Follow the Energy, Find the Path: Riemannian Metrics from Energy-Based Models

Louis Bethune[*]
Apple

David Vigouroux
IRT Saint Exupéry, ANITI, IMT Atlantique

Yilun Du
Harvard University

Rufin VanRullen
CNRS

Thomas Serre
Brown University

Victor Boutin[*]
CNRS

Abstract

What is the shortest path between two data points lying in a high-dimensional space? While the answer is trivial in Euclidean geometry, it becomes significantly more complex when the data lies on a curved manifold—requiring a Riemannian metric to describe the space's local curvature. Estimating such a metric, however, remains a major challenge in high dimensions.

In this work, we propose a method for deriving Riemannian metrics directly from pretrained Energy-Based Models (EBMs)—a class of generative models that assign low energy to high-density regions. These metrics define spatially varying distances, enabling the computation of geodesics—shortest paths that follow the data manifold's intrinsic geometry. We introduce two novel metrics derived from EBMs and show that they produce geodesics that remain closer to the data manifold and exhibit lower curvature distortion, as measured by alignment with ground-truth trajectories. We evaluate our approach on increasingly complex datasets: synthetic datasets with known data density, rotated character images with interpretable geometry, and high-resolution natural images embedded in a pretrained VAE latent space. Our results show that EBM-derived metrics consistently outperform established baselines, especially in high-dimensional settings.

Our work is the first to derive Riemannian metrics from EBMs, enabling data-aware geodesics and unlocking scalable, geometry-driven learning for generative modeling and simulation.

1 Introduction

What is the shortest path between two data points in a high-dimensional space? In Euclidean geometry, the answer is a straight line. But in modern machine learning, where data often lies on unknown curved manifolds within a high-dimensional space, straight lines slice through regions without data (see linear interp. in Fig. 1). Capturing the true geometry of data is therefore critical in fields where distance-based analyses depend on underlying structure, such as vision [1–3], language [4, 5], biology [6], and cognitive science [7, 8]. Riemannian geometry offers a principled way to navigate these spaces by introducing a smoothly varying local metric, the Riemannian metric, which encodes how space bends and stretches [9]. Within this framework, the shortest path between two points is no longer a straight line, but a geodesic—a curve that follows the intrinsic curvature of the manifold. Computing geodesics requires knowing the underlying Riemannian metric, but estimating such a metric for complex, high-dimensional data remains a major challenge in machine learning.

[*]Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

A promising strategy for deriving Riemannian metrics is to take a data-driven approach—learning the metric directly from the data itself. This approach estimates the data density and turns it into a Riemannian metric that contracts high-density regions and dilates low-density ones, aligning the geometry with the data manifold [10] (see § 2 for more details). Existing methods, such as kernel-based estimators [11], normalizing flows [12], and density-based constructions [13], have succeeded in low-dimensional settings. However, their performance often degrades in high dimensions, where sparse local sampling makes it hard to capture reliable geometric structure [14, 15]. Meanwhile, recent advances in generative AI [16–18] have produced models capable of capturing complex data distributions in high-dimensional spaces with remarkable accuracy. *If these models can learn the data distribution, can they also reveal its underlying geometry?*

In this article, we answer affirmatively by proposing to derive Riemannian metrics from pretrained Energy-Based Models (EBMs) [16, 19, 20]. EBMs are a flexible class of generative models that define an energy function E_θ, parameterized by a neural network, assigning low energy to likely data points (i.e., $p_\theta(\mathbf{x}) \propto \exp(-E_\theta(\mathbf{x}))$).We show that the energy landscape of an EBM encodes a rich geometric structure and can be leveraged to derive effective Riemannian metrics. Specifically, we introduce two novel conformal Riemannian metrics—metrics that scale the identity by a positive scalar function: $\mathbf{G}_{\mathbf{E}_\theta}$ proportional to the energy itself, and $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$, proportional to the inverse unnormalized density. We evaluate both against established alternatives ($\mathbf{G}_{\mathbf{RBF}}$ [13] and $\mathbf{G}_{\mathbf{LAND}}$ [11]) across datasets of increasing complexity—from toy distributions with known geodesics (see § 4.2), to rotated character images where the manifold structure is partially known (see § 4.3), and finally to high-dimensional natural images where no ground truth geometry is available (see § 4.4). Throughout this work, we adopt the common choice of equipping the data space with a density-based Riemannian metric, thereby defining the geometry of the manifold in terms of data concentration. We show that EBM-based metrics yield geodesics that (i) remain closer to the data manifold and (ii) better reflect its intrinsic curvature (see Fig. 1 for a visualization of the geodesics).

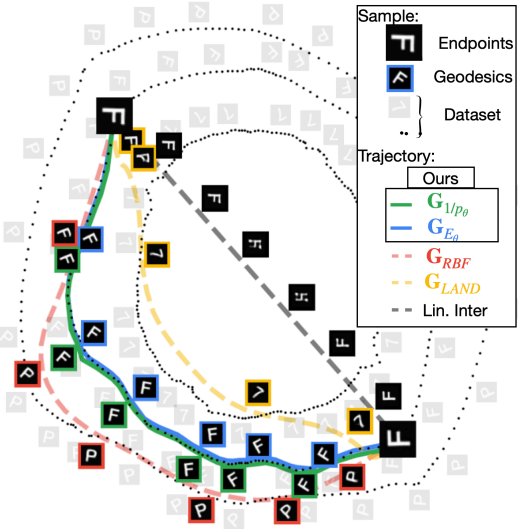


Figure 1: **Geodesics visualization for the URC dataset**. Trajectories and samples are projected in the PCA space for visualization.

Overall, our contributions are summarized as follows:

- We propose a novel framework based on pretrained Energy-Based Models (EBMs) to derive Riemannian metrics. In particular, we introduce two novel conformal metrics $\mathbf{G}_{\mathbf{E}_\theta}$ and $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$, based on the data log-likelihood and data density, respectively.
- We demonstrate that these EBM-derived metrics yield geodesics that remain closer to the data manifold and better reflect its curvature.
- We show that the proposed EBM-based metrics scale more robustly than prior approaches.

By grounding Riemannian metrics in generative AI, we hope to initiate a new paradigm for understanding and navigating the hidden geometry of high-dimensional data spaces.

2 Related Work

The many facets of data geometry: A variety of approaches have been proposed to study the geometry of data:

- *Information Geometry*: This historical approach is rooted in the work of Rao [21] and Amari [22]. It connects statistics and differential geometry by interpreting the Fisher information [23] as a Riemannian metric on the manifold of parameters of a statistical model. In contrast, our

work derives Riemannian metrics directly from the data space using the energy or the likelihood of an EBM.

- *Data-Space induced metrics*: Closer to our work, this approach estimates Riemannian metrics directly from samples. The LAND metric [11] derives a local metric tensor from the empirical covariance of nearby points. The RBF metric [13] defines a conformal metric using an RBF network trained as a parametric KDE, learning centres, widths, and weights so its output forms an unnormalised data density. Both serve as baselines in our study (see § 4.1) and have recently been used for geodesic fitting via flow matching [24]. The (unpublished) work of Perone [25] was also a key inspiration, proposing to build metrics from the score function of a generative model—an idea also explored by Diepeveen et al. [26].
- *Latent-Space induced metrics*: Another line of work uses pullback geometry [27–32], mapping the Euclidean metric from a network's latent space to the data space—typically through the Jacobian of a VAE encoder [33]. While our method operates in the latent space of a VAE in high-dimensional settings, the metric is derived from the energy of the EBM and remains independent of the VAE encoder.
- *Generative modeling on a pre-defined manifold:* Recent approaches such as flow-based models [34, 35] and Schrödinger bridges [36, 37] learn transport paths between distributions, sometimes defined over Riemannian manifolds [38–41]. These methods assume a known, fixed manifold geometry (e.g., a hypersphere) and design generative models to operate within that structure. In contrast, our approach starts from a generative model—an EBM—and derives the Riemannian metric itself from the model, allowing the geometry to emerge from the data.

For a more detailed review of the related work (including topological data analysis, symmetries, computer graphics, or metric learning), see Supp. A and [42, 14].

Energy-Based Models (EBMs): EBMs, trained via maximum likelihood [16] (see § 3.2), are particularly well-suited for deriving Riemannian metrics. Their contrastive training, combined with Langevin dynamics sampling, encourages learning a *global* energy landscape that assigns meaningful values across the entire ambient space, including regions far from the data manifold. In contrast, normalizing flows [43] are limited by their invertible architecture [44, 45] and tend to perform poorly on out-of-distribution data [46], sometimes leaking probability mass outside the support [47]. EBMs trained with diffusion losses [48] or distilled from diffusion models [49] generate high-quality samples, but their energy function depends on a time-indexed noise scale, limiting them to local rather than global energy landscapes. This makes them unsuitable for defining a consistent Riemannian metric. Prior work has used the global energy landscape of EBMs trained via maximum likelihood for trajectory planning in robotics [50], though not in the context of geodesics.

3 Method

Notation: Scalars are denoted by plain lowercase (e.g., x), vectors by bold lowercase (e.g., $\mathbf{x} \in \mathbb{R}^D$), and matrices by bold uppercase (e.g., $\mathbf{X}$). Let $\mathbf{I}$ be the identity matrix of $\mathbb{R}^{D\times D}$. $\mathcal{S}^D_{++}$ is the set of symmetric $D \times D$ positive definite matrices. Let $\mathcal{M}$ be a Riemannian manifold, with tangent space at $\mathbf{x} \in \mathcal{M}$ denoted $\mathcal{T}^{\mathcal{M}}_{\mathbf{x}}$. Herein, we assume that $\mathcal{M}$ is embedded in a D-dimensional Euclidian space ($\mathcal{M} \subset \mathbb{R}^D$).

3.1 A primer on Riemannian geometry

A Riemannian manifold $(\mathcal{M}, \mathbf{G})$ is a smooth manifold $\mathcal{M}$ (i.e., a set locally homeomorphic to $\mathbb{R}^D$) equipped with a Riemannian metric $\mathbf{G} : \mathcal{M} \to \mathcal{S}^D_{++}$. $\mathbf{G}$ defines a smoothly changing inner product on the tangent space $\mathcal{T}^{\mathcal{M}}_{\mathbf{x}}$ at each point $\mathbf{x} \in \mathcal{M} : \langle \mathbf{u}, \mathbf{v}\rangle_{\mathbf{x}} = \mathbf{u}^\top \mathbf{G}(\mathbf{x})\mathbf{v}$, with $\mathbf{u}, \mathbf{v} \in \mathcal{T}^{\mathcal{M}}_{\mathbf{x}}$ [9]. The length of a curve $\boldsymbol{\gamma} : [0, 1] \to \mathcal{M}$ linking two points $\mathbf{x_0} = \boldsymbol{\gamma}(0)$ and $\mathbf{x_1} = \boldsymbol{\gamma}(1)$ ($\mathbf{x_0}, \mathbf{x_1} \in \mathcal{M}$), is measured as:

$$L(\boldsymbol{\gamma}) = \int_0^1 \sqrt{\langle \dot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t)\rangle_{\boldsymbol{\gamma}(t)}}\, dt. \tag{1}$$

In Eq. 1, $\dot{\boldsymbol{\gamma}}(t)$ denotes the velocity vector of the curve $\boldsymbol{\gamma}(t)$, which lies in the tangent space at that point (i.e., $\dot{\boldsymbol{\gamma}}(t) \in \mathcal{T}^{\mathcal{M}}_{\boldsymbol{\gamma}(t)}$). The minimizer of Eq. 1 is called a *geodesic*; it represents the (locally) shortest path between $\mathbf{x_0}$ and $\mathbf{x_1}$. In this work, we minimize the kinetic energy functional instead of the length (see Eq. 2). Although both functionals yield the same geodesics up to a parametrization,

minimizing the kinetic energy functional results in a constant Riemannian speed parametrization[2]. This property simplifies optimization and improves numerical stability [9, 13].

$$\boldsymbol{\gamma}^\star(t) = \arg\min_{\boldsymbol{\gamma}} \mathcal{E}[\boldsymbol{\gamma}] \;\text{ s.t. }\; \mathcal{E}[\boldsymbol{\gamma}] = \frac{1}{2}\int_0^1 \langle \dot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t)\rangle_{\boldsymbol{\gamma}(t)}\, dt. \tag{2}$$

In the Euclidean case ($\mathcal{M} = \mathbb{R}^D$, $\mathbf{G}(\mathbf{x}) = \mathbf{I}$), $\mathcal{E}$ is equivalent to the kinetic energy of a unit-mass particle moving along $\boldsymbol{\gamma}(t)$, hence the name kinetic energy functional.

To avoid the computational cost of solving Eq. 2 for each new pair $(\mathbf{x}_0, \mathbf{x}_1)$ at inference time, we follow [24] and approximate the geodesic with a neural interpolant $\boldsymbol{\varphi}_\eta$ (with parameters η).

$$\mathbf{x}_{t,\eta} = (1-t)\mathbf{x}_0 + t\mathbf{x}_1 + 2t(1-t)\boldsymbol{\varphi}_\eta(\mathbf{x}_0, \mathbf{x}_1, t). \tag{3}$$

This parameterization satisfies the boundary conditions ($\mathbf{x}_{0,\eta}$=$\mathbf{x}_0$, $\mathbf{x}_{1,\eta}$=$\mathbf{x}_1$). In Eq. 3, $\boldsymbol{\varphi}_\eta$ serves as a nonlinear correction to the linear path, allowing the learned path to bend toward the data manifold. We train a single interpolant network $\boldsymbol{\varphi}_\eta$ over batches of random endpoint pairs so it can approximate geodesics between arbitrary points (see Algo. 1). Intuitively, our geodesic interpolant begins with a straight line between the endpoints and uses a neural network to compute a smooth curvature relative to this baseline—bending the path toward regions of higher data density, much like pulling a string taut over a curved surface that reflects the geometry of the data. Unlike Kapusniak et al. [24], who use full autodifferentiation to compute $\dot{\mathbf{x}}_{t,\eta}$, we opt for finite difference instead. We found this approach more stable and accurate when using a fine-time discretization.

Algorithm 1: Training geodesic interpolant

Input: Endpoints pairs: ($\{\mathbf{x}_0\}$, $\{\mathbf{x}_1\}$), Interp. net.: $\boldsymbol{\varphi}_\eta$, Metric: $\mathbf{G}$, Time steps: T

dt=$\frac{1}{\mathrm{T}-1}$, t=[0:1:dt]

while training **do**

- $\mathbf{x}_0 \sim \{\mathbf{x}_0\}$ and $\mathbf{x}_1 \sim \{\mathbf{x}_1\}$ ## sample batch of pairs
- $\mathbf{x}_{t,\eta} = (1-t)\mathbf{x}_0 + t\mathbf{x}_1 + 2t(1-t)\boldsymbol{\varphi}_\eta(\mathbf{x}_0, \mathbf{x}_1, t)$
- $\dot{\mathbf{x}}_{t,\eta} = \dfrac{\mathbf{x}_{t+1,\eta} - \mathbf{x}_{t,\eta}}{\mathrm{dt}}$ ## finite difference
- $\mathcal{L}(\eta) = \mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1}\left[\dfrac{1}{2}\displaystyle\sum_{t=0}^{1}\left[\dot{\mathbf{x}}_{t,\eta}^\top \mathbf{G}(\mathbf{x}_{t,\eta})\dot{\mathbf{x}}_{t,\eta}\right]\mathrm{dt}\right]$
- Update η using gradient $\nabla_\eta \mathcal{L}$

Although Algo. 1 approximates geodesics for a given metric $\mathbf{G}$, the trajectories may initially deviate from the data manifold—especially early in training, when they are initialized as straight lines in the ambient space. However, if (i) the eigenvalues of $\mathbf{G}$ are large when off-manifold and (ii) small when on-manifold, then the interpolated points $\mathbf{x}_t$ are progressively drawn toward the manifold during optimization [24, 13]. In other words, an effective $\mathbf{G}$ should penalize off-manifold directions and encourage paths through high-density paths, steering the geodesics along true data geometry. This insight suggests that defining the metric as a decreasing function of the data probability (e.g., $\mathbf{G}(\mathbf{x}) \propto p(\mathbf{x})^{-1} \cdot \mathbf{I}$) can effectively steer trajectories toward high-density regions. In practice, however, the true data distribution is unknown and only observed through samples. In this work, we use an EBM to approximate the data distribution.

3.2 Energy-Based Models

Let $p_{\mathcal{M}}$ be the true data distribution supported on the manifold $\mathcal{M}$, such that $\int_{\mathbf{x}\in\mathcal{M}} p_{\mathcal{M}}(\mathbf{x})d\mathbf{x} = 1$. In practice, we do not have access to $p_{\mathcal{M}}$ directly, but only to a finite set of samples $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$drawn from it. These samples define the empirical distribution $p_{\mathcal{D}}$, which we use to train our models.

Energy-Based Models (EBMs) provide a flexible framework for modeling complex, unnormalized probability distributions—making them particularly well-suited for data concentrated on low-dimensional manifolds. Here we define the energy function $E_\theta(\mathbf{x}) \in \mathbb{R}$, parameterized with a neural network with weights θ. This energy induces a probability distribution of the form:

$$p_\theta(\mathbf{x}) = \frac{\exp\big(-E_\theta(\mathbf{x})\big)}{Z(\theta)} \quad \text{where} \quad Z(\theta) = \int \exp\big(-E_\theta(\mathbf{x})\big)d\mathbf{x}. \tag{4}$$

[2]With length fixed, the strictly convex energy $E = \frac{1}{2}\int_0^1 v(t)^2 dt$ attains its minimum—by Jensen's inequality—only when the speed $v(t)$ is constant.

Our goal is to train the EBM so that p_θ approximates the data distribution $p_{\mathcal{M}}$. To do so, we minimize the negative log-likelihood w.r.t to the empirical distribution: $\mathcal{L}_{ML}(\theta) = \mathbb{E}_{\mathbf{x} \sim p_D}[-\log p_\theta(\mathbf{x})]$. Although the partition function $Z(\theta)$ is intractable, previous works have shown that the gradient of this objective can be estimated without computing $Z(\theta)$ explicitly [51, 52] (see Supp. B.1 for the demonstration), a loss known as *contrastive divergence*:

$$\nabla_\theta \mathcal{L}_{ML} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_\mathcal{D}}[E_\theta(\mathbf{x}^+)] - \mathbb{E}_{\mathbf{x}^- \sim p_\theta}[E_\theta(\mathbf{x}^-)] \tag{5}$$

where $\mathbf{x}^+$ are data samples and $\mathbf{x}^-$ are samples drawn from the model distribution p_θ using Langevin dynamics. We adopt the training procedure of [16], which is known to scale well (see Supp. B for the full pseudo-code). From this point on, we refer to E_θ as a pre-trained energy function.

EBM can be hard to train in high-dimensional pixel space, especially because of the sampling procedure [53–55]. For complex tasks, we follow standard practice and operate in the latent space of a pretrained VAE [56], where all baselines are evaluated for fairness. To improve the EBM training training stability, we regularize the contrastive divergence loss with a denoising term, which preserves the global structure of the energy landscape while enhancing convergence—a technique we find both effective and broadly applicable.

3.3 EBM-derived Riemannian Metrics

Here, we describe the EBM-derived metrics $\mathbf{G}_{\mathbf{E}_\theta}$, $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$. For details on the baseline Riemannian metrics $\mathbf{G}_{\mathbf{LAND}}$,$\mathbf{G}_{\mathbf{RBF}}$, see § 4.1. To ensure a fair comparison —and following standard practice in the field [42, 57, 58]— all metrics are cast using a shared parametric form:

$$\mathbf{G}(\boldsymbol{x}) = \begin{cases} \alpha\, \mathbf{h}(\boldsymbol{x}) + \beta & \text{for } \mathbf{G}_{\mathbf{E}_\theta}, \\ (\alpha\, \mathbf{h}(\boldsymbol{x}) + \beta)^{-1} & \text{for } \mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}, \mathbf{G}_{\mathbf{LAND}}, \mathbf{G}_{\mathbf{RBF}}, \end{cases}$$

where $\mathbf{h}(\mathbf{x})$ is a metric-specific, positive-definite function (either scalar, diagonal, or matrix), and α, β are calibration constants. These constants are chosen so that the metric scale to $\mathbf{I}$ on the data manifold and to $10^3 \cdot \mathbf{I}$ in low-density regions[3]. This allows fair comparison across metric choices without introducing significant sensitivity to hyperparameter tuning. Further details about the metric calibration procedure are provided in Supp. C.1. Importantly, all EBM-derived metrics are *conformal*, they take the form $\lambda(\mathbf{x})\mathbf{I}$, where λ is a scalar function. In other words, they scale the identity matrix uniformly in all directions, resulting in isotropic metrics:

- $\mathbf{G}_{\mathbf{E}_\theta}$ defines a Riemannian metric by directly scaling the raw energy of a pretrained EBM. This is the simplest —yet surprisingly effective—formulation we consider:

  $$\mathbf{G}_{\mathbf{E}_\theta}(\mathbf{x}) = (\alpha * E_\theta(\mathbf{x}) + \beta) \cdot \mathbf{I}. \tag{6}$$

  Intuitively, high-energy (low-density) regions receive a larger metric, penalizing movement away from the data. Note that E_θ is an affine rescaling of the negative log-likelihood $-\log p_\mathcal{D}$.
- $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$ leverages the inverse of an unnormalized probability estimate:

  $$\mathbf{G}_{1/p_\theta}(\mathbf{x}) = (\alpha * \exp(-E_\theta(\mathbf{x})) + \beta)^{-1} \cdot \mathbf{I}. \tag{7}$$

  Compared to $\mathbf{G}_{\mathbf{E}_\theta}$, this metric applies an inverse to a decreasing exponential, forming a strong barrier against low-density regions. It stays small near the data manifold but rises sharply elsewhere, acting as a repulsive force. Its key advantages are: (i) a clear probabilistic interpretation via the unnormalized density, and (ii) direct comparability to $\mathbf{G}_{\mathbf{LAND}}$ and $\mathbf{G}_{\mathbf{RBF}}$ as they share the same inverse form.

In the next section, we introduce the baseline Riemannian metrics used for comparison. We also empirically evaluate their behavior across datasets of increasing complexity, focusing on how they capture the underlying manifold and shape geodesic paths.

4 Experiments

4.1 Baseline Riemannian Metrics

$\mathbf{G}_{\mathbf{RBF}}$ [13, 24] and $\mathbf{G}_{\mathbf{LAND}}$ [11] are established metrics from the Riemannian geometry literature:

[3]Note that this multiplicative factor amounts to a change of unit, to ensure reasonable scaling of the lengths, but the induced geodesics are only determined by the ratio α/β.

- $\mathbf{G}_{\text{LAND}}$, also known as the LAND metric [11], is a nonparametric Riemannian metric that adapts to the local geometry of the dataset. Around each point $\mathbf{x}$, it estimates a Gaussian distribution by weighting all data points $\{\mathbf{x}_i\}_{i=1}^N$ according to their distance to $\mathbf{x}$:

$$\mathbf{G}_{\text{LAND}}(\mathbf{x}) = (\alpha \operatorname{diag}(\mathbf{h}(\mathbf{x})) + \beta \mathbf{I})^{-1} \text{ s.t } h^{(j)}(\mathbf{x}) = \sum_{i=1}^{N} (x_i^{(j)} - x^{(j)})^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x_i}||^2}{2\sigma^2}\right) \quad (8)$$

  Here, $h^{(j)}(\mathbf{x})$ measures the local variance along dimension j, weighted by a Gaussian kernel with bandwidth σ. $\mathbf{G}_{\text{LAND}}$ is the only diagonal (i.e., non-conformal) metric we consider, allowing it to model local anisotropy. While flexible and model-free, LAND has practical drawbacks: it requires the full dataset at inference, is sensitive to the choice of σ, and can behave non-smoothly near sharp neighborhood transitions (see Supp. C.2 for examples).
- $\mathbf{G}_{\text{RBF}}$ is a conformal Riemannian metric in which h is a weighted sum of Radial Basis Functions (RBFs) centered on K cluster centroids $\{\hat{\mathbf{x}}_k\}_{k=1}^K$ computed via K-means [13]:

$$\mathbf{G}_{\text{RBF}}(\mathbf{x}) = (\alpha \cdot h(\mathbf{x}) + \beta)^{-1} \cdot \mathbf{I}, \quad h(\mathbf{x}) = \sum_{k=1}^{K} w_k \exp\left(-\frac{1}{2} \cdot \lambda_k \|\mathbf{x} - \hat{\mathbf{x}}_k\|^2\right).$$

  The weights w_k are trained so that $h(\mathbf{x}) \approx 1$ on the data manifold, and λ_k is set from inter-cluster distances (see Supp. C.3). This yields a smooth, efficient approximation of the data geometry and scales better than LAND [24]. However, it may miss fine-grained structure, especially in regions of complex or uneven density. Like other methods based on Euclidean distance (and K-means), it suffers from the curse of dimensionality. Its accuracy depends on K, λ_k, and centroid placement (illustrated in Supp. C.3).

The scaling constants (α, β) are introduced to ensure consistent dynamic range across metrics and have minimal impact on convergence or geodesic quality; the number of discretization steps ($T = 100$) is chosen as a trade-off between efficiency and accuracy, consistent with prior work. We evaluate $\mathbf{G}_{\mathbf{1/p}_\theta}$, $\mathbf{G}_{\mathbf{E}_\theta}$, $\mathbf{G}_{\text{RBF}}$, and $\mathbf{G}_{\text{LAND}}$ on three datasets of increasing complexity. Circular Mixture of Gaussians offers full control and ground-truth geodesics. The rotated characters dataset is higher-dimensional but still allows quantitative evaluation. Animal Faces is made of higher-dimensional images but with no ground truth. This progression tests metric performance as data complexity grows. The code to reproduce all our experiments is available at `https://github.com/VictorBoutin/RiemannEBM`.

4.2 Circular Mixture of Gaussians

We consider two toy datasets built using a mixture of Gaussians arranged along a semicircle. In the first, called Uniform Circular Gaussians (UCG), the Gaussian components have equal weights (see Fig. 2a). In the second, Weighted Circular Gaussians (WCG), the weights are non-uniform, with higher density near the center of the arc, as reflected by the contour intensity shown in Fig. 2c. For both datasets, we have access to the closed-form probability distribution of the data, denoted $p_{\mathcal{M}}$ (see Supp. D.1 for details of $p_{\mathcal{M}}$). We first train an Energy-Based Model (EBM) on each dataset to derive the metrics $\mathbf{G}_{\mathbf{E}_\theta}$ and $\mathbf{G}_{\mathbf{1/p}_\theta}$ (see Supp. D.2 for training details). Then, we apply Algo. 1 to both datasets using all Riemannian metrics described above. Additionally, we include two baseline Riemannian metrics derived directly from the true distribution $p_{\mathcal{M}}$:

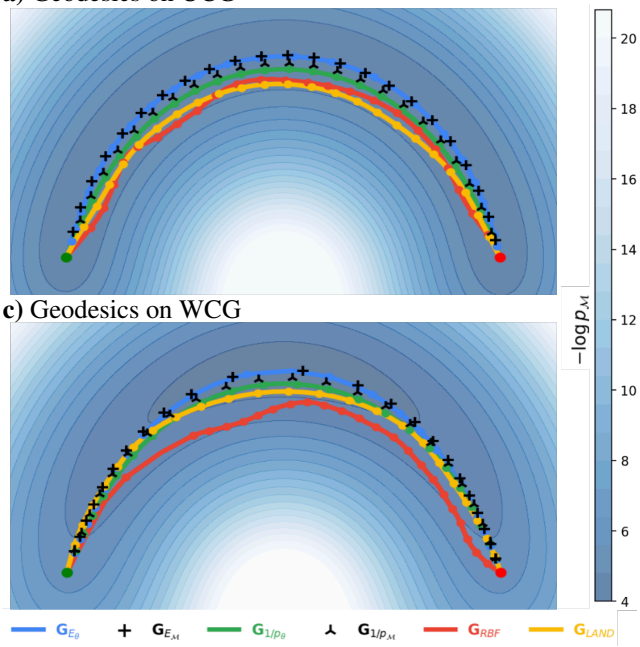
$$\mathbf{G}_{\mathrm{E}_{\mathcal{M}}}(\mathbf{x}) = -\alpha * \log p_{\mathcal{M}}(\mathbf{x})\mathbf{I} + \beta \quad \text{and} \quad \mathbf{G}_{1/p_{\mathcal{M}}}(\mathbf{x}) = (\alpha * p_{\mathcal{M}}(\mathbf{x}) + \beta)^{-1} \cdot \mathbf{I} \quad (9)$$

Eq. 9 uses calibration constants α and β, computed as in other metrics. Some geodesics obtained for the 6 different metrics are shown in Fig. 2a and Fig. 2c for the UCG and WCG datasets, respectively. We refer the reader to Supp. D for details on network architectures and hyperparameters.

To evaluate geodesic quality, we use two evaluation metrics. The first is the accumulated probability along the geodesic path, $p_{\mathcal{M}}(\boldsymbol{\gamma}^\star) = \sum_{t=1}^{T} p_{\mathcal{M}}(\mathbf{x}_{t,\eta^\star})$. It measures how closely the trajectory aligns with the data manifold — the higher the better. The second is the RMSE to a baseline geodesic computed using the true distribution $p_{\mathcal{M}}$, matched by metric type (e.g., $\mathbf{G}_{\mathbf{E}_\theta}$ vs. $\mathbf{G}_{\mathrm{E}_{\mathcal{M}}}$). All quantitative results are averaged over $1,000$ geodesics with distinct endpoints (See Fig. 2b and d). $\mathbf{G}_{\mathbf{E}_\theta}$ achieves the highest accumulated probability, indicating closest alignment with the data manifold, while $\mathbf{G}_{\mathbf{1/p}_\theta}$ yields the lowest RMSE to its baseline—best approximating the ground-truth geodesic. Both EBM-based metrics consistently outperform other methods across evaluation criteria.

To test how different metrics behave when the density varies along the data manifold, we switch

a) Geodesics on UCG



b) Geodesics evaluation on UCG

Metric	$p_{\mathcal{M}}(\gamma^\star)$ (↑)	RMSE (↓)
$\boldsymbol{G}_{E_\mathcal{M}}$	0.79	-
$\boldsymbol{G}_{1/p_\mathcal{M}}$	0.77	-
$\mathbf{G}_{\mathbf{E}_\theta}$	**0.78**	0.12
$\mathbf{G}_{\mathbf{1/p}_\theta}$	0.73	**0.10**
$\mathbf{G}_{\mathbf{LAND}}$	0.60	0.38
$\mathbf{G}_{\mathbf{RBF}}$	0.61	0.39

c) Geodesics on WCG

d) Geodesics evaluation on WCG

Metric	$p_{\mathcal{M}}(\gamma^\star)$ (↑)	RMSE (↓)
$\boldsymbol{G}_{E_\mathcal{M}}$	0.67	-
$\boldsymbol{G}_{1/p_\mathcal{M}}$	0.73	-
$\mathbf{G}_{\mathbf{E}_\theta}$	**0.67**	0.18
$\mathbf{G}_{\mathbf{1/p}_\theta}$	**0.67**	**0.14**
$\mathbf{G}_{\mathbf{LAND}}$	0.65	0.34
$\mathbf{G}_{\mathbf{RBF}}$	0.47	2.2

Figure 2: **Geodesics on UCG and WCG datasets. (a, c)**: Some geodesics obtained on UCG **(a)** and WCG **(c)**, for 6 different Riemannian metrics. The contour plots represent the energy landscape given by $-\log p_{\mathcal{M}}$. **(b, d)** Quantitative evaluation of geodesics on UCG **(b)** and WCG **(d)**. We report (i) the accumulated probability along the geodesic (the higher the better) and ii) RMSE between each geodesic and its corresponding baseline (i.e., $\boldsymbol{G}_{E_\mathcal{M}}$ for $\mathbf{G}_{\mathbf{E}_\theta}$, and $\boldsymbol{G}_{1/p_\mathcal{M}}$ for $\mathbf{G}_{\mathbf{1/p}_\theta}$, $\mathbf{G}_{\mathbf{LAND}}$ and $\mathbf{G}_{\mathbf{RBF}}$). See Supp. D.3 for the 2-σ error.

from the uniformly populated UCG semicircle to the Weighted Circular Gaussian (WCG), whose samples cluster near the arc's centre. As shown in Fig. 3, log-based metrics ($\mathbf{G}_{\mathbf{E}_\theta}$, $\boldsymbol{G}_{E_\mathcal{M}}$) accentuate the manifold curvature more than $1/p$-based ones ($\mathbf{G}_{\mathbf{1/p}_\theta}$, $\mathbf{G}_{\mathbf{RBF}}$, $\mathbf{G}_{\mathbf{LAND}}$, $\boldsymbol{G}_{1/p_\mathcal{M}}$), producing larger steps in high-density regions. This is because $-\log p$ diverges as $p \to 0$, amplifying distortions and speed variations.

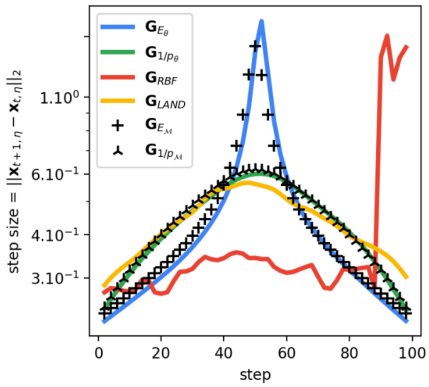
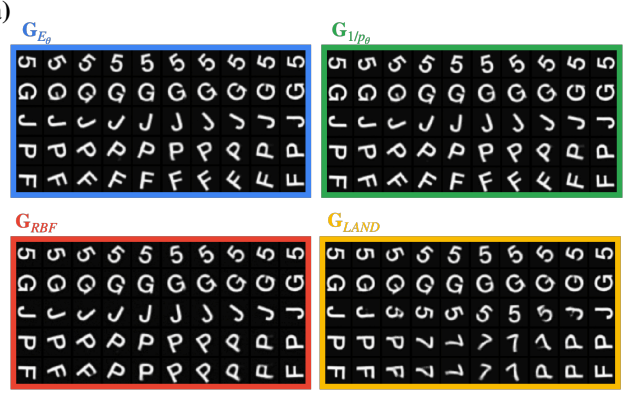


Figure 3: **Step size along geodesics in the WCG dataset**. Log-based metrics ($\mathbf{G}_{\mathbf{E}_\theta}$ and $\boldsymbol{G}_{E_\mathcal{M}}$) produce sharper variations, reflecting stronger sensitivity to density curvature.

4.3 Rotated Characters

We use an image dataset of seven rotated, non-symmetric characters in two variants: Uniform Rotated Characters (URC), with evenly distributed angles, and Biased Rotated Characters (BRC), concentrated near $0°$. In this subsection, all computations are done in the 64-dimensional latent space of a regularized autoencoder trained with a triplet loss, ensuring that small angular differences yield short latent distances. This setup provides a unique middle ground: although the underlying Riemannian metric is unknown, we can treat the smooth in-plane rotation between two instances of the same character as a proxy for the ground-truth geodesic. Thanks to the triplet loss, the latent space is structured so that nearby points correspond to slight rotations of the same character, making the shortest path between two orientations a meaningful approximation of the true geodesic in the task-relevant transformation space. Separate EBMs and interpolant networks are trained for each dataset variant. Full experimental details (datasets, architectures, and hyperparameters) are provided in Supp. E.

a)



b)

Metric	$\mathcal{D}$-RMSE (↓)	$\gamma^\star$-RMSE (↓)
linear interp.	2.96	3.52
$\mathbf{G}_{\mathbf{E}_\theta}$	**0.11**	**0.40**
$\mathbf{G}_{\mathbf{1/p}_\theta}$	0.14	0.44
$\mathbf{G}_{\mathbf{LAND}}$	0.66	2.39
$\mathbf{G}_{\mathbf{RBF}}$	0.36	0.86

Figure 4: **Geodesics on the URC dataset. (a)** Geodesics computed with different Riemannian metrics, projected into pixel space for visualization. $\mathbf{G}_{\mathbf{RBF}}$ and $\mathbf{G}_{\mathbf{LAND}}$ often deviate from the intended path, sometimes drifting toward other characters (e.g., the letter *F*). **(b)** Quantitative evaluation using two metrics: (i) $\mathcal{D}$-RMSE, which measures proximity to the dataset manifold, and (ii) γ-RMSE, which measures the deviation from an ideal smooth rotation. See Supp. E.6 for the 2-σ error.

In Fig. 4a, we visualize geodesics computed on the URC dataset, projected back into pixel space (see Supp. E.5 for additional results on both URC and BRC). EBM-based metrics ($\mathbf{G}_{\mathbf{E}_\theta}$ and $\mathbf{G}_{\mathbf{1/p}_\theta}$) yield smooth rotations that preserve character identity, while $\mathbf{G}_{\mathbf{RBF}}$ and especially $\mathbf{G}_{\mathbf{LAND}}$ often deviate from the intended trajectory. To illustrate these failures, Fig. 1 shows all geodesics projected into PCA space for a case involving the letter F. While $\mathbf{G}_{\mathbf{E}_\theta}$ and $\mathbf{G}_{\mathbf{1/p}_\theta}$ remain on the manifold of rotated F instances, linear interpolation cuts through low-density regions, and $\mathbf{G}_{\mathbf{RBF}}$ and $\mathbf{G}_{\mathbf{LAND}}$ drift toward other character classes. To quantify this, we use two metrics: $\mathcal{D}$-RMSE, which measures the average distance from each geodesic point to its nearest neighbor in the dataset—lower values indicate better adherence to the data manifold; and γ-RMSE, which evaluates how closely the geodesic follows an ideal smooth rotation between endpoints. All results are averaged over $1,000$ geodesics with random endpoint orientations. As shown in Fig. 4b, EBM-based metrics consistently outperform others; $\mathbf{G}_{\mathbf{RBF}}$ performs reasonably well, while $\mathbf{G}_{\mathbf{LAND}}$ shows large deviations on both metrics. Overall, these results suggest that EBM-based metrics scale more effectively to high-dimensional data than alternative approaches.

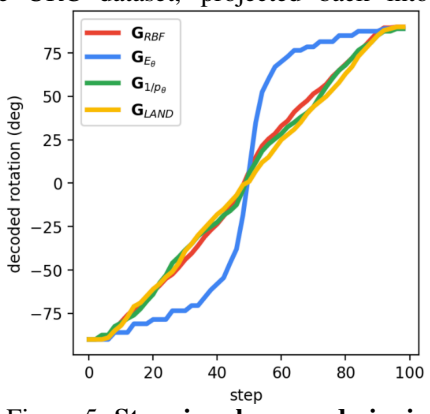


Figure 5: **Step size along geodesics in the WCG dataset**. Log-based metric ($\mathbf{G}_{\mathbf{E}_\theta}$) produces sharper variations, reflecting stronger sensitivity to density curvature.

As in the previous section, we examine how different metrics influence a geodesic's ability to follow the manifold's curvature. We focus on the BRC dataset, where orientations are biased toward 0°, creating sharper curvature near that region. To assess this, we decode the orientation at each time step along geodesics connecting fixed endpoints. As shown in Fig. 5, geodesics under $\mathbf{G}_{\mathbf{E}_\theta}$ rotate significantly faster near 0° than those under $\mathbf{G}_{\mathbf{1/p}_\theta}$ and $\mathbf{G}_{\mathbf{RBF}}$, reflecting stronger sensitivity to density variations.

At first glance, it may seem counterintuitive that trajectories following the geodesics move faster in high-density regions. However, this is consistent with minimizing the kinetic energy $\mathcal{E}$ in Eq.2, which enforces constant Riemannian speed (i.e., the quantity $||\dot{\gamma}(t)||_{\gamma(t)}$ is preserved along the trajectory) but not a constant Euclidean speed (i.e., $||\dot{\gamma}(t)||$ is not constant). Since EBM-derived metrics assign lower Riemannian cost in high-density regions, maintaining constant Riemannian speed requires moving faster in Euclidean terms through these regions. The faster rotation near 0°, observed in Fig.5 and Fig. 3, thus reflects the lower Riemannian cost of traveling through high-density regions. These results confirm and extend our previous findings: metrics based on energy (i.e., proportional to $-\log p$) more effectively capture the curvature of the data manifold.

4.4 Animal Faces

We now evaluate our method on the Animal Faces High Quality (AFHQ) dataset [59], using the latent space of the pretrained Stable Diffusion v1 VAE [18] (latent dimension: $4 \times 16 \times 16$). An EBM is trained to model the distribution of latent codes, and Algo. 1 is used to compute geodesics between a cat and a dog representation. We compare the resulting paths to two baselines: (i) linear interpolation and (ii) spherical interpolation (slerp) [60], which is known to better preserve the structure of VAE latent spaces under Gaussian priors (see Supp. F.6). Full experimental details are in Supp. F.

Fig. 6 illustrates geodesics computed in the latent space of a pretrained VAE and projected back into image space (see F.5 for additional samples as well as samples for $\mathbf{G}_{\mathbf{LAND}}$ and linear interpolation). Qualitatively, we observe that geodesics computed with the $\mathbf{G}_{\mathbf{1/p}_\theta}$ metric best adhere to the data manifold. The $\mathbf{G}_{\mathbf{E}_\theta}$ metric also shows noticeable improvements over the other metrics. Despite extensive tuning, $\mathbf{G}_{\mathbf{RBF}}$ and $\mathbf{G}_{\mathbf{LAND}}$ produce trajectories only slightly better than linear interpolation—suggesting these parametric metrics struggle to scale in high dimensions, consistent with prior findings [11, 13].

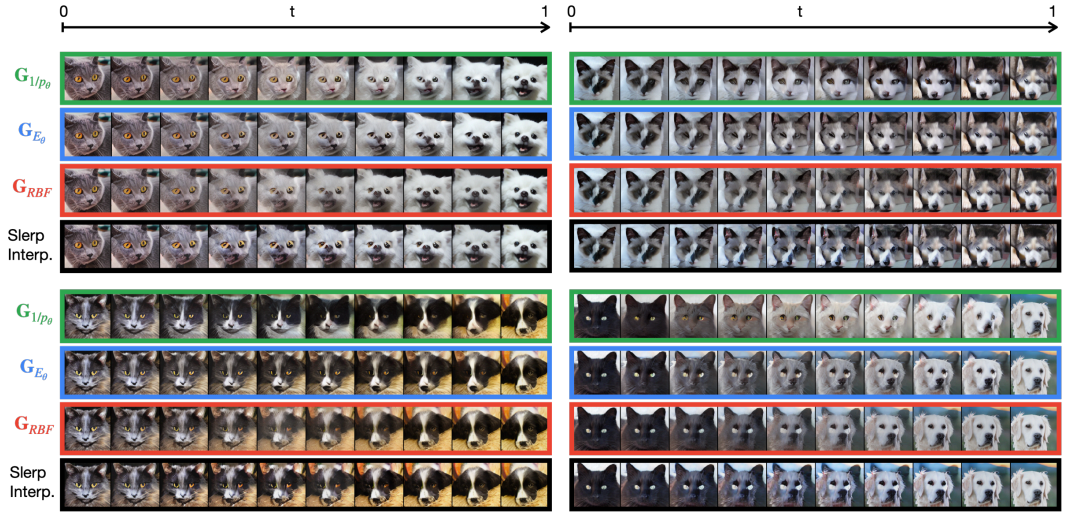


Figure 6: **Geodesics on the AFHQ dataset.** Each block shows an interpolated trajectory between two animal images (cats and dogs), projected back into image space for visualization. We compare geodesics computed with two EBM-based metrics ($\mathbf{G}_{\mathbf{1/p}_\theta}$, $\mathbf{G}_{\mathbf{E}_\theta}$), a parametric RBF-based metric ($\mathbf{G}_{\mathbf{RBF}}$), and spherical interpolation (slerp). Results using $\mathbf{G}_{\mathbf{LAND}}$, linear interpolation, and additional examples are provided in Supp. F.5.

To quantitatively assess geodesic quality, we report FID scores [61] in Table 1, computed over $50,000$ trajectories that interpolate from randomly chosen cat images to randomly chosen dog images. The results are consistent with qualitative observations: $\mathbf{G}_{\mathbf{1/p}_\theta}$ and $\mathbf{G}_{\mathbf{E}_\theta}$ yield the lowest FIDs, followed by the model-free slerp baseline, then $\mathbf{G}_{\mathbf{RBF}}$, $\mathbf{G}_{\mathbf{LAND}}$, and linear interpolation. Note that the FID measures how aligned individual samples are with the training distribution—on-manifold alignment—but does not assess whether the full trajectory respects the true manifold curvature. Unfortunately, AFHQ lacks ground-truth geometry for such evaluation.

Metric	FID (↓)
Linear interp.	42.47
Slerp interp.	32.67
$\mathbf{G}_{\mathbf{E}_\theta}$	20.79
$\mathbf{G}_{\mathbf{1/p}_\theta}$	**16.47**
$\mathbf{G}_{\mathbf{LAND}}$	39.17
$\mathbf{G}_{\mathbf{RBF}}$	37.98

Table 1: **FID along geodesics for different Riemannian metrics**. FID is computed at each trajectory point to assess on-manifold alignment. See Supp. F.4 for the 2-σ error.

5 Conclusion

In this work, we use pretrained Energy-Based Models (EBMs) to derive conformal Riemannian metrics, $\mathbf{G}_{\mathbf{E}_\theta}$ and $\mathbf{G}_{\mathbf{1/p}_\theta}$, and we compare them to established alternatives ($\mathbf{G}_{\mathbf{LAND}}$ [11] and

$\mathbf{G}_{\mathbf{RBF}}$ [13]). On both synthetic and high-dimensional data, EBM-derived metrics yield geodesics that stay closer to the data manifold and better capture its curvature—especially with $\mathbf{G}_{\mathbf{E}_\theta}$.

We focus on conformal metrics, which scale the identity by a scalar field to encode density. While more complex, non-conformal and anisotropic metrics (e.g., the Stein metric [25]) are accessible from the EBM score, we found that conformal metrics offer comparable performance with simpler interpretation and reduced computational cost, justifying our focus in this work. Future work may explore these extensions with regularization or structural priors to ensure smoothness and scalability (See Supp.G for a discussion of limitations and Supp.H for broader impact). To keep computational cost manageable, we train the EBM in the latent space of a pretrained autoencoder and compute geodesics using finite-difference optimization, two design choices that substantially reduce complexity and memory use without compromising performance.

Although this article is primarily methodological, it points to promising applications. One example is the mental rotation task, in which humans mentally rotate objects to match a target [62]. In such experiments, reaction times tend to decrease with training [63], suggesting that repeated exposure sharpens internal representations around training examples. These refined representations may concentrate in high-density regions, where mental transformations occur more quickly. As shown in Fig. 3 and 5, our geodesics naturally accelerate in such high-density regions, echoing these psychophysical findings. Modeling mental simulation as geodesics on Riemannian manifolds shaped by a generative model offers a principled computational framework to understand human cognition. It provides a way to formalize and test the hypothesis that the human cognition relies on generative models to support flexible inference [64–68]. Our approach is also particularly relevant for neuroscience, where datasets are high-dimensional, often sparsely sampled, and where understanding the geometry of neural population activity is central to scientific insight. In such settings, high-fidelity geodesics are essential for capturing the true structure of neural trajectories—approximations may distort the manifold and lead to misinterpretation of brain dynamics. While training EBMs is costly, the benefits in terms of interpretability and geometric accuracy make this approach compelling for applications where precision is critical.

As machine learning models are increasingly used to capture complex data distributions, understanding the geometry of their latent spaces becomes essential. Our work contributes to this effort by showing that geometry can serve as a useful tool for building models that better reflect data structure, align with human perception, and shed light on cognitive processes.

Acknowledgments

This work was supported by ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANR-19-PI3A-0004). Part of this work was carried out within the DEEL project, which is part of IRT Saint Exupéry and the ANITI AI cluster. The authors acknowledge the financial support from DEEL's industrial and academic members and the "France 2030" program (NR-10-AIRT-01 and ANR-23-IACL-0002). Additional support for TS provided by ONR (N00014-24-1-2026 and REPRISM MURI N00014-24-1-2603) and NSF (IIS-2402875).

References

[1] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[2] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 17–32. Springer, 2014.

[3] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9*, pages 589–600. Springer, 2006.

[4] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[5] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

[6] Hongsong Feng, Sean Cottrell, Yuta Hozumi, and Guo-Wei Wei. Multiscale differential geometry learning of networks with applications to single-cell rna sequencing data. *Computers in Biology and Medicine*, 171:108211, 2024.

[7] Kazuya Horibe, Gentaro Taga, and Koichi Fujimoto. Geodesic theory of long association fibers arrangement in the human fetal cortex. *Cerebral Cortex*, 33(17):9778–9786, 2023.

[8] Peter D Neilson, Megan D Neilson, and Robin T Bye. A riemannian geometry theory of three-dimensional binocular visual perception. *Vision*, 2(4):43, 2018.

[9] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 2. Springer, 1992.

[10] Søren Hauberg, Oren Freifeld, and Michael Black. A geometric take on metric learning. *Advances in Neural Information Processing Systems*, 25, 2012.

[11] Georgios Arvanitidis, Lars K Hansen, and Søren Hauberg. A locally adaptive normal distribution. *Advances in Neural Information Processing Systems*, 29, 2016.

[12] Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. *Advances in neural information processing systems*, 33:442–453, 2020.

[13] Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. Geometrically enriched latent spaces. *arXiv preprint arXiv:2008.00565*, 2020.

[14] Samuel Gruffaz and Josua Sassen. Riemannian metric learning: Closer to you than you imagine. *arXiv preprint arXiv:2503.05321*, 2025.

[15] Guy Lebanon. Learning riemannian metrics. *arXiv preprint arXiv:1212.2474*, 2012.

[16] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

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

[18] 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.

[19] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009.

[20] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[21] C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 235–247. Springer, 1992.

[22] Shun-Ichi Amari. A foundation of information geometry. *Electronics and Communications in Japan (Part I: Communications)*, 66(6):1–10, 1983.

[23] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

[24] Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 37: 135011–135042, 2025.

[25] Christian S. Perone. The geometry of data: the missing metric tensor and the stein score [part ii]. `https://blog.christianperone.com/2024/11/the-geometry-of-data-part-ii/`, November 2024. Terra Incognita.

[26] Willem Diepeveen, Georgios Batzolis, Zakhar Shumaylov, and Carola-Bibiane Schönlieb. Score-based pullback riemannian geometry. *arXiv preprint arXiv:2410.01950*, 2024.

[27] Georgios Arvanitidis, Miguel González-Duque, Alison Pouplin, Dimitris Kalatzis, and Søren Hauberg. Pulling back information geometry. *arXiv preprint arXiv:2106.05367*, 2021.

[28] Hadi Beik-Mohammadi, Søren Hauberg, Georgios Arvanitidis, Gerhard Neumann, and Leonel Rozo. Learning riemannian manifolds for geodesic motion skills. *arXiv preprint arXiv:2106.04315*, 2021.

[29] Dimitris Kalatzis, David Eklund, Georgios Arvanitidis, and Søren Hauberg. Variational autoencoders with riemannian brownian motion priors. *arXiv preprint arXiv:2002.05227*, 2020.

[30] Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim GJ Rudner, and Smita Krishnaswamy. Geometry-aware generative autoencoder for warped riemannian metric learning and generative modeling on data manifolds. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[31] Georgios Arvanitidis, Soren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1506–1515. PMLR, 2019.

[32] Nutan Chen, Francesco Ferroni, Alexej Klushyn, Alexandros Paraschos, Justin Bayer, and Patrick van der Smagt. Fast approximate geodesics for deep generative models. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Deep Learning: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part II 28*, pages 554–566. Springer, 2019.

[33] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*, 2018.

[34] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

[35] Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen, Evangelos A Theodorou, and Weilie Nie. Augmented bridge matching. *arXiv preprint arXiv:2311.06978*, 2023.

[36] Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrodinger bridge. In *International conference on machine learning*, pages 10794–10804. PMLR, 2021.

[37] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrodinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

[38] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Friso de Kruiff, Erik Bekkers, Ozan Öktem, Carola-Bibiane Schönlieb, and Willem Diepeveen. Pullback flow matching on data manifolds. *arXiv preprint arXiv:2410.04543*, 2024.

[40] 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.

[41] James Thornton, Michael Hutchinson, Emile Mathieu, Valentin De Bortoli, Yee Whye Teh, and Arnaud Doucet. Riemannian diffusion schrodinger bridge. *arXiv preprint arXiv:2207.03024*, 2022.

[42] Gabriel Peyre, Mickael Pechaud, Renaud Keriven, Laurent D Cohen, et al. Geodesic methods in computer vision and graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(3–4):197–397, 2010.

[43] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[44] Zhifeng Kong and Kamalika Chaudhuri. The expressive power of a class of normalizing flow models. In *International conference on artificial intelligence and statistics*, pages 3599–3609. PMLR, 2020.

[45] Felix Draxler, Peter Sorrenson, Lea Zimmermann, Armand Rousselot, and Ullrich Köthe. Free-form flows: Make any architecture a normalizing flow. In *International Conference on Artificial Intelligence and Statistics*, pages 2197–2205. PMLR, 2024.

[46] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

[47] Keegan Kelly, Lorena Piedras, Sukrit Rao, and David Roth. Variations and relaxations of normalizing flows. *arXiv preprint arXiv:2309.04433*, 2023.

[48] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.

[49] James Thornton, Louis Béthune, Ruixiang ZHANG, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Controlled generation with distilled diffusion energy models and sequential monte carlo. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[50] Yilun Du, Toru Lin, and Igor Mordatch. Model-based planning with energy-based models. In *Conference on Robot Learning*, pages 374–383. PMLR, 2020.

[51] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[52] Oliver Woodford. Notes on contrastive divergence. *Department of Engineering Science, University of Oxford, Tech. Rep*, 4, 2006.

[53] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5272–5280, 2020.

[54] David Duvenaud, Jacob Kelly, Kevin Swersky, Milad Hashemi, Mohammad Norouzi, and Will Grathwohl. No mcmc for me: Amortized samplers for fast and stable training of energy-based models. In *International Conference on Learning Representations (ICLR)*, 2021.

[55] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021.

[56] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33: 21994–22008, 2020.

[57] Søren Hauberg. Only bayes should learn a manifold (on the estimation of differential geometric structure from data). *arXiv preprint arXiv:1806.04994*, 2018.

[58] Georgios Arvanitidis, Bogdan M Georgiev, and Bernhard Schölkopf. A prior-based approximate latent riemannian metric. In *International Conference on Artificial Intelligence and Statistics*, pages 4634–4658. PMLR, 2022.

[59] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[60] Karpathy Andrej. Walk in stable diffusion. `https://gist.github.com/karpathy/00103b0037c5aaea32fe1da1af553355`, 2022.

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

[62] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[63] Lynn A Cooper and Roger N Shepard. Chronometric studies of the rotation of mental images. In *Visual information processing*, pages 75–176. Elsevier, 1973.

[64] Karl Friston. A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3): 70–87, 2006.

[65] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[66] Victor Boutin, Angelo Franciosini, Frédéric Chavane, and Laurent U Perrinet. Pooling strategies in v1 can account for the functional and structural diversity across species. *PLOS Computational Biology*, 18(7):e1010270, 2022.

[67] Victor Boutin, Lakshya Singhal, Xavier Thomas, and Thomas Serre. Diversity vs. recognizability: Human-like generalization in one-shot generative models. *Advances in Neural Information Processing Systems*, 35:20933–20946, 2022.

[68] Victor Boutin, Rishav Mukherji, Aditya Agrawal, Sabine Muzellec, Thomas Fel, Thomas Serre, and Rufin Van-Rullen. Latent representation matters: Human-like sketches in one-shot drawing tasks. In *38th Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[69] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[70] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

[71] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[72] Richard D Lange, Devin Kwok, Jordan Kyle Matelsky, Xinyue Wang, David Rolnick, and Konrad Kording. Deep networks as paths on the manifold of neural representations. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 102–133. PMLR, 2023.

[73] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2): 255–308, 2009.

[74] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70(1-39):19, 2012.

[75] Olympio Hacquard and Vadim Lebovici. Euler characteristic tools for topological data analysis. *Journal of Machine Learning Research*, 25(240):1–39, 2024.

[76] Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

[77] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356, 2004.

[78] Herbert Edelsbrunner and Dmitriy Morozov. *Persistent homology: theory and practice*. eScholarship, University of California, 2013.

[79] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.

[80] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[81] Taco S Cohen and Max Welling. Steerable cnns. In *International Conference on Learning Representations*, 2017.

[82] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pages 1321–1330. PMLR, 2019.

[83] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.

[84] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[85] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022.

[86] Cédric Rommel, Thomas Moreau, and Alexandre Gramfort. Deep invariant networks with differentiable augmentation layers. *Advances in Neural Information Processing Systems*, 35: 35672–35683, 2022.

[87] Jianke Yang, Robin Walters, Nima Dehmamy, and Rose Yu. Generative adversarial symmetry discovery. In *International Conference on Machine Learning*, pages 39488–39508. PMLR, 2023.

[88] Moshe Lichtenstein, Gautam Pai, and Ron Kimmel. Deep eikonal solvers. In *Scale Space and Variational Methods in Computer Vision: 7th International Conference, SSVM 2019, Hofgeismar, Germany, June 30–July 4, 2019, Proceedings 7*, pages 38–50. Springer, 2019.

[89] Qijian Zhang, Junhui Hou, Yohanes Adikusuma, Wenping Wang, and Ying He. Neurogf: A neural representation for fast geodesic distance and path queries. *Advances in Neural Information Processing Systems*, 36:19485–19501, 2023.

[90] Louis Béthune, Paul Novello, Guillaume Coiffier, Thibaut Boissin, Mathieu Serrurier, Quentin Vincenot, and Andres Troya-Galvis. Robust one-class classification with signed distance function using 1-lipschitz neural networks. In *International Conference on Machine Learning*, pages 2245–2271. PMLR, 2023.

[91] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[92] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[93] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[94] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[95] Zengyi Li, Yubei Chen, and Friedrich T Sommer. Learning energy-based models in high-dimensional spaces with multiscale denoising-score matching. *Entropy*, 25(10):1367, 2023.

[96] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *International Conference on Machine Learning*, pages 3732–3747. PMLR, 2020.

[97] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[98] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in artificial intelligence*, pages 574–584. PMLR, 2020.

[99] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[100] Tobias Schroder, Zijing Ou, Jen Lim, Yingzhen Li, Sebastian Vollmer, and Andrew Duncan. Energy discrepancies: a score-independent loss for energy-based models. *Advances in Neural Information Processing Systems*, 36:45300–45338, 2023.

[101] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015.

[102] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=S1g7tpEYDS`.

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

[104] Barrett O'neill. *Semi-Riemannian geometry with applications to relativity*, volume 103. Academic press, 1983.

[105] Joel W Robbin and Dietmar A Salamon. *Introduction to differential geometry*. Springer Nature, 2022.

A Extended related work

Several tools have been developed to study the geometrical properties of distributions. We survey some prominent approaches below.

Information geometry, initiated by the seminal works of [21, 22], was the first to apply ideas from differential geometry to the field of statistics. Unlike our present work, the goal was not to understand the geometry of the data $\boldsymbol{x}$, but rather to understand the geometry of a smooth manifold $\theta \in \Theta$ of parameters of an estimator p_θ. In particular, starting from the Taylor expansion of *reverse* Kullback-Leibler [69] divergence to $\boldsymbol{p_\theta}$, in the neighborhood of p_θ itself, with $\theta' = \theta + \epsilon$, we get

$$D_{KL}(p_{\theta'} \| \boldsymbol{p_\theta}) \approx \underbrace{D_{KL}(p_\theta \| \boldsymbol{p_\theta})}_{=0} + \underbrace{\nabla_1 D_{KL}(p_\theta \| \boldsymbol{p_\theta})}_{=\mathbf{0}} \epsilon + \epsilon^T \nabla_1^2 D_{KL}(p_\theta \| \boldsymbol{p_\theta}) \epsilon. \tag{10}$$

One can show that, since $\theta' = \theta$ is the global minimum of this function, the first-order term vanishes and the second-order term $\nabla_1^2 D_{KL}(p_\theta \| \boldsymbol{p_\theta})$ must be a positive definite form - i.e an inner product. This quantity, called *Fisher information* [23], gives Θ the structure of a Riemannian manifold. The Riemannian gradient associated with this manifold yields a second-order optimization method coined *natural gradient descent* [70], that has been proved helpful in deep learning [71]. Our method inherits some spirit of this approach, since we define a local inner product as a function of the density, to give a Riemannian structure to the data manifold. However, we focus on the geometry of the *data* $\boldsymbol{x}$, not the geometry of the model's parameters $\boldsymbol{\theta}$.

Riemannian structure of data manifolds has already been proposed in the past. For example, the seminal LAND metric [11] is a non-conformal metric built from the samples, with the intent of generalizing multivariate normal distributions to manifolds. The RBF metric [13] is a conformal metric, derived from a kernel density estimator, with some learnable coefficients. More recently, Kapusniak et al. [24] proposed to use those metrics and learn a flow matching algorithm to fit geodesics in the data manifold. The Jacobian of a generative model also defines a metric [33]. The (unpublished) work of Perone [25] has been inspirational for our contribution. They use the Stein score function to build the metric, an approach also chosen by [26] - although restricted to unimodal densities.

Pullback geometry of latent manifolds is an active research area. [72] studies the manifold of representations of a given network, while [30] builds a generative autoencoder to represent the manifold. Shortest paths are computed with fixed-point methods [31], or using a discrete graph [32]. While we may rely on the latent space of a VAE for some challenging tasks, studying latent representations of a neural network is beyond the scope of our work.

On-manifold generative models can be found in the literature. For example, we can mention flow and bridge matching approaches [34, 35], which learn a flow between a source and a target distribution, including on Riemannian manifolds [38, 39]. In particular, the Schrödinger bridge [36, 37] focuses on an optimization problem involving paths in the space of probability distributions, and was also generalized to non-Euclidean geometries [40, 41]. These works differ significantly from ours: they assume the Riemannian manifold to be given, not chosen, and they build a generative model on top of it. To the contrary, given a special class of generative models to represent the data, we *choose* the metric to build the manifold.

Topological data analysis [73, 74] studies the topological properties of the data manifold. This field aims to estimate some topological invariants such as the Euler characteristic [75] and persistent Betti numbers [76] (which are the number of connected components, number of closed loops, etc.) from a finite sample. It relies on tools such as persistent homology [77–79] to design algorithms. This approach typically focuses on the *global* properties: it assumes that the data accumulate on a well-defined manifold, from which these high-level features must be computed. To the contrary, our approach focuses on the *local* structure defined by the metric, while the global structure is inherited from the induced geodesics. Furthermore, we consider the whole ambient space for our manifold, tweaking only the metric to account for low-density regions.

Symmetries and geometry in representations have gathered considerable attention from the deep learning community, warranting no fewer than 3 workshops at Neurips alone [4]. Symmetries are

[4] `https://www.neurreps.org`

operations under which a structure is left invariant, or equivariant. In particular, some neural architectures are leveraged to reflect priors about the underlying symmetries of the data [80–84]. In other cases, symmetries are discovered and learned from observations [85–87]. Unlike these approaches, we do not seek symmetries in data, and we make minimal assumptions about the model; we are mainly interested in the density to build the structure.

Non-Euclidean 2D and 3D manifolds are first-class citizens in computer graphics. The works of [88, 89] define a way to find shortest paths over such manifolds. However, this requires solving the Eikonal equation, which is prohibitively expensive in high dimensions or restricted to Euclidean geometries [90]. Geodesics can be learned, but this is restricted to low dimensions [89]. These setups are beyond the scope of our work, as we focus on higher-dimensional and sparsely populated spaces, and no discrete meshes can be built from samples.

Metric learning (or *distance learning*) is another field whose purpose is to learn a distance function between samples, typically in a weakly-supervised manner with contrastive losses [91–93]. Often, these distances cannot be realized as a geodesic distance and are intended for a specific task, like classification or retrieval.

B Energy-Based Model

B.1 Derivation of the Gradient of the EBM Log-Likelihood

The demonstration below is adapted from [52] to fit our notation. Even though this mathematical derivation is not crucial for a good understanding of our work, we include it to make sure our article is self-contained and complete.

We consider an Energy-Based Model (EBM) defining a probability distribution via the Boltzmann form:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)} \quad \text{with} \quad Z(\theta) = \int \exp(-E_\theta(\mathbf{x}))\, d\mathbf{x}.$$

Our goal is to minimize the negative log-likelihood with respect to the empirical data distribution $p_{\mathcal{D}}$:

$$\mathcal{L}_{\text{ML}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}[-\log p_\theta(\mathbf{x})].$$

We first expand the log-probability:

$$-\log p_\theta(\mathbf{x}) = E_\theta(\mathbf{x}) + \log Z(\theta).$$

Taking the gradient with respect to θ:

$$\nabla_\theta \mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_\theta E_\theta(\mathbf{x}) + \nabla_\theta \log Z(\theta)\right].$$

The derivative of the log-partition function could be simplified:

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_\theta Z(\theta) \\
&= \frac{1}{Z(\theta)} \nabla_\theta \int \exp(-E_\theta(\mathbf{x}))\, d\mathbf{x} \\
&= -\frac{1}{Z(\theta)} \int \exp(-E_\theta(\mathbf{x})) \nabla_\theta E_\theta(\mathbf{x})\, d\mathbf{x} \\
&= -\int p_\theta(\mathbf{x}) \nabla_\theta E_\theta(\mathbf{x})\, d\mathbf{x} \\
&= -\mathbb{E}_{\mathbf{x}\sim p_\theta}\left[\nabla_\theta E_\theta(\mathbf{x})\right].
\end{aligned}$$

Substituting this back into the gradient of the loss:

$$\nabla_\theta \mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_\theta E_\theta(\mathbf{x})\right] - \mathbb{E}_{\mathbf{x}\sim p_\theta}\left[\nabla_\theta E_\theta(\mathbf{x})\right].$$

In practice, we denote the $\mathbf{x}^+$ the "positive" samples from the empirical data distribution p_D, and $\mathbf{x}^-$ the "negative" samples from the model:

$$\nabla_\theta \mathcal{L}_{\text{ML}} \approx \mathbb{E}_{\mathbf{x}^+\sim p_{\mathcal{D}}}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^-\sim p_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right].$$

B.2 EBM training algorithm

To train our Energy-Based Models (EBMs), we follow the approach of [16]. Algo. 2 details the general training procedure:

Algorithm 2: Training Energy-Based Model using Langevin Dynamics

Input: Training dataset :$\mathcal{D}$, learning rate η, Replay Buffer $\mathcal{B}$, Langevin step size α, noise scale σ, number of Langevin steps L

while Training **do**

- $\mathbf{x}^+ \sim \mathcal{D}$ sample from the dataset
- $\mathbf{x}^0 \sim \mathcal{B}$ # sample from a replay buffer with probability 95%
- ## Refine negative samples using Langevin dynamics
- **for** $t \leftarrow 1$ **to** L **do**
  - $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t - \alpha \nabla_{\mathbf{x}^t} E_\theta(\mathbf{x}^t) + \omega \quad$ with $\omega \sim \mathcal{N}(0, \sigma)$
- $\mathbf{x}^- = \mathbf{x}^L$.detach()
- $\nabla_\theta \mathcal{L}_{\text{ML}} \approx \mathbb{E}_{\text{Batch}} \left[\nabla_\theta E_\theta(\mathbf{x}_i^+) - \nabla_\theta E_\theta(\mathbf{x}_i^-) \right]$ ## Compute the ML loss
- $\mathcal{L}_{REG}(\theta) = \mathbb{E}_{\text{Batch}} \left[\nabla_\theta E_\theta(\mathbf{x}_i^+)^2 + \nabla_\theta E_\theta(\mathbf{x}_i^-)^2 \right]$ ## Compute Regularization loss
- $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{ML}} - \eta \nabla_\theta \mathcal{L}_{REG}$## update parameters with gradient descent
- $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{x}^+$

In all experiments, we use $L = 100$ Langevin steps with step size $\alpha = 1$ and noise scale $\sigma = 10^{-2}$. The energy function is optimized using the Adam optimizer [94] with a learning rate of $\eta = 10^{-4}$. In addition to the maximum likelihood (ML) loss, we include a regularization term that encourages the energy values to remain close to zero, a technique shown to be effective in prior work [16].

We observed that training can be unstable, particularly for high-dimensional datasets. We attribute this instability to the lack of gradient supervision: the loss is not backpropagated through the Langevin dynamics to reduce memory usage. To mitigate this, we introduce a small Denoising Score Matching (DSM) loss—only for the AFHQ dataset—which provides weak supervision of the energy gradient. This additional regularization loss is similar to the DSM loss in [95]. We found this trick to strongly improve stability without degrading the performance.

The energy network architecture is adapted to the complexity of each dataset. Full details are provided in Appendix D.2, E.3, and F.2. Following Li et al. [95], we design the output layer of the energy function to take a quadratic form.

B.3 Other training procedure in literature

EBM can also be trained by minimizing the so-called *Stein discrepancy* [96], Denoising Score Matching [97], Sliced Score Matching [98], Noise Contrastive Estimation [99]. A related objective to contrastive divergence is *energy discrepancy* [100]. We refer the reader [20], for a complete review of the different methods to train EBMs.

C Riemannian Metrics

C.1 Calibration

We normalize each metric using calibration coefficients α and β, with two goals: (i) ensuring that the Riemannian metric averages to the identity matrix $\mathbf{I}$ on the manifold, and (ii) aligning the overall scale of all metrics to allow fair comparisons. Here are more details on the calibration procedure:

First, we randomly sample data pairs $(\mathbf{x}_0, \mathbf{x}_1)$ from the dataset $\mathcal{D}$ (it corresponds to the geodesics endpoints) and generate linear interpolations between them using:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1 \tag{11}$$

Second, we define two sets of samples: $\mathcal{S}_{\mathcal{M}}$, which contains the endpoints $\mathbf{x}_0$ and $\mathbf{x}1$ *lying on the data manifold*, and $\mathcal{S}_{\bar{\mathcal{M}}}$, which contains the midpoints at $t = \frac{1}{2}$. These sets are then used to estimate the calibration coefficients α and β:

$$\mathbf{G}(\mathbf{x}) = \alpha\,\mathbf{h}(\boldsymbol{x}) + \beta \quad \text{s.t} \quad \begin{cases} \alpha = \dfrac{g_{\max} - g_{\min}}{\frac{1}{|\mathcal{S}_{\bar{\mathcal{M}}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\bar{\mathcal{M}}}} h(\mathbf{x}) - \frac{1}{|\mathcal{S}_{\mathcal{M}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\mathcal{M}}} h(\mathbf{x})} \\ \beta = g_{\min} - \alpha \cdot \frac{1}{|\mathcal{S}_{\mathcal{M}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\mathcal{M}}} h(\mathbf{x}) \end{cases}$$

$$\mathbf{G}(\mathbf{x}) = (\alpha\,\mathbf{h}(\boldsymbol{x}) + \beta)^{-1} \quad \text{s.t} \quad \begin{cases} \alpha = \dfrac{1/g_{\max} - 1/g_{\min}}{\frac{1}{|\mathcal{S}_{\bar{\mathcal{M}}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\bar{\mathcal{M}}}} h(\mathbf{x}) - \frac{1}{|\mathcal{S}_{\mathcal{M}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\mathcal{M}}} h(\mathbf{x})} \\ \beta = \dfrac{1}{g_{\min}} - \alpha \cdot \frac{1}{|\mathcal{S}_{\mathcal{M}}|}\sum_{\mathbf{x}\in\mathcal{S}_{\mathcal{M}}} h(\mathbf{x}) \end{cases}$$

This calibration strategy adjusts the metric based on both on-manifold and off-manifold regions. It ensures that all metrics operate within a comparable dynamic range and promotes a useful geometric prior: lower metric values near the data manifold and higher values farther away. As a result, geodesics are encouraged to stay close to high-density areas, aligning the geometry with the data distribution.

C.2 LAND metric

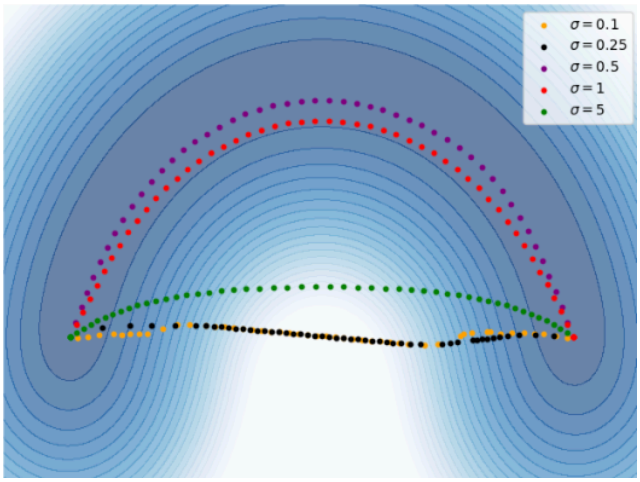


Figure 7: **Effect of the bandwidth** σ on the geodesics obtained with the LAND metric. Here we have explored five σ values ($\sigma \in \{0.1, 0.25, 0.5, 1, 5\}$). We observed that σ has a major impact on the shape of the geodesics.

We remind the land metric formula (see Eq. 8):

$$\mathbf{G}_{\text{LAND}}(\mathbf{x}) = (\alpha\,\text{diag}(\mathbf{h}(\mathbf{x})) + \beta\mathbf{I})^{-1} \text{ s.t } h^{(j)}(\mathbf{x}) = \sum_{i=1}^{N}(x_i^{(j)} - x^{(j)})^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x_i}||^2}{2\sigma^2}\right) \tag{12}$$

This metric is highly sensitive to the choice of the σ parameter, which controls the "locality" of the metric. A small σ results in a very local metric that is strongly influenced by nearby points, while a large σ smooths the metric by averaging over a wider region. This directly affects the trade-off between how closely geodesics follow the data manifold and how smooth or stable they are. In practice, we observe that σ has a major impact on the shape of the geodesics, as shown in Fig.7, confirming earlier findings by[11]. To illustrate this, we plot geodesics for five different values of σ ($\sigma \in \{0.1, 0.25, 0.5, 1, 5\}$) and find that they closely follow the data manifold only within a narrow range, particularly around $\sigma = 0.5$.

C.3 RBF metric

We first remind the RBF formula :

$$\mathbf{G}_{\text{RBF}}(\mathbf{x}) = (\alpha \cdot h(\mathbf{x}) + \beta)^{-1} \cdot \mathbf{I}, \quad h(\mathbf{x}) = \sum_{k=1}^{K} w_k \exp\left(-0.5 \cdot \lambda_k \|\mathbf{x} - \hat{\mathbf{x}}_k\|^2\right).$$

In the equation, the $\{\hat{\mathbf{x}}\}_{i=1}^{K}$ are centroids evaluated using a K-Means algorithm. Following [13], the bandwidth (λ_k) using the inter-distance to prototype (see Eq. 13):

$$\lambda_k = \frac{1}{2}\left(\frac{\kappa}{2K}\sum_{k=1}^{K}||\mathbf{x} - \hat{\mathbf{x}}_\mathbf{k}||\right)^{-2} \tag{13}$$

The bandwidth, λ_k, controls the spatial extent of each radial basis function. In Eq. 13, κ is a tunable hyperparameter controlling how concentrated or spread out the RBFs are. Intuitively, a larger κ results in narrower kernels (stronger locality) while a smaller one yields wider coverage. This trade-off is explored via hyperparameter search. The weights w_k modulate the relative contribution of each RBF to the resulting scalar field. These weights are optimized to ensure that $h(x)$ remains close to 1 on the training data, using the following loss:

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}||1 - h(\mathbf{x_i})||^2 \tag{14}$$

This encourages the RBF combination to approximate a constant value (here, 1) across the data distribution, ensuring consistency and stability of the field on the manifold.

In Fig. 8, we evaluate how the number of centroids K affects the shape of the geodesics. The results show that geodesics are highly sensitive to this parameter. When K is too small, the geodesics fail to follow the data manifold accurately. Conversely, when K is too large, the trajectories become overly sinuous—passing through many centroids that are not necessarily aligned with the true manifold.

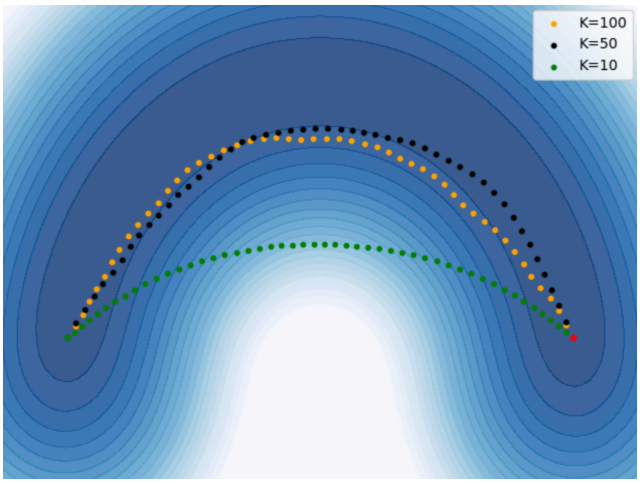


Figure 8: **Effect of the number of centroids** K on the geodesics obtained with the RBF metric ($K \in \{10, 50, 100\}$). We observed that K has a major impact on the shape of the geodesics.

D Experimental details on the Circular Mixture of Gaussian datasets

D.1 Datasets

To design our toy datasets, we have used a mixture of K (2D) Gaussians. Specifically, $K = 200$ in all our datasets. The resulting probability distribution is therefore:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \mathbf{I}), \tag{15}$$

where $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \mathbf{I})$ denotes a 2D isotropic Gaussian centered at $\boldsymbol{\mu}_k$. Here, $\mathbf{I}$ is the identity matrix of size 2×2. In both datasets, the centers of the Gaussians are uniformly positioned along a semi-circle or Radius R (here $R = 8$). Specifically, the centers are given by:

$$\boldsymbol{\mu}_k = R \cdot \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix} \quad \text{with} \quad \theta_k = \frac{k}{K} \cdot \pi, \quad k = 0, \ldots, K-1. \tag{16}$$

The only difference between the Uniform Circular Gaussian (UCG) dataset and the Weighted Circular Gaussian dataset (WCG) is the weighting coefficient $\{\pi_k\}_{k=1}^{K}$

Uniform Circular Gaussian dataset. Here all the weights are similar and equal to $1/K$. As a result, the energy landscape forms a semi-circular basin with constant depth (see contour plot of Fig. 2a for an illustration of the energy landscape).

Weighted Circular Gaussian dataset. In this setting, the mixture weights vary, concentrating the distribution toward the center of the arc. The weights are symmetric with respect to the horizontal axis, producing an energy landscape with a semi-circular shape and slopes symmetric around the arc's midpoint (see the contour plot in Fig.2c). Fig.9 shows the weights π_k as a function of orientation, with all weights summing to 1. This setup generates a curved, non-uniform density with higher mass near the center of curvature (i.e., at 90 degrees), allowing us to introduce a controlled curvature in the data manifold and assess how well different metrics capture it.

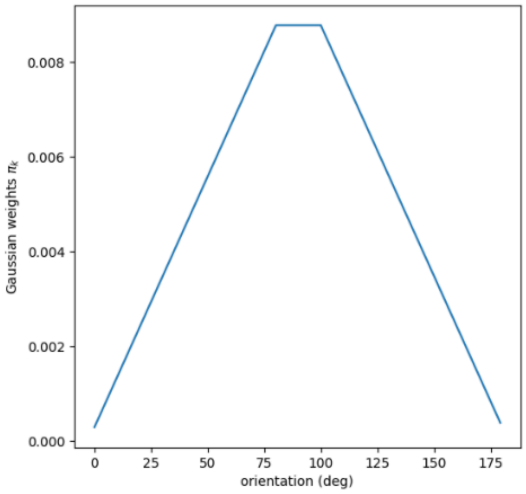


Figure 9: **Profile of the Gaussians weights π_k**

D.2 Neural networks architectures and Hyperparameters on the Circular Mixture of Gaussian Dataset

Here, we describe the architecture of the energy function (see Table 2), the interpolant network (see Table 3), and the hyperparameters used for the G_{LAND} and G_{RBF} metrics. Note that the architectures and settings are the same for both the UCG and WCG datasets.

Energy-Based Model Table 2 summarizes the architecture used for the energy function of the EBM. The output is designed to follow a quadratic form, similar to the approach in [95], which we found improves performance across all datasets. To assess whether the EBM successfully learns the target distribution, we visualize the learned energy landscapes for both the UCG and WCG datasets (see Fig. 10a and Fig. 10b, respectively). For reference, we also include the ground-truth energy landscapes of the target distributions (see Fig. 10c and Fig. 10d for UCG and WCG, respectively). We observe that the EBM accurately captures the overall shape of the energy landscape for both distributions. However, in the WCG dataset, the true energy spans a broader range than the EBM's learned energy. This discrepancy is partially corrected by the normalization procedure described in Appendix C.1.

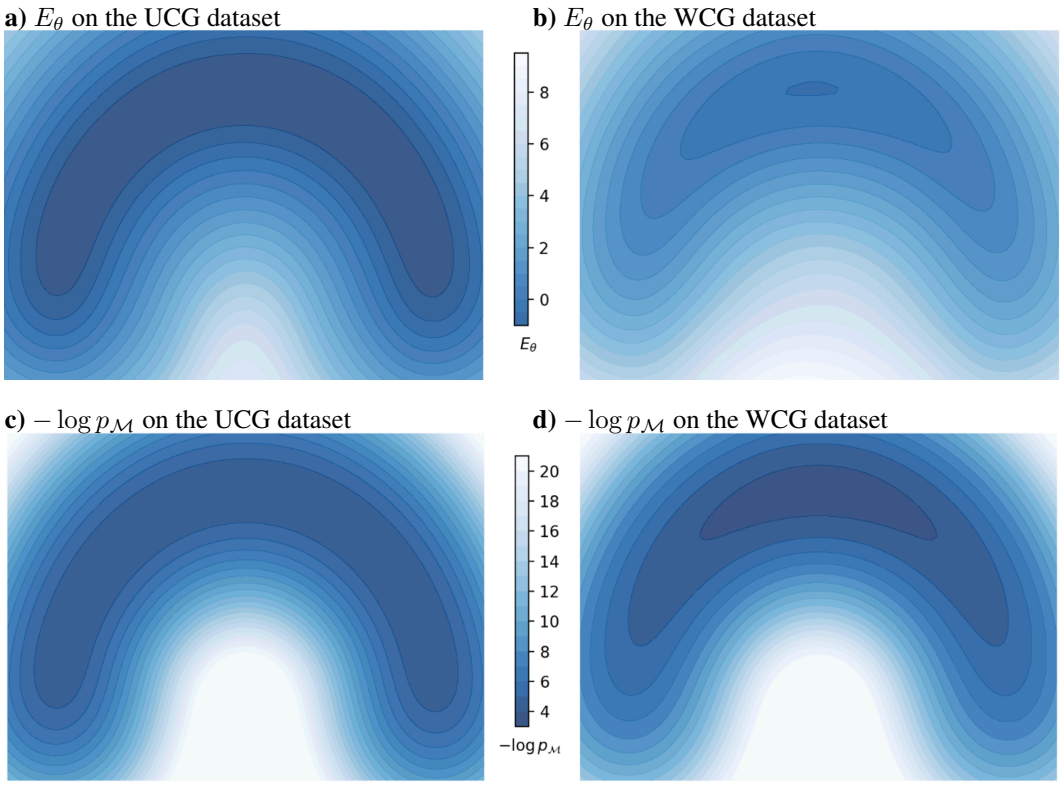


Figure 10: **Energy Landscape on the UCG and WCG datasets. (a, b)** shows the energy landscape learned by the EBMs on the UCG and WCG datasets, respectively. **(c, d)** shows the true energy landscape (i.e., $-\log p_{\mathcal{M}}$) on the UCG and WCG datasets, respectively.

Interpolant Network Table 3 summarizes the architecture used for the interpolant network (i.e., $\varphi_{t,\eta}$ in Algo. 1 and Eq. 3). For all datasets, we use an autoencoder-like architecture for the interpolant, following a similar approach to [24].

Nb. Layers	Layer type
1	Linear (2, 32) SiLU
4	Linear (32, 32) SiLU
1	Linear (32, 32)
1	Three output heads: Linear (32, 1) for f_1 Linear (32, 1) for f_2 Linear (32, 1) for f_3
output	$f_1(x) \cdot f_2(x) + f_3(x^2)$

Table 2: MLP architecture of the energy function on both UCG and WCG datasets.

NB. Layers	Layer type
1	Linear (3, 32) SiLU
1	Linear (32, 64) SiLU
1	Linear (64, 64) SiLU
1	Linear (64, 32) SiLU
1	Linear (32, 3)

Table 3: MLP architecture of the interpolant network $\varphi_{t,\eta}$for WCG dataset.

LAND metric We performed a hyperparameter search to tune the σ parameter. We found that $\sigma = 1$ yielded the best performance. Parameters are similar for both UCG and WCG.

RBF metric We conducted a hyperparameter search to tune both the number of centroids K and the scaling factor κ. The best results were obtained with $K = 30$ and $\kappa = 1$. Parameters are similar for both UCG and WCG.

D.3 Quantitative evaluation with error bars

In Fig. 11, we report the same quantitative results as in Fig. 2, now including 2-σ error bars. The standard deviation σ is computed over evaluation metrics, each averaged on a different set of randomly sampled trajectories (five sets in total).

a) Geodesics evaluation on UCG

Metric	$p_{\mathcal{M}}(\gamma^\star)$ (↑)	RMSE (↓)
$\boldsymbol{G}_{E_{\mathcal{M}}}$	0.79 ± 0.02	-
$\boldsymbol{G}_{1/p_{\mathcal{M}}}$	0.77 ± 0.04	-
$\mathbf{G}_{\mathbf{E}_\theta}$	$\mathbf{0.78} \pm 0.03$	0.12 ± 0.02
$\mathbf{G}_{\mathbf{1/p}_\theta}$	0.73 ± 0.01	$\mathbf{0.10} \pm 0.03$
$\mathbf{G}_{\mathbf{LAND}}$	0.60 ± 0.07	0.38 ± 0.05
$\mathbf{G}_{\mathbf{RBF}}$	0.61 ± 0.06	0.39 ± 0.1

b) Geodesics evaluation on WCG

Metric	$p_{\mathcal{M}}(\gamma^\star)$ (↑)	RMSE (↓)
$\boldsymbol{G}_{E_{\mathcal{M}}}$	0.67 ± 0.05	-
$\boldsymbol{G}_{1/p_{\mathcal{M}}}$	0.73 ± 0.07	-
$\mathbf{G}_{\mathbf{E}_\theta}$	$\mathbf{0.67} \pm 0.06$	0.18 ± 0.07
$\mathbf{G}_{\mathbf{1/p}_\theta}$	$\mathbf{0.67} \pm 0.09$	$\mathbf{0.14} \pm 0.06$
$\mathbf{G}_{\mathbf{LAND}}$	0.65 ± 0.11	0.34 ± 0.05
$\mathbf{G}_{\mathbf{RBF}}$	0.47 ± 0.14	2.2 ± 0.1

Figure 11: **Quantitative evaluation of the geodesics on the UCG and WCG datasets.** We report (i) the accumulated probability along the geodesic (the higher the better) and ii) RMSE between each geodesic and its corresponding baseline (the lower the better). Values after the $\pm$ sign indicate the 2-σ error.

E Experimental details on the Rotated Character Dataset

E.1 Datasets

The Rotated Character Datasets consist of 7 printed characters (5, G, F, P, J, 7, 2), represented as black-and-white images of size 32×32. These characters were selected for two main reasons: (i) they are commonly used in psychophysics experiments [63], and (ii) they are asymmetric and visually distinct, which helps avoid ambiguities in the resulting geodesic trajectories. Fig. 12 shows all characters in their unrotated form.

Figure 12: **Original (non-rotated) samples from the Rotated Character Dataset**

The only difference between the Uniform Rotated Character (URC) and Biased Rotated Character (BRC) datasets lies in the distribution of character orientations.

Uniform Rotated Character (URC) In this setting, character orientations are sampled uniformly across the full range of $[-179°, 180°]$, using a one-degree step. This ensures that each possible orientation within this interval is equally likely. Importantly, the distribution is consistent across all characters, meaning that each character appears with the same uniform spread of rotations.

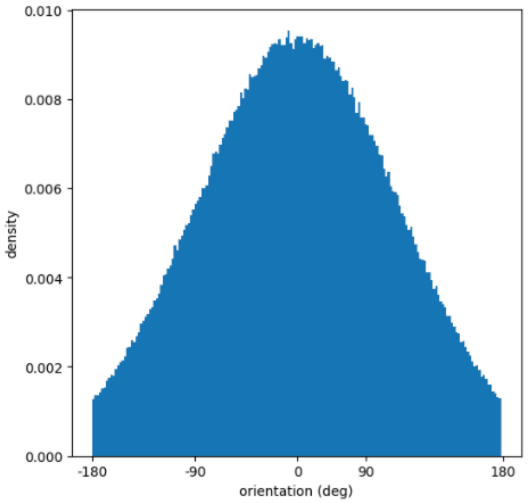


Figure 13: **Distribution of orientation for the BRC dataset**

Biased Rotated Character (BRC) Here, orientations follow a truncated Gaussian distribution centered at $0°$, designed to mimic natural rotation statistics (see Fig. 13). Unlike the Mixture of Gaussian datasets, we do not have access to a closed-form expression for the underlying distribution $p_{\mathcal{M}}$, but we do control its empirical form. This setup introduces a controlled curvature in the data manifold, allowing us to assess how well different metrics adhere to it.

E.2 Architecture and algorithm of the Triplet Loss autoencoder

We computed geodesics in the latent space of an autoencoder trained with a Triplet Loss [101]. This approach is motivated by the fact that image space is inherently non-Euclidean, making it poorly suited for defining meaningful distances. In contrast, the latent space of our autoencoder is explicitly regularized so that Euclidean distances correspond to differences in orientation. By treating the latent space as the ambient space for geodesic computation, we align with the assumption that the data manifold is embedded in an Euclidian Manifold. The training procedure is described in

Algorithm 3: Autoencoder with Triplet regularization

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, \theta_i)\}$, Encoder E_ϕ, Decoder D_ψ
while training **do**
  Sample bath of triplet B=$(\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$ from $\mathcal{D}$
  # Same character; θ_p close to θ_a, θ_n farther
  $\mathbf{z}_a = E_\phi(\mathbf{x}_a), \quad \mathbf{z}_p = E_\phi(\mathbf{x}_p), \quad \mathbf{z}_n = E_\phi(\mathbf{x}_n)$
  $\mathcal{L}_{\text{rec}} = \|D_\psi(\mathbf{z}_a) - \mathbf{x}_a\|^2$
  $\Delta\theta_p = |\theta_a - \theta_p|, \quad \Delta\theta_n = |\theta_a - \theta_n|$
  $\mathcal{L}_{\text{T}} = \mathbb{E}_{\text{B}}\left((\|\mathbf{z}_a - \mathbf{z}_p\| - \alpha\Delta\theta_p)^2 + (\|\mathbf{z}_a - \mathbf{z}_n\| - \alpha\Delta\theta_n)^2\right)$
  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda \cdot \mathcal{L}_{\text{T}}$
  Update (ϕ, ψ) using gradient $\nabla\mathcal{L}_{\text{total}}$

Algo. 3, and the encoder and decoder architectures—based on the Regularized Autoencoder (RAE) framework [102]—are detailed in Table 4 and Table 5, respectively.

We trained the model using the Adam optimizer [94] with a learning rate of 1×10^{-4} and a batch size of 128. In Algorithm 3, we set $\alpha = 1$ and $\lambda = 0.1$. For the architecture, the number of input features (i.e., the number of channels in the first convolutional layer) was set to $F = 128$. In Table 4 and Table 5, the notation "Conv2D(n_c, n_f, 3, 1)" refers to a convolutional layer with n_c input channels, n_f output channels, a kernel size of 3, and padding of 1. Similarly, "ConvTr2D" denotes a transposed convolution. The RAE blocks are modules introduced in [102], referred to here as RaeBlockDown and RaeBlockUp, and are used for efficient downsampling and upsampling, respectively.

Nb. Layers	Layer Type
1	Conv2d (1, F, 3, 1)
1	RaeBlockDown (F, $2F$) ReLU
1	Conv2d ($2F$, $2F$, 3, 1)
1	RaeBlockDown ($2F$, $4F$) ReLU
1	Conv2d ($4F$, $4F$, 3, 1) ReLU
1	Linear ($4F * 8 * 8$,z)

Table 4: Encoder architecture of the autoencoder. F is the number of features ($F = 128$), and z is the size of the latent space ($z = 64$).

Nb. Layers	Layer Type
1	ConvTr2d (z, $4F$, 8, 0) ReLU
1	Conv2d ($4F$, $4F$, 3, 1)
1	RaeBlockUp ($4F$, $2F$) ReLU
1	Conv2d ($2F$, $2F$, 3, 1)
1	RaeBlockUp ($2F$, F) ReLU
1	Conv2d (F, F, 3, 1)
1	Conv2d (F, 1, 4, 1) Tanh

Table 5: Decoder architecture of the autoencoder. F is the number of features ($F = 128$), and z is the size of the latent space ($z = 64$).

E.3 Architecture of the energy function and the interpolant network on the Rotated Character Dataset

The architecture of the energy function used in the EBM is shown in Table 6, and the architecture of the interpolant network is provided in Table 7. These architectures are used for both the URC and BRC datasets. The EBM was trained using the procedure described in Algorithm 2, and Fig.14 shows samples generated by the EBM at the end of training. All EBM training hyperparameters match those described in SectionB.2. For both the EBM and interpolant training, we use a batch size of 128. The interpolant network is optimized with Adam, using a learning rate of 1×10^{-4}.

Nb. Layers	Layer Type
1	Linear (64, 128) SiLU
1	Linear (128, 512) SiLU
6	Linear (512, 512) SiLU
1	Linear (512, 64)
1	Three output heads: Linear (64, 1) for f_1 Linear (64, 1) for f_2 Linear (64, 1) for f_3
Output	$f_1(x) \cdot f_2(x) + f_3(x^2)$

Table 6: Archiecture of the EBM energy function on both URC and BRC datasets

Nb. Layers	Layer Type
1	Linear (64*3, 128) SiLU
1	Linear (128, 128) SiLU
1	Linear 128, 128) SiLU
1	Linear 128, 128) SiLU
1	Linear 128, 128) SiLU
1	Linear 128, 64) SiLU

Table 7: Architecture of the interpolant network used on the URC and BRC dataset.

E.4 Hyperparameters of the LAND and RBF metric

LAND metric We performed a hyperparameter search to tune the σ parameter. We found that $\sigma = 0.4$ yielded the best performance. Parameters are similar for both the URC and BRC datasets.

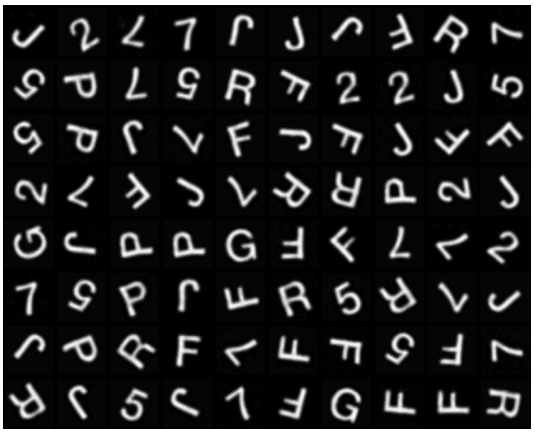


Figure 14: **Samples from the EBM train on URC.** These samples are generated by applying Langevin dynamics to the energy function learned by the EBM.

RBF metric We conducted a hyperparameter search to tune both the number of centroids K and the scaling factor κ. The best results were obtained with $K = 300$ and $\kappa = 0.75$. Parameters are similar for both the URC and BRC datasets.

E.5 Additional geodesics

URC dataset: In Fig. 15 we show additional geodesics on the URC dataset.

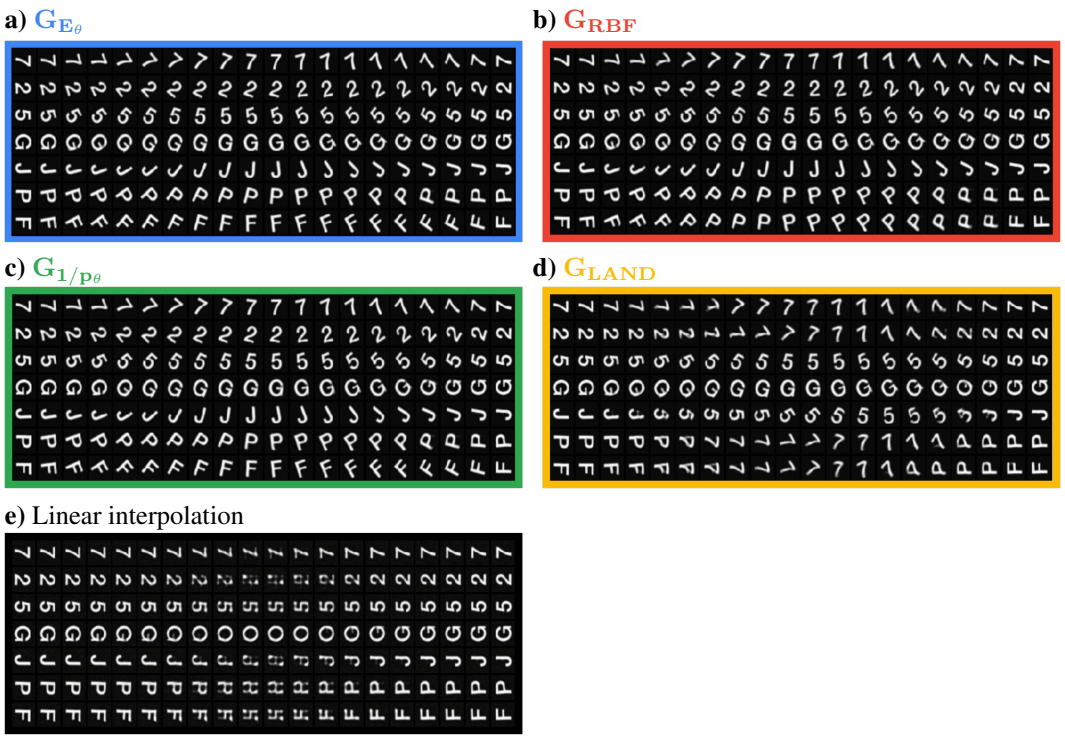


Figure 15: **Geodesics on the URC dataset.** Geodesics are computed using four different metrics: **a)** G_{E_θ}, **b)** G_{RBF}, **c)** G_{1/p_θ}, **d)** G_{LAND}. For comparison, a simple linear interpolation is shown in **e)**. The trajectory are computed in the latent space of the autoencoder and projected into pixel space for visualization. Each trajectory is subsampled at 20 time steps for clarity.

BRC dataset: In Fig. 16 we show additional geodesics on the BRC dataset.

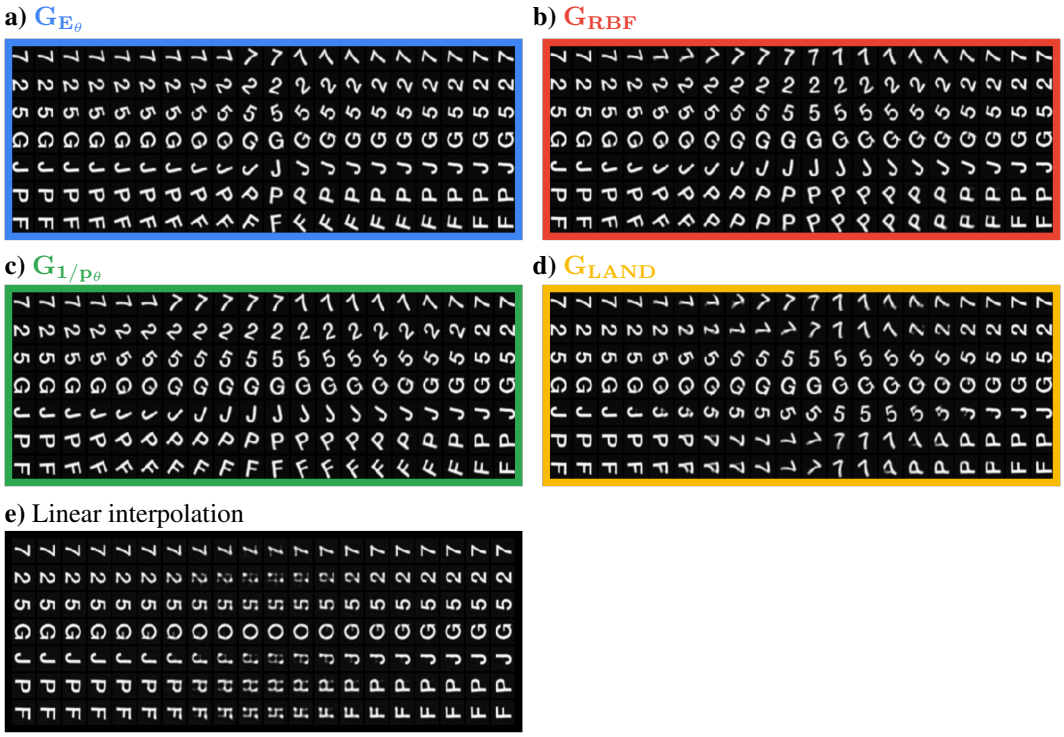


Figure 16: **Geodesics on the BRC dataset.** Geodesics are computed using four different metrics: **a)** G_{E_θ}, **b)** G_{RBF}, **c)** G_{1/p_θ}, **d)** G_{LAND}. For comparison, a simple linear interpolation is shown in **e)**. The trajectory are computed in the latent space of the autoencoder and projected into pixel space for visualization. Each trajectory is subsampled at 20 time steps for clarity.

E.6 Quantitative evaluation with error bars

In Table. 8, we report the same quantitative results as in Fig. 4, now including 2-σ error bars. The standard deviation σ is computed over evaluation metrics, each averaged on a different set of randomly sampled trajectories (five sets in total).

Metric	$\mathcal{D}$-RMSE (↓)	$\gamma^\star$-RMSE (↓)
linear interp.	2.96 ± 0.42	3.52 ± 0.21
G_{E_θ}	$\mathbf{0.11} \pm 0.01$	$\mathbf{0.40} \pm 0.03$
G_{1/p_θ}	0.14 ± 0.02	0.44 ± 0.07
G_{LAND}	0.66 ± 0.12	2.39 ± 0.51
G_{RBF}	0.36 ± 0.06	0.86 ± 0.17

Table 8: **Quantitative evaluation on the URC dataset with the 2σ error.** Quantitative evaluation using two metrics: (i) $\mathcal{D}$-RMSE, which measures proximity to the dataset manifold, and (ii) γ-RMSE, which measures the deviation from an ideal smooth rotation. Values after the $\pm$ sign indicate the 2-σ error.

F Experimental details on the Rotated Character Dataset

F.1 Dataset

In this section, we conduct experiments on the Animal Faces High-Quality (AFHQ) dataset introduced by [59]. The full dataset contains 15,000 images across three categories: cats, dogs, and wild animals. For our experiments, we restrict the dataset to the cat and dog classes only, each comprising approximately 5,000 images. This choice avoids introducing curvature in the data manifold that could arise from the relatively small number of samples in the wild animal category. All images are cropped, aligned, and have a resolution of 512×512 pixels. AFHQ is widely used for image-to-image translation and style transfer, and its diversity in pose, breed, and appearance makes it well-suited for smooth interpolation tasks. See Fig. 17 for example images from the AFHQ dataset.

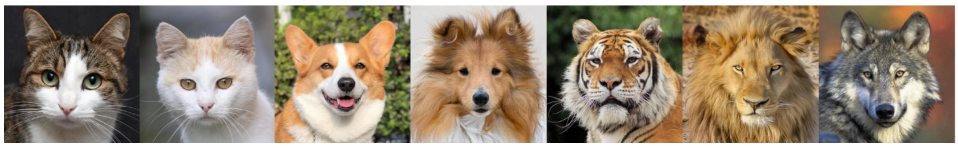


Figure 17: **Samples from the AFHQ dataset [59]**

For the experiments in this section, we compute geodesics in the latent space of a pretrained Variational Autoencoder (VAE). Specifically, we use the VAE from Stable Diffusion V1 [18]. The latent representations have a spatial size of $4 \times 16 \times 16$.

F.2 Architecture of the energy function and the interpolant network on the AFHQ dataset

Energy Function: The architecture used for the energy function is detailed in Table 9. We set the number of input channels to $n_c = 4$, matching the dimensionality of the latent representation, and use $F = 256$ feature channels in the first convolutional layer. The network follows a simple sequence of downsampling convolutional layers, which we found to yield more stable training than ResNet-style architectures. The EBM is trained using Algorithm 2, with the same hyperparameters as in Section B.2. To further improve training stability, we add a denoising score matching regularization term and use a cosine learning rate scheduler.

Nb. Layers	Layer Type
1	Conv2d (n_c, F, 3, 1, 1) SiLU
1	Conv2d (F, F, 3, 1, 1) SiLU
1	Conv2d (F, $2F$, 4, 2, 1) SiLU
1	Conv2d ($2F$, $2F$, 3, 1, 1) SiLU
1	Conv2d ($2F$, $4F$, 4, 2, 1) SiLU
1	Conv2d ($4F$, $4F$, 3, 1, 1) SiLU
1	Conv2d ($4F$, $8F$, 4, 2, 1) SiLU
1 1 1	Conv2d ($8F$, 1, 2, 1, 0): for f_1 Conv2d ($8F$, 1, 2, 1, 0): for f_2 Conv2d ($8F$, 1, 2, 1, 0): for f_3
Output	$f_1(x) \cdot f_2(x) + f_3(x^2)$

Table 9: **Architecture of the energy function**. F denotes the base number of feature channels, and n_c is the number of input channels. The final energy is computed using three parallel output heads. The notation Conv2d(n_c, n_f, k, s, p) refers to a 2D convolutional layer with n_c input channels, n_f output channels, a kernel size of k, stride s, and padding p.

In Fig. 18, we show randomly selected samples generated by the EBM after training. The Fréchet Inception Distance (FID) of the model is measured to be 9.89.

Figure 18: **Samples from the EBM trained on the AFHQ dataset.** These samples are generated by applying Langevin dynamics to the energy function learned by the EBM.

Interpolant Network: We use the U-Net architecture from [103], following the same hyperparameter settings.

F.3 Hyperparameters of the LAND and RBF metric

LAND metric We performed a hyperparameter search to tune the σ parameter. We found that $\sigma = 10$ yielded the best performance.

RBF metric We conducted a hyperparameter search to tune both the number of centroids K and the scaling factor κ. The best results were obtained with $K = 1000$ and $\kappa = 3$.

F.4 FIDs with error bars

In Table. 10, we include 2-σ error bars. The standard deviation σ is computed over different sets of randomly sampled trajectories (five sets in total).

Metric	FID (↓)
Linear interp.	42.47 ± 3.17
Slerp interp.	32.67 ± 2.33
$\mathbf{G}_{\mathbf{E}_\theta}$	20.79 ± 2.17
$\mathbf{G}_{\mathbf{1/P}_\theta}$	$\mathbf{16.47} \pm 1.04$
$\mathbf{G}_{\mathbf{LAND}}$	39.17 ± 3.63
$\mathbf{G}_{\mathbf{RBF}}$	37.98 ± 2.46

Table 10: **FID along geodesics for different Riemannian metrics**. FID is computed at each trajectory point to assess on-manifold alignment. Values after the $\pm$ sign indicate the 2-σ error.

F.5 Additional geodesics on AFHQ

Riemanian metric: $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$

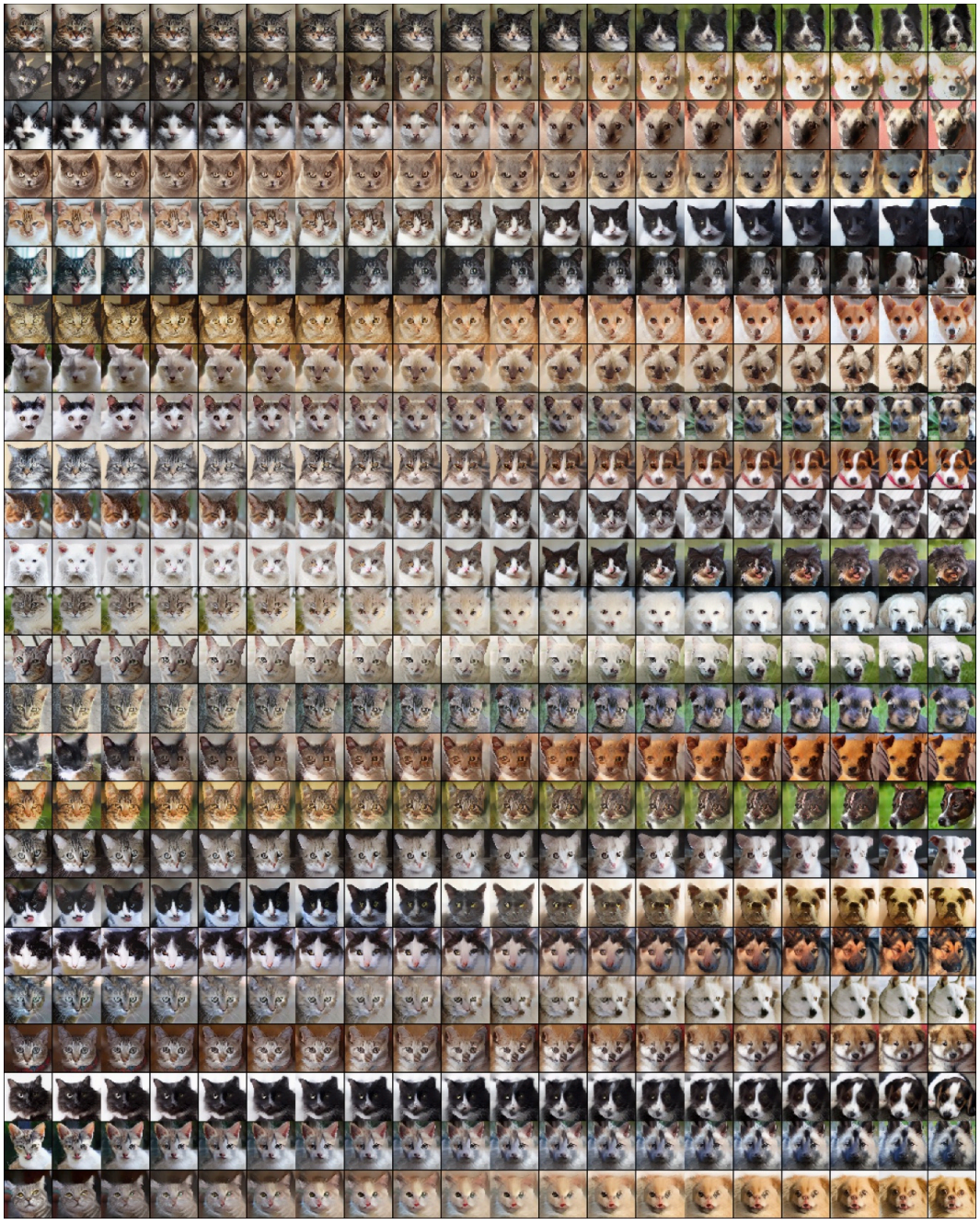


Figure 19: **Geodesics on the AFHQ dataset using $\mathbf{G}_{\mathbf{1}/\mathbf{p}_\theta}$.** Each row shows a geodesic in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each geodesic, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

Riemanian metric: $\mathbf{G}_{\mathbf{E}_\theta}$

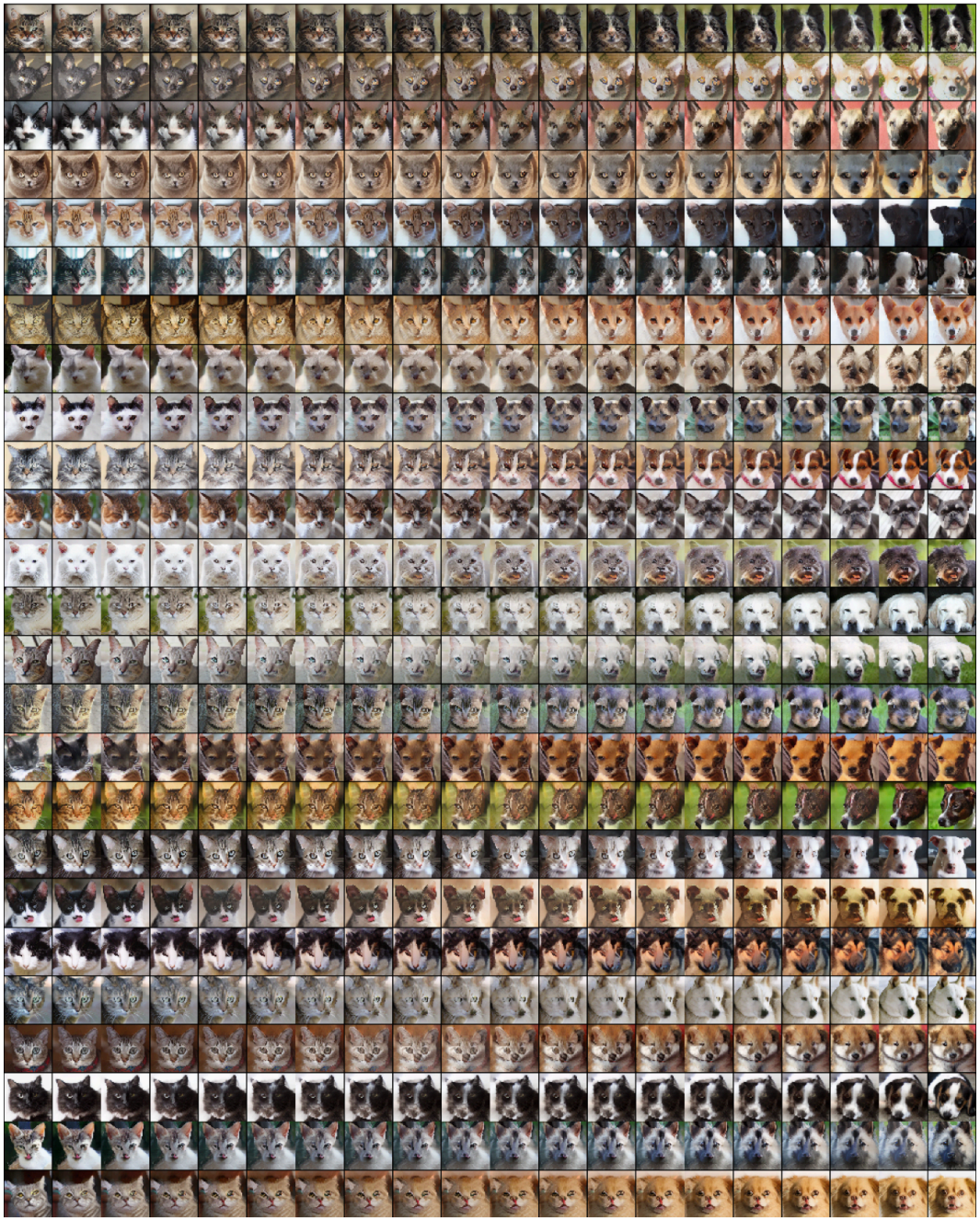


Figure 20: **Geodesics on the AFHQ dataset using $\mathbf{G}_{\mathbf{E}_\theta}$.** Each row shows a geodesic in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each geodesic, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

Riemanian metric: $\mathbf{G}_{\mathbf{RBF}}$

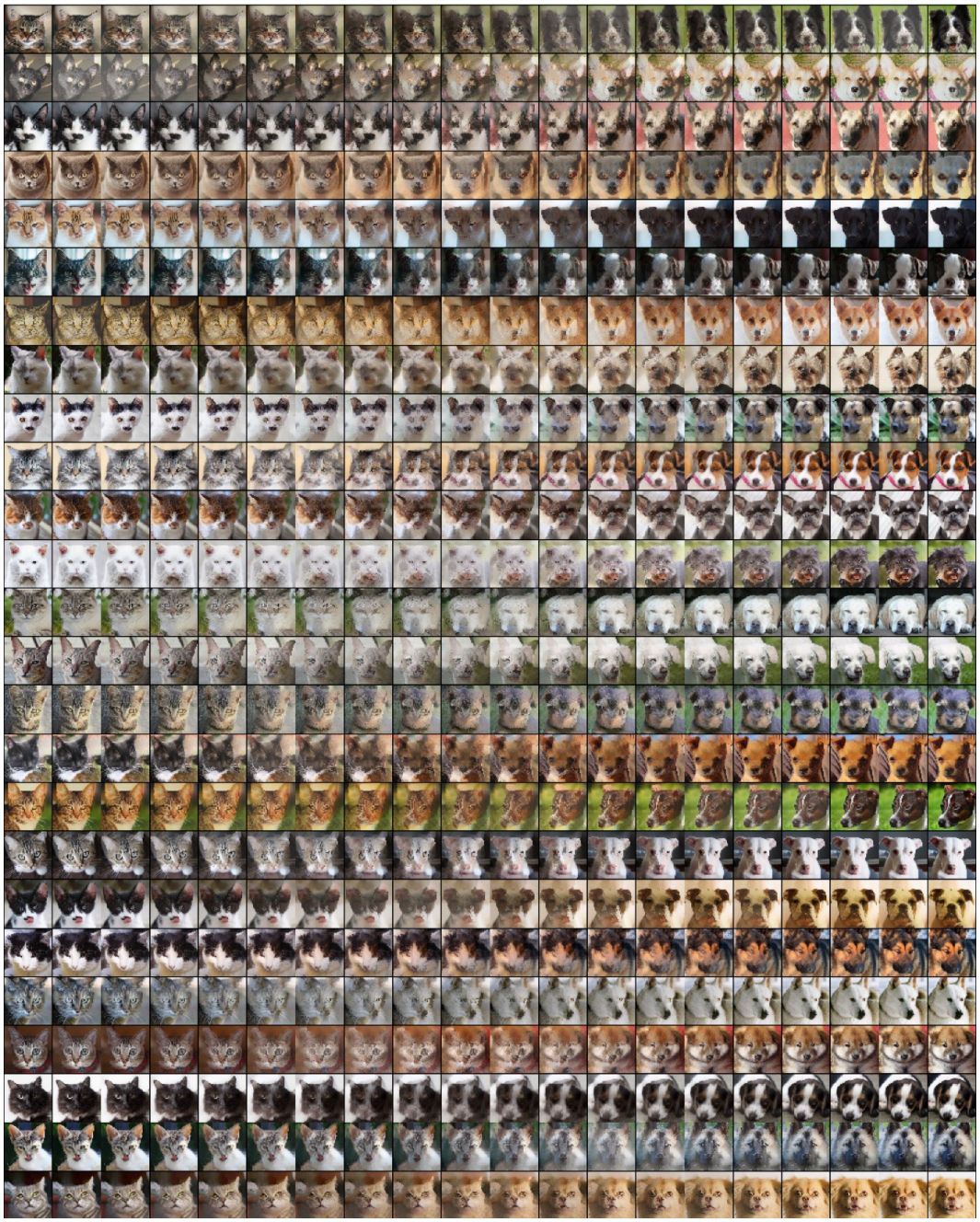


Figure 21: **Geodesics on the AFHQ dataset using $\mathbf{G}_{\mathbf{RBF}}$.** Each row shows a geodesic in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each geodesic, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

Riemanian metric: $\mathrm{G}_{\mathrm{LAND}}$

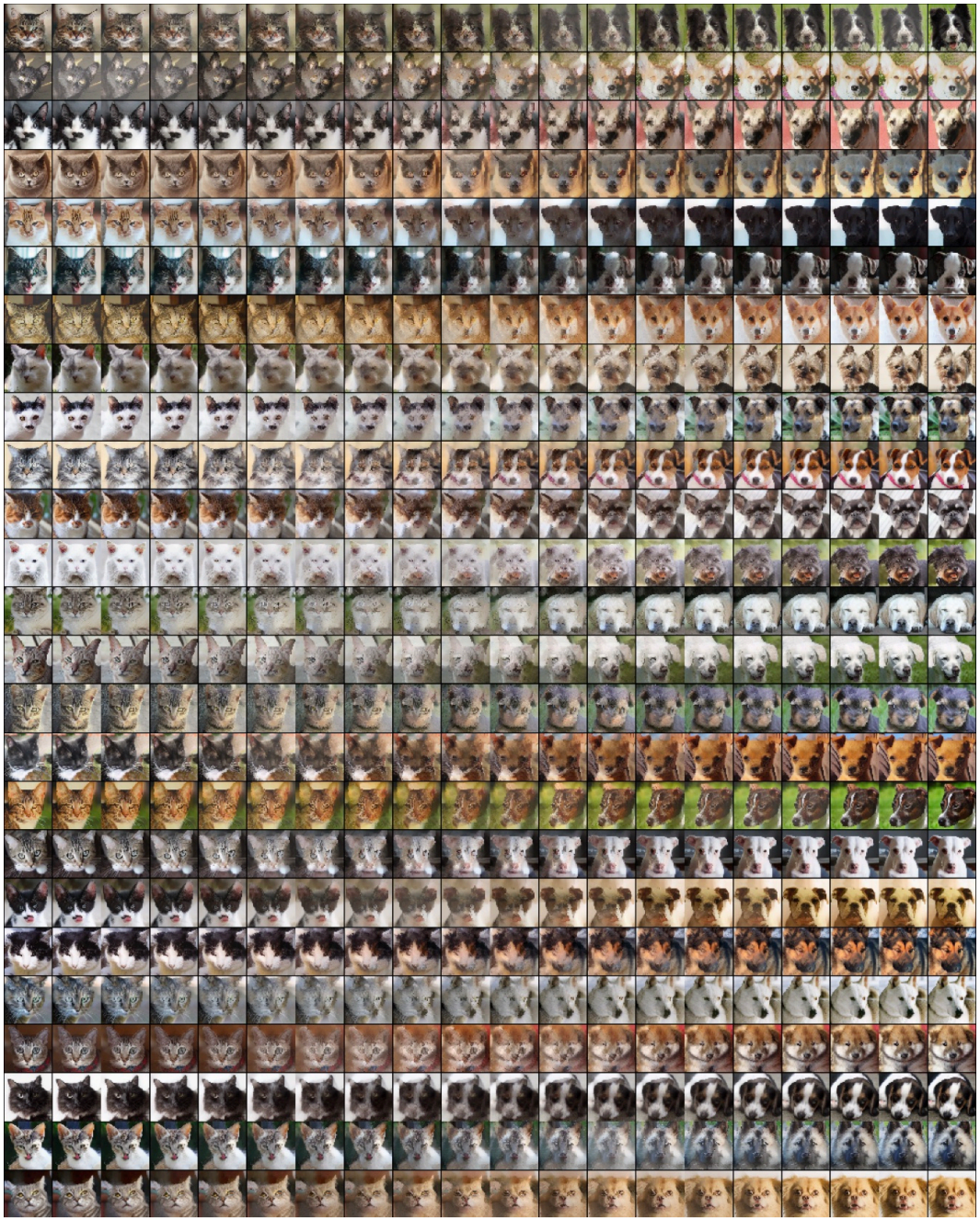


Figure 22: **Geodesics on the AFHQ dataset using $\mathrm{G}_{\mathrm{LAND}}$.** Each row shows a geodesic in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each geodesic, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

Linear interpolation

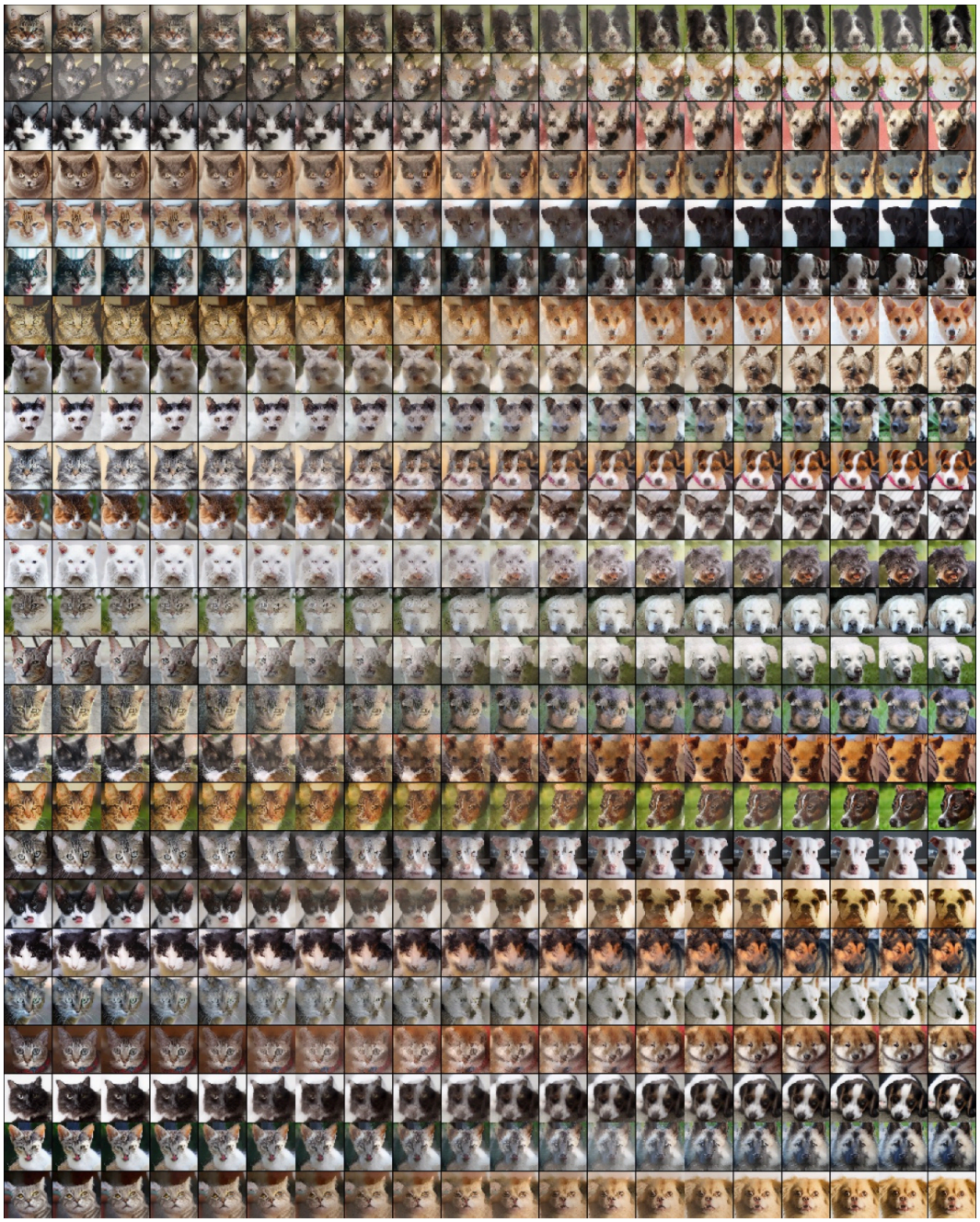


Figure 23: **Linear interpolation on the AFHQ dataset.** Each row shows an interpolation in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each interpolation, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

Slerp interpolation

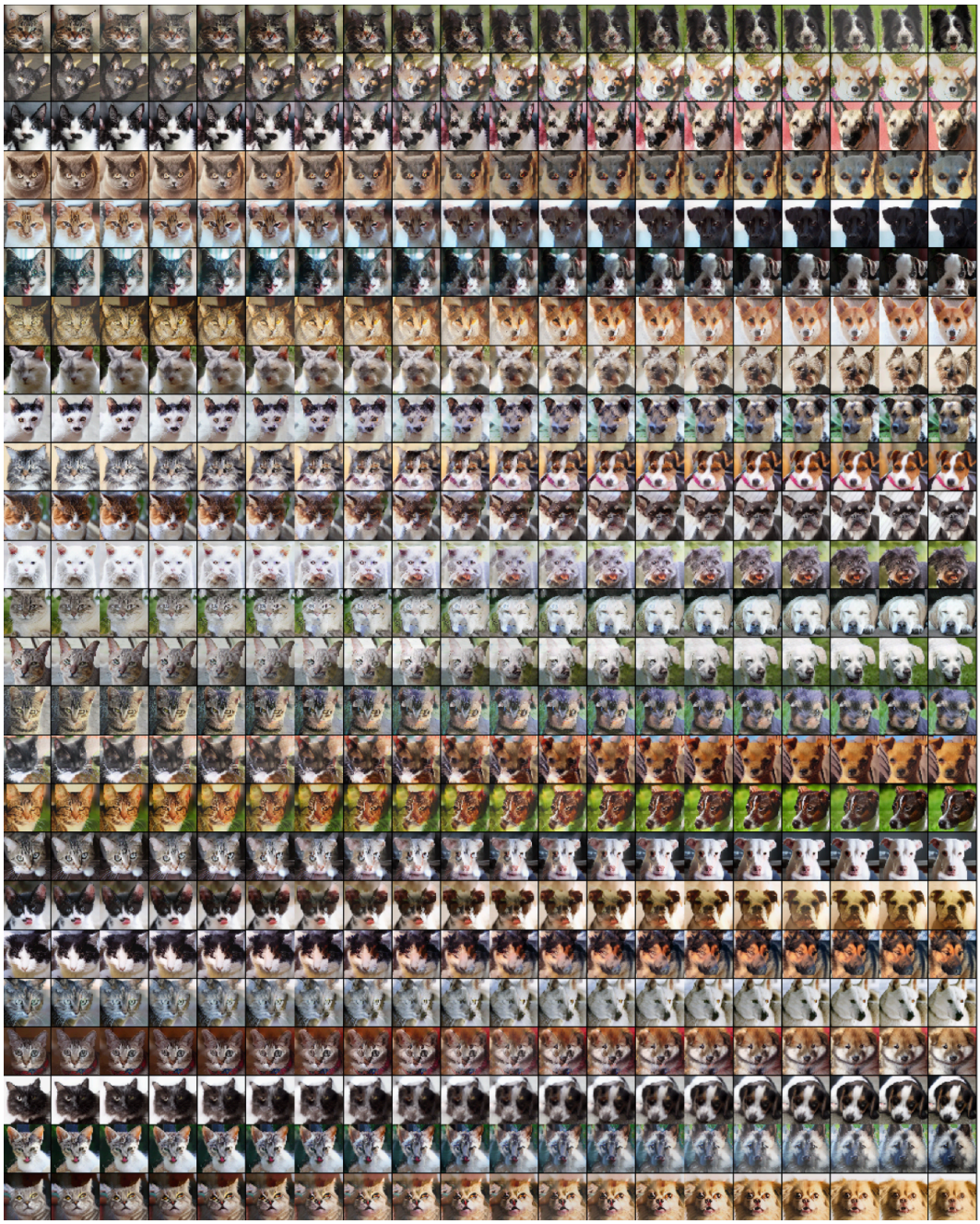


Figure 24: **Spherical interpolation (Slerp) on the AFHQ dataset.** Each row shows an interpolation in latent space between a randomly sampled cat image (start point) and a dog image (end point). Columns correspond to time steps along each interpolation, from left (start) to right (end). Images are obtained by decoding the latent representations back into pixel space.

F.6 About the Spherical interpolation

Given two points $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^D$ lying on the unit hypersphere (i.e., $\|\mathbf{x}_0\| = \|\mathbf{x}_1\| = 1$), the spherical interpolation between them is defined as:

$$\text{slerp}(t; \mathbf{x}_0, \mathbf{x}_1) = \frac{\sin((1-t)\theta)}{\sin\theta}\mathbf{x}_0 + \frac{\sin(t\theta)}{\sin\theta}\mathbf{x}_1, \quad t \in [0,1],$$

where θ is the angle between $\mathbf{x}_0$ and $\mathbf{x}_1$, given by:

$$\theta = \arccos\left(\frac{\langle \mathbf{x}_0, \mathbf{x}_1 \rangle}{\|\mathbf{x}_0\| \, \|\mathbf{x}_1\|}\right).$$

In practice, when interpolating latent codes from a Variational Autoencoder (VAE), the latent vectors $\mathbf{x}_0$ and $\mathbf{x}_1$ are typically drawn from a standard normal prior and do not lie on the unit sphere. To apply slerp, we first normalize the vectors:

$$\tilde{\mathbf{x}}_0 = \frac{\mathbf{x}_0}{\|\mathbf{x}_0\|}, \qquad \tilde{\mathbf{x}}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|},$$

and compute θ as:

$$\theta = \arccos\left(\langle \tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1 \rangle\right).$$

This interpolation method, introduced by [60], has proven particularly effective for interpolating in the latent space of VAEs. The intuition behind its success is that it implicitly assumes the data manifold lies on a hypersphere. While this may seem restrictive, the assumption is reasonable in practice. In a VAE, each latent coordinate x_i is drawn from a standard Normal distribution: $x_i \sim \mathcal{N}(0,1)$ $(1 < i < D)$. As a result, the squared norm, $||\mathbf{x}||^2 = \sum_{i=1}^{D} x_i^2$ follows a chi-squared distribution with D degree of freedom. This distribution is known to concentrate tightly around D, effectively placing most latent codes near the surface of a hypersphere. To validate this empirically, we visualize the distribution of $||\mathbf{x}||^2$ for all latent codes of the AFHQ dataset (see Fig. 25). We observe that this distribution is concentrated on D—$D = 1024$ for VAE with latent space of size $4 \times 16 \times 16$.

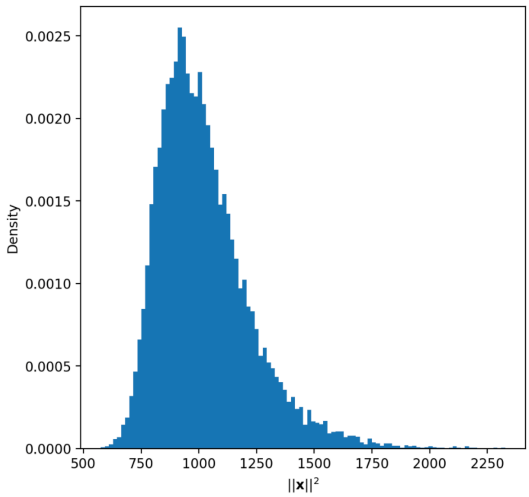


Figure 25: **Distribution of $||\mathbf{x}||^2$ on the AFHQ dataset**

To conclude, slerp interpolation is well-suited for VAE latent spaces because it aligns with their underlying geometric structure.

F.7 Physical interpretation

We refer the reader to [104] or [105] for a detailed background on differential geometry.

Geodesic equation. Assume that the manifold $\mathcal{M}$ is the ambient D-dimensional Euclidean space ($\mathcal{M} = \mathbb{R}^D$). We equipped the manifold $\mathcal{M}$ with a conformal Riemannian metric $\mathbf{G}(\mathbf{x}) = \dfrac{1}{p(\mathbf{x})} \cdot \mathbf{I}$, with p the probability density of the data, and $\mathbf{I}$ the identity matrix of $\mathbb{R}^{D\times D}$. Let $\boldsymbol{\gamma}(t)$ be a geodesic (i.e. $\gamma : [0,1] \to \mathbb{R}^D$). We denote the instantaneous speed of the geodesic at time t, $\dot{\boldsymbol{\gamma}}(t)$, and its acceleration $\ddot{\boldsymbol{\gamma}}(t)$. Said otherwise, $\dot{\boldsymbol{\gamma}}(t)$ and $\ddot{\boldsymbol{\gamma}}(t)$ denote $\frac{\partial \boldsymbol{\gamma}}{\partial t}(t)$ and $\frac{\partial^2 \boldsymbol{\gamma}}{\partial t^2}(t)$ respectively.

The geodesic equation is the 2nd-order ODE written as:

$$\ddot{\boldsymbol{\gamma}}^k(t) + \sum_{i,j} \Gamma^k_{i,j}(\boldsymbol{\gamma}(t)) \cdot \dot{\boldsymbol{\gamma}}^i(t) \cdot \dot{\boldsymbol{\gamma}}^j(t) = 0 \tag{17}$$

In Eq.17, $\ddot{\boldsymbol{\gamma}}^k(t)$ and $\dot{\boldsymbol{\gamma}}^k(t)$ denotes the k-th coordinate of $\ddot{\boldsymbol{\gamma}}(t)$ and $\dot{\boldsymbol{\gamma}}(t)$, respectively (here $1 < k < D$). $\Gamma^k_{i,j}$ are the Christoffel symbols, they are derived from the Riemannian metric and encode how it bends and curves the space. $\Gamma^k_{i,j}$ tells how much the change in direction in the i-th and j-th coordinate causes acceleration in the k-th coordinate ($1 < i, j, k < D$). Said differently, i, j refer to the coordinate direction along which the particule is moving, and k refers to the coordinate direction where the motion causes effect (i.e. curvature induces acceleration).

Christoffel symbols for conformal metric. The Christoffel symbols for a conform metric $\mathbf{G}(\mathbf{x}) = \lambda(\mathbf{x}) \cdot \mathbf{I}$ (with λ a scalar function):

$$\Gamma^k_{ij}(\mathbf{x}) = \frac{1}{2\lambda(\mathbf{x})} \left(\delta_{j,k}\, \partial_i \lambda(\mathbf{x}) + \delta_{i,k}\, \partial_j \lambda(\mathbf{x}) - \delta_{ij}\, \partial_k \lambda(\mathbf{x})\right) \tag{18}$$

In Eq. 18, $\partial_i \lambda(\mathbf{x}) = \dfrac{\partial \lambda(\mathbf{x})}{\partial x^i}$ (i.e. the partial derivative of $\lambda(\mathbf{x})$ with respect to the i-th coordinate), and $\delta_{j,k}$ is the Kronecker symbol ($\delta_{j,k} = 1$ if $j = k$ and $\delta_{j,k} = 0$ otherwise). If one plugs Eq. 18 in the right hand side of Eq. 17:

$$\begin{aligned}
\sum_{i,j} \Gamma^k_{ij}\big(\boldsymbol{\gamma}(t)\big)\, \dot{\boldsymbol{\gamma}}^i(t)\, \dot{\boldsymbol{\gamma}}^j(t) = \frac{1}{2\lambda\big(\boldsymbol{\gamma}(t)\big)} \cdot \Bigg[& \sum_i \partial_i \lambda\big(\boldsymbol{\gamma}(t)\big)\, \dot{\boldsymbol{\gamma}}^i(t)\, \dot{\boldsymbol{\gamma}}^k(t) \\
& + \sum_j \partial_j \lambda\big(\boldsymbol{\gamma}(t)\big)\, \dot{\boldsymbol{\gamma}}^k(t)\, \dot{\boldsymbol{\gamma}}^j(t) \\
& - \sum_i \partial_k \lambda\big(\boldsymbol{\gamma}(t)\big)\, \dot{\boldsymbol{\gamma}}^i(t)^2 \Bigg] \\
= \frac{1}{2\lambda\big(\boldsymbol{\gamma}(t)\big)} \cdot \Big[& 2\dot{\boldsymbol{\gamma}}^k(t) \big\langle \nabla \lambda\big(\boldsymbol{\gamma}(t)\big), \dot{\boldsymbol{\gamma}}(t) \big\rangle - \partial_k \lambda\big(\boldsymbol{\gamma}(t)\big)\, \|\dot{\boldsymbol{\gamma}}(t)\|^2 \Big],
\end{aligned} \tag{19}$$

where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ are the usual *Euclidean* inner product and norms, respectively.

So Eq. 17, becomes :

$$\ddot{\boldsymbol{\gamma}}^k(t) = -\frac{\dot{\boldsymbol{\gamma}}^k(t)}{\lambda\big(\boldsymbol{\gamma}(t)\big)} \big\langle \nabla \lambda\big(\boldsymbol{\gamma}(t)\big), \dot{\boldsymbol{\gamma}}(t) \big\rangle + \frac{1}{2\lambda\big(\boldsymbol{\gamma}(t)\big)} \partial_k \lambda\big(\boldsymbol{\gamma}(t)\big)\, \|\dot{\boldsymbol{\gamma}}(t)\|^2 \tag{20}$$

Pulling everything together. If one plugs our definition of the Riemannian metric (i.e. $\lambda\big(\boldsymbol{\gamma}(t)\big) = \dfrac{1}{p\big(\boldsymbol{\gamma}(t)\big)}$, and therefore $\dfrac{\nabla \lambda\big(\boldsymbol{\gamma}(t)\big)}{\lambda\big(\boldsymbol{\gamma}(t)\big)} = -\nabla \log p\big(\boldsymbol{\gamma}(t)\big)$), Eq. 20 becomes:

$$\ddot{\boldsymbol{\gamma}}(t) = \langle \nabla_\gamma \log p(\boldsymbol{\gamma}(t)), \dot{\boldsymbol{\gamma}}(t) \rangle \cdot \dot{\boldsymbol{\gamma}}(t) \; - \; \frac{1}{2} \|\dot{\boldsymbol{\gamma}}(t)\|^2 \cdot \nabla_\gamma \log p(\boldsymbol{\gamma}(t)) \tag{21}$$

Eq. 21 is similar in form to Newton's second law. The acceleration of a particle (of unit mass) is governed by a velocity-dependent force built from the Stein Score (i.e. $\nabla_\gamma \log p(\boldsymbol{\gamma}(t))$). More speficically:

- $\langle \nabla \log p(\boldsymbol{\gamma}(t)), \dot{\boldsymbol{\gamma}}(t) \rangle \cdot \dot{\boldsymbol{\gamma}}(t)$ describes a "force" aligned with the particle velocity direction. This term acts like an anisotropic drag or propulsion term: i) it speeds up the particle when it goes toward a high-density region and ii) it slows down the particle going the other way.
- $-\frac{1}{2}\|\dot{\boldsymbol{\gamma}}(t)\|^2 \cdot \nabla \log p(\boldsymbol{\gamma}(t))$ is a force in the direction of the stein score (pointing toward low density regions). It behaves like a repulsive force, pushing the particle toward areas with low probability. The faster the particle moves, the stronger the force.

The "force" seems to depend on the velocity $\dot{\boldsymbol{\gamma}}(t)$, which is typical of inertial forces (i.e, forces that depend on a given frame). This is an artifact from the affine parametrization of the geodesic, which ensures constant speed along the trajectory.

Newtonian formalism. Note that the variable t in previous equations is the geometrical "time". This variable t stems from the affine parametrization (e.g. see Eq. 3) and is not related to the physical time. To make Eq. 21 compatible with the "physical" time, denoted s, one can consider the following change of variable:

$$\frac{\partial s}{\partial t}(t) = p(\boldsymbol{\gamma}(t)) \quad \text{or equivalently} \quad \frac{\partial t}{\partial s}(s) = \frac{1}{p(\boldsymbol{\gamma}(t(s)))} \tag{22}$$

This change of variable implies that when moving through space according to arc-length s, the geometric time t runs more slowly in low-density regions and faster in high-density ones. This change of variable is particularly handy to interpret Eq. 21 as Newtonian motion. Let's therefore consider the following reparametrization: $\boldsymbol{\gamma}\big(t(s)\big) = \mathbf{x}(s)$, where $\mathbf{x}$ is the new trajectory parametrized by the physical time s. So:

$$\dot{\boldsymbol{\gamma}}(t) = \frac{\partial}{\partial t}\boldsymbol{\gamma}(t) = \frac{\partial}{\partial t}\mathbf{x}(s(t)) = \frac{\partial \mathbf{x}}{\partial s} \cdot \frac{\partial s}{\partial t} = \dot{\mathbf{x}}(s) \cdot p(\mathbf{x}(s)) \tag{23}$$

$$\begin{aligned}
\ddot{\boldsymbol{\gamma}}(t) &= \frac{\partial}{\partial t}\left(\dot{\mathbf{x}}(s) \cdot p(\mathbf{x}(s))\right) = \left(\frac{\partial}{\partial s}\left(\dot{\mathbf{x}}(s) \cdot p(\mathbf{x}(s))\right)\right) \cdot \frac{\partial s}{\partial t} \\
&= \left(\ddot{\mathbf{x}}(s) \cdot p(\mathbf{x}(s)) + \langle \nabla p(\mathbf{x}(s)), \dot{\mathbf{x}}(s) \rangle \cdot \dot{\mathbf{x}}(s)\right) \cdot p(\mathbf{x}(s)) \\
&= p(\mathbf{x})^2 \ddot{\mathbf{x}} + p(\mathbf{x}) \langle \nabla p(\mathbf{x}), \dot{\mathbf{x}} \rangle \dot{\mathbf{x}}
\end{aligned} \tag{24}$$

Now plugging Eq. 24 and Eq. 23 in Eq. 21:

$$\begin{aligned}
& p(\mathbf{x})^2 \ddot{\mathbf{x}} + p(\mathbf{x}) \langle \nabla p(\mathbf{x}), \dot{\mathbf{x}} \rangle \dot{\mathbf{x}} \\
&\qquad = \langle \nabla \log p(\mathbf{x}), \dot{\boldsymbol{\gamma}}(t) \rangle \cdot \dot{\boldsymbol{\gamma}}(t) - \frac{1}{2}\|\dot{\boldsymbol{\gamma}}(t)\|^2 \cdot \nabla \log p(\mathbf{x}) \\
&\qquad = \langle \nabla \log p(\mathbf{x}), p(\mathbf{x})\dot{\mathbf{x}} \rangle \cdot (p(\mathbf{x})\dot{\mathbf{x}}) - \frac{1}{2}\|p(\mathbf{x})\dot{\mathbf{x}}\|^2 \cdot \nabla \log p(\mathbf{x}) \\
&\qquad = p(\mathbf{x})^2 \langle \nabla \log p(\mathbf{x}), \dot{\mathbf{x}} \rangle \dot{\mathbf{x}} - \frac{1}{2} p(\mathbf{x})^2 \|\dot{\mathbf{x}}\|^2 \cdot \nabla \log p(\mathbf{x}) \\
\Rightarrow \quad & \ddot{\mathbf{x}} = -\frac{1}{2}\|\dot{\mathbf{x}}\|^2 \underbrace{\nabla \log p(\mathbf{x})}_{\text{Stein score}}.
\end{aligned}$$

This equation can be interpreted through Newton's second law: it describes the motion of a particle $\mathbf{x}$ following a geodesic in the Riemanannian manifold $\big(\mathcal{M}, \frac{1}{p(\mathbf{x})}\big)$, where $p(\mathbf{x})$ denotes the data density. The particle experiences a force $-\frac{1}{2}\|\dot{\mathbf{x}}\|^2 \nabla \log p(\mathbf{x})$, pushing away from regions of high probability. The term $||\mathbf{x}||^2$ modulates the forces magnitude and plays a role analogous to momentum, strengthening the pull when the particle moves quickly. While this is not a literal physical system—here the particle is a data point, and has no mass, it provides a useful analogy for understanding the dynamics of trajectories shaped by data geometry.

G Limitations

While our approach provides a promising framework for deriving Riemannian metrics from EBMs, several limitations should be acknowledged:

- First, we restrict our study to conformal metrics, which uniformly scale the identity matrix and thus cannot capture directional (anisotropic) structure in the data manifold. While this simplifies optimization, it limits expressivity in settings where geometry varies across directions—something more expressive, score-based metrics may help resolve.
- Second, our method relies on pretrained EBMs that assign meaningful energy values across the entire space. Training such models is challenging in high-dimensional settings due to the computational cost of sampling (e.g., Langevin dynamics), and performance can degrade if the energy landscape is poorly shaped or overfitted.
- Third, although we demonstrate improvements over strong baselines, our evaluation of geodesic quality remains largely indirect—relying on alignment with proxy measures (e.g., density, rotation smoothness, FID). In complex datasets like natural images, the absence of ground-truth geometry makes rigorous evaluation difficult.
- Fourth, our approach assumes that the data distribution is adequately captured by the EBM, yet in practice, misestimation of density—especially in underrepresented regions—may distort the metric and lead to suboptimal paths.
- Finally, while we demonstrate promising results on several datasets, our experiments are constrained to pretrained generative models and fixed feature spaces (e.g., VAE latents), and generalizing to end-to-end learnable architectures remains unexplored.

Future work may address these limitations by developing scalable score-based metrics, improving EBM training stability, integrating richer evaluation protocols, and extending the framework to broader model classes and learning settings.

H Broader Impact

This work advances our understanding of data geometry by connecting generative modeling and Riemannian geometry, with potential implications across machine learning, neuroscience, and cognitive science. By enabling principled geodesic computation in high-dimensional spaces, our approach could support safer interpolation in generative models, improve motion planning in robotics, or inform models of human cognition. However, care should be taken when applying such methods to sensitive domains, as learned energy landscapes may inherit biases present in training data.

I Computational ressources

All experiments were conducted on NVIDIA RTX 3090 GPUs (32 GB memory). Training on the toy dataset was fast, with both the EBM and interpolant completing in a few minutes. For the Rotated Characters dataset, EBM training took 8 GPU hours and the interpolant 30 minutes. On the AFHQ dataset, training required 6 GPU days for the EBM and 24 GPU hours for the interpolant. Including extensive hyperparameter searches and trial-and-error development, the total compute usage amounted to approximately 123,000 GPU hours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claim is that EBM-derived metrics stay closer to the data manifold and better capture the geometry of the data compared to alternative metrics. In all experimental settings this claim is verified (see Fig. 2, Fig. 4, Table 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have briefly discussed limitations in the conclusion section of the main article. But we have included an addtional a full section in the supplementary information (see Supp. G) to expand those limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: This paper does not present new theoretical results, but it builds on and leverages existing theoretical insights from prior work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Due to space constraints, we could not include all experimental details in the main paper. However, the supplementary material provides a thorough description of each experiment, including neural network architectures, all hyperparameters, additional samples, and the pseudo-codes for the main algorithms we used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets we used are open access. In addition, upon acceptance we will release the github code to reproduce all the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   In the supplementary materials (see Supp. D, Supp. E and Supp. F), we have extensively reported experimental details about the datasets, the type of optimizers we used, and the hyperparameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported the 2σ error bar for all quantitative metrics on the Supplementary information (see Supp. D.3, Supp. F and Supp. F.4).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of computing workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included a section describing the computational resources we use for all experiments in the supplementary information (see Supp. I).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All the research conducted in this article conforms to the Neurips Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed the broader impact of our research in Supp. H

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: We don't think our work poses a significant risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In terms of datasets, we use the Mixture of Gaussians (not under license), the AFHQ dataset (under CC BY 4.0 Licence), and the rotated letter (based and sklearn letters). In addition, we use the interpolant training algorithms, and the contrastive divergence to train EBMS. All the creators of these assets have been credited by citing the corresponding articles.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] .

    Justification: Our only new asset is the code used to run all experiments, which will be released publicly under the MIT license upon acceptance. All other assets are fully documented in this article.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: No human experiments or crowdsourcing are involved in this article.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA] .

Justification: No human experiments or crowdsourcing are involved in this article.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.