J. M. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2012.

Lee, J. M. *Introduction to Riemannian Manifolds*, volume 176 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2018.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Mathieu, E. and Nickel, M. Riemannian continuous normalizing flows. In *Advances in Neural Information Processing Systems*, 2020.

Mostowsky, P., Dutordoir, V., Azangulov, I., Jaquier, N., Hutchinson, M. J., Ravuri, A., Rozo, L., Terenin, A., and Borovitskiy, V. The geometric kernels package: Heat and matérn kernels for geometric learning on manifolds, meshes, and graphs. *Journal of Machine Learning Research*, 2025.

Ozakin, A. and Gray, A. Submanifold density estimation. In *Advances in Neural Information Processing Systems*, 2009.

Park, M., Hwang, J., Yoo, S., Yeo, K., and Sung, M. Pairflow: Closed-form source-target coupling for few-step generation in discrete flow models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Pelletier, B. Kernel density estimation on riemannian manifolds. *Statistics & Probability Letters*, 73(3):297–304, 2005.

Roos, D., Davis, O., Eijkelboom, F., Bronstein, M., Welling, M., İsmail İlkan Ceylan, Ambrogioni, L., and van de Meent, J.-W. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Steinert, F., Said, S., and Mostajeran, C. Universal kernels via harmonic analysis on riemannian symmetric spaces. *arXiv preprint arXiv:2506.19245*, 2025.

Subbarao, R. and Meer, P. Nonlinear mean shift over riemannian manifolds. *International Journal of Computer Vision*, 2009.

Turan, E. and Ovsjanikov, M. Generative drifting is secretly score matching: a spectral and variational perspective. *arXiv preprint arXiv:2603.09936*, 2026.

Weber, R. M. The score-difference flow for implicit generative modeling. *Transactions on Machine Learning Research*, 2023.

Woo, D., Skreta, M., Park, S., Neklyudov, K., and Ahn, S. Riemannian meanflow. *arXiv preprint arXiv:2602.07744*, 2026.

Wu, H.-T. and Wu, N. Strong uniform consistency with rates for kernel density estimators with general kernels on manifolds. *Information and Inference: A Journal of the IMA*, 11, 2021.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. S. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/C7SC02664A.

Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., and Park, T. One-step diffusion with distribution matching distillation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zaghen, O., Eijkelboom, F., Pouplin, A., Liu, C., Welling, M., van de Meent, J.-W., and Bekkers, E. J. Riemannian variational flow matching for material and protein design. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhou, L., Parger, M., Haque, A., and Song, J. Terminal velocity matching. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ethics statement

Generative models, can have harmful societal consequences, most notably through the dissemination of disinformation, as well as by amplifying harmful stereotypes and implicit biases. In this work, we aim to advance understanding of drifting models, a specific class of generative models. Although such insights could eventually contribute to improving these models and thereby potentially increase opportunities for misuse, our research does not introduce ethical risks beyond those already associated with generative AI.

Reproducibility statement

We include the source code in our submission, which allows for reproducing the results. Our claims made in the main text are proven in the appendices. Experiment details can be found in Appendix E.

Disclosure of LLM Usage

We have used Large Language Models to polish writing on a sentence level.

A. Notation

In this section, we report the notations that are used in the paper and the rest of the appendix. For a more in depth introduction to Riemannian geometry refer to (Lee, 2018) or the Appendix section in (Zaghen et al., 2026).

Symbol	Name	Description
p	data distribution	Target distribution.
q_θ	model distribution	Pushforward distribution induced by the generator f_θ.
f_θ	generator	One-step map from latent noise to samples.
$\widehat{p}_k$	smoothed data density	Kernel-smoothed version of p.
$\widehat{q}_{\theta,k}$	smoothed model density	Kernel-smoothed version of q_θ.
$\mathrm{sg}(\cdot)$	stop-gradient	Operator that blocks gradients through its argument for backpropagation.
V_{p,q_θ}	drifting field	Original drift field by (Deng et al., 2026).
V^{∇}_{p,q_θ}	kernel-gradient drift	Drift field induced by kernel gradients.
D_{KL}	KL divergence	Kullback–Leibler divergence.
V^{+}_{p}	Original attraction drift	Original attraction drift by (Deng et al., 2026) that attracts models samples to data distribution p.
$V^{-}_{q_\theta}$	Original repulsion drift	Original repulsion drift by (Deng et al., 2026) that repulse models samples to from each other.
G^{+}_{p}	Gradient attraction drift	Attraction field of model samples towards data distribution p.
$G^{-}_{q_\theta}$	Gradient repulsion drift	Repulsion field of model samples from each other.
η	step size	Scale of the drift update.
τ	temperature	Kernel bandwidth.
ν	smoothness	Smoothness parameter of the Matérn kernel.
$\mathcal{M}$	manifold	Smooth Riemannian manifold.
g	metric	Riemannian metric on $\mathcal{M}$; $\langle\cdot,\cdot\rangle = g_p(\cdot,\cdot)$ and $\lvert\cdot\rvert = \sqrt{g_p(\cdot,\cdot)}$.
$T_x\mathcal{M}$	tangent space	Tangent space at $x \in \mathcal{M}$.
$\nabla^{\mathcal{M}}_x$	Riemannian gradient	Intrinsic gradient with respect to x.
$_x$	exponential map	Maps tangent vectors in $T_x\mathcal{M}$ back to $\mathcal{M}$. Within the injectivity radius, this map is a diffeomorphism.
$_x$	logarithm map	Inverse of $_x$ locally (within the injectivity radius). Maps points in $\mathcal{M}$ to tangent vectors in $T_x\mathcal{M}$).
$d_g(x,y)$	geodesic distance	Distance induced by the metric g.
$\mathrm{inj}(x)$	injectivity radius	The injectivity radius at $x \in \mathcal{M}$ is the supremum of all values $r > 0$ such that the exponential map from the ball $B_r(0) \subset T_x\mathcal{M}$ to the manifold $\mathcal{M}$ is injective.
$\mathbb{S}^2$	sphere	Unit 2-sphere used for spherical data.
$\mathbb{H}^2$	hyperboloid	Hyperbolic 2D manifold used in synthetic experiments.
Δ^d	simplex	Probability simplex for categorical data.
$\mathring{\Delta}^d$	simplex interior	Relative interior of the probability simplex.
$\mathbb{S}^d_+$	positive sphere orthant	Fisher–Rao embedding space of the simplex.

B. Discrete data generation on the probability simplex

In this section we provide further details on the generation of discrete data in the probability simplex with the Fisher-Rao metric. This construction is based on the work of (Davis et al., 2024). Consider the d-dimensional probability simplex

$$\Delta^d = \left\{ x \in \mathbb{R}^{d+1} \,\middle|\, \mathbf{1}^\top x = 1,\ x \geq 0 \right\},$$

and its relative interior

$$\mathring{\Delta}^d = \left\{ x \in \Delta^d \,\middle|\, x_i > 0 \ \ \forall i \right\}.$$

A categorical distribution over $K = d + 1$ categories can be represented as a point in Δ^d, where the vertices of the simplex correspond to hard categories. We can endow this manifold with a Riemannian metric: The Fisher-Rao metric. Under the convention

$$g_x(u, v) = \sum_{i=0}^{d} \frac{u_i v_i}{x_i},$$

the square-root map

$$\phi(x) = 2\,(\sqrt{x_0}, \ldots, \sqrt{x_d})$$

is an isometry from $\mathring{\Delta}^d$ to the positive orthant of the sphere of radius 2. Equivalently, under a rescaled Fisher–Rao metric, the map $x \mapsto \sqrt{x}$ isometrically identifies the simplex with the positive orthant of the unit sphere. For a sequence of length L with K categories per position, the continuous state space is the product manifold

$$\left(\mathbb{S}_+^{K-1}\right)^L.$$

The generator outputs points on this product manifold, kernel-gradient drifting is performed using the Riemannian formulation above, and final discrete samples are obtained by projecting each simplex point to a category, for example by taking the nearest vertex or the largest coordinate. Thus the same construction that makes drifting intrinsic on manifolds also provides a principled way to apply one-step drifting to discrete data.

C. What goes wrong with the original formulation?

The argument is very similar to the one presented in (Lai et al., 2026). Let's take the original drift formulation (1), generalized to the manifold setting (this just involves replacing the direction $y - x$, by ${}_x y$). So the drift then looks like

$$V_{p,q_\theta}(x) = \frac{\mathbb{E}_{y\sim p}[k(x,y)\,{}_x(y)]}{\mathbb{E}_{y\sim p}[k(x,y)]} - \frac{\mathbb{E}_{x'\sim q_\theta}[k(x,x')\,{}_x(x')]}{\mathbb{E}_{x'\sim q_\theta}[k(x,x')]}, \qquad V_{p,q_\theta}(x) \in T_x,$$

where $k(x, y)$ is a exponential geodesic radial kernel

$$k(x,y) = \exp\left(\phi\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right)$$

for a smooth function $\phi : \mathbb{R} \to \mathbb{R}$. Slightly abusing notation, we write $k\left(\frac{d_g^2(x,y)}{\tau^2}\right)$ for the geodesic radial kernel $k(x, y)$. The mean-shift direction is then

$$\nabla \log \hat{p}(x) = \frac{-2}{\tau^2} \frac{\mathbb{E}_{y\sim p}\left[k\left(\frac{d_g^2(x,y)}{\tau^2}\right)\phi'\left(\frac{d_g^2(x,y)}{\tau^2}\right)_x(y)\right]}{\mathbb{E}_{y\sim p}\left[k\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right]}$$

where we have used the identity

$$\nabla_x d_g^2(x,y) = -2\,{}_x(y).$$

Note that this construction also follows from the Kernel density estimation (KDE) on Riemannian manifolds (Subbarao & Meer, 2009). Now, we apply the formula

$$\mathbb{E}[f(x)g(x)] = \mathrm{Cov}(f(x), g(x)) + \mathbb{E}[f(x)]\mathbb{E}[g(x)]$$

to get

$$\nabla \log \hat{p}(x) = \frac{1}{\tau^2} \frac{\mathbb{E}_{y\sim p}\left[k\left(\frac{d_g^2(x,y)}{\tau^2}\right)_x(y)\right]}{\mathbb{E}_{y\sim p}\left[k\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right]} \underbrace{\mathbb{E}_{y\sim p}\left[\phi'\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right]}_{A_p(x)} + \frac{1}{\tau^2} \underbrace{\frac{\mathrm{Cov}\left(k\left(\frac{d_g^2(x,y)}{\tau^2}\right)_x(y), \phi'\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right)}{\mathbb{E}_{y\sim p}\left[k\left(\frac{d_g^2(x,y)}{\tau^2}\right)\right]}}_{\delta_p(x)}.$$

The same decomposition applies to q_θ, and we get

$$\nabla \log \frac{\hat{p}(x)}{\hat{q}_\theta(x)} = \tau^2\big(V_p^+(x)A_p(x) - V_{q_\theta}^-(x)A_{q_\theta}(x)\big) + \delta_p(x) - \delta_{q_\theta}(x).$$

Thus, in the Gaussian case, the drifting field is exactly the mean-shift direction, but this is not necessarily true for non-Gaussian kernels due to the extra factors A_p, A_{q_θ} and covariance $\delta_p, \delta_{q_\theta}$. The curvature of the manifold enters in through the exponential map. The larger the curvature, the larger is the Riemannian volume distortion, amplifying the mismatch between the mean-shift direction and the true smoothed score direction.

D. Theoretical guarantees

Lemma D.1. *Consider the gradient-drifting field V_{p,q_θ}^∇ as defined in (6). Then,*

$$p = q_\theta \quad \Rightarrow \quad V_{p,q_\theta}^\nabla(x) = 0 \quad \textit{for all } x.$$

Proof. The proof follows from (Deng et al., 2026). We add it for completeness. Since k is a positive kernel, it is easy to see that $V_{p,q_\theta}^\nabla(x)$ is an antisymmetric drifting field, i.e. $V_{p,q_\theta}^\nabla(x) = -V_{q_\theta,p}^\nabla(x)$ for all x. Then,

$$V_{p,q_\theta}^\nabla(x) = V_{q_\theta,p}^\nabla(x) = -V_{p,q_\theta}^\nabla(x).$$

□

Here we present the formal version of Proposition 3.1.

Proposition 3.1 (formal) Let $p :\to [0,\infty)$ and $q_\theta :\to [0,\infty)$ be probability densities with respect to the Riemannian volume measure $d\mathrm{vol}_g$, and consider the normalized kernel $k : \times \to (0,\infty)$. Assume that $\mathcal{M}$ is compact and that given an $x \in$ there exists an open neighborhood U of x such that:

1. The map $x \mapsto k(x,y)$ is C^1 on U for almost every $y \in \mathcal{M}$.
2. For every $x \in U$, the function $y \mapsto k(x,y)\,p(y)$ and $y \mapsto k(x,y)\,q_\theta(y)$ are integrable on .
3. There exists two integrable functions $g \in L^1\big(p\,d\mathrm{vol}_g\big)$ and $h \in L^1\big(q_\theta\,d\mathrm{vol}_g\big)$ such that for all $x \in U$ and for almost every y,
$$\|\nabla_x k(x,y)\|_g \le g(y) \qquad\qquad \|\nabla_x k(x,y)\|_g \le h(y).$$

Then, if

$$\hat{p}(x) = \int_M k(x,y)\,p(y)\,d\mathrm{vol}_g(y), \quad \hat{q}_\theta(x) = \int_M k(x,y)\,q_\theta(y)\,d\mathrm{vol}_g(y)$$

are the smoothed distributions, the drift operator can be expressed as

$$V_{p,q_\theta}(x) = \nabla_x \log \frac{\hat{p}(x)}{\hat{q}_\theta(x)}.$$

Proof. Let

$$\hat{p}(x) = \int_M k(x,y)\, p(y)\, d\mathrm{vol}_g(y).$$

Then,

$$\nabla \log \hat{p}(x) = \frac{\nabla_x \int k(x,y)\, p(y)\, d\mathrm{vol}_g(y)}{\int k(x,y)\, p(y)\, d\mathrm{vol}_g(y)} = \frac{\int \nabla_x k(x,y)\, p(y)\, d\mathrm{vol}_g(y)}{\int k(x,y)\, p(y)\, d\mathrm{vol}_g(y)} = \frac{\mathbb{E}_{y\sim p}[\nabla_x k(x,y)]}{\mathbb{E}_{y\sim p}[k(x,y)]}$$

Where in the second step we have used the Dominated Convergence Theorem to differentiate under the integral sign. Note that in manifolds, we should use a partition of unity to express the integral as a sum of integrals over $\mathbb{R}^n$, and then differentiate under the integral signs there (see Proposition 16.33 of (Lee, 2012) for details). The same procedure is repeated for q_θ to obtain

$$\nabla \log \hat{q}_\theta(x) = \frac{\mathbb{E}_{y\sim q_\theta}[\nabla_x k(x,y)]}{\mathbb{E}_{y\sim q_\theta}[k(x,y)]}.$$

Finally, we get

$$V_{p,q_\theta}(x) = \nabla \log \frac{\hat{p}(x)}{\hat{q}_\theta(x)}.$$

□

Note that exponential geodesic kernels satisfy Assumption 2 because they are bounded. However, these kernels are not necessarily smooth outside the cut locus as the distance function is not bijective there.

Below is the proof of Proposition 3.1.

Proposition 3.3 Assume that the kernel k is characteristic and satisfies the assumptions from Proposition 3.1. If

$$V^{\nabla}_{p,q_\theta}(x) = 0 \quad \text{for all } x,$$

then $p = q_\theta$.

Proof. By Proposition 3.1, $V^{\nabla}_{p,q} = 0$ implies

$$\nabla_x \log \widehat{p}_k(x) = \nabla_x \log \widehat{q}_k(x).$$

As k is a normalized kernel, this implies

$$\widehat{p}_k(x) = \widehat{q}_k(x).$$

Since k is characteristic, this implies $p = q$. □

Finally we present the proof of Proposition 3.4. Note that this proof only holds when the support of the distribution lies in the Euclidean space, as we are assuming the iterative updates are additive. For future work, we would like to extend this proof to manifolds.

Proposition 3.4 The smoothened gradient drift direction gives the steepest infinitesimal decrease of the KL-divergence between the smoothed distributions $\hat{p}(x)$ and $\hat{q}_\theta(x)$. This corresponds to the Wasserstein gradient flow of $D_{\mathrm{KL}}(\hat{q}_{\theta,\tau} \,\|\, \hat{p}_\tau)$.

Proof. The proof follows from (Weber, 2023). Consider the smoothed densities $p_\tau = p * k_\tau$ and $q_{\theta,\tau} = q_\theta * k_\tau$, where k_τ is a smoothing kernel. Equivalently, if $\xi \sim \mathcal{N}(0, I)$, $x = f_i(\xi) \sim q_\theta$, and $u \sim k_\tau$ is independent noise with density k_τ, then

$$z = x + u \sim q_{\theta,\tau}.$$

Consider the drift update in the smoothed variable $z_{i+1} = z_i + V_{p_\tau, q_{\theta,\tau}}(z_i)$, then the KL-divergence between $q_{\theta,\tau}$ and p_τ is

$$D_{\mathrm{KL}}(q_{\theta,\tau} \,\|\, p_\tau) = \mathbb{E}_{z\sim q_{\theta,\tau}} \left[\log q_{\theta,\tau}(z) - \log p_\tau(z)\right],$$

and varies according to its functional derivative,

$$\nabla_\varepsilon D_{\mathrm{KL}}(q_{\theta,\tau} \,\|\, p_\tau)\big|_{\varepsilon=0} = -\mathbb{E}_{z\sim q_{\theta,\tau}} \left[\mathrm{Tr}(A_{p_\tau} V_{p_\tau, q_{\theta,\tau}}(z))\right],$$

where

$$A_{p_\tau} f(z) = \nabla_z \log p_\tau(z)\, V_{p_\tau, q_{\theta,\tau}}(z)^\top + \nabla_z V_{p_\tau, q_{\theta,\tau}}(z)$$

is the Stein operator (Gorham & Mackey, 2015). By applying Stein's identity, we obtain

$$\begin{aligned}
\mathbb{E}_{z\sim q_{\theta,\tau}} \left[\mathrm{Tr}(A_{p_\tau} V_{p_\tau, q_{\theta,\tau}}(z))\right] &= \mathbb{E}_{z\sim q_{\theta,\tau}} \left[\nabla_z \log p_\tau(z)^\top V_{p_\tau, q_{\theta,\tau}}(z)\right] \\
&\quad - \mathbb{E}_{z\sim q_{\theta,\tau}} \left[\nabla_z \log q_{\theta,\tau}(z)^\top V_{p_\tau, q_{\theta,\tau}}(z)\right] \\
&= \mathbb{E}_{z\sim q_{\theta,\tau}} \left[\left(\nabla_z \log p_\tau(z) - \nabla_z \log q_{\theta,\tau}(z)\right)^\top V_{p_\tau, q_{\theta,\tau}}(z)\right],
\end{aligned}$$

which is the inner product of the score difference and the flow vector $V_{p_\tau, q_{\theta,\tau}}(z)$. Maximizing the reduction in the KL divergence corresponds to maximizing this inner product. Since the inner product of two vectors is maximized when they are parallel, choosing $V_{p_\tau, q_{\theta,\tau}}(z)$ to output a vector parallel to the score difference will decrease the KL divergence as fast as possible. □

Note that although $D_{\mathrm{KL}}(q_{\theta,\tau} \,\|\, p_\tau) \le D_{\mathrm{KL}}(q_\theta \,\|\, p)$, we have that $D_{\mathrm{KL}}(q_{\theta,\tau} \,\|\, p_\tau) = 0$ if and only if $p = q_\theta$ (as long as the kernel is characteristic). In addition, minimizing the KL divergence of the distributions directly is not usually feasible since it may diverge to infinity of p and q_θ can have unequal support, so looking at the smoothed distributions is the correct proxy.

E. Additional experimental results

E.1. Synthetic experiments

We provide qualitative visualizations of the synthetic experiments to complement the quantitative results in Table 1. Figure 6 compares the final generated distributions for the four drifting variants: Euclidean base, Euclidean gradient, manifold base, and manifold gradient. The visual results support the trends observed quantitatively. These are, the gradient formulation generally leads to more accurate samples, and incorporating the manifold structure can provide additional improvements in some settings. Figures 7, 8, 9, and 10 provide a more detailed view by showing the evolution of the generated distributions during training at selected checkpoints.

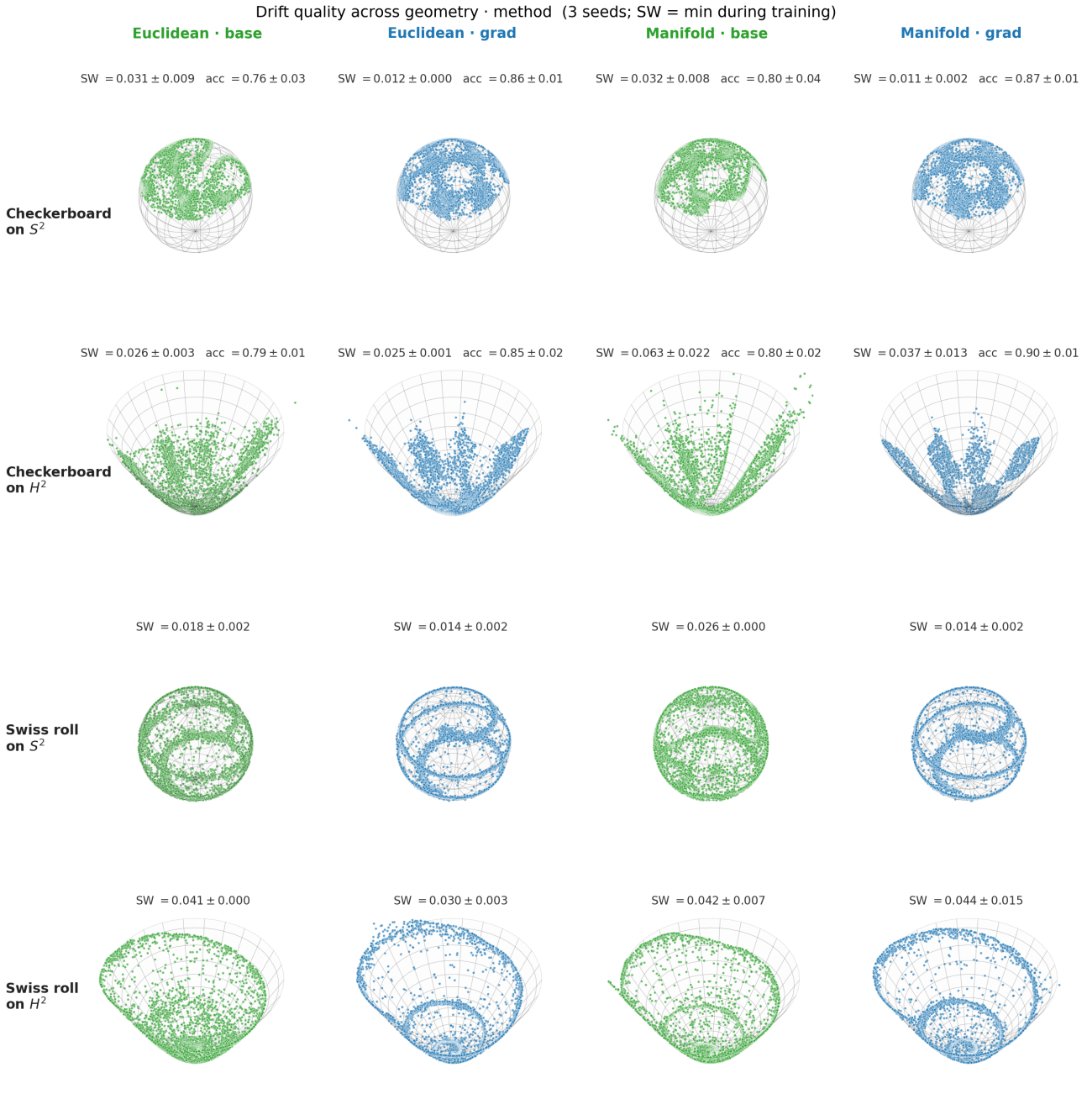

Figure 6. Final generated distributions for the checkerboard and swissroll targets on the sphere $\mathbb{S}^2$ and hyperboloid $\mathbb{H}^2$. We compare the four drifting variants considered in the toy experiments.

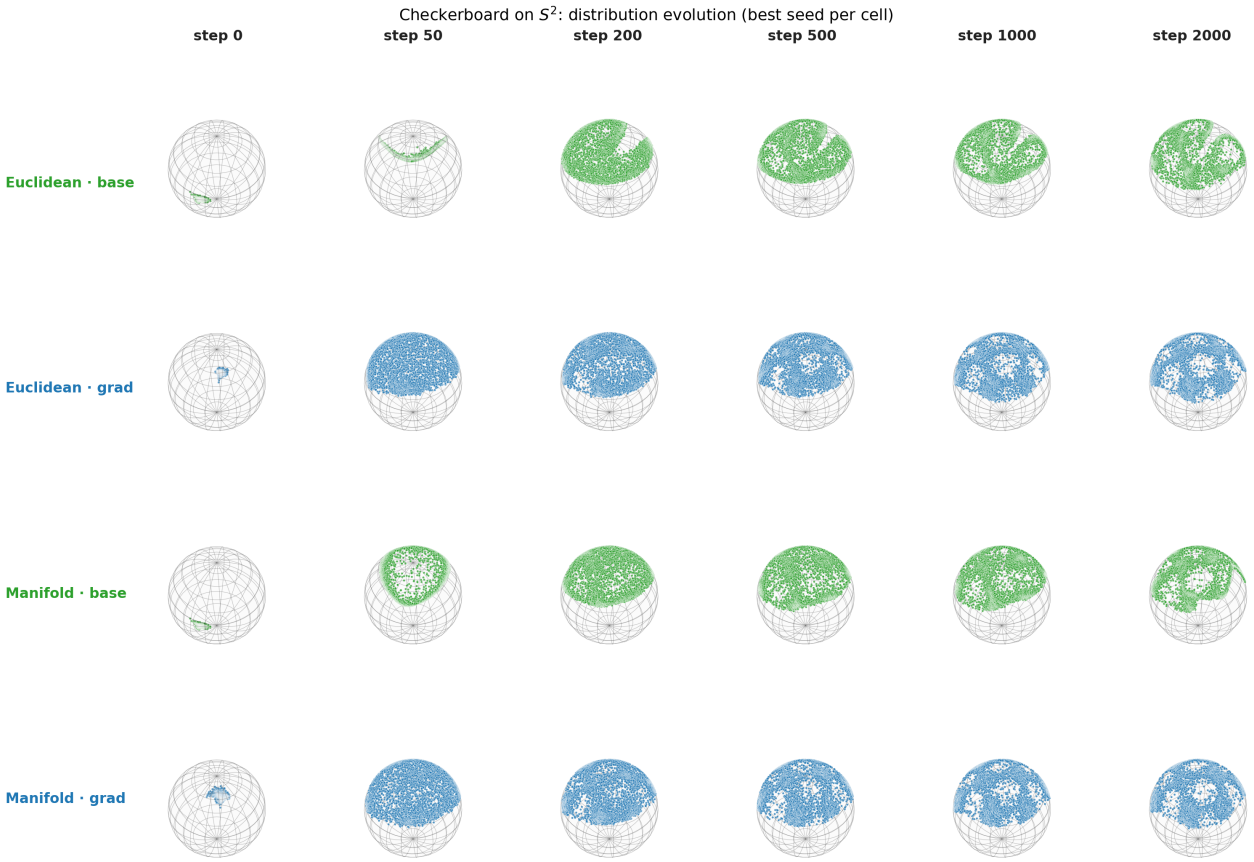

Figure 7. Evolution of the generated distribution during training for the checkerboard target on the spherical manifold $\mathbb{S}^2$.

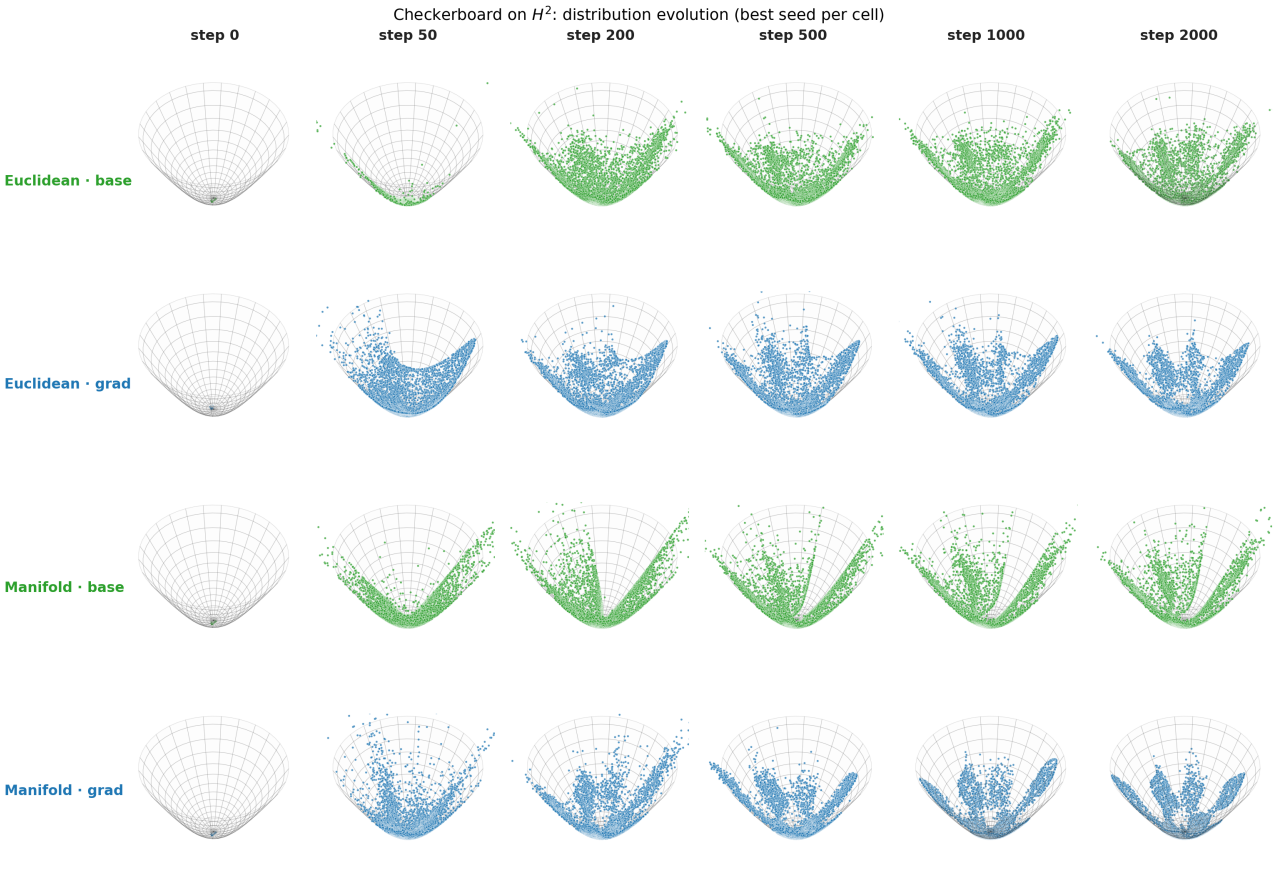

Figure 8. Evolution of the generated distribution during training for the checkerboard target on the hyperbolic manifold $\mathbb{H}^2$.

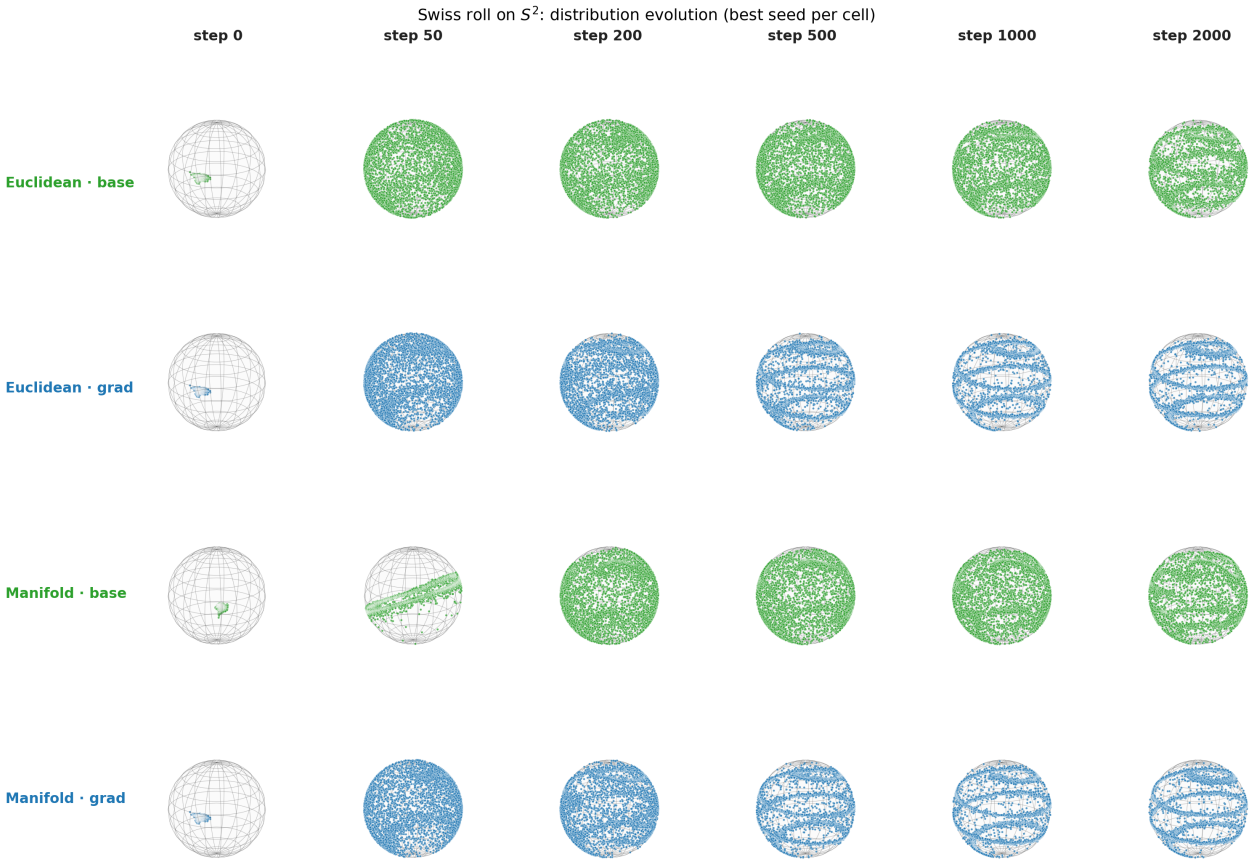

Figure 9. Evolution of the generated distribution during training for the swissroll target on the spherical manifold $\mathbb{S}^2$.

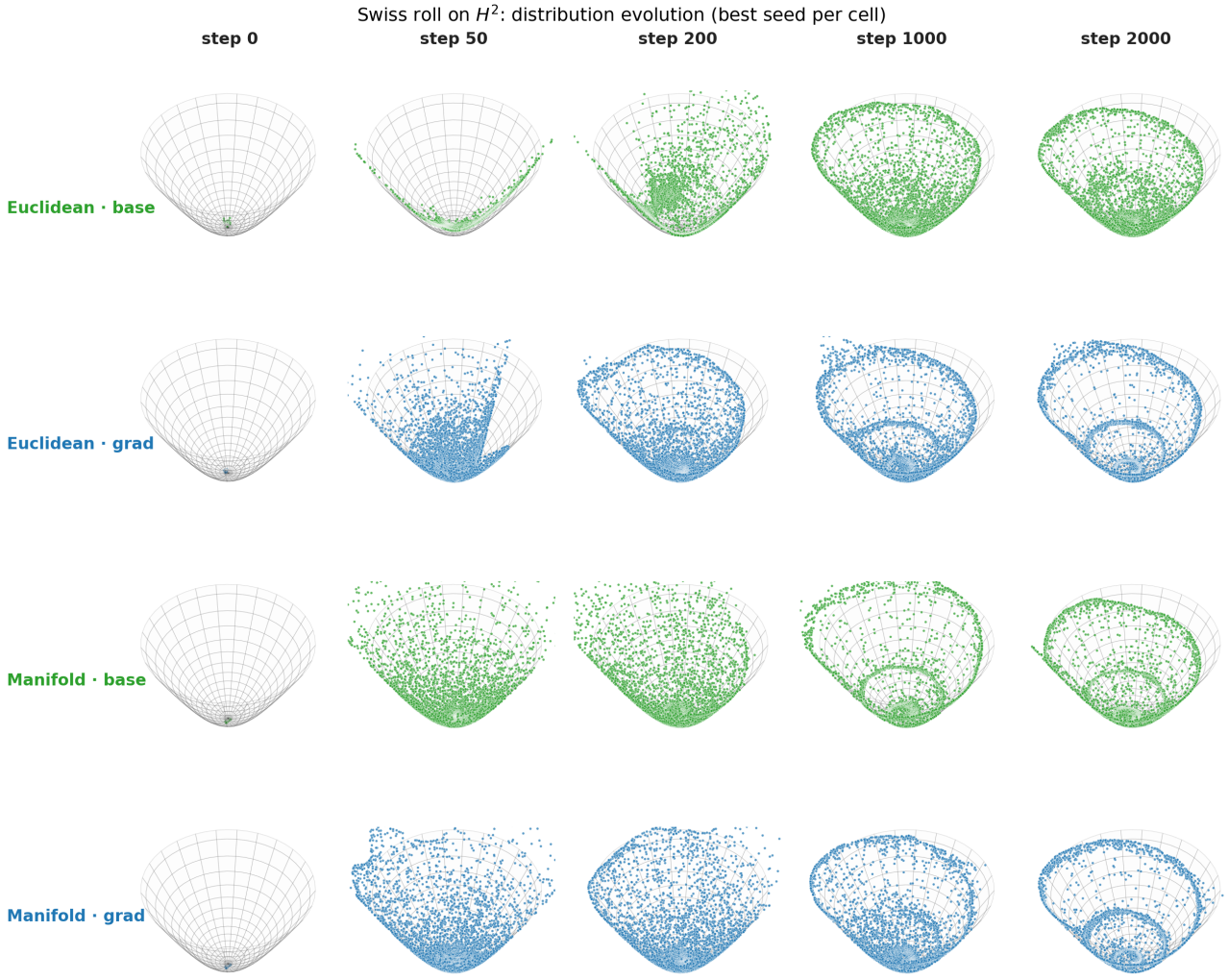

Figure 10. Evolution of the generated distribution during training for the swissroll target on the hyperbolic manifold $\mathbb{H}^2$.

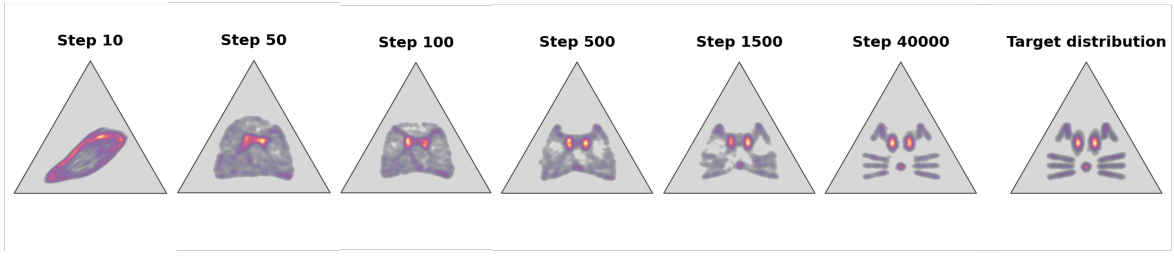

Figure 11. Evolution of the 'cat' distribution on the probabiblity simplex during training.

E.2. Geospatial data on the sphere

In addition to the results reported in Section 5, we present here a more detailed comparison of the drifting variants. Beyond MMD, we also report geodesic Sinkhorn distance and 1-NN accuracy, as described in Appendix F. Together, these additional metrics provide a broader view of the relative behavior of the different drifting formulations and further illustrate the

advantages of the gradient-based approach.

Overall, the results are consistent with the trends observed in the main text. In particular, the gradient-based variants tend to improve not only MMD, but also Sinkhorn distance and 1-NN performance across several datasets. This is especially clear for the Euclidean Laplace Gradient model on *Volcano* and *Earthquake*, and for the spherical gradient variants on *Flood*. On *Fire*, the picture is more mixed, but the gradient formulation still remains competitive across metrics. These results support the conclusion that the proposed reformulation leads to more reliable transport dynamics beyond a single evaluation criterion.

Method	**Volcano**		
	MMD ↓	**Sink.** ↓	**NN1** ↓
Euclidean Laplace (Deng et al., 2026)	0.146 ± 0.022	0.352 ± 0.035	0.843 ± 0.044
Euclidean Laplace Gradient	0.128 ± 0.013	0.283 ± 0.019	0.711 ± 0.032
Euclidean Gaussian Gradient	0.143 ± 0.035	0.334 ± 0.046	0.766 ± 0.029
Spherical Laplace	0.113 ± 0.021	0.320 ± 0.025	0.801 ± 0.014
Spherical Laplace Gradient	0.113 ± 0.014	0.313 ± 0.029	0.780 ± 0.026
Spherical Gaussian Gradient	0.112 ± 0.012	0.317 ± 0.015	0.790 ± 0.018
Spectral ($\nu = 2.5$)	0.126 ± 0.022	0.309 ± 0.031	0.769 ± 0.017

Table 5. Additional comparison for Volcano dataset.

Method	**Earthquake**		
	MMD ↓	**Sink.** ↓	**NN1** ↓
Euclidean Laplace (Deng et al., 2026)	0.044 ± 0.013	0.157 ± 0.014	0.694 ± 0.016
Euclidean Laplace Gradient	0.038 ± 0.009	0.151 ± 0.011	0.686 ± 0.018
Euclidean Gaussian Gradient	0.047 ± 0.006	0.162 ± 0.007	0.706 ± 0.008
Spherical Laplace	0.043 ± 0.002	0.179 ± 0.003	0.732 ± 0.003
Spherical Laplace Gradient	0.056 ± 0.008	0.184 ± 0.009	0.703 ± 0.005
Spherical Gaussian Gradient	0.158 ± 0.070	0.326 ± 0.120	0.707 ± 0.016
Spectral ($\nu = 2.5$)	0.037 ± 0.006	0.166 ± 0.004	0.704 ± 0.012

Table 6. Additional comparison for Earthquake dataset.

Method	**Fire**		
	MMD ↓	**Sink.** ↓	**NN1** ↓
Euclidean Laplace (Deng et al., 2026)	0.036 ± 0.002	0.142 ± 0.009	0.778 ± 0.011
Euclidean Laplace Gradient	0.030 ± 0.005	0.129 ± 0.006	0.749 ± 0.008
Euclidean Gaussian Gradient	0.048 ± 0.020	0.166 ± 0.015	0.846 ± 0.001
Spherical Laplace	0.049 ± 0.006	0.180 ± 0.007	0.809 ± 0.011
Spherical Laplace Gradient	0.047 ± 0.003	0.165 ± 0.011	0.770 ± 0.015
Spherical Gaussian Gradient	0.039 ± 0.003	0.220 ± 0.003	0.852 ± 0.002
Spectral ($\nu = 2.5$)	0.029 ± 0.004	0.170 ± 0.008	0.825 ± 0.010

Table 7. Additional comparison for Fire dataset.

Method	Flood		
	MMD ↓	Sink. ↓	NN1 ↓
Euclidean Laplace (Deng et al., 2026)	0.064 ± 0.010	0.203 ± 0.019	0.692 ± 0.021
Euclidean Laplace Gradient	0.067 ± 0.009	0.196 ± 0.013	0.655 ± 0.017
Euclidean Gaussian Gradient	0.072 ± 0.001	0.245 ± 0.010	0.841 ± 0.040
Spherical Laplace	0.064 ± 0.005	0.223 ± 0.006	0.713 ± 0.007
Spherical Laplace Gradient	0.053 ± 0.004	0.193 ± 0.014	0.672 ± 0.020
Spherical Gaussian Gradient	0.058 ± 0.001	0.191 ± 0.007	0.659 ± 0.018
Spectral ($\nu = 2.5$)	0.070 ± 0.009	0.241 ± 0.006	0.724 ± 0.010

Table 8. Additional comparison for Flood dataset.

F. Experimental Setup

Hardware. All experiments are carried out on NVIDIA A100 GPUs.

F.1. Synthetic experiments

We consider two complementary synthetic setups: a kernel-smoothness ablation in Euclidean space and a geometry ablation on constant-curvature manifolds. All synthetic models are optimized with Adam using learning rate 10^{-3}.

Kernel-smoothness ablation. For Fig. 5, we use a 2D swiss-roll target in $\mathbb{R}^2$ and sweep a Matérn kernel over

$$\nu \in \{0.5, 1, 1.5, 2, 2.5, 5, 100\},$$

treating $\nu = 100$ as a finite approximation to the Gaussian limit. We compare displacement-based drift with our kernel-gradient drift. The backbone is a $(16, 16)$ MLP with SiLU activations. We train for 3000 steps with batch size 256, kernel bandwidth $T = 0.2$, and step cap $\eta_{\max} = 1$.

Manifold toy examples. For Tab. 1 and Fig. 6, we use checkerboard and swiss-roll targets on $\mathbb{S}^2$ and $\mathbb{H}^2$. We compare a 2×2 design crossing

$$\textit{geometry} \in \{\textsc{Euclidean}, \textsc{Manifold}\} \qquad \text{and} \qquad \textit{method} \in \{\textsc{Base}, \textsc{Gradient}\}.$$

The EUCLIDEAN variants use ambient distances and additive updates, while the MANIFOLD variants use geodesic distances and the exponential map. All variants use a (geodesic) Laplace kernel. The backbone is a width-256, depth-5 MLP. We train for 2000 steps with batch size 2048 and gradient clipping at 1.0. Full hyperparameter configurations are provided in the accompanying repository.

Metrics. For the kernel-smoothness ablation, we report classifier two-sample test (C2ST) accuracy. A small MLP is trained to distinguish real from generated samples; values close to 0.5 indicate that the two distributions are difficult to distinguish. For the manifold toy examples, we report Sliced Wasserstein-2 distance (lower is better) and, on checkerboard targets, tile accuracy: the fraction of samples that, after being unwrapped to the source 2D chart, fall in a black square of the underlying 4×4 grid (higher is better). Sliced Wasserstein-2 is computed *extrinsically* on the ambient embeddings ($\mathbb{R}^3$ for $\mathbb{S}^2$, and the Minkowski coordinates treated as $\mathbb{R}^3$ for $\mathbb{H}^2$), using random linear projections rather than an intrinsic manifold variant; since all compared methods produce samples in the same ambient space, this remains a fair head-to-head comparison. All results are averaged over 3 seeds and reported as mean $\pm$ standard deviation.

F.2. Geospatial data on the sphere.

Data. We evaluate our method on the Earth benchmark introduced by Mathieu & Nickel (2020), which consists of geospatial event distributions supported on the sphere $\mathbb{S}^2$. We consider the four standard datasets: *earthquake*, *volcano*, *fire*, and *flood*. Following Woo et al. (2026), we use a fixed random split with 80% of the data for training, 10% for validation, and 10% for testing.

Optimization. We train all models for up to 20 epochs, or until earlier convergence, using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 10^{-2}. The learning rate is linearly warmed up during the first 2500 optimization steps and kept constant afterwards. We maintain an exponential moving average (EMA) of model parameters with decay 0.999, and report results using the EMA model. Gradients are clipped to norm 1.0. A single run on an A100 GPU takes approximately 20 minutes.

Metrics. Our primary selection metric is a kernel two-sample statistic on $\mathbb{S}^2$ computed with the geodesic Gaussian kernel $k(x, y) = \exp\big(-d_{\mathbb{S}^2}(x, y)^2\big)$. As discussed in Section 4.1, exponential-of-squared-geodesic kernels are not positive-definite on curved manifolds in general, so this statistic is not a strict MMD; we use it as a smooth discrepancy for relative comparison between methods. Empirically, all reported values are non-negative across runs, and rankings are consistent with the geodesic Sinkhorn distance and 1-NN two-sample accuracy, which do not rely on a positive-definite kernel. The Sinkhorn metric uses entropically regularized optimal transport between empirical real and generated distributions, with geodesic cost on the sphere. For 1-NN accuracy, we pool generated and real samples, assign each point the label of its nearest neighbor under geodesic distance, and report the resulting classification accuracy; values closer to 50% indicate that the two distributions are harder to distinguish.

F.3. Promoter DNA generation

Data. We follow the promoter design benchmark introduced by Avdeyev et al. (2023). The task is conditional generation of human promoter sequences of length $L = 1024$ over the nucleotide alphabet $\{A, C, G, T\}$. Each sequence is conditioned on a per-position regulatory signal track derived from CAGE transcription start site profiles, denoted `signal_1c`. This signal corresponds to the plus-strand CAGE BigWig from the FANTOM CAT release, with strand normalization applied: for negative-strand TSSs, both the sequence and signal are reverse-complemented so that all examples are presented in a consistent orientation.

The dataset consists of the top 100k most highly expressed transcription start sites, as ranked in the FANTOM CAT annotation. Following Woo et al. (2026), we use a chromosome-based split: chromosomes 8 and 9 for testing, chromosome 10 for validation, and all remaining autosomes for training.

DNA representation. For the Euclidean drift model, ground-truth sequences are represented as one-hot vectors in $\mathbb{R}^{L\times 4}$, while generated samples are relaxed categorical distributions on the product simplex Δ_3^L. For the spherical drift model, we map each categorical distribution to the positive orthant of $\mathbb{S}^3$ using the square-root embedding $p \mapsto \sqrt{p}$. Generation therefore takes place on the product manifold $(\mathbb{S}^3_+)^L$.

Architecture and optimization. We use the same dilated 1D-CNN backbone as Woo et al. (2026), removing time-conditioning components since drifting models do not require a time variable. The network consists of 20 residual blocks with channel width 256, GroupNorm, SiLU activations, and dilation schedule $[1, 1, 4, 16, 64] \times 4$. This is preceded by a kernel-size 9 input convolution and followed by two 1×1 output convolutions that produce per-position logits over the four nucleotides. The same backbone is used for all drifting formulations; geometry-specific operations are applied outside the network to the projected samples. Training uses Adam with an exponential moving average (EMA) of decay 0.999, and all reported metrics are computed on the EMA model. A single run on an A100 GPU takes approximately one hour.

Metrics. We evaluate generated promoters using 6-mer Pearson correlation, following Avdeyev et al. (2023). We aggregate 6-mer counts over the generated and reference corpora, normalize them to relative frequencies, and compute Pearson correlation over the union of observed 6-mers. Generated samples are discretized via per-position $\arg\max$ before evaluation. This metric measures whether generated sequences reproduce the local sequence composition of real promoters and serves as our primary model-selection criterion.

F.4. Molecular generation on QM9

Data. We evaluate unconditional molecular generation on QM9 (Wu et al., 2018). The task is to generate small molecules represented as SMILES strings, which are then parsed into molecular graphs for evaluation. We follow the same preprocessing and train/validation/test split protocol as Park et al. (2026).

Molecule representation. Although molecules are naturally represented as graphs, applying our framework directly to graph representations is complicated by permutation symmetry: a single molecule admits many equivalent node orderings, and the drifting objective would treat these isomorphic copies as distinct points to be pushed apart. Resolving this would require permutation-aware neighborhoods, which we leave to future work. Here, following Park et al. (2026), we sidestep this difficulty and operate on SMILES strings, viewing each molecule as a categorical sequence. Note that this is a strictly harder setting than most graph-based approaches, which typically condition on the number of atoms; SMILES generation is unconditional on molecule size.

Concretely, each molecule is represented as a SMILES string of fixed length $N = 32$ over a vocabulary V of size $K = 40$, padded when necessary. Generation therefore takes place over V^N. The Euclidean drift model operates on one-hot vectors in $\mathbb{R}^{N\times K}$, with generated samples relaxed to the product simplex $(\Delta^{K-1})^N$. The spherical drift model maps each per-position categorical distribution to the positive orthant of $\mathbb{S}^{K-1}$ via the square-root embedding $p \mapsto \sqrt{p}$, so generation takes place on the product manifold $(\mathbb{S}^{K-1}_+)^N$.

Architecture and optimization. We adopt the graph backbone and training protocol of Park et al. (2026), adapting the flow-based model to the one-step drifting objective. The network predicts relaxed categorical distributions for SMILES, while geometry-specific operations are applied outside the backbone.At evaluation, generated strings are discretized by taking the per-variable $\arg\max$.

Metrics. We report validity and uniqueness. Validity is the fraction of generated samples that yield a fully sanitizable RDKit molecule that can be converted to SMILES and uniqueness is the fraction of distinct molecules among the valid samples, identified by the canonical SMILES of the largest connected fragment.