Pulling back information geometry

Georgios Arvanitidis[*1,2] **Miguel González-Duque**[*3] **Alison Pouplin**[*1]
Dimitris Kalatzis[*1] **Søren Hauberg**[*1]
[1] Technical University of Denmark, Section for Cognitive Systems, Lyngby, Denmark
[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[3] IT University of Copenhagen, Creative AI Lab, Copenhagen, Denmark

Abstract

Latent space geometry has shown itself to provide a rich and rigorous framework for interacting with the latent variables of deep generative models. The existing theory, however, relies on the decoder being a Gaussian distribution as its simple reparametrization allows us to interpret the generating process as a random projection of a deterministic manifold. Consequently, this approach breaks down when applied to decoders that are not as easily reparametrized. We here propose to use the Fisher-Rao metric associated with the space of decoder distributions as a reference metric, which we pull back to the latent space. We show that we can achieve meaningful latent geometries for a wide range of decoder distributions for which the previous theory was not applicable, opening the door to 'black box' latent geometries.

1 Introduction

Generative models such as *variational autoencoders (VAEs)* (Kingma and Welling, 2014; Rezende et al., 2014) and *generative adversarial networks (GANs)* (Goodfellow et al., 2014) provide state-of-the-art density estimators for high dimensional data. The underlying assumption is that data $\mathbf{x} \in \mathcal{X}$ lie near a low-dimensional manifold $\mathcal{M} \subset \mathcal{X}$, which is parametrized through a low-dimensional *latent representation* $\mathbf{z} \in \mathcal{Z}$. As data is finite and noisy, we only recover a probabilistic estimate of the true manifold, which, in VAEs, is represented through a decoder distribution $p(\mathbf{x}|\mathbf{z})$. Our target is the geometry of this *random manifold*.

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. [*]Denotes equal contribution.

The geometry of the manifold has been shown to carry great value when systematically interacting with the latent representations, as it provides a stringent solution to the *identifiability problem* that plagues latent variable models (Tosi et al., 2014; Arvanitidis et al., 2018; Hauberg, 2018). For example, this geometry has allowed VAEs to discover latent evolutionary signals in proteins (Detlefsen et al., 2020), provide efficient robot controls (Scannell et al., 2021; Chen et al., 2018b; Beik-Mohammadi et al., 2021), improve latent clustering abilities (Yang et al., 2018; Arvanitidis et al., 2018) and more. The fundamental issue with these geometric approaches is that the studied manifold is inherently a stochastic object, but classic differential geometry only supports the study of *deterministic* manifolds. To bridge the gap, Eklund and Hauberg (2019) have shown how VAEs with a Gaussian decoder family can be viewed as a random projection of a deterministic manifold, thereby making the classic theories applicable to the random manifold.

A key strength of VAEs is that they can model data from diverse modalities through the choice of decoder distribution $p(\mathbf{x}|\mathbf{z})$. For discrete data, we use categorical decoders, while for continuous data we may opt for a Gaussian, a Gamma or whichever distribution best suits the data. However, for non-Gaussian decoders, there exists no useful approach for treating the associated random manifold as deterministic, which prevents us from systematically interacting with the latent representations without being subjected to identifiability issues. This limitation motivates the current work.

In this paper, we provide a general framework that allows us to interact with the geometry of almost any random manifold. The key, and simple idea is to reinterpret the decoder as spanning a deterministic manifold in the space of probability distributions $\mathcal{H}$, rather than a random manifold in the observation space (see Fig. 1). Calling on classical *information geometry* (Amari, 2016; Nielsen, 2020), we show that the learned manifold is a Riemannian manifold of $\mathcal{H}$, and provide

the corresponding computational tools. The approach is applicable to any family of decoders for which the KL-divergence can be differentiated, allowing us to work with a wide range of models from a single codebase.

2 The geometry of generative models

As a starting point, consider the deterministic generative model given by a prior $p(\mathbf{z})$ and a decoder $f : \mathcal{Z} = \mathbb{R}^d \to \mathcal{X} = \mathbb{R}^D$, which is assumed to be a smooth immersion. The latent representation $\mathbf{z}$ of an observation $\mathbf{x}$ is generally not *identifiable*, meaning that one can recover different latent representations that give rise to equally good density estimates. For example, let $g : \mathcal{Z} \to \mathcal{Z}$ be a smooth invertible function such that $\mathbf{z} \sim p(\mathbf{z}) \Leftrightarrow g(\mathbf{z}) \sim p(\mathbf{z})$, then the latent representation $g(\mathbf{z})$ coupled with the decoder $f \circ g^{-1}$ gives the same density estimate as $\mathbf{z}$ coupled with f (Hauberg, 2018). Practically speaking, the identifiability issue implies that it is improper to view the latent space $\mathcal{Z}$ as being Euclidean, as any reasonable view of $\mathcal{Z}$ should be invariant to reparametrizations g.

The classic geometric solution to the identifiability problem is to define any quantity of interest in the observation space $\mathcal{X}$ rather than the latent space $\mathcal{Z}$. For example, the length of a curve $\gamma : [0, 1] \to \mathcal{Z}$ in the latent space can be defined as its length measured in $\mathcal{X}$ on the manifold $\mathcal{M} = f(\mathcal{Z})$ with $N \to +\infty$ as:

$$\begin{aligned} \mathrm{L}(\gamma) &= \sum_{n=1}^{N-1} \|f(\gamma(t_{n+1})) - f(\gamma(t_n))\| = \int_0^1 \|\dot{f}(\gamma(t))\| \mathrm{d}t \\ &= \int_0^1 \sqrt{\dot{\gamma}(t)^\intercal \mathbf{J}_f(\gamma(t))^\intercal \mathbf{J}_f(\gamma(t)) \dot{\gamma}(t)} \mathrm{d}t, \end{aligned} \tag{1}$$

where $t_n = {}^n/_N$ and $t_{n+1} = {}^{n+1}/_N$ and we used the chain rule $\partial_t f(\gamma(t)) = \mathbf{J}_f(\gamma(t))\dot{\gamma}(t)$ with $\dot{\gamma}(t) = \partial_t \gamma(t)$ being the curve derivative, and $\mathbf{J}_f(\gamma(t)) \in \mathbb{R}^{D \times d}$ the Jacobian of f at $\gamma(t)$. This construction shows how we may calculate lengths in the latent space with respect to the metric of the observation space, which is typically assumed to be the Euclidean, but other options exist (Arvanitidis et al., 2021). In this way, the symmetric positive definite matrix $\mathbf{J}_f(\gamma(t))^\intercal \mathbf{J}_f(\gamma(t))$ is denoted by $\mathbf{M}(\gamma(t)) \in \mathbb{R}^{d \times d}_{\succ 0}$ and captures the geometry of $\mathcal{M}$ in $\mathcal{Z}$. This is known as the *pullback metric* as it pulls the Euclidean metric from $\mathcal{X}$ into $\mathcal{Z}$. As the Jacobian spans the d-dimensional tangent space at the point $\mathbf{x} = f(\mathbf{z})$, we may interpret $\mathbf{M}(\mathbf{z})$ as an inner product $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{M}} = \mathbf{u}^\intercal \mathbf{M}(\mathbf{z}) \mathbf{v}$ over this tangent space, given us all the ingredients to define *Riemannian manifolds*:

Definition 2.1. *A Riemannian manifold is a smooth manifold $\mathcal{M}$ together with a Riemannian metric $\mathbf{M}(\mathbf{z})$, which is a positive definite matrix that changes smoothly throughout space and defines an inner product on the tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{M}$.*

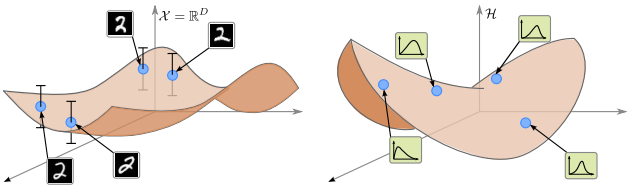

Figure 1: Traditionally (left), we view the learned manifold as a stochastic manifold in the observation space. We propose (right) to view the learned manifold as a deterministic manifold embedded in the space of decoder distributions, which is equipped with a Fisher-Rao metric based on information geometry.

We see that the decoder naturally spans a Riemannian manifold and the latent space $\mathcal{Z}$ can be considered as the *intrinsic coordinates*. Technically, we can consider any Euclidean space as the intrinsic coordinates of an abstract $\mathcal{M}$ using a suitable metric $\mathbf{M}(\mathbf{z})$, which is implicitly induced by an abstract f. Since the Riemannian length of a latent curve (1), by construction, is invariant to reparametrizations, it is natural to extend this view with a notion of *distance*. We say that the distance between two points $\mathbf{z}_0, \mathbf{z}_1 \in \mathcal{Z}$ is simply the length of the shortest connecting path, $\mathrm{dist}(\mathbf{z}_0, \mathbf{z}_1) = \min_\gamma \mathrm{L}(\gamma)$. Calculating distances implies finding the shortest path. One can show (Gallot et al., 2004) that length minimizing curves also have minimal *energy*:

$$\mathrm{E}(\gamma) = \int_0^1 \|\dot{f}(\gamma(t))\|^2 \mathrm{d}t = \int_0^1 \dot{\gamma}(t)^\intercal \mathbf{M}(\gamma(t)) \dot{\gamma}(t) \mathrm{d}t, \tag{2}$$

which is a locally convex functional. Shortest paths can then be found by direct energy minimization (Yang et al., 2018) or by solving the associated system of ordinary differential equations (ODEs) (Hennig and Hauberg, 2014; Arvanitidis et al., 2019) (see supplementary materials for additional details).

2.1 Stochastic decoders

As previously discussed, deterministic decoders directly induce a Riemannian geometry in the latent space. However, most models of interest are stochastic and there is significant evidence that this stochasticity is important to faithfully capture the intrinsic structure of data (Hauberg, 2018). When the decoder is a smooth stochastic process, e.g. as in the Gaussian Process Latent Variable Model (GP-LVM) (Lawrence, 2005), Tosi et al. (2014) laid the foundations for modeling a stochastic geometry. Most contemporary models, such as VAEs, assume independent noise, making this theory inapplicable. Arvanitidis et al. (2018) proposed an extension of this stochastic geometry to VAEs with

Gaussian decoders, which take the form

$$f(\mathbf{z}) = \mu(\mathbf{z}) + \sigma(\mathbf{z}) \odot \epsilon$$
$$= \begin{bmatrix}\mathbb{I}_D & \text{diag}(\epsilon)\end{bmatrix} \begin{bmatrix}\mu(\mathbf{z}) \\ \sigma(\mathbf{z})\end{bmatrix} = \mathbf{P}_\varepsilon\ h(\mathbf{z}), \quad (3)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_D)$. Here we have written the Gaussian decoder in its reparametrized form. This can be viewed as a random projection of a deterministic manifold spanned by h with projection matrix $\mathbf{P}_\epsilon$ (Eklund and Hauberg, 2019), which can easily be given a geometry. The associated Riemannian metric,

$$\mathbf{M}(\mathbf{z}) = \mathbf{J}_\mu(\mathbf{z})^\intercal \mathbf{J}_\mu(\mathbf{z}) + \mathbf{J}_\sigma(\mathbf{z})^\intercal \mathbf{J}_\sigma(\mathbf{z}), \quad (4)$$

gives shortest paths that follow the data as distances grow with the model uncertainty (Arvanitidis et al., 2018; Hauberg, 2018). An example of a shortest path $\gamma(t) \in \mathcal{Z}$ computed under this metric is shown in Fig. 2 and the respective curve on the corresponding expected manifold $\mu(\gamma(t)) \in \mathcal{M} \subset \mathcal{X}$.

Previous work has, thus, focused on *pulling back* the Euclidean metric from the observation space to the latent space using the reparametrization of the Gaussian decoder. This is, however, intrinsically linked with the simple reparametrization of the Gaussian, and this strategy can only extend to location-scale distributions. We propose an alternative, principled way of dealing with stochasticity by changing the focus from the observation space $\mathcal{X}$ to the parameter space $\mathcal{H}$ associated to the distribution of the decoder, leveraging the metrics defined in classical information geometry.

3 Information geometric latent metric

So far we have seen how we can endow the latent space $\mathcal{Z}$ with meaningful distances only when our stochastic decoders are reparameterizable and their codomain is the observation space $\mathcal{X}$. Ideally, we would like a more general framework of computing shortest path distances for a more general class of distributions.

We first note that the codomain of a VAE decoder is the parameter space $\mathcal{H}$ of a probability density function. In particular, depending on the type of data we specify a likelihood $p(\mathbf{x}|\eta)$ with parameters $\eta \in \mathcal{H}$, which we can rewrite as $p(\mathbf{x}|\mathbf{z})$ using the mapping $h : \mathcal{Z} \to \mathcal{H}$.

With this in mind, we can ask what is a natural distance in the latent space $\mathcal{Z}$ between two infinitesimally near points $\mathbf{z}_1$ and $\mathbf{z}_2 = \mathbf{z}_1 + \epsilon$ when measured in $\mathcal{H}$. Since our latent codes map to distributions we can define the (infinitesimal) distance through the KL-divergence:

$$\text{dist}^2(\mathbf{z}_1, \mathbf{z}_2) = \text{KL}(p(\mathbf{x}|\mathbf{z}_1),\ p(\mathbf{x}|\mathbf{z}_2)). \quad (5)$$

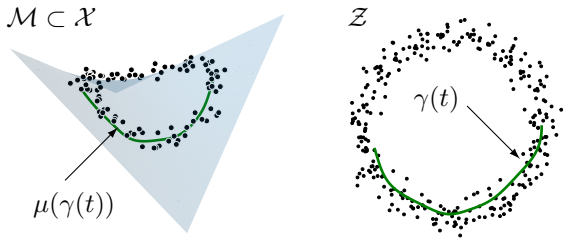

Figure 2: A conceptual example of a Riemannian manifold $\mathcal{M} = \mu(\mathcal{Z})$ lying in $\mathcal{X}$ and the corresponding latent space $\mathcal{Z}$, together with an associated shortest paths.

So we can define the length of a curve $\gamma : [0, 1] \to \mathcal{Z}$ as

$$\text{L}(\gamma) = \lim_{N\to\infty} \sum_{n=1}^{N-1} \text{KL}(p(\mathbf{x}|\gamma(t_n)), p(\mathbf{x}|\gamma(t_{n+1})))^{\frac{1}{2}}, \quad (6)$$

and distances could be defined as before. This would satisfy our desiderata of a deterministic notion of similarity in the latent space that is applicable to wide range of decoder distributions.

This construction may seem arbitrary, but in reality it carries deeper geometric meaning. *Information geometry* (Nielsen, 2020) considers families of probabilistic densities $p(\mathbf{x}|\eta)$ as represented by their parameters $\eta \in \mathcal{H}$, such that $\mathcal{H}$ is constructed as a statistical manifold equipped with the Fisher-Rao metric, which infinitesimally coincides with the KL divergence in (5). This is known to be a Riemannian metric over $\mathcal{H}$ that takes the following form:

$$\mathbf{I}_\mathcal{H}(\eta) = \int_\mathcal{X} [\nabla_\eta \log p(\mathbf{x}|\eta) \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal] p(\mathbf{x}|\eta) \text{d}\mathbf{x}. \quad (7)$$

When the parameter space $\mathcal{H}$ is equipped with this metric, we call it a *statistical manifold*.

Definition 3.1. *A statistical manifold consists of the parameter space $\mathcal{H}$ of a probability density function $p(\mathbf{x}|\eta)$ equipped with the Fisher-Rao information matrix $\mathbf{I}_\mathcal{H}(\eta)$ as a Riemannian metric.*

Note that the geometry induced by the Fisher-Rao metric is predefined and can be seen as a modeling decision, since it is related to the chosen likelihood and does not change with data.

As previously mentioned, a known result in Information Geometry is that the Fisher-Rao metric coincides with the KL-divergence locally (Nielsen, 2020; Amari, 2016):

Proposition 3.1. *The Fisher-Rao metric is the second order approximation of the KL-divergence between perturbed distributions:*

$$\text{KL}(p(\mathbf{x}|\eta), p(\mathbf{x}|\eta + \delta\eta)) = \frac{1}{2} \delta\eta^\intercal \mathbf{I}_\mathcal{H}(\eta) \delta\eta + o(\delta\eta^2). \quad (8)$$

The central idea put forward in this paper is to consider the decoder as a map $h : \mathcal{Z} \to \mathcal{H}$ instead of $f : \mathcal{Z} \to \mathcal{X}$, and let $\mathcal{H}$ be equipped with the appropriate Fisher-Rao metric. The VAE can then be interpreted as spanning a manifold $h(\mathcal{Z})$ in $\mathcal{H}$ and the latent space $\mathcal{Z}$ can be endowed with the corresponding metric. We detail this approach in the sequel.

3.1 The Riemannian pull-back metric

Our construction implies that the length of a latent curve $\gamma : [0, 1] \to \mathcal{Z}$ when mapped through h can be measured in the parameter space $\mathcal{H}$ using the Fisher-Rao metric therein as

$$\mathrm{L}(\gamma) = \int_0^1 \sqrt{\partial_t h(\gamma(t))^\intercal \mathbf{I}_{\mathcal{H}}(h(\gamma(t)))\partial_t h(\gamma(t))}\mathrm{d}t, \quad (9)$$

with $\mathbf{M}$ the pullback metric:

Proposition 3.2. *Let $h : \mathcal{Z} \to \mathcal{H}$ be an immersion that parametrizes the likelihood. Then, the latent space $\mathcal{Z}$ is equipped with the Riemannian pull-back metric $\mathbf{M}(\mathbf{z}) = \mathbf{J}_h^\intercal(\mathbf{z})\mathbf{I}_{\mathcal{H}}(h(\mathbf{z}))\mathbf{J}_h(\mathbf{z})$.*

Proof. See appendix, Prop. C.1. □

Note that instead of considering the parameters $\eta \in \mathcal{H}$ of the probabilistic density function $p(\mathbf{x}|\eta)$ that approximates the data, we can consider the latent variable $\mathbf{z}$ as the actual parameters of the model. This view is equivalent to the one explained above, and the corresponding pull-back metric is directly the Fisher-Rao metric endowed in the latent space $\mathcal{Z}$:

Proposition 3.3. *The pullback metric $\mathbf{M}(\mathbf{z})$ is identical to the Fisher-Rao metric obtained over the parameter space $\mathcal{Z}$ as $\mathbf{M}(\mathbf{z}) = \int_{\mathcal{X}} [\nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z}) \nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z})^\intercal] \, p(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{x}$.*

Proof. See appendix, Prop. C.2. □

Therefore, pulling back the Fisher-Rao metric from $\mathcal{H}$ into $\mathcal{Z}$ enables us to compute length minimizing curves which are indentifiable (see Sec. 2). The advantange of this approach is that it applies to any type of decoders and data, as the actual distance is measured over the manifold spanned by h in the parameter space $\mathcal{H}$. So shortest paths between probability distributions move optimally on this manifold while taking the geometry of $\mathcal{H}$ into account through the Fisher-Rao metric.

Computing shortest paths directly in $\mathcal{H}$ need not result in a sensible sequence of probability density functions $p(\mathbf{x}|\eta)$. To ensure that the shortest paths computed under our metric stay within the support of the data, we carefully design our decoder h to extrapolate to uncertain distributions outside the support of the data (see supplements for additional details).

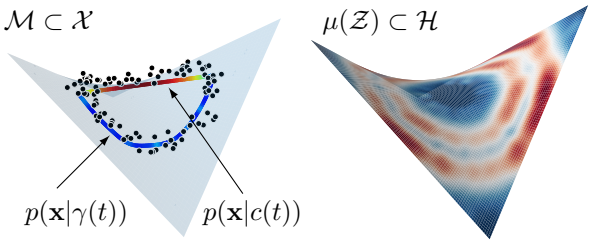

Figure 3: *Left*: The optimal $\gamma(t)$ under $\mathbf{M}(\mathbf{z})$ results to distributions that respect the structure of data, while the curve $c(t)$ with minimal length in $\mathcal{H}$ does not as it leaves $\mathcal{M}$. Red and green signal high and low variance respectively. *Right*: A part of the spanned manifold $h(\mathcal{Z}) = [\mu(\mathcal{Z}), \sigma(\mathcal{Z})] \in \mathcal{H}$ colored by $|\mathbf{M}(\mathbf{z})|$. Note that we design $\sigma(\mathbf{z})$ to increase far from data, which ensures that $\gamma(t)$ stays within their support.

In Fig. 3 we compare a shortest path $\gamma : [0, 1] \to \mathcal{Z}$ under the proposed metric $\mathbf{M}(\mathbf{z})$ against a curve $c : [0, 1] \to \mathcal{H}$ with minimal length. We consider a Gaussian likelihood with isotropic covariance. We show the resulting sequence of means for both interpolants color-coded by the corresponding variances. As expected $c(t)$ does not take into account the given data, but only respects the geometry of $\mathcal{H}$ implied by the likelihood.

3.2 Efficient shortest path computation

An essential task in computational geometry is to compute shortest paths. This can be achieved by minimizing curve energy (2) or solving the corresponding system of ODEs (see supplementary material). The latter, however, requires inordinate computational resources, since the evaluation of the system relies on the Jacobian of the decoder and its derivatives.

Bearing in mind that the metric is an approximation of the KL divergence between perturbations (8), the energy is directly expressed as a sum of KL divergence terms along a discretized curve γ:

$$\mathrm{E}(\gamma) \propto \lim_{N \to \infty} \sum_{n=1}^{N-1} \mathrm{KL}(p(\mathbf{x}|\gamma(t_n)), p(\mathbf{x}|\gamma(t_{n+1}))). \quad (10)$$

The proof can be found in the appendix, Prop. A.2. A simple algorithm for computing shortest paths is to minimize (10) with respect to the parameters of the curve γ. Here we represent γ as a cubic spline with fixed end-points. Then standard free-form optimization can be applied to minimize this energy.

3.3 Example: categorical decoders

The motivation for our approach is that, while several options for decoders exist in VAEs depending on the type of the given data, we could only capture and use the learned geometry in a principled way with Gaussian decoders. Our proposed methodology is more general.

For a constructive example, assume that $\mathbf{x}$ is a categorical variable. We can select a generalized Bernoulli likelihood $p(\mathbf{x}|\mathbf{z})$, such that $h(\mathbf{z}) = (\eta_1, \cdots, \eta_D)$ where each η_i represents the probability of x_i being 1. Thus, the parameters η lie on the unit simplex $\mathcal{H}$, and the distance under the corresponding Fisher-Rao metric between points on the simplex coincides with the spherical distance between the points $\sqrt{\eta}$ on the unit sphere,

$$\text{dist}(\eta, \eta') = \arccos\left(\sqrt{\eta}^\intercal \sqrt{\eta'}\right). \tag{11}$$

We derive in detail this previously known result in the supplementary materials.

Given a curve $\gamma : [0,1] \to \mathcal{Z}$ we can approximate the energy by using the small angle approximation $\cos\theta \approx 1 - {}^{\theta^2}/_2 \Leftrightarrow \theta^2 \approx 2 - 2\cos\theta$ to give

$$\text{E}(\gamma) = \sum_{n=1}^{N-1} \left(2 - 2\sqrt{h(\gamma(t_n))}^\intercal \sqrt{h(\gamma(t_{n+1}))}\right), \tag{12}$$

for sufficiently fine discretization with $t_n = {}^n/_N$ and $t_{n+1} = {}^{n+1}/_N$. This gives a particular simple expression for the energy, which we can minimize in order to compute the shortest path.

3.4 Black-box random geometry

In general, we can derive suitable expressions for computing metrics and energies for families of decoders, doing so is tedious, error-prone and time-consuming. This limits the practical use of the developed theory.

Drawing inspiration from *black-box variational inference* (Ranganath et al., 2014), we propose a notion of *black-box random geometry*. Assume that we have access to a differentiable KL divergence for our choice of decoder distribution. We can then apply the methodology presented in Sec. 3.2 to compute shortest paths.

In practice, modern libraries such as PyTorch (Paszke et al., 2019) have this functionality implemented for several distributions. When we do not have closed-form expression for the KL divergence, we can resort to Monte Carlo estimates thereof. More specifically, we can estimate the KL divergence by generating samples from the likelihood based on the re-parametrization trick, which allows us to get derivatives with automatic differentiation.

Interestingly, apart from finding the shortest path through the KL formulation, we can also approximate the actual metric tensor $\mathbf{M}(\mathbf{z})$. As we have discussed above, evaluating explicitly this metric is not a trivial task in many cases. One problem is that we need access to the Jacobian of the parametrization h, which is typically a deep neural network, so the computation is not always straightforward. Alternatively, one could use that the Fisher-Rao metric is the Hessian of the KL-divergence (8), but such approaches fare poorly with current tools for automatic differentiation, where higher-order derivatives are often incompatible with batching. Furthermore, the Fisher-Rao metric itself may be intractable depending on the chosen likelihood $p(\mathbf{x}|\eta)$. Nevertheless, we show that the KL formulation (8) allows us to approximate the latent metric as:

Proposition 3.4. *We define perturbations vectors as* $\delta e_i = \varepsilon \cdot \mathbf{e_i}$, *with* $\varepsilon \in \mathbb{R}_+$ *a small infinitesimal quantity, and* $\mathbf{e_i}$ *a canonical basis vector in* $\mathbb{R}^d$. *For better clarity, we rename* $\text{KL}(p(\mathbf{x}|\mathbf{z}), p(\mathbf{x}|\mathbf{z}+\delta\mathbf{z})) = \text{KL}_\mathbf{z}(\delta\mathbf{z})$ *and we note* $\mathbf{M}_{ij} = \mathbf{M}_{ji}$ *the components of* $\mathbf{M}(\mathbf{z})$. *We can then approximate by a system of equations the diagonal and non-diagonal elements of the metric:*

$$\mathbf{M}_{ii} \approx 2\ \text{KL}_\mathbf{z}(\delta\mathbf{e_i})/\varepsilon^2$$
$$\mathbf{M}_{ji} \approx \left(\text{KL}_\mathbf{z}(\delta\mathbf{e_i}+\delta\mathbf{e_j}) - \text{KL}_\mathbf{z}(\delta\mathbf{e_i}) - \text{KL}_\mathbf{z}(\delta\mathbf{e_j})\right)/\varepsilon^2.$$

See Prop. C.4 in the appendix for a proof. Note that this formulation only requires h to be a smooth immersion. This is particularly useful, as the metric is used for other purposes on a Riemannian manifold and not exclusively for computing shortest paths. For example, relying on $\mathbf{M}(\mathbf{z})$ we can compute the exponential map by solving the corresponding ODE system as an initial value problem. Assuming a fully differentiable KL divergence, then the approximated metric is also differentiable. This is all that is required for practical usage of differential geometry, and thus, we have a reasonable notion of *black-box random geometry*.

4 Experiments

4.1 Pulling back Euclidean and Fisher-Rao metric with Gaussian decoders

We start our experiments by comparing our proposed way of inducing geometry in latent spaces with the existing theory: pulling back the Euclidean metric using a stochastic Gaussian decoder (see (4)). We also include in this comparison the effect of regularizing the uncertainty quantification in the learned geometries. In this regularization, we use transition networks (Detlefsen et al., 2019) to ensure high uncertainty outside the support of the data (see Sec. C.2 in the supplementary material).

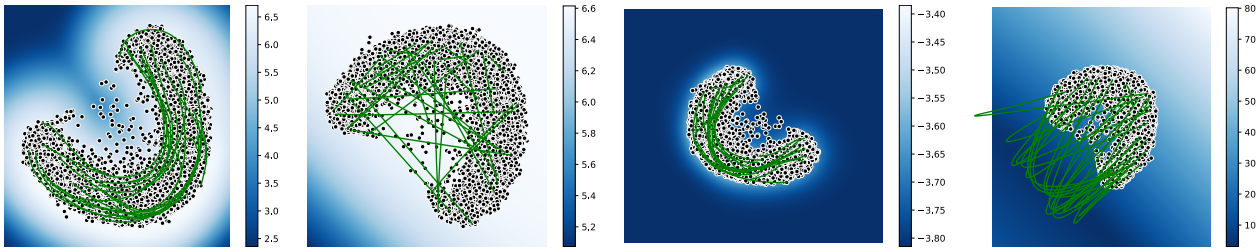

Figure 4: Pulling back the Euclidean and Fisher-Rao metrics with Gaussian decoders. Left to right: Euclidean pull-back with regularized uncertainty, Euclidean pull-back with a NN to model uncertainty, Fisher-Rao pull-back with regularized uncertainty, Fisher-Rao pull-back with a NN to model uncertainty.

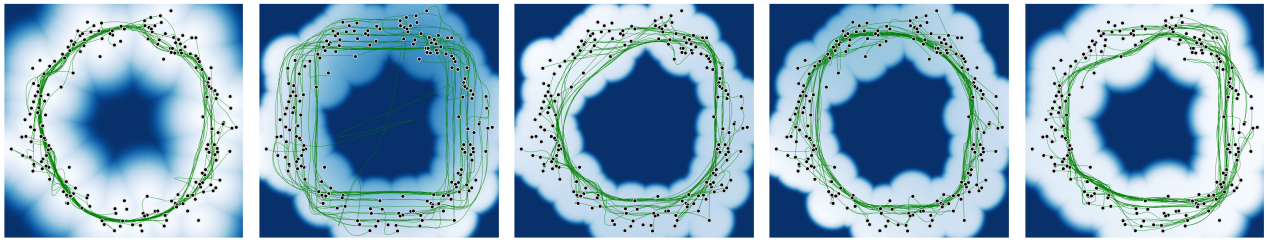

Figure 5: Pulling back the metric from different parameter spaces. From left to right: Normal, Bernoulli, Beta, Dirichlet and Exponential. White areas represent low entropy of the decoded distribution, while blue areas represent higher entropy. Notice that the Bernoulli latent space is darker blue (i.e. more entropic) because distributions with parameters around $1/2$ are near uniform.

In this experiment, we train four VAEs on a subset of the MNIST dataset composed of only the digits with label 1. Two of these VAEs implement a standard Gaussian decoder, and we induce a metric in the latent space by pulling the Euclidean metric back using the Jacobian of the decoder. In the other two, we consider the output of the decoder as lying in a statistical manifold and approximate the pullback of the Fisher-Rao metric by using the KL divergence locally. In each of these two sets, one of the decoders implements the uncertainty regularization described above.

Fig. 4 shows the latent spaces of these four decoders, illuminated by the volume measure. In each of this latent spaces, we analyze the geometry induced by the respective pullbacks by computing and plotting several shortest paths. This figure illustrates two key findings: (1) Our approach is on par with the existing literature in learning geometric structure, which can be seen by comparing the first and third latent spaces (Euclidean vs. Fisher Rao, respectively), and (2) Performing uncertainty regularization plays an instrumental role on learning a sensible geometric structure, which can be seen when comparing the first and second latent spaces (both coming from the Euclidean pullback, with and without regularization respectively), and similarly for the third and fourth.

4.2 The Fisher-Rao pullback metric for various distributions with toy data

For our second experiment, we induced a geometry on a known latent space (given by noisy circular data in $\mathcal{Z}_{\text{toy}} = \mathbb{R}^2$) by *pulling back* the Fisher-Rao metric from the parameter space of different distributions, showcasing the potential for computing shortest paths efficiently, even in non-Gaussian settings. The statistical manifolds from which we pull the metric are associated with multivariate versions of the Normal, Bernoulli, Beta, Dirichlet and Exponential distributions. For this approximation to follow the support of the data we need to ensure that our mapping $\mathcal{Z}_{\text{toy}} \to \mathcal{H}$ extrapolates to high uncertainty outside our training codes (see Fig. 4). To do so, we perform uncertainty regularization for each one of the decoded distributions (see supplementary materials for implementation details).

In Fig. 5 we show the toy latent space alongside several shortest paths computed using the pullback of the Fisher-Rao metric from the statistical manifolds associated with the Gaussian, Bernoulli, Beta, Dirichlet and Exponential distributions. We parametrize the curves as cubic splines and minimize their energy using automatic differentiation (see Sec. 3.2). These results show that the approximated pulled-back metric induces a meaningful geometry in this latent space, which recovers the true circular structure of the data. In the

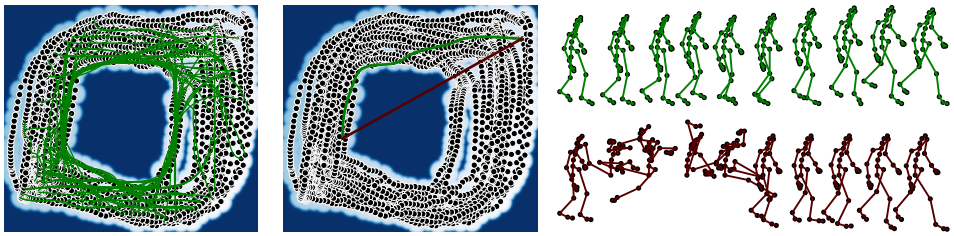

Figure 6: *Left:* Geodesics in the latent space of a von Mises-Fisher decoder. *Middle*: Shortest path (green) vs. linear (red). *Right*: decoding the shortest path (green) vs. the linear interpolation (red) as poses (i.e. the product of von Mises-Fisher distributions). Our path follows the trend of the data manifold, while the linear path traverses regions with no data support.

case of the Bernoulli distribution, we notice that some of the paths fail to converge. We hypothesize that our uncertainty regularization (which decodes to the uniform distribution outside the support) is not strong enough since Bernoulli distributions with parameters close to 1/2 are already highly entropic.

4.3 Motion capture data with products of von Mises-Fisher distributions

As a further demonstration of our black-box random geometry, we consider a model of human motion capture data. Here we observe a time series, where each time point represent a 'skeleton' corresponding to a human pose. As only pose, and not shape, changes over time, individual limbs on the body only change position and orientation, but not length. Each limb is then a point on a sphere in $\mathbb{R}^3$ with radius given by the limb length. Following Tournier et al. (2009) we view the skeleton representation space as a product of spheres. From this, we build a VAE where the decoder distribution is a product of von Mises-Fisher distributions. To ensure a sensible uncertainty estimates in the decoder, we enforce that the concentration parameter extrapolate to a small constant.

In this case, we do not have easily accessible Fisher-Rao metrics, so we lean on the KL formulation from Sec. 3.4. Since, the KL does not have a closed-form expression for the von Mises-Fisher distribution, we resort to a Monte Carlo estimate thereof. This is realisable with off-the-shelf tools (Davidson et al., 2018).

Fig. 6 shows the latent representation of a motion capture sequence of a person walking (Seq. `69_06` from `http://mocap.cs.cmu.edu/`) with shortest paths superimposed. We see that our paths follow the trend of the data, and reflect the underlying periodic nature of the observed walking motion. We pick two random points in the latent space, and traverse both the shortest path and the straight line implied by a Euclidean interpretation of the latent space. As we traverse, we sample from the decoder distribution, thereby producing two new motion sequences, which appear in Fig. 6. As can be seen, the straight line traverses uncharted territory of the latent space and end up creating an implausible motion. This is in contrast to the shortest path, that consistently generates meaningful poses.

4.4 Numerical approximation of the Fisher-Rao pullback metric

Prop. 3.4 provide an approximation to the metric and we test its accuracy as per (8). We discretize the latent space for the just-described von Mises-Fisher decoder and, for each $\mathbf{z}$ in this grid, we both approximate $\mathbf{M}(\mathbf{z})$ and compute the expected value of $\|\mathrm{KL}(p(\mathbf{x}|\mathbf{z}), p(\mathbf{x}|\mathbf{z}+\delta\mathbf{z})) - \frac{1}{2}\delta\mathbf{z}^\top\mathbf{M}(\mathbf{z})\delta\mathbf{z}\|$ for several samples of $\delta\mathbf{z}$, uniformly distributed around the circle of radius $\varepsilon = 0.1$. Notice that we do not have a ground truth to compare against, and that this error will always be off by $o(\delta\mathbf{z}^2)$. Fig. 7 shows the average error, where we can see that the approximate metric is well-estimated both within and outside the support of the data. The error, however, grows at the boundaries of the support, where the distribution is changing from a concentrated von Mises-Fisher to a uniform distribution. It is worth mentioning that we observe some approximated metrics have negative determinant, showing that our numerical approximations are imprecise at the boundary. These results warrant further research on more stable ways of approximating pulled back metrics under our proposed approach.

4.5 Statistical models on manifolds

We demonstrate the usefulness of the approximated metrics, by fitting a distribution to data in the latent space, which requires normalization according to the measure induced by the metric. In particular, we fit a locally adaptive normal distribution (LAND) (Arvanitidis et al., 2016), which extends the Gaussian distribution to learned manifolds. The probability density function is $\rho(\mathbf{z}) = C(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \cdot \exp\left(-0.5 \cdot \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{z})^\intercal \boldsymbol{\Gamma} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{z})\right)$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean, $\boldsymbol{\Gamma} \in \mathbb{R}^{d\times d}_{\succ 0}$ is the preci-

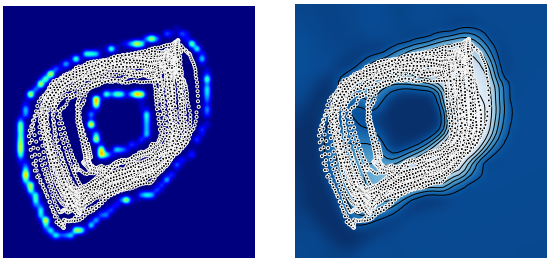

Figure 7: *Left*: Average error of the approximated metric in the von Mises-Fisher latent space. Darker colors indicate lower error (less than ε^2), while higher values are clear. *Right*: The LAND density well-adapts to the nonlinear structure of the latent representations due to the shortest paths behavior.

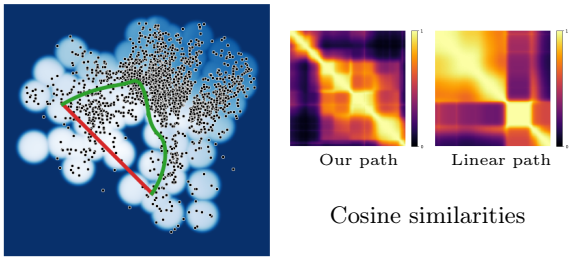

Figure 8: Our path (green) follows users with similar preferences, as similarity is only locally high. Instead, the line (red) does not respect the learned structure resulting to users with no specific preferences.

sion matrix and $C(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ the normalization constant. The operator $\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{z})$ returns the scaled initial velocity $\mathbf{v} = \dot{\gamma}(0) \in \mathbb{R}^d$ of the shortest connecting path with $\gamma(1) = \mathbf{z}$ and $||\mathbf{v}|| = \mathrm{Length}(\gamma)$. In Fig. 7 we show the LAND density on the learned latent representations under the approximated Riemannian metric from Sec. 4.4. Since shortest paths follow the data, so does the density ρ. See supplementary material for details.

4.6 Movie preferences via latent interpolants

In addition, we explored the latent space of the movie-users rating dataset MovieLens 25M (https://grouplens.org/datasets/movielens/25m/). In particular, we consider a Bernoulli VAE to model if a user has watched a movie among the 60 most popular in the dataset. Also, we considered only users who have seen less than 30 movies. The implementation and preprocessing details can be found in the supplementary material. Our VAE decodes to 60 Bernoulli parameters that are conditionally independent given the latent code $\mathbf{z}$, which state the likelihood that a given user has seen these movies. Latent codes in this space, then, can be seen as individual users with certain movie preferences.

We then computed the shortest path between two points by considering the pulled-back Fisher-Rao (see Sec. 3.2), and we compare against a straight line interpolation. We consider the cosine similarity of the decoded outputs. This cosine similarity measures whether two users (encoded as points in the latent space) have similar preferences according to our model. In Fig. 8 we see that our path follows users with similar movie preferences locally, while the linear interpolation failed to capture a local notion of preference.

5 Related work

The literature is rich on deterministic generative models such as autoencoders (Rumelhart et al., 1986) and generative adversarial networks (Goodfellow et al., 2014),and a series of papers have investigated such deterministic decoders (Shao et al., 2018; Chen et al., 2018a; Laine, 2018). However, our work is not applicable to this setting. As demonstrated in Sec. 4.1 stochasticity is essential to shape the latent space according to the data manifold. Hauberg (2018) argues model uncertainty plays a role much akin to topology in classic geometry, in that it, practically, allows us to deviate from the Euclidean topology of the latent space.

Our constructions rely on information geometry and in particular Fisher-Rao metrics (Nielsen, 2020). While our work is within the spirit of information geometry, it does not represent typical usage of this theory. Information geometry has been widely used in the context of optimisation with *natural gradients* (Martens, 2014; Martens and Grosse, 2015), Markov Chain Monte Carlo methods (Girolami and Calderhead, 2011) and hypothesis testing (Nielsen, 2020). The key difference between natural gradients and our work is the space we wish to explore: in the case of the natural gradients, the shortest path is obtained on the space of the weights of the neural networks, while we aim to explore the latent space of a VAE. It can also be noted that Information geometry provides a rich family of alternative divergences over the here-applied KL-divergence. We did not investigate their usage in our context.

To make use of the here-developed tools, we may lean on techniques for statistics on manifolds. These provide generalizations of a long list of classic statistical algorithms (Zhang and Fletcher, 2013; Hauberg, 2015; Fletcher, 2011). We refer the reader to Pennec (2006) for a gentle introduction to this line of research.

6 Conclusion and discussion

We have proposed a new approach for getting a well-defined and useful geometry in the latent space of generative models with stochastic decoders. The theory is easy to apply and readily generalize to a large family of decoder distributions. The latent geometry gives access to a series of operations on latent variables that are invariant to reparametrizations of the latent space, and therefore are not subject to a large class of identifiability issues. Such operational representations have already shown great value in applications ranging from biology (Detlefsen et al., 2020) to robotics (Scannell et al., 2021). We have here focused on the Fisher-Rao metric, but other geometries over distributions may apply equally well, e.g. the Wasserstein geometry may be interesting to explore.

Limitations. The largest practical hurdle with the proposed methodology, is that it only works well for decoders with well-calibrated uncertainties. That is, the decoder should yield high entropy in regions of little training data to ensure that shortest paths follow the trend of the data. This constraint is shared with existing approaches (Arvanitidis et al., 2018). Some heuristics exists (Detlefsen et al., 2019), but principled approaches are currently lacking.

Acknowledgements

This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606). It also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research, innovation programme (757360) and from a research grant (15334) from VILLUM FONDEN.

References

S. Amari. *Information Geometry and Its Applications*. Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 4431559779.

G. Arvanitidis, L. K. Hansen, and S. Hauberg. A locally adaptive normal distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, jun 2016.

G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent Space Oddity: on the Curvature of Deep Generative Models. In *International Conference on Learning Representations (ICLR)*, 2018.

G. Arvanitidis, S. Hauberg, P. Hennig, and M. Schober. Fast and robust shortest paths on manifolds learned from data. In *Artificial Intelligence and Statistics (AISTATS)*, 2019.

G. Arvanitidis, S. Hauberg, and B. Schölkopf. Geometrically Enriched Latent Spaces. In *Artificial Intelligence and Statistics (AISTATS)*, 2021.

H. Beik-Mohammadi, S. Hauberg, G. Arvanitidis, G. Neumann, and L. Rozo. Learning riemannian manifolds for geodesic motion skills. In *Robotics: Science and Systems (R:SS)*, 2021.

A. L. Brigant and S. Puechmorel. The fisher-rao geometry of beta distributions applied to the study of canonical moments, 2019.

N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer, and P. Smagt. Metrics for deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1550. PMLR, 2018a.

N. Chen, A. Klushyn, A. Paraschos, D. Benbouzid, and P. van der Smagt. Active learning based on data uncertainty and model sensitivity, 2018b.

T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

N. S. Detlefsen, M. Jørgensen, and S. Hauberg. Reliable training and estimation of variance networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

N. S. Detlefsen, S. Hauberg, and W. Boomsma. What is a meaningful representation of protein sequences?, 2020.

D. Eklund and S. Hauberg. Expected path length on random manifolds. In *arXiv preprint*, 2019.

T. Fletcher. Geodesic regression on riemannian manifolds. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 75–86, 2011.

S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian metrics*, pages 51–127. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

S. Hauberg. Principal curves on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

S. Hauberg. Only bayes should learn a manifold. 2018.

S. Hauberg, O. Freifeld, and M. J. Black. A geometric take on metric learning. In *Advances in Neural Information Processing Systems (NeurIPS) 25*, 2012.

P. Hennig and S. Hauberg. Probabilistic solutions to differential equations and their application to riemannian statistics. In *Proceedings of the 17th international Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 33, 2014.

T. Hillen, K. Painter, A. Swan, and A. Murtha. Moments of von mises and fisher distributions and applications. *Mathematical Biosciences and Engineering*, 14:673–694, 12 2016. doi: 10.3934/mbe.2017038.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations, ICLR*, 2014.

S. Laine. Feature-based metrics for exploring the latent space of generative models. 2018.

N. Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *J. Mach. Learn. Res.*, 2005.

J. Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

F. Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10):1100, Sep 2020. ISSN 1099-4300. doi: 10.3390/e22101100. URL http://dx.doi.org/10.3390/e22101100.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*. 2019.

X. Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision*, 2006.

R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323, 1986.

A. Scannell, C. H. Ek, and A. Richards. Trajectory Optimisation in Learned Multimodal Dynamical Systems Via Latent-ODE Collocation. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2021.

H. Shao, A. Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 315–323, 2018.

J. Tomczak. Fisher information matrix for Gaussian and categorical distributions. https://www.ii.pwr.edu.pl/~tomczak/PDF/[JMT]Fisher_inf.pdf, 2012. Online; accessed 17 Mai 2021.

A. Tosi, S. Hauberg, A. Vellido, and N. D. Lawrence. Metrics for Probabilistic Geometries. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Reveret. Motion compression using principal geodesics analysis. In *Computer Graphics Forum*, volume 28, pages 355–364. Wiley Online Library, 2009.

T. Yang, G. Arvanitidis, D. Fu, X. Li, and S. Hauberg. Geodesic clustering in deep generative models. In *arXiv preprint*, 2018.

M. Zhang and T. Fletcher. Probabilistic principal geodesic analysis. *Advances in Neural Information Processing Systems*, 26:1178–1186, 2013.

Supplementary Material: Pulling back information geometry

A Additional details for information geometry

In this section we provide additional information regarding information geometry. We note that many of these proposition are already know in the literature, however, we include them for completion and for the paper to be standalone.

The Fisher-Rao metric is positive definite only if it is non-singular, and then, defines a Riemannian metric (Nielsen, 2020). In this paper, we assume that the observation $\mathbf{x} \in \mathcal{X}$ is a random variable following a probability distribution $p(\mathbf{x})$ such that $\mathbf{x} \sim p(\mathbf{x}|\eta)$, and any smooth changes of the parameter η would alter the observation $\mathbf{x}$. This way, the Fisher-Rao metric used in our paper is non-singular and the statistical manifold $\mathcal{H}$ is a Riemannian manifold.

A known result in *information geometry* (Nielsen, 2020; Amari, 2016) is that the Fisher-Rao metric is the first order approximation of the KL-divergence, as recall in Proposition A.1. Using this fact, we can define the Fisher-Rao distance and energy in function of the KL-divergence, leading to Proposition A.2.

Proposition A.1. *The Fisher-Rao metric is the first order approximation of the KL-divergence between perturbed distributions:*

$$\mathrm{KL}(p(\mathbf{x}|\eta),\ p(\mathbf{x}|\eta+\delta\eta)) = \frac{1}{2}\delta\eta^\top \mathbf{I}_{\mathcal{H}}(\eta)\delta\eta + o(\delta\eta^2),$$

with $\mathbf{I}_{\mathcal{H}}(\eta) = \int p(\mathbf{x}|\eta)\left[\nabla_\eta \log p(\mathbf{x}|\eta)\nabla_\eta \log p(\mathbf{x}|\eta)^\intercal\right]\mathrm{d}\mathbf{x}$.

Proof. Let's decompose $\log p(\mathbf{x}|\eta+\delta\eta)$ using the Taylor expansion:

$$\log p(\mathbf{x}|\eta+\delta\eta) = \log p(\mathbf{x}|\eta) + \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal\delta\eta + \frac{1}{2}\delta\eta^\intercal \mathrm{Hess}_\eta\left[\log p(\mathbf{x}|\eta)\right]\delta\eta + o(\delta\eta^2),$$

where the Hessian is $\mathrm{Hess}_\eta\left[\log p(\mathbf{x}|\eta)\right] = \frac{\mathrm{Hess}_\eta[p(\mathbf{x}|\eta)]}{p(\mathbf{x}|\eta)} - \nabla_\eta \log p(\mathbf{x}|\eta)\nabla_\eta \log p(\mathbf{x}|\eta)^\intercal$ and the $\nabla_\eta \log p(\mathbf{x}|\eta) = \frac{\nabla_\eta p(\mathbf{x}|\eta)}{p(\mathbf{x}|\eta)}$.

Also $\int \nabla_\eta p(\mathbf{x}|\eta)\mathrm{d}\mathbf{x} = \nabla_\eta \int p(\mathbf{x}|\eta)\mathrm{d}\mathbf{x} = 0$ and $\int \mathrm{Hess}_\eta[p(\mathbf{x}|\eta)]\mathrm{d}\mathbf{x} = \mathrm{Hess}_\eta[\int p(\mathbf{x}|\eta)\mathrm{d}\mathbf{x}] = 0$.

Replacing all those expressions to the first equation finally gives:

$$\begin{aligned}
\mathrm{KL}(p(\mathbf{x}|\eta),\, p(\mathbf{x}|\eta+\delta\eta)) &= \int p(\mathbf{x}|\eta)\log p(\mathbf{x}|\eta)\mathrm{d}\mathbf{x} - \int p(\mathbf{x}|\eta)\log p(\mathbf{x}|\eta+\delta\eta)\mathrm{d}\mathbf{x} \\
&= -\int p(\mathbf{x}|\eta)\left(\nabla_\eta \log p(\mathbf{x}|\eta)^\intercal\delta\eta + \frac{1}{2}\delta\eta^\intercal \mathrm{Hess}_\eta[\log p(\mathbf{x}|\eta)]\delta\eta + o(\delta\eta^2)\right)\mathrm{d}\mathbf{x} \\
&= \frac{1}{2}\delta\eta^\intercal\left[\int p(\mathbf{x}|\eta)\left[\nabla_\eta \log p(\mathbf{x}|\eta)\nabla_\eta \log p(\mathbf{x}|\eta)^\intercal\right]\mathrm{d}\mathbf{x}\right]\delta\eta + o(\delta\eta^2).
\end{aligned}$$

□

Definition A.1. *We consider a curve $\gamma(t)$ and its derivative $\dot\gamma(t)$ on the statistical manifold such that, $\forall t \in [0,1], \gamma(t) = \eta_t \in \mathcal{H}$. The manifold is equipped with the Fisher-Rao metric. The length and the energy functionals are defined with respect to the metric $\mathbf{I}_{\mathcal{H}}(\eta)$:*

$$\mathrm{Length}(\gamma) = \int_0^1 \sqrt{\dot\gamma(t)^\intercal \mathbf{I}_{\mathcal{H}}(\eta)\dot\gamma(t)}\mathrm{d}t \quad \textit{and} \quad \mathrm{Energy}(\gamma) = \int_0^1 \dot\gamma(t)^\intercal \mathbf{I}_{\mathcal{H}}(\eta)\dot\gamma(t)\mathrm{d}t.$$

Locally length-minimising curves between two connecting points are called geodesics. These can be found by minimizing the energy using the Euler-Lagrange equations which gives the following system of 2^{nd} order nonlinear ordinary differential equations (ODEs) (Arvanitidis et al., 2018)

$$\ddot{\gamma}(t) = -\frac{1}{2}\mathbf{I}_{\mathcal{H}}^{-1}(\gamma(t))\Big[2(\dot{\gamma}(t)^{\intercal} \otimes \mathbb{I}_d)\frac{\partial \text{vec}[\mathbf{I}_{\mathcal{H}}(\gamma(t))]}{\partial \gamma(t)}\dot{\gamma}(t) - \frac{\partial \text{vec}[\mathbf{I}_{\mathcal{H}}(\gamma(t))]}{\partial \gamma(t)}^{\intercal}(\dot{\gamma}(t) \otimes \dot{\gamma}(t))\Big]. \tag{13}$$

Proposition A.2. *The KL-divergence between two close elements of the curve γ is defined as:* $\text{KL}(p_t,\ p_{t+\delta t}) = \text{KL}(p(\mathbf{x}|\gamma(t)),\ p(\mathbf{x}|\gamma(t+\delta t)))$. *The length and the energy functionals can be approximated with respect to this KL-divergence:*

$$\text{Length}(\gamma) \approx \sqrt{2\sum_{t=1}^{T}\text{KL}(p_t, p_{t+\delta t})} \quad \textit{and} \quad \text{Energy}(\gamma) \approx \frac{2}{\delta t}\sum_{t=1}^{T}\text{KL}(p_t, p_{t+\delta t})$$

Proof. On the statistical manifold, we have $\gamma(t+\delta t) = \gamma(t) + \delta t\dot{\gamma}(t)$. The KL-divergence between perturbed distributions can be defined as: $\text{KL}(p_t, p_{t+\delta t}) = \text{KL}(p(\mathbf{x}|\gamma(t)), p(\mathbf{x}|\gamma(t+\delta t))) = \text{KL}(p(\mathbf{x}|\eta_t), p(\mathbf{x}|\eta_t + \delta\eta_t))$, with $\eta_t = \gamma(t)$ and $\delta\eta_t = \delta t\ \dot{\gamma}(t)$. Then, we obtain:

$$\text{KL}(p_t, p_{t+\delta t}) = \frac{1}{2}\delta t^2\ \dot{\gamma}(t)^{\intercal}\mathbf{I}_{\mathcal{H}}(\eta_t)\dot{\gamma}(t) + o(\delta t^2).$$

The length and energy terms appear in the following equations:

$$\int_0^1 \text{KL}(p_t, p_{t+\delta t})\text{d}t = \frac{\delta t^2}{2}\int_0^1 \dot{\gamma}(t)^{\intercal}\mathbf{I}_{\mathcal{H}}(\eta_t)\dot{\gamma}(t)\text{d}t + o(\delta t^2) = \frac{\delta t^2}{2}\ \text{Energy}(\gamma) + o(\delta t^2),$$
$$\int_0^1 \sqrt{\text{KL}(p_t, p_{t+\delta t})}\text{d}t = \frac{\delta t}{\sqrt{2}}\int_0^1 \sqrt{\dot{\gamma}(t)^{\intercal}\mathbf{I}_{\mathcal{H}}(\eta_t)\dot{\gamma}(t)}\text{d}t + o(\delta t^2) = \frac{\delta t}{\sqrt{2}}\ \text{Length}(\gamma) + o(\delta t^2).$$

If we want approximate any continuous function f with a discrete sequence, by partitioning it in T small segments, such that: $\delta t \approx \frac{1}{T}$, we have: $\int_0^1 f(t)dt \approx \sum_{t=1}^{T} f(t)\delta t$, which in our case gives:

$$\text{Length}(\gamma) \approx \sqrt{2\sum_{t=1}^{T}\text{KL}(p_t, p_{t+\delta t})} \quad \text{and} \quad \text{Energy}(\gamma) \approx \frac{2}{\delta t}\sum_{t=1}^{T}\text{KL}(p_t, p_{t+\delta t}).$$

□

A.1 The Fisher-Rao metric for several distributions

Distributions	Probability density functions	Parameters	Fisher-Rao matrix
Normal	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}$	μ, σ^2	$\mathbf{I}_{\mathcal{N}}(\mu, \sigma^2)$
Bernoulli	$\theta^{\mathbf{x}}(1-\theta)^{1-\mathbf{x}}$	θ	$\mathbf{I}_{\mathcal{B}}(\theta)$
Categorical	$\prod_{k=1}^{K}\theta_k^{\mathbf{x}_k}$	$\theta_1, \ldots, \theta_K$	$\mathbf{I}_{\mathcal{C}}(\theta_1, \ldots, \theta_K)$
Gamma	$\frac{\beta^{\alpha}\mathbf{x}^{\alpha-1}e^{-\beta\mathbf{x}}}{\Gamma(\alpha)}$	α, β	$\mathbf{I}_{\mathcal{G}}(\alpha, \beta)$
Von Mises-Fisher, for $\mathbb{S}^2$	$\frac{\kappa}{4\pi\sinh\kappa}\exp(\kappa\mu^{\intercal}\mathbf{x})$	κ, μ	$\mathbf{I}_{\mathcal{S}}(\kappa, \mu)$
Beta	$\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\mathbf{x}^{\alpha-1}(1-\mathbf{x})^{\beta-1}$	α, β	$\mathbf{I}_{\mathcal{B}}(\alpha, \beta)$

Table 1: List of distributions

With the notations of Table 1, the Fisher-Rao matrices of the the univariate Normal, Bernoulli and Categorical are:

$$\mathbf{I}_{\mathcal{N}}(\mu,\sigma^2)=\begin{pmatrix}\frac{1}{\sigma^2} & 0\\ 0 & \frac{1}{2\sigma^2}\end{pmatrix},\quad \mathbf{I}_{\mathcal{B}}(\theta)=\frac{1}{\theta(1-\theta)},\quad \mathbf{I}_{\mathcal{C}}(\theta_1,\dots,\theta_K)=\begin{pmatrix}1/\theta_1 & 0 & \dots & 0\\ 0 & 1/\theta_2 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & 1/\theta_K\end{pmatrix}$$

In addition, the Fisher-Rao matrices of the Gamma, Von Mises-Fisher and the Beta distributions are:

$$\begin{aligned}\mathbf{I}_{\mathcal{G}}(\alpha,\beta)&=\begin{pmatrix}\frac{\alpha}{\beta^2} & -\frac{1}{\beta}\\ -\frac{1}{\beta} & \Psi_1(\alpha)\end{pmatrix},\\ \mathbf{I}_{\mathcal{S}}(\kappa,\mu)&=\begin{pmatrix}\kappa K(\kappa)(\mathbb{1}-3\mu\mu^\top)+\kappa^2\mu\mu^\top & (\kappa K(\kappa)^2-\frac{2}{k}K(\kappa)+1)\mu\\ (\kappa K(\kappa)^2-\frac{2}{k}K(\kappa)+1)\mu^\top & 3K(\kappa)^2-\frac{2}{\kappa}K(\kappa)+1\end{pmatrix},\\ \mathbf{I}_{\mathcal{B}}(\alpha,\beta)&=\begin{pmatrix}\Psi_1(\alpha)-\Psi_1(\alpha+\beta) & -\Psi_1(\alpha+\beta)\\ -\Psi_1(\alpha+\beta) & \Psi_1(\beta)-\Psi_1(\alpha+\beta)\end{pmatrix},\end{aligned}$$

with $\Psi_1(\alpha)=\frac{\partial^2\ln\Gamma(\alpha)}{\partial\alpha}$ the trigamma function, and $K(\kappa)=\coth\kappa-\frac{1}{\kappa}$.

Proof. The univariate Normal, Bernoulli and Categorical have already been studied by Tomczak (2012), and the Beta distribution by Brigant and Puechmorel (2019). We will then focus our proof on the Gamma and the Von-Mises Fisher distributions.

In order to bypass unnecessary details, we will use the following notations, we redefine the Fisher-Rao as: $\mathbf{I}(\eta)=\mathbb{E}_x[g(\eta,x)g(\eta,x)^\intercal]$, with $g(\eta,x)=\nabla_\eta\ln p(x|\eta)$ the Fisher score. We call $G=g(\eta,x)g(\eta,x)^\top$, and G_{ij} the matrix elements.

Gamma distribution:

We have $p(x|\alpha,\beta)=\Gamma(\alpha)^{-1}\beta^\alpha x^{\alpha-1}e^{-\beta x}$, which leads to:

$$\begin{aligned}\ln p(x|\alpha,\beta)&=-\ln\Gamma(\alpha)+\alpha\ln\beta+(\alpha-1)\ln x-\beta x,\\ \frac{\partial\ln p}{\partial\alpha}&=-\Psi(\alpha)+\ln\beta+\ln x,\\ \frac{\partial\ln p}{\partial\beta}&=\frac{\alpha}{\beta}-x.\end{aligned}$$

Then:

$$\begin{aligned}G_{11}&=\left(\frac{\partial\ln p}{\partial\alpha}\right)^2=(\Psi_0(\alpha)+\ln\beta)^2+2(\Psi(\alpha)+\ln\beta)lnx+\ln^2 x,\\ G_{22}&=\left(\frac{\partial\ln p}{\partial\beta}\right)^2=\left(\frac{\alpha}{\beta}\right)^2-2\frac{\alpha}{\beta}x+x^2,\\ G_{12}=G_{21}&=\frac{\partial\ln p}{\partial\alpha}\cdot\frac{\partial\ln p}{\partial\beta}=(\Psi(\alpha)+\ln\beta)\left(\frac{\alpha}{\beta}-x\right)+\frac{\alpha}{\beta}\ln x-x\ln x.\end{aligned}$$

We know that $\mathbb{E}[x]=\frac{\alpha}{\beta}$. We can compute, using your favorite symbolic computation software, the following moments:

$$\begin{aligned}\mathbb{E}[\ln x]&=-\ln\beta+\Psi(\alpha)\\ \mathbb{E}[x\ln x]&=\frac{\alpha}{\beta}(\Psi(\alpha+1)-\ln\beta)\\ \mathbb{E}[\ln^2 x]&=(\ln\beta-\Psi(\alpha))^2+\Psi_1(\alpha)\end{aligned}$$

Replacing the moments for the following equations: $\mathbb{E}[G_{11}]$, $\mathbb{E}[G_{22}]$ and $\mathbb{E}[G_{12}]$ will finally give the Fisher-Rao matrix.

Von Mises Fisher distribution, for $\mathbb{S}^2$:

We have $p(\mathbf{x}|\mu,\kappa) = C_3(\kappa)\exp(\kappa\mu^\intercal\mathbf{x})$, with $C_3(\kappa) = \kappa(4\pi\sinh\kappa)^{-1}$. Here, μ is a 3-dimensional vector with $\|\mu\| = 1$.

$$\begin{aligned}\ln p(\mathbf{x}|\mu,\kappa) &= \ln\kappa - \ln 4\pi - \ln\sinh(\kappa) + \kappa\mu^\intercal\mathbf{x}\\ \nabla_\mu \ln p &= \kappa\mathbf{x}\\ \frac{\partial \ln p}{\partial\kappa} &= \kappa^{-1} - \coth(\kappa) + \mu^\intercal\mathbf{x}.\end{aligned}$$

Here, the Fisher-Rao matrix $\mathbf{I}_\mathcal{S}$ will be composed of block matrices, such that: $\mathbf{I}_\mathcal{S} = \mathbb{E}[G]$, with G_{11} a 3×3-matrix, G_{22} a scalar, and $G_{12} = G_{21}^\top$ a 3-dimensional vector.

$$\begin{aligned}G_{11} &= \nabla_\mu \ln p \nabla_\mu \ln p^\top = \kappa^2\mathbf{x}\mathbf{x}^\intercal\\ G_{22} &= \left(\frac{\partial \ln p}{\partial\kappa}\right)^2 = K(\kappa)^2 + 2K(\kappa)\mu^\top\mathbf{x} + (\mu^\intercal\mathbf{x})^2\\ G_{12} = G_{21}^\top &= \frac{\partial \ln p}{\partial\kappa}\cdot\nabla_\mu \ln p = (K(\kappa) + \mu^\intercal\mathbf{x})\,\kappa\mathbf{x},\end{aligned}$$

with $K(\kappa) = \coth(\kappa) - \frac{1}{\kappa}$.

We know from Hillen et al. (2016) that the mean and variance of the Von Mises Fisher distribution in the 3-dimensional case is: $\mathbb{E}[\mathbf{x}] = K(\kappa)\mu$ and $\mathrm{Var}[\mathbf{x}] = \frac{1}{\kappa}K(\kappa)\mathbb{1} + (1 - \frac{\coth(\kappa)}{\kappa} + \frac{2}{\kappa^2} - \coth^2(\kappa))\mu\mu^\top$. We can then deduce the following meaningful moments:

$$\begin{aligned}\mathbb{E}[\mathbf{x}\mathbf{x}^\top] &= \mathrm{Var}[\mathbf{x}] + \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^\top = \left(1 - \frac{3}{\kappa}K(\kappa)\right)\mu\mu^\top + \frac{1}{\kappa}K(\kappa)\mathbb{1},\\ \mathbb{E}[\mu^\intercal\mathbf{x}] &= \mu^\top\mathbb{E}[\mathbf{x}] = K(\kappa)\mu^\top\mu = K(\kappa)\\ \mathbb{E}[(\mu^\intercal\mathbf{x})^2] &= \mu^\top\mathrm{Var}[\mathbf{x}]\mu + \mathbb{E}[\mu^\intercal\mathbf{x}]^2 = 1 - \frac{2}{\kappa}K(\kappa),\\ \mathbb{E}[\mu^\intercal\mathbf{x}\mathbf{x}] &= \mathbb{E}[\mu\mathbf{x}\mathbf{x}^\intercal] = \mathbb{E}[\mathbf{x}\mathbf{x}^\intercal]\mu = \left(1 - \frac{3}{\kappa}K(\kappa)\right)\mu + \frac{1}{\kappa}K(\kappa)\mu.\end{aligned}$$

Replacing those moments in the following expressions: $\mathbb{E}[G_{11}]$, $\mathbb{E}[G_{22}]$, $\mathbb{E}[G_{12}]$ directly gives the Fisher-Rao metric. □

B Curve energy approximation for categorical data

In this section we present the details of the example in Section 3.3. In particular, we the steps to derive an approximation to the energy of a latent curve in closed form, which is suitable for applying automatic differentiation. This is particularly useful for our setting, since it allows us to consider our framework as a Black Box Random Geometry processing toolbox.

Let a random variable $\mathbf{x}\in\mathbb{R}^D$ that follows a generalized Bernoulli likelihood $p(\mathbf{x}|\eta)$, so the vector $\mathbf{x}\in\mathbb{R}^D$ is of the form $\mathbf{x} = (0,\cdots,1,\cdots,0)$ with $\sum_i x_i = 1$. The parameters $\eta\in\mathbb{R}^D$ are given as $\eta = h(\mathbf{z})$, with $\eta_i \geq 0\ \forall i$ and $\sum_i \eta_i = 1$ so we know that the parameters lie on the unit simplex. Actually, they represent the probability the corresponding dimension to be 1 on a random draw. Also, the $p(\mathbf{x}|\mathbf{z}) = \eta_1^{[x_1]}\cdots\eta_D^{[x_D]}$, where $[x_i] = 1$ if $x_i = 1$ else $[x_i] = 0$ which can be seen as an indicator function. The $\log p(\mathbf{x}|\eta) = \sum_i [x_i]\log(\eta_i)$ and $\nabla_\eta \log p(\mathbf{x}|\eta) = \left(\frac{[x_1]}{\eta_1},\ldots,\frac{[x_D]}{\eta_D}\right)$. Due to the outer product we have to compute the following expectations

$$\mathbb{E}_\mathbf{x}\left[\frac{[x_i]}{\eta_i}\frac{[x_j]}{\eta_j}\right] = 0, \quad \text{if} \quad i\neq j, \tag{14}$$

$$\mathbb{E}_\mathbf{x}\left[\left(\frac{[x_i]}{\eta_i}\right)^2\right] = \frac{1}{\eta_i}, \quad \text{if} \quad i = j, \tag{15}$$

because the $[x_i]$ and $[x_j]$ cannot be 1 on the same time, while the $\mathbb{E}_{\mathbf{x}}[[x_i]^2] = \eta_i$ as it shows the number of times $x_i = 1$. So the Fisher-Rao metric of $\mathcal{H}$ is equal to $\mathbf{I}_{\mathcal{H}}(\eta) = \text{diag}\,(1/\eta_1, \dots, 1/\eta_D)$. Note that the shortest paths between two distributions must be on the unit simplex in $\mathcal{H}$, while on the same time respecting the geometry of the Fisher-Rao metric.

We can easily parametrize the unit simplex by $[\eta_1, \dots, \eta_{D-1}, \widetilde{\eta}_D]$ with

$$\widetilde{\eta}_D(\eta_1, \dots, \eta_{D-1}) = 1 - \sum_{i=1}^{D-1} \eta_i. \tag{16}$$

This allows to pullback the Fisher-Rao metric in the latent space $[\eta_1, \dots, \eta_{D-1}]$ as we have described in this paper. Intuitively, the $\mathbf{z} = [\eta_1, \dots, \eta_{D-1}]$ and the function h is the parametrization of the simplex. Hence, we are able to compute the shortest path using the induced metric.

However, there is a simpler way to compute this path. We know that the element-wise square root of the parameters η gives a point on the positive orthonant of the unit sphere as $y_i = \sqrt{\eta_i} \;\Rightarrow\; \sum_i y_i^2 = \sum_i \sqrt{\eta_i}^2 = 1$. We also know that the shortest path on a sphere is the great-circle. Therefore, the distance between two distributions parametrized by η and η' on the unit simplex in $\mathcal{H}$, can be equivalently measured using the great-circle distance between their square roots as

$$\text{dist}(\eta, \eta') = \arccos \sqrt{\eta}^\intercal \sqrt{\eta'}. \tag{17}$$

In this way, we can approximate the energy of a curve $c(t)$ in the latent space as follows

$$\begin{aligned} \text{Energy}[c] &\approx \sum_{n=1}^{N-1} \text{dist}^2(h(c(n/N)), h(c(n+1/N))) = \sum_{n=1}^{N-1} \arccos^2 \sqrt{h(c(n/N))}^\intercal \sqrt{h(c(n+1/N))} \\ &= \sum_{n=1}^{N-1} \left(2 - 2\sqrt{h(c(n/N))}^\intercal \sqrt{h(c(n+1/N))}\right), \end{aligned} \tag{18}$$

where we used at the last step the small angle approximation $\cos\theta \approx 1 - \frac{\theta^2}{2} \Leftrightarrow \theta^2 \approx 2 - 2\cos\theta$. Note that this formulation is suitable for our proposed method to compute shortest paths (see Section 3.2).

The derivation above represents the conceptual strategy, while in general we proposed to use the KL divergence approximation result (8) in place of the great-circle distance. Intuitively, when the KL divergence has an analytic solution, we can derive an analogous energy approximation. Even if the solution of the KL is intractable, we can still use our approach as long as we can estimate the KL using Monte Carlo and propagate the gradient through the samples using a re-parametrization scheme or a score function estimator.

C Information geometry in generative modeling

In this section we present the additional technical information related to the pullback Fisher-Rao metric in the latent space of a VAE.

C.1 Details for the pullback metric in the latent space

We call h the non linear function, typically parametrized as deep neural networks, that maps the variables from the latent space $\mathcal{Z}$ to the parameter space $\mathcal{H}$, such that: $h(\mathbf{z}) = \eta$, with $\mathbf{z} \in \mathcal{Z}$ and $\eta \in \mathcal{H}$. Furthermore, the data $\mathbf{x} \in \mathcal{X}$ is reconstructed such that it follows a specific distribution: $\mathbf{x} \sim p(\mathbf{x}|\eta)$, with $p(\mathbf{x}|\eta)$ being for instance a Bernoulli or Gaussian distribution. The parameter space $\mathcal{H}$ is a statistical manifold equipped with Fisher-Rao metric: $\mathbf{I}_{\mathcal{H}}(\eta) \triangleq \int p(\mathbf{x}|\eta) \left[\nabla_\eta \log p(\mathbf{x}|\eta) \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal\right] \mathrm{d}\mathbf{x}$. We denote by $\mathbf{J}_h$ the Jacobian of h.

Proposition C.1. *The latent space $\mathcal{Z}$ is equipped with the Riemannian pullback metric tensor:*

$$\mathbf{M}(\mathbf{z}) \triangleq \mathbf{J}_h(\mathbf{z})^\intercal \mathbf{I}_{\mathcal{H}}(h(\mathbf{z})) \mathbf{J}_h(\mathbf{z}).$$

Proof. The parameter space is a statistical manifold equipped with the Fisher-Rao metric $\mathbf{I}_{\mathcal{H}}(\eta)$, thus the scalar product at η between two vectors $d\eta_1, d\eta_2 \in \mathcal{H}$ is: $\langle d\eta_1, d\eta_2 \rangle_{\mathbf{I}_{\mathcal{H}}(\eta)} = d\eta_1^\intercal \mathbf{I}_{\mathcal{H}}(\eta) d\eta_2$. For two vectors $d\mathbf{z}_1, d\mathbf{z}_2 \in \mathcal{Z}$, we have at $\eta = f(\mathbf{z})$ that: $\langle d\eta_1, d\eta_2 \rangle_{\mathbf{I}_{\mathcal{H}}(\eta)} = \langle \mathbf{J}_h(\mathbf{z}) d\mathbf{z}_1, \mathbf{J}_h(\mathbf{z}) d\mathbf{z}_2 \rangle_{\mathbf{I}_{\mathcal{H}}(\eta)} = d\mathbf{z}_1^\intercal (\mathbf{J}_h(\mathbf{z})^\intercal \mathbf{I}_{\mathcal{H}}(h(\mathbf{z})) \mathbf{J}_h(\mathbf{z})) d\mathbf{z}_2$.

$\mathbf{I}_{\mathcal{H}}(h(\mathbf{z}))$ is a Riemannian metric tensor by definition, and it is then positive definite. Furthermore, $h : \mathcal{Z} \to \mathcal{H}$ is a smooth immersion, and so $\mathbf{J}_h(\mathbf{z})$ is full-rank. It follows that $\mathbf{J}_h(\mathbf{z})^\intercal \mathbf{I}_{\mathcal{H}}(h(\mathbf{z}))\mathbf{J}_h(\mathbf{z})$ is positive definite. Hence $\mathbf{M}(\mathbf{z})$ is a Riemannian metric tensor. □

Proposition C.2. *Our pullback metric* $\mathbf{M}(\mathbf{z})$ *is actually equal to the Fisher-Rao metric obtained over the parameter space* $\mathcal{Z}$*:*

$$\mathbf{M}(\mathbf{z}) = \mathbf{I}_{\mathcal{Z}}(\mathbf{z}) \triangleq \int p(\mathbf{x}|\mathbf{z}) \left[\nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z}) \nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z})^\intercal\right] \mathrm{d}\mathbf{x}$$

Proof. We will show that $\mathbf{I}_{\mathcal{Z}}(\mathbf{z}) = \mathbf{J}_f(\mathbf{z})^\intercal \mathbf{I}_{\mathcal{H}}(\eta)\mathbf{J}_f(\mathbf{z})$. Let's consider the definition of the Fisher-Rao metric in $\mathcal{Z}$:

$$\begin{aligned}
\mathbf{I}_{\mathcal{Z}}(\mathbf{z}) &= \int \nabla_{\mathbf{z}} \log p(\mathbf{x} \mid \mathbf{z}) \cdot \nabla_{\mathbf{z}} \log p(\mathbf{x} \mid \mathbf{z})^\intercal p(\mathbf{x} \mid \mathbf{z}) \mathrm{d}\mathbf{x} && (19)\\
&= \int \mathbf{J}_f(\mathbf{z})^\intercal \nabla_\eta \log p(\mathbf{x}|\eta) \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal \mathbf{J}_f(\mathbf{z}) p(\mathbf{x}|\eta) \mathrm{d}\mathbf{x} && (20)\\
&= \mathbf{J}_f(\mathbf{z})^\intercal \left[\int_{\mathcal{X}} \nabla_\eta \log p(\mathbf{x}|\eta) \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal p(\mathbf{x}|\eta) \mathrm{d}\mathbf{x}\right] \mathbf{J}_f(\mathbf{z}) && (21)\\
&= \mathbf{J}_f(\mathbf{z})^\intercal \mathbf{I}_{\mathcal{H}}(f(\mathbf{z})) \mathbf{J}_f(\mathbf{z}) = \mathbf{M}(\mathbf{z})
\end{aligned}$$

where we use the fact that $\eta = f(\mathbf{z})$ so the $\nabla_{\mathbf{z}} \log p(\mathbf{x}|f(\mathbf{z})) = \mathbf{J}_f(\mathbf{z})^\intercal \cdot \nabla_\eta \log p(\mathbf{x}|\eta)$

The same argument can be proved as follows:

$$\begin{aligned}
\langle d\eta, \mathbf{I}_{\mathcal{H}}(\eta) d\eta \rangle &= \langle \mathbf{J}_f(\mathbf{z}) d\mathbf{z}, \mathbf{I}_{\mathcal{H}}(f(\mathbf{z})) \mathbf{J}_f(\mathbf{z}) d\mathbf{z} \rangle && (22)\\
&= \langle \mathbf{J}_f(\mathbf{z}) d\mathbf{z}, \int \nabla_\eta \log p(\mathbf{x}|\eta) \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal p(\mathbf{x}|\eta) \mathrm{d}\mathbf{x}\ \mathbf{J}_f(\mathbf{z}) d\mathbf{z} \rangle && (23)\\
&= \langle d\mathbf{z}, \int \mathbf{J}_f(\mathbf{z})^\intercal \cdot \nabla_\eta \log p(\mathbf{x}|\eta)\ \nabla_\eta \log p(\mathbf{x}|\eta)^\intercal \cdot \mathbf{J}_f(\mathbf{z}) p(\mathbf{x}|\eta) \mathrm{d}\mathbf{x}\ d\mathbf{z} \rangle && (24)\\
&= \langle d\mathbf{z}, \int \nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z}) \nabla_{\mathbf{z}} \log p(\mathbf{x}|\mathbf{z})^\intercal p(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{x}\ d\mathbf{z} \rangle = \langle d\mathbf{z}, \mathbf{I}_{\mathcal{Z}}(\mathbf{z}) d\mathbf{z} \rangle && (25)
\end{aligned}$$

□

In section A.1, we have seen how to derive a close-form expression of the Fisher-Rao metric for a one-dimensional observation x that follows a specific distribution. In practice, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$ is a multi-dimensional variable where each dimension represents, for instance, a pixel when working with images or a feature when working with tabular data. Each feature, x_i with $i = 1 \cdots D$, is obtained for a specific set of parameters $\{\eta_i\}$. We assume that the features follow the same distribution $\mathcal{D}$, such that: $x_i \sim p(x_i|\eta_i)$, and $p(\mathbf{x}|\eta) = \prod_{i=1}^D p(x_i|\eta_i)$.

Proposition C.3. *If the features follow the same distribution* $\mathcal{D}$*, such that:* $x_i \sim p(x_i|\eta_i)$ *and* $p(\mathbf{x}|\eta) = \prod_{i=1}^D p(x_i|\eta_i)$*, then the Fisher-Rao metric* $\mathbf{I}_{\mathcal{H}}(\eta)$ *is a block matrix where the diagonal terms are the Fisher-Rao matrices* $\mathbf{I}_{\mathcal{H},i}$ *obtained for each data feature* x_i*:*

$$\mathbf{I}_{\mathcal{H}}(\eta) = \begin{pmatrix} \mathbf{I}_{\mathcal{H},1} & 0 & \dots & 0 \\ 0 & \mathbf{I}_{\mathcal{H},2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{I}_{\mathcal{H},D} \end{pmatrix}$$

Proof. We have $x_i \sim p(x_i|\eta_i)$ and $\mathbf{I}_{\mathcal{H},i} = \int p(x_i|\eta_i) \left[\nabla_{\eta_i} \log p(x_i|\eta_i) \nabla_{\eta_i} \log p(x_i|\eta_i)^\intercal\right] dx_i$. Also, we assumed: $p(\mathbf{x}|\eta) = \prod_{i=1}^D p(x_i|\eta_i)$. We then have: $\log p(\mathbf{x}|\eta) = \sum_{i=1}^D \log p(x_i|\eta_i)$, and the Fisher score: $\nabla_\eta \log p(\mathbf{x}|\eta) = \nabla_\eta \sum_{i=1}^D \log p(x_i|\eta_i) = [\nabla\eta_1 \ln p(x_1|\eta_1), \dots, \nabla\eta_D \ln p(x_1|\eta_D)]^\intercal$.

The matrix $\mathbf{I}_{\mathcal{H}}(\eta)$ is thus a $D \times D$ block matrix, where the (i, j)-block element is:

$$I_{ij} = \int p(x_i|\eta_i) \left[\nabla_{\eta_i} \log p(x_i|\eta_i) \nabla_{\eta_i} \log p(x_j|\eta_j)^\intercal\right] dx_i.$$

Let's note that:

$$\int p(x_i|\eta_i)\nabla_{\eta_i}\log p(x_i|\eta_i)dx_i = \int p(x_i|\eta_i)\frac{\nabla_{\eta_i}p(x_i|\eta_i)}{p(x_i|\eta_i)}dx_i = \nabla_{\eta_i}\int p(x_i|\eta_i)dx_i = 0.$$

When $i = j$, we have $\mathbf{I}_{ii} = \mathbf{I}_{\mathcal{H},i}$, with $\mathbf{I}_{\mathcal{H},i}$ being the Fisher-Rao metric obtained for: $x_i \sim p(x_i|\eta_i)$.
When $i \neq j$, we have: $\mathbf{I}_{ij} = \nabla\log p(x_j|\eta_j)^\intercal \int p(x_i|\eta_i)\nabla_{\eta_i}\log p(x_i|\eta_i)dx_i = 0$. □

Then, for example, if we are dealing with binary images, and make the assumption that each pixel x_i follows a Bernoulli distribution: $p(x_i|\eta_i) = \eta^{x_i}(1-\eta_i)^{1-x_i}$, then according to Section A.1 and Proposition C.3, the Fisher-Rao matrix that endows the parameter space $\mathcal{H}$ is:

$$\mathbf{I}_{\mathcal{H}}(\eta) = \begin{pmatrix} \frac{1}{\eta_1(1-\eta_1)} & 0 & \dots & 0 \\ 0 & \frac{1}{\eta_2(1-\eta_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\eta_D(1-\eta_D)} \end{pmatrix}.$$

We have seen that in theory, we can obtain a close form expression for the pullback metric, if the probability distribution is known. In practice, we can directly infer the metric using the approximation of the KL-divergence.

Proposition C.4. *We define perturbations vectors as: $\delta e_i = \varepsilon \cdot \mathbf{e_i}$, with $\varepsilon \in \mathbb{R}_+$ a small infinitesimal quantity, and ($\mathbf{e_i}$) a canonical basis vector in $\mathbb{R}^d$. For better clarity, we rename $\mathrm{KL}(p(\mathbf{x}|\mathbf{z}), p(\mathbf{x}|\mathbf{z}+\delta\mathbf{z})) = \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{z})$ and we note $\mathbf{M}_{ij}$ the components of $\mathbf{M}(\mathbf{z})$. We can then approximate by a system of equations the diagonal and non-diagonal elements of the metric:*

$$\begin{aligned} \mathbf{M}_{ii} &\approx 2\ \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i})/\varepsilon^2 \\ \mathbf{M}_{ij} = \mathbf{M}_{ji} &\approx \left(\mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i}+\delta\mathbf{e_j}) - \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i}) - \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_j})\right)/\varepsilon^2. \end{aligned}$$

Proof. From Proposition A.1, we know that:

$$\mathrm{KL}_{\mathbf{z}}(\delta\mathbf{z}) = \frac{1}{2}\delta\mathbf{z}^\top\mathbf{M}(\mathbf{z})\delta\mathbf{z} + o(\delta\mathbf{z}^2).$$

Let's take $\delta e_i = \varepsilon \cdot \mathbf{e_i}$. On one hand, we have: $\delta e_i^\top \mathbf{M}(\mathbf{z})\delta e_i = \varepsilon^2\mathbf{M}_{ii}$. On the second hand, we also have: $\delta e_i^\top \mathbf{M}(\mathbf{z})\delta e_i \approx 2\mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i})$, which gives us the equation to infer the diagonal elements of the metric.

Now, let's take $\delta e_i + \delta e_j = \varepsilon\cdot(\mathbf{e_i}+\mathbf{e_j})$. Then, we have: $(\delta e_i+\delta e_j)^\top\mathbf{M}(\mathbf{z})(\delta e_i+\delta e_j) = \varepsilon^2(\mathbf{M}_{ii}+\mathbf{M}_{jj}+\mathbf{M}_{ij}+\mathbf{M}_{ji})$. We also know that $\mathbf{M}_{ji} = \mathbf{M}_{ij}$. Again, we also have: $(\delta e_i+\delta e_j)^\top\mathbf{M}(\mathbf{z})(\delta e_i+\delta e_j) \approx 2\mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i}+\delta\mathbf{e_j})$.

We can replace the terms $\mathbf{M}_{ii}$ and $\mathbf{M}_{jj}$ in the equation obtained above with the KL-divergence for the diagonal terms. Which finally gives us: $\mathbf{M}_{ij} = \mathbf{M}_{ji} \approx \left(\mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i}+\delta\mathbf{e_j}) - \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_i}) - \mathrm{KL}_{\mathbf{z}}(\delta\mathbf{e_j})\right)/\varepsilon^2$.

□

C.2 Uncertainty quantification and regularization

As discussed in the main text, we carefully design our mappings from latent space to parameter space such that they model the training codes according to the learned decoders, and extrapolate to uncertainty outside the support of the data. This, we refer to as **uncertainty regularization**. In this section we explain it in detail. The core idea of this uncertainty regularization is imposing a "slider" that forces the distribution $p(\mathbf{x}|\mathbf{z})$ to change when $\mathbf{z}$ is far from the training latent codes. For this, we use a combination of KMeans and the sigmoid activation function.

We start by encoding our training data, arriving at a set of latent codes $\{\mathbf{z}_n\}_{n=1}^N \subseteq \mathcal{Z}$. We then train KMeans($k$) on these latent codes (where k is a hyperparameter that we tweak manually), arriving at k cluster centers $\{\mathbf{c}_j\}_{j=1}^k$. These cluster centers serve as a proxy for "closeness" to the data: we know that a latent code $\mathbf{z} \in \mathcal{Z}$ is near the support if $D(\mathbf{z}) := \min_j \left\{\|\mathbf{z}-\mathbf{c}_j\|^2\right\}$ is close to 0.

Distribution	$h\colon \mathcal{Z}_{\text{toy}} \to \mathcal{H}$	Extrapolation mechanism
Normal	$\mu(\mathbf{z}) = 10 \cdot f_3(\mathbf{z}), \quad \sigma(\mathbf{z}) = 10 \cdot \text{Softplus}(f_3(\mathbf{z}))$	$\sigma(\mathbf{z}) \to \infty$
Bernoulli	$p(\mathbf{z}) = \text{Sigmoid}(f_{15}(\mathbf{z}))$	$p(\mathbf{z}) = 1/2$
Beta	$\alpha(\mathbf{z}) = 10 \cdot \text{Softplus}(f_3(\mathbf{z})), \quad \beta(\mathbf{z}) = 10 \cdot \text{Softplus}(f_3(\mathbf{z}))$	$(\alpha(\mathbf{z}), \beta(\mathbf{z})) = (1, 1)$
Dirichlet	$\alpha(\mathbf{z}) = \text{Softplus}(f_3(\mathbf{z}))$	$\alpha(\mathbf{z}) = 1$
Exponential	$\lambda(\mathbf{z}) = \text{Softplus}(f_3(\mathbf{z}))$	$\lambda(\mathbf{z}) \to 0$

Table 2: This table shows the implementations of the decode and extrapolate functions in Eq. (27) for all the distributions studied in our second experiment (see Sec. 4.2). Here we represent a randomly initialized neural network with f_i, where i represents the size of the co-domain. For example, in the case of the Dirichlet distribution, we use a randomly initialized neural network to compute the parameters α of the distribution and, since these have to be positive, we pass the output of this network through a Softplus activation; moreover, since the Dirichlet distribution is approximately uniform when all its parameters equal 1, our extrapolation mechanism consists of replacing the output of the network with a constant vector of ones.

The next step in our regularization process is to reweight our decoded distributions such that we decode to high uncertainty when $D(\mathbf{z})$ is large, and we decode to our learned distributions when $D(\mathbf{z}) \approx 0$. This mapping from $[0, \infty) \to (0, 1)$ can be constructed using a modified sigmoid function Detlefsen et al. (2020, 2019), consider indeed

$$\tilde{\sigma}_\beta(d) = \text{Sigmoid}\left(\frac{d - c \cdot \text{Softplus}(\beta)}{\text{Softplus}(\beta)}\right), \tag{26}$$

where $\beta \in \mathbb{R}$ is another hyperparameter that we manually tweak, and $c \approx 7$.

With this translated sigmoid, we have that $\tilde{\sigma}_\beta(D(\mathbf{z}))$ is close to 0 when $\mathbf{z}$ is close to the support of the data (i.e. close to the cluster centers), and it converges to 1 when $D(\mathbf{z}) \to \infty$. $\tilde{\sigma}_\beta(D(\mathbf{z}))$ serves, then, as a slider that indicates closeness to the training codes. This reweighting takes the following form:

$$\text{reweight}(\mathbf{z}) = (1 - \tilde{\sigma}_\beta(D(\mathbf{z})))h(\mathbf{z}) + \tilde{\sigma}_\beta(D(\mathbf{z}))\,\text{extrapolate}(\mathbf{z}), \tag{27}$$

where $h(\mathbf{z}) = \eta \in \mathcal{H}$ represents our learned networks in parameter space, and $\text{extrapolate}(\mathbf{z})$ returns the parameters of the distribution that maximize uncertainty (e.g. $\sigma \to \infty$ in the case of an isotropic Gaussian, $p \to 1/2$ in the case of a Bernoulli, and $\kappa \to 0$ in the case of the von Mises-Fisher).

For the particular case of the experiment in which we pull back the Fisher-Rao metric from the parameter space of several distributions (see 4.2), Table 2 provides the exact extrapolation mechanisms and implementations of $h(\mathbf{z})$.

D Details for our implementation and experiments

In this section we present the technical details that we used in our implementation and experiments. We are currently implementing an open-source version of our code here.

D.1 What we mean when we say black-box random geometry

Before we dive into the specific details of our experiments, it is worth noting that they were all made using the same *interface*. This is precisely what we mean when we say that our results open the doors for black-box random geometry: We can define a `curve_energy` method that is agnostic to the distribution our models decode to.

To hammer this point home, consider the following interface, written in Python:

```python
class StatisticalManifold:
    def __init__(self, model: torch.nn.Module):
        # A model with regularized uncertainty (see Uncertainty Quantification)
        self.model = model
        assert "decode" in dir(model)
```

Pulling back the Euclidean vs. Fisher-Rao (Sec. 4.1)

Module	MLP
	Encoder
μ	Linear(728, 2)
	Decoder
μ	Linear(2, 728)
σ_{UR}	RBF(), PosLinear(500, 1), Reciprocal(), PosLinear(1, 728)
$\sigma_{\text{no UR}}$	Linear(2, 728), Softplus()
Optimizer	Adam ($\alpha = 1 \times 10^{-5}$)
Batch size	32

Table 3: This table shows the Variational Autoencoder used in our first experiment (see Sec. 4.1). The network for approximating the standard deviation σ leverages ideas from Arvanitidis et al. (2018), in which an RBF network is trained on latent codes using centers positioned through KMeans. The operation PosLinear(a, b) represents the usual Linear transformation with a inputs and b outputs, but considering only positive weights. To compare between having and not having uncertainty regularization, we use two different approximations of the standard deviation in the decoder: σ_{UR} when performing meaningful uncertainty quantification, and $\sigma_{\text{no UR}}$ otherwise.

```
6
7       def curve_energy(self, curve: CubicSpline) -> torch.Tensor:
8           # An energy function that can be minimized using autodifferentiation.
9
10          dt = (curve[1] - curve[0])
11          dist1 = self.model.decode(curve[:-1])
12          dist2 = self.model.decode(curve[1:])
13          kl = kl_divergence(dist1, dist2)
14          energy = kl.sum() * (2 * dt ** -1)
15
16          return energy
```

Notice that the user need only provide a `model` that implements a `decode` function which is expected to return a distribution with proper uncertainty estimates (as described in Sec. C.2). Line 14 is a direct implementation of our derived expression for the energy (see Prop. A.2). Most distributions of interest are available in the Torch submodule `torch.distributions`, and similar implementations could be done for other frameworks.

D.2 Shortest path approximation with cubic splines

As we described in the main paper, we use an approximate solution for the shortest paths based on cubic splines. Let a cubic spline $c_{\boldsymbol{\psi}}(t) = [1, t, t^2, t^3]^\intercal [\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3]$ with parameters $\boldsymbol{\psi}_i \in \mathbb{R}^{d\times 1}$. Also, in our implementation the actual curve is a piecewise cubic spline and we optimize the K control points $\mathbf{c}_k$ as well. We optimize the parameters using the approximation of the curve energy $\{\boldsymbol{\psi}_k^*, \mathbf{c}_k^*\}_{k=1}^K = \text{argmin}_{\boldsymbol{\psi}} \text{Energy}[c_{\boldsymbol{\psi}}]$. In general, we can use Prop. A.2 as long as we can propagate the gradient through the KL or as in (18) if an explicit closed form solution exists. In this case, we are able to use automatic differentiation for the optimization of the parameters (as discussed in Sec. D.1).

In practical terms, we compute these shortest paths by creating a uniform grid in latent space and computing, only once, the curve energy for the edges of this grid. After this expensive computation (which only needs to be performed once) we can use shortest-paths algorithms in graphs to create a suitable initialization of the geodesic. We fit a cubic spline to this initialization and then optimize its parameters further.

D.3 Models used

In this section we describe, in detail, the models that we used for our experiments (see Sec.4). All the networks that we used are Multi-Layer Perceptrons implemented in PyTorch.

First, Table 3 shows the Variational Autoencoder implemented for the experiment described in Sec. 4.1. In the

Toy latent spaces (Sec. 4.2)

Distribution	Module	MLP	Seed for randomness	β in $\tilde{\sigma}_\beta$
Normal	μ	Linear(2,3)	1	-2.5
	σ	Linear(2,3), Softplus()		
Bernoulli	p	Linear(2,15), Sigmoid()	1	-3.5
Beta	α	Linear(2,3), Softplus()	1	-4.0
	β	Linear(2,3), Softplus()		
Dirichlet	α	Linear(2,3), Softplus()	17	-4.0
Exponential	λ	Linear(2,3), Softplus()	17	-4.0

Table 4: This table describes the neural networks used for the experiment presented in Sec. 4.2. Following the notation of PyTorch, Linear(a, b) represents an MLP layer with a input nodes and b output nodes. In each of these networks, we implement the reweighting operation described in Sec. C.2, and we describe the β hyperparameter present in the modified sigmoid function (Eq. (26)). This networks were not trained in any way, and they were initialized using the provided seed.

Decoding to a von Mises-Fisher Distribution (Sec 4.3, 4.4, 4.5)

Module	MLP
	Encoder (Normal dist.)
μ	Linear(3×26, 90), Linear(90, 2)
σ	Linear(3×26, 90), Linear(90, 2), Softplus()
	Decoder (vMF dist.)
μ	Linear(2, 90), Linear(90, 3×26), Linear(3×26, 3×26)
κ	Linear(2, 90), Linear(90, 3×26), Linear(3×26, 26), Softplus()
Optimizer	Adam ($\alpha = 1 \times 10^{-3}$)
Batch size	16
β in $\tilde{\sigma}_\beta$	-5.5
KL annealing	0.01
Extrapolation mechanism	$\kappa \to 0.1$

Table 5: This table shows the Variational Autoencoder used in our last two experiments (see Sec. 4.3, 4.4). Our motion capture data tracked 26 different bones, and thus we decode to a product of 26 different von Mises-Fisher distributions.

Decoding to a Bernoulli Distribution (Sec 4.6)	
Module	**MLP**
	Encoder (Normal dist.)
μ	Linear(60, 16), Tanh(), Linear(16, 16), Tanh(), Linear(16, 2)
σ	Linear(60, 16), Tanh(), Linear(16, 16), Tanh(), Linear(16, 2), Softplus()
	Decoder (Bernoulli dist.)
p	Linear(2, 16), Tanh(), Linear(16, 16), Tanh(), Linear(16, 60), Sigmoid()
Optimizer	Adam ($\alpha = 1 \times 10^{-3}$, $\omega = 1 \times 10^{-7}$)
Batch size	256
β in $\tilde{\sigma}_\beta$	-3.0
KL annealing	0.01
Extrapolation mechanism	$p \to 1/2$

Table 6: This table shows the Variational Autoencoder used in the movie rating experiement (see Sec. 4.6). The MovieLens 25M dataset has been preprocessed such that it is composed of 10000 users rating if they have seen some of 60 selected movies. We only select users that have seen more than two movies and less than 30 movies, to avoid outliers and aim for a more realistic scenario. We used the same extrapolation mechanism described in the toy experiments for the Bernoulli: having the probits be 1/2 (see Sec. C.2).

computations *without* uncertainty regularization, we used a simpler model for the uncertainty quantification (namely, a single Linear layer, followed by a Softplus activation). For our second experiment involving a toy latent space, we also provide the implementation of the respective MLPs in Table 4. Finally, Table 5 and 6 respectively represents the VAE trained for the experiments related to motion capture (Sec. 4.3) and movie rating (Sec. 4.6). For the motion capture experiments, we are training a VAE that decodes to a von Mises Fisher distribution, and for the movie rating experiments, we decode to a Bernoulli distribution.

All of these VAEs were trained by maximizing the Evidence Lower Bound with different values for KL annealing which can be read from the different tables. For example, Table 5 shows that the KL annealing constant was chosen to be 0.01.

D.4 Metric approximation and KL by sampling

When visualising our latent space as a statistical manifold, we can obtain a direct approximation of the metric using the KL-divergence between two close distributions (Proposition 3.4). We will show here, in simple cases, how our metric approximation compares to close-form expressions.

In the following experiment, our statistical manifold is the parameter space of known distributions (Beta and Normal). Their Fisher-Rao matrices are well-known (Sec. A.1), and we approximate them by computing the KL-divergence of sampled distributions. We call $\mathbf{M}_t$ the theoretical metric and $\mathbf{M}_a$ the approximated metric, and we note $\varepsilon_r = \frac{\|\mathbf{M}_t - \mathbf{M}_a\|}{\|\mathbf{M}_t\|}$ the relative error between the theoretical and approximated matrices. Here, $\|\cdot\|$ denotes the Frobenius norm. For the Normal distribution, we empirically obtain: $\varepsilon_r = 5.32 \cdot 10^{-4} \pm 9.63 \cdot 10^{-4}$, and for the Beta distribution, we have: $\varepsilon_r = 1.73 \cdot 10^{-5} \pm 1.17 \cdot 10^{-5}$.

D.5 Computational complexity

Proposition A.1 shows the system of equations required to approximate the pullback metric in the latent space. Each KL operation requires 2 forward passes from the decoder to compute, so first we establish the lower bound on the time complexity of the decoder forward pass. Ignoring all activation function related operations, for an MLP with H hidden layers, N-dimensional network output, K-dimensional hidden layer output and single M-dimensional vector input, this lower bound is:

$$\Omega\left(MK_1 + K_H N + \sum_{i=1}^{H-1} M_i M_{i+1}\right) \tag{28}$$

For each diagonal element $\mathbf{M}_{ii}$ of the metric tensor we need to compute a single KL divergence, which will require two forward passes through the decoder network giving us a (lower bounded) time complexity of $\Omega\left[2\left(MK_1 + K_H N + \sum_{i=1}^{H-1} M_i M_{i+1}\right)\right]$ for each element. For the off-diagonal elements we will need to compute the KL three times which corresponds to six forward passes through the decoder network. which yields a (lower bounded) time complexity $\Omega\left[6\left(MK_1 + K_H N + \sum_{i=1}^{H-1} M_i M_{i+1}\right)\right]$ per element.

D.6 Information for the movie preferences experiment

For this experiment we used the MovieLens 25M dataset (https://grouplens.org/datasets/movielens/25m/). Each cell of the data matrix represents the rating of a user (row) from 1 to 5 for the corresponding movie (column). In order to fit a Bernouli VAE we considered the matrix as binary i.e. if a user has seen a movie (1) or not (0). We then selected the 60 most popular movies, as well as, 10000 users who have seen between 2 and 30 of these movies. We also verified that all the movies have been seen from at least 600 users. In this way we reduced the size of the dataset, obtaining a realistic scenario where: 1) some movies are more popular than the others, and 2) we do not include users that have seen 0 or almost all the movies. We show in Fig. 9 the number of views for each movie and the number of movies each user has seen. In Table 6 we present the details for the Bernouli VAE.

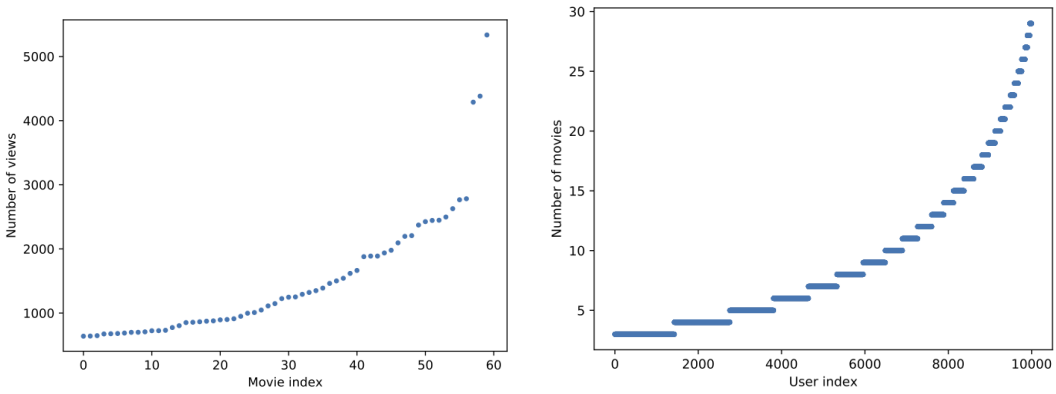

Figure 9: The numbers of views for the movies and the users.

D.7 Information for fitting the LAND model

The locally adaptive normal distribution (LAND) (Arvanitidis et al., 2016) is the extension of the normal distribution on Riemannian manifolds learned from data. Pennec (2006) first derived this distribution on predefined manifolds as the sphere and also showed that it is the maximum entropy distribution given a mean and a precision matrix. The flexibility of this probability density relies on the shortest paths. However, the computational demand to fit this model is relatively high, especially in our case, since we need to use an approximation scheme to find the shortest paths.

In particular, we compute the *logarithmic map* $\mathbf{v} = \text{Log}_{\mathbf{x}}(\mathbf{y})$ by first finding the shortest path between $\mathbf{x}$ and $\mathbf{y}$, and then, rescaling the initial velocity as $\mathbf{v} = \frac{\dot{c}(0)}{||\dot{c}(0)||}\text{Length}(c)$, which ensures that $||\mathbf{v}|| = \text{Length}(c)$. In addition, for the estimation of the normalization constant we use the *exponential map* $\text{Exp}_{\mathbf{x}}(\mathbf{v}) = c_{\mathbf{v}}(t)$, which is the inverse operator that generates the shortest path with $c(1) = \mathbf{y}$ taking the rescaled initial velocity $\mathbf{v}$ as input. Also, we should be able to evaluate the metric. While the logarithmic map can be approximated using our approach (Section D.2), for the exponential map we need to solve the ODEs system (13) as an initial value problem (IVP). Note that we fit the LAND using gradient descent, which implies that the computation of these operators is the main computational bottleneck.

We provided a method in Proposition 3.4, which enables us to approximate the pullback metric in the latent space of a generative model using the corresponding KL divergence. Even if this is a sensible approach, in practice, the computational cost is relatively high as we might need to estimate the KL using Monte Carlo. For example, this is the case when the likelihood is the von Mises-Fisher. This further implies that fitting the LAND using this approach is prohibited due to the computational cost. Especially, since we need to evaluate many times the metric and its derivative for the computation of each exponential map. Hence, in order to fit the LAND efficiently, we used the following approximation based on Hauberg et al. (2012).

First we construct a uniformly spaced grid in the latent space. Then, we evaluate the metric using Proposition 3.4 for each point on the grid getting a set $\{\mathbf{z}_s, \mathbf{M}_s\}_{s=1}^S$ of metric tensors. Thus, we can estimate the metric at any point $\mathbf{z}$ as

$$\mathbf{M}(\mathbf{z}) = \sum_{s=1}^{S} \widetilde{w}_s(\mathbf{z})\mathbf{M}_s, \quad \text{with} \quad \widetilde{w}_s(\mathbf{z}) = \frac{w_s(\mathbf{z})}{\sum_{j=1}^{S} w_s(\mathbf{z})} \quad \text{and} \quad w_s(\mathbf{z}) = \exp\left(-\frac{||\mathbf{z}_s - \mathbf{z}||^2}{2\sigma^2}\right) \tag{29}$$

where $\sigma > 0$ the bandwidth parameter. This is by definition a Riemannian metric as a weighted sum of Riemannian metrics with a smooth weighting function. In this way, we can approximate the pullback of the Fisher-Rao metric in the latent space $\mathcal{Z}$ in order to perform the necessary computations more efficiently.