Geometrically Enriched Latent Spaces

Arvanitidis, Georgios; Hauberg, Søren; Schoelkopf, Bernhard

Published in:
Proceedings of the 24th International Conference on Artificial Intelligence and Statistics

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Geometrically Enriched Latent Spaces

Georgios Arvanitidis
MPI for Intelligent Systems, Tübingen

Søren Hauberg
DTU Compute, Lyngby

Bernhard Schölkopf
MPI for Intelligent Systems, Tübingen

Abstract

A common assumption in generative models is that the generator immerses the latent space into a Euclidean ambient space. Instead, we consider the ambient space to be a Riemannian manifold, which allows for encoding domain knowledge through the associated Riemannian metric. Shortest paths can then be defined accordingly in the latent space to both follow the learned manifold and respect the ambient geometry. Through careful design of the ambient metric we can ensure that shortest paths are well-behaved even for deterministic generators that otherwise would exhibit a misleading bias. Experimentally we show that our approach improves the interpretability and the functionality of learned representations both using stochastic and deterministic generators.

1 Introduction

Unsupervised representation learning has made tremendous progress with generative models such as *variational autoencoders (VAEs)* (Kingma and Welling, 2014; Rezende et al., 2014) and *generative adversarial networks (GANs)* (Goodfellow et al., 2014). These, and similar, models provide a flexible and efficient parametrization of the density of observations in an ambient space $\mathcal{X}$ through a typically lower dimensional latent space $\mathcal{Z}$.

While the latent space $\mathcal{Z}$ constitutes a compressed representation of the data, it is by no means unique. Like most other latent variable models, these generative models are subject to *identifiability problems*, such that different representations can give rise to identical densities (Bishop, 2006). This implies that straight lines in $\mathcal{Z}$ are not shortest paths in any meaningful sense, and therefore do not constitute natural interpolants. To overcome this issue, it has been proposed to endow the latent space with a *Riemannian metric* such that curve lengths are measured in the ambient observation space $\mathcal{X}$ (Tosi et al., 2014; Arvanitidis et al., 2018). In other words, this ensures that any smooth invertible transformation of $\mathcal{Z}$ does not change the distance between a pair of points, as long as the ambient path in $\mathcal{X}$ remains the same. This approach immediately solves the identifiability problem.

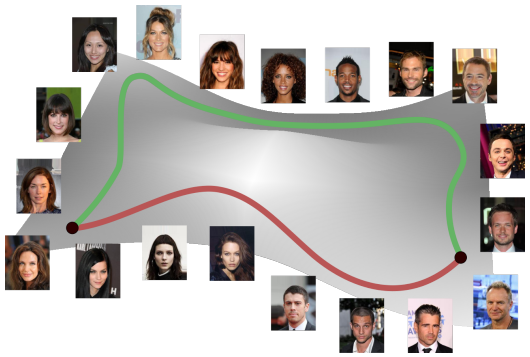

Figure 1: The proposed shortest path (—) favors the *smiling* class, while the standard shortest path (—) merely minimizes the distance on the data manifold.

While distances in $\mathcal{X}$ are well-defined and give rise to an identfiable latent representation, they need not be particularly useful. We take inspiration from *metric learning* (Weinberger et al., 2006; Arvanitidis et al., 2016) and propose to equip the ambient observation space $\mathcal{X}$ with a Riemannian metric and measure curve lengths in latent space accordingly. With this approach it is straight-forward to steer shortest paths in latent space to avoid low-density regions, but also to incorporate higher level semantic information. For instance, Fig. 1 shows a shortest path under an ambient metric that favors images of *smiling* people. In such a way, we can control, and potentially unbias, distance based methods by utilizing domain knowledge, for example in an individual fairness scenario. Hence, we get both identifiable and useful latent representations.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

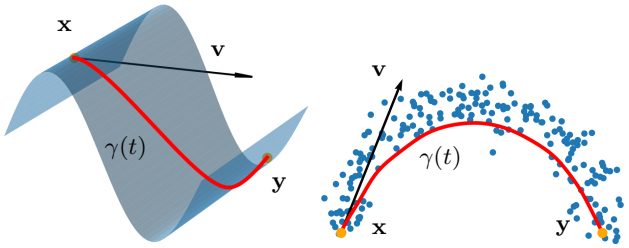

Figure 2: Examples of a tangent vector (⟶) and a shortest path (—) on an embedded $\mathcal{M} \subset \mathcal{X}$ (*left*) and on an ambient $\mathcal{X}$ (*right*).

In summary, we consider the ambient space of a generative model as a Riemannian manifold, where the metric can be defined by the user in order to encode high level information about the problem of interest. In such a way, the resulting shortest paths in the latent space move optimally on the data manifold, while respecting the geometry of the ambient space. This can be useful in scenarios where a domain expert wants to control the shortest paths in an interpretable way. In addition, we propose a simple method to construct diagonal metrics in the ambient space, as well as an architecture for the generator in order to extrapolate meaningfully. We show how this enables us to properly capture the geometry of the data manifold in deterministic generators, which is otherwise infeasible.

2 Applied Riemannian geometry intro

We are interested in Riemannian manifolds (do Carmo, 1992), which constitute well-defined metric spaces, where the inner product is defined only locally and changes smoothly throughout space. In a nutshell, these are smooth spaces where we can compute shortest paths, which prefer regions where the magnitude of the inner product is small. In this work, we show how to use these geometric structures in machine learning, where it is commonly assumed that data lie near a low dimensional manifold in an ambient observation space.

Definition 1. *A Riemannian manifold is a smooth manifold $\mathcal{M}$, equipped with a positive definite Riemannian metric $\mathbf{M}(\mathbf{x})$ $\forall$ $\mathbf{x} \in \mathcal{M}$, which changes smoothly and defines a local inner product on the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ at each point $\mathbf{x} \in \mathcal{M}$ as $\langle \mathbf{v}, \mathbf{u} \rangle_{\mathbf{x}} = \langle \mathbf{v}, \mathbf{M}(\mathbf{x})\mathbf{u} \rangle$ with $\mathbf{v}, \mathbf{u} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$.*

A smooth manifold is a topological space, which locally is homeomorphic to a Euclidean space. An intuitive way to think of a d-dimensional smooth manifold is as an embedded non-intersecting surface $\mathcal{M}$ in an ambient space $\mathcal{X}$ for example $\mathbb{R}^D$ with $D > d$ (see Fig. 2 left). In this case, the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ is a d-dimensional vector space tangential to $\mathcal{M}$ at the point $\mathbf{x} \in \mathcal{M}$. Hence, $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$ is a vector $\mathbf{v} \in \mathbb{R}^D$ and actually the Riemannian metric is $\mathbf{M}_{\mathcal{X}} : \mathcal{M} \to \mathbb{R}^{D \times D}_{\succ 0}$. Thus, the simplest approach is to assume that $\mathcal{X}$ is equipped with the Euclidean metric $\mathbf{M}_{\mathcal{X}}(\mathbf{x}) = \mathbb{I}_D$ and its restriction is utilized as the Riemannian metric on $\mathcal{T}_{\mathbf{x}}\mathcal{M}$. Since the choice of $\mathbf{M}_{\mathcal{X}}(\cdot)$ has a direct impact on $\mathcal{M}$, we can utilize other metrics designed to encode high-level semantic information (see Sec. 3).

Another view is to consider as smooth manifold the whole ambient space $\mathcal{X} = \mathbb{R}^D$. Hence, the $\mathcal{T}_{\mathbf{x}}\mathcal{X} = \mathbb{R}^D$ is centered at $\mathbf{x} \in \mathbb{R}^D$ and again the simplest Riemannian metric is the Euclidean $\mathbf{M}_{\mathcal{X}}(\mathbf{x}) = \mathbb{I}_D$. However, we are able to use other suitable metrics that simply change the way we measure distances in $\mathcal{X}$ (see Sec. 3). For instance, given a set of points in $\mathcal{X}$ we can construct a metric with small magnitude near the data to pull the shortest paths towards them (see Fig. 2 right).

For a d-dimensional embedded manifold $\mathcal{M} \subset \mathcal{X}$, a collection of *chart maps* $\phi_i : \mathcal{U}_i \subset \mathcal{M} \to \mathbb{R}^d$ is used to assign *local intrinsic coordinates* to neighborhoods $\mathcal{U}_i \subset \mathcal{M}$, and for simplicity, we assume that a global chart map $\phi(\cdot)$ exists. By definition, when $\mathcal{M}$ is smooth the $\phi(\cdot)$ and its inverse $\phi^{-1} : \phi(\mathcal{M}) \subset \mathbb{R}^d \to \mathcal{M} \subset \mathcal{X}$ exist and are smooth maps. Thus, a $\mathbf{v}_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$ can be expressed as $\mathbf{v}_{\mathbf{x}} = \mathbf{J}_{\phi^{-1}}(\mathbf{z})\mathbf{v}_{\mathbf{z}}$, where $\mathbf{z} = \phi(\mathbf{x}) \in \mathbb{R}^d$ and $\mathbf{v}_{\mathbf{z}} \in \mathbb{R}^d$ are the representations in the intrinsic coordinates. Also, the Jacobian $\mathbf{J}_{\phi^{-1}}(\mathbf{z}) \in \mathbb{R}^{D \times d}$ defines a basis that spans the $\mathcal{T}_{\mathbf{x}}\mathcal{M}$, and thus, we represent the ambient metric $\mathbf{M}_{\mathcal{X}}(\cdot)$ in the intrinsic coordinates as

$$\begin{aligned} \langle \mathbf{v}_{\mathbf{x}}, \mathbf{v}_{\mathbf{x}} \rangle_{\mathbf{x}} &= \langle \mathbf{v}_{\mathbf{z}}, \mathbf{J}_{\phi^{-1}}(\mathbf{z})^{\intercal} \mathbf{M}_{\mathcal{X}}(\phi^{-1}(\mathbf{z})) \mathbf{J}_{\phi^{-1}}(\mathbf{z}) \mathbf{v}_{\mathbf{z}} \rangle \\ &= \langle \mathbf{v}_{\mathbf{z}}, \mathbf{M}(\mathbf{z}) \mathbf{v}_{\mathbf{z}} \rangle = \langle \mathbf{v}_{\mathbf{z}}, \mathbf{v}_{\mathbf{z}} \rangle_{\mathbf{z}}, \end{aligned} \tag{1}$$

with $\mathbf{M}(\mathbf{z}) = \mathbf{J}_{\phi^{-1}}(\mathbf{z})^{\intercal} \mathbf{M}_{\mathcal{X}}(\phi^{-1}(\mathbf{z})) \mathbf{J}_{\phi^{-1}}(\mathbf{z}) \in \mathbb{R}^{d \times d}_{\succ 0}$ being smooth. As we discuss below, we should be able to evaluate the intrinsic $\mathbf{M}(\mathbf{z})$ in order to find length minimizing curves on $\mathcal{M}$. However, when $\mathcal{M}$ is embedded the chart maps are usually unknown, as well as a global chart rarely exists. In contrast, for ambient like manifolds the global chart is $\phi(\mathbf{x}) = \mathbf{x}$, which is convenient to use in practice.

In general, one of the main utilities of a Riemannian manifold $\mathcal{M} \subseteq \mathcal{X}$ is to enable us compute shortest paths therein. Intuitively, the norm $\sqrt{\langle d\mathbf{x}, d\mathbf{x} \rangle_{\mathbf{x}}}$ represents how the infinitesimal displacement vector $d\mathbf{x} \approx \mathbf{x}' - \mathbf{x}$ on $\mathcal{M}$ is locally scaled. Thus, for a curve $\gamma : [0, 1] \to \mathcal{M}$ that connects two points $\mathbf{x} = \gamma(0)$ and $\mathbf{y} = \gamma(1)$, the length on $\mathcal{M}$ or equivalently in $\phi(\mathcal{M})$ using that $\gamma(t) = \phi^{-1}(c(t))$ and Eq. 1 is measured as

$$\begin{aligned} \mathit{length}[\gamma(t)] &= \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} dt \\ &= \int_0^1 \sqrt{\langle \dot{c}(t), \mathbf{M}(c(t)) \dot{c}(t) \rangle} dt = \mathit{length}[c(t)], \end{aligned} \tag{2}$$

where $\dot{\gamma}(t) = \partial_t \gamma(t) \in \mathcal{T}_{\gamma(t)}\mathcal{M}$ is the velocity of the curve and accordingly $\dot{c}(t) \in \mathcal{T}_{c(t)}\phi(\mathcal{M})$. The minimizers of this functional are the *shortest paths*, also known as *geodesics*. We find them by solving a system of 2nd order nonlinear ordinary differential equations (ODEs) defined in the intrinsic coordinates. Notably, for ambient like manifolds the trivial chart map enables us to compute the shortest paths in practice by solving the ODEs system. In a certain sense, the behavior of the shortest paths is to avoid high measure $\sqrt{|\mathbf{M}(c(t))|}$ regions. For general manifolds, the analytical solution is intractable, so we rely on approximate solutions (Hennig and Hauberg, 2014; Yang et al., 2018; Arvanitidis et al., 2019). For further information on Riemannian geometry and the ODE system see Appendix 1.

2.1 Unifying the two manifold perspectives

In all related works, the ambient space $\mathcal{X}$ is considered as a Euclidean space. Instead, we propose to consider $\mathcal{X}$ as a Riemannian manifold. This allows us to encode high-level information through the associated metric, which constitutes an interpretable way to control the shortest paths. In order to find such a path on an embedded $\mathcal{M} \subset \mathcal{X}$, we have to solve a system of ODEs defined in the intrinsic coordinates. However, when $\mathcal{M}$ is embedded the chart maps are commonly unknown. Hence, the usual trick is to utilize another manifold $\mathcal{Z}$ that has a trivial chart map and to represent the geometry of $\mathcal{M}$ therein. Thus, we find the curve in $\mathcal{Z}$, which corresponds to the actual shortest path on $\mathcal{M}$.

In particular, assume an embedded d-dimensional manifold $\mathcal{M} \subset \mathcal{X}$ within a Riemannian manifold $\mathcal{X} = \mathbb{R}^D$ with metric $\mathbf{M}_{\mathcal{X}}(\cdot)$, a Euclidean space $\mathcal{Z} = \mathbb{R}^{d_{\mathcal{Z}}}$ called as *latent space* and a smooth function $g : \mathcal{Z} \to \mathcal{X}$ called as *generator*. Since $g(\cdot)$ is smooth, $\mathcal{M}_{\mathcal{Z}} = g(\mathcal{Z}) \subset \mathcal{X}$ is an immersed $d_{\mathcal{Z}}$-dimensional smooth (sub)manifold[1]. In general, we assume that $d_{\mathcal{Z}} = d$ and also that $g(\cdot)$ approximates closely the true embedded $\mathcal{M}_{\mathcal{Z}} \approx \mathcal{M}$, while if $d_{\mathcal{Z}} < d$ then $g(\cdot)$ can only approximate a submanifold on $\mathcal{M}$. Consequently, the Jacobian matrix $\mathbf{J}_g(\mathbf{z}) \in \mathbb{R}^{D \times d_{\mathcal{Z}}}$ is a basis that spans the $\mathcal{T}_{\mathbf{x}=g(\mathbf{z})}\mathcal{M}_{\mathcal{Z}}$, and maps a tangent vector $\mathbf{v}_{\mathbf{z}} \in \mathcal{T}_{\mathbf{z}}\mathcal{Z}$ to a tangent vector $\mathbf{v}_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}=g(\mathbf{z})}\mathcal{M}_{\mathcal{Z}}$. Thus, as before the restriction of $\mathbf{M}_{\mathcal{X}}(\cdot)$ on $\mathcal{T}_{\mathbf{x}}\mathcal{M}_{\mathcal{Z}}$ induces a metric in the latent space $\mathcal{Z}$ as

$$\langle \mathbf{v}_{\mathbf{x}}, \mathbf{v}_{\mathbf{x}} \rangle_{\mathbf{x}} = \langle \mathbf{v}_{\mathbf{z}}, \mathbf{J}_g(\mathbf{z})^\intercal \mathbf{M}_{\mathcal{X}}(g(\mathbf{z})) \mathbf{J}_g(\mathbf{z}) \mathbf{v}_{\mathbf{z}} \rangle. \quad (3)$$

The induced Riemannian metric in the latent space $\mathbf{M}(\mathbf{z}) = \mathbf{J}_g(\mathbf{z})^\intercal \mathbf{M}_{\mathcal{X}}(g(\mathbf{z})) \mathbf{J}_g(\mathbf{z})$ is known as the *pull-back* metric. Essentially, it captures the intrinsic geometry of the immersed $\mathcal{M}_{\mathcal{Z}}$, while taking into account the geometry of $\mathcal{X}$. Therefore, the space $\mathcal{Z}$ together with $\mathbf{M}(\mathbf{z})$ constitutes a Riemannian manifold, but since $\mathcal{Z} = \mathbb{R}^{d_{\mathcal{Z}}}$ the chart map and the $\mathcal{T}_{\mathbf{z}}\mathcal{Z}$ are trivial. Consequently, we can evaluate the metric $\mathbf{M}(\mathbf{z})$ in intrinsic coordinates, which enables us to compute shortest paths on $\mathcal{Z}$ by solving the ODEs system. Intuitively, these paths in $\mathcal{Z}$ move optimally on $\mathcal{M}_{\mathcal{Z}}$, while simultaneously respecting the geometry of the ambient space $\mathcal{X}$. Also, note that $g(\cdot)$ is *not* a chart map, and hence, it is easier to learn. Moreover, if the data manifold has a special topology, we have to choose the latent space accordingly (Davidson et al., 2018; Mathieu et al., 2019). For further discussion and the theoretical properties of $g(\cdot)$ see Appendix 2.

3 Data learned Riemannian manifolds

We discuss previous approaches that apply differential geometry, which largely inspire our work. Briefly, we present two related Riemannian metric learning methods, and then, we propose a simple technique to construct similar metrics in the ambient space $\mathcal{X}$. This constitutes a principled and interpretable way to encode high-level information as domain knowledge in our models. Also, we present the related work where the structure of an embedded data manifold is properly captured in the latent space of stochastic generators.

3.1 Riemannian metrics in ambient space

Assume that a set of points $\{\mathbf{x}_n\}_{n=1}^N$ in $\mathcal{X} = \mathbb{R}^D$ is given. The Riemannian metric learning task is to learn a positive definite metric tensor $\mathbf{M}_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}_{\succ 0}^{D \times D}$ that changes smoothly across the space. The actual behavior of the metric depends on the problem that we want to model. For example, if we want the shortest paths to stay on the data manifold (see Fig. 2 right) the meaningful behavior for the metric is that the measure $\sqrt{|\mathbf{M}_{\mathcal{X}}(\cdot)|}$ should be small near the data and large as we move away. Similarly, in Fig. 1 the ambient metric is designed such that its measure is small near the data points with smiling face, and thus, the shortest paths tend to follow this semantic constraint. In such a way, we can regulate the inductive bias of a model, by including high-level information through the ambient metric and controlling the shortest paths accordingly.

One of the first approaches to learn such a Riemannian metric was presented by Hauberg et al. (2012), where $\mathbf{M}_{\mathcal{X}}(\mathbf{x})$ is the convex combination of a predefined set of metrics, using a smooth weighting function. In particular, at first K metrics are estimated $\{\mathbf{M}_k\}_{k=1}^K \in \mathbb{R}_{\succ 0}^{D \times D}$ centered at the locations $\{\mathbf{c}_k\}_{k=1}^K \in \mathbb{R}^D$. Then,

[1] A function $g : \mathcal{Z} \to \mathcal{M} \subset \mathcal{X}$ is an immersion if the $\mathbf{J}_g : \mathcal{T}_{\mathbf{z}}\mathcal{Z} \to \mathcal{T}_{g(\mathbf{z})}\mathcal{M}$ is injective (full rank) $\forall \mathbf{z} \in \mathcal{Z}$. Intuitively, $\mathcal{M}$ is a $\dim(\mathcal{Z})$-dimensional surface and intersections are allowed. While $g(\cdot)$ is an embedding if it is an injective function, which means that interections are not allowed. An immersion locally is an embedding.

we can evaluate the metric at a new point as

$$\mathbf{M}_{\mathcal{X}}(\mathbf{x}) = \sum_{k=1}^{K} w_k(\mathbf{x})\mathbf{M}_k, \quad w_k(\mathbf{x}) = \frac{\tilde{w}_k(\mathbf{x})}{\sum_{j=1}^{K} \tilde{w}_j(\mathbf{x})}, \quad (4)$$

where the kernel $\tilde{w}_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_k\|_2^2}{2\sigma^2}\right)$ with bandwidth $\sigma \in \mathbb{R}_{>0}$ is a smooth function, and thus, the metric is smooth as a linear combination of smooth functions. A practical example for the base metrics $\mathbf{M}_k$ is the local Linear Discriminant Analysis (LDA), where local metrics are learned using labeled data such that to separate well the classes locally (Hastie and Tibshirani, 1994). In a similar spirit, a related approach is the Large Margin Nearest Neighbor (LMNN) classifier (Weinberger et al., 2006). Note that the domain of metric learning provides a huge list of options that can be considered (Suárez et al., 2018).

Similarly, in an unsupervised setting Arvanitidis et al. (2016) proposed to construct the Riemannian metric in a non-parametric fashion as the the inverse of the local diagonal covariance. In particular, for a given point set $\{\mathbf{x}_n\}_{n=1}^N$ at a point $\mathbf{x}$ the diagonal elements of the metric $\mathbf{M}_{\mathcal{X}}(\cdot)$ are equal to

$$M_{\mathcal{X}_{dd}}(\mathbf{x}) = \left(\sum_{n=1}^{N} w_n(\mathbf{x})(x_{nd} - x_d)^2 + \varepsilon\right)^{-1}, \quad (5)$$

with $w_n(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_n-\mathbf{x}\|_2^2}{2\sigma^2}\right)$, where the parameter $\sigma \in \mathbb{R}_{>0}$ controls the curvature of the Riemannian manifold i.e., how fast the metric changes, and $\varepsilon > 0$ is a small scalar to upper bound the metric. Although these are quite flexible and intuitive metrics, selecting the parameter σ is a challenging task (Arvanitidis et al., 2017), especially due to the curse of dimensionality (Bishop, 2006) and the sample size N due to the non-parametric regime.

The proposed Riemannian metrics. Inspired by the approaches described above and Peyré et al. (2010), we propose a general and simple technique to easily construct metrics in $\mathcal{X}$, which allows to encode information depending on the problem of interest. An unsupervised diagonal metric can be defined as

$$\mathbf{M}_{\mathcal{X}}(\mathbf{x}) = (\alpha \cdot h(\mathbf{x}) + \varepsilon)^{-1} \cdot \mathbb{I}_D, \quad (6)$$

where $\mathbb{I}_D$ is the identity matrix with dimension D, the function $h(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}_{>0}$ with behavior $h(\mathbf{x}) \to 1$ when $\mathbf{x}$ is near the data manifold, otherwise $h(\mathbf{x}) \to 0$, and $\alpha, \epsilon \in \mathbb{R}_{>0}$ are scaling factors to lower and upper bound the metric, respectively. One simple but very effective approach is to use a positive Radial Basis Function (RBF) network (Que and Belkin, 2016) as $h(\mathbf{x}) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x})$ with weights $\mathbf{w} \in \mathbb{R}_{>0}^K$ and $\phi_k(\mathbf{x}) = \exp(-0.5 \cdot \lambda_k \cdot \|\mathbf{x} - \mathbf{x}_k\|_2^2)$ with bandwidth $\lambda_k \in \mathbb{R}_{>0}$. Also, $h(\mathbf{x})$ can be the probability density function of the given data. Usually, the true density function is unknown and difficult to learn, however, we can approximate it roughly by utilizing a simple model as the Gaussian Mixture Model (GMM) (Bishop, 2006). Such a Riemannian metric pulls the shortest paths towards areas of $\mathcal{X}$ with high $h(\mathbf{x})$ (see Fig. 2 right).

In a similar context, a supervised version can be defined where the function $h(\mathbf{x})$ represents cost, while in Eq. 6 we do not use the inversion. In this way, shortest paths will tend to avoid regions of the ambient space $\mathcal{X}$ where the cost function is high. For instance, in Fig. 1 we can think of a cost function that is high over all the non-smiling data points. Of course, such a cost function can be learned as an independent regression or classification problem, while in other cases can even be given by the problem or a domain expert. Further details about all the metrics above in Appendix 3.

3.2 Riemannian metrics in latent space

As discussed in Sec. 2.1, we can capture the geometry of the given embedded data manifold $\mathcal{M} \subset \mathcal{X}$ by learning a smooth generator $g : \mathcal{Z} \to \mathcal{M}_{\mathcal{Z}} \subset \mathcal{X}$ such that $\mathcal{M}_{\mathcal{Z}} \approx \mathcal{M}$. In previous works has been shown how to learn in practice such a function $g(\cdot)$, and also, the mild conditions it has to follow so that the induced Riemannian metric to capture properly the structure of $\mathcal{M}$. In the latent space $\mathcal{Z} = \mathbb{R}^d$ we call as latent codes or representations the points $\{\mathbf{z}_n \in \mathcal{Z} \mid \mathbf{x}_n = g(\mathbf{z}_n),\ n = 1, \ldots, N\}$.

First Tosi et al. (2014) considered the Gaussian Process Latent Variable Model (GP-LVM) (Lawrence, 2005), where $g(\cdot)$ is a stochastic function defined as a multi-output Gaussian process $g \sim \mathcal{GP}(\mathbf{m}(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$. This stochastic generator induces a random Riemannian metric in $\mathcal{Z}$ and the expected metric is used for computing shortest paths. The advantage of such a stochastic generator is that the metric magnitude increases analogous to the uncertainty of $g(\cdot)$, which happens in regions of $\mathcal{Z}$ where there are no latent codes. Apart from this desired behavior, this metric is not very practical due to the GPs computational cost.

Another set of approaches known as deep generative models, parameterizes $g(\cdot)$ as a deep neural network (DNN). On the one hand we have the explicit density models, where the marginal likelihood can be computed, with main representatives the VAE (Kingma and Welling, 2014; Rezende et al., 2014) and the normalizing flow models (Dinh et al., 2016; Rezende and Mohamed, 2015). On the other hand we have the implicit density models for which the marginal likelihood is intractable, as is the GAN (Goodfellow et al., 2014).

Recently, Arvanitidis et al. (2018) showed that we are able to properly capture the structure of the embedded data manifold $\mathcal{M} \subset \mathcal{X}$ in the latent space $\mathcal{Z}$ of a VAE under the condition of having meaningful uncertainty quantification for the generative process. In particular, the standard VAE assumes a Gaussian likelihood $p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \mu(\mathbf{z}),\ \mathbb{I}_D \cdot \sigma^2(\mathbf{z}))$ with a prior $p(\mathbf{z}) = \mathcal{N}(0, \mathbb{I}_d)$. Hence, the generator can be written as $g(\mathbf{z}) = \mu(\mathbf{z}) + \text{diag}(\varepsilon) \cdot \sigma(\mathbf{z})$ where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_D)$ and $\mu : \mathcal{Z} \to \mathcal{X}$, $\sigma : \mathcal{Z} \to \mathbb{R}^D_{>0}$ are usually parametrized with DNNs. However, simply parametrizing $\sigma(\cdot)$ with a DNN does not directly imply meaningful uncertainty quantification, because in general the DNN extrapolates arbitrarily to regions of $\mathcal{Z}$ with no latent codes. Thus, the proposed solution in Arvanitidis et al. (2018) is to model the inverse variance $\beta(\mathbf{z}) = (\sigma^2(\mathbf{z}))^{-1}$ with a positive Radial Basis Function (RBF) network (Que and Belkin, 2016), which implies that moving further from the latent codes increases the uncertainty. Under this stochastic generator, the expected Riemannian metric in $\mathcal{Z}$ can be written as

$$\begin{aligned}\mathbf{M}(\mathbf{z}) &= \mathbb{E}_{p(\varepsilon)}[\mathbf{J}_{g_\varepsilon}(\mathbf{z})^\intercal \mathbf{J}_{g_\varepsilon}(\mathbf{z})] \\ &= \mathbf{J}_\mu(\mathbf{z})^\intercal \mathbf{J}_\mu(\mathbf{z}) + \mathbf{J}_\sigma(\mathbf{z})^\intercal \mathbf{J}_\sigma(\mathbf{z}), \end{aligned} \tag{7}$$

where $g_\varepsilon(\cdot)$ implies that ε is kept fixed $\forall\ \mathbf{z} \in \mathcal{Z}$, which ensures a smooth mapping, and hence, differentiability (Eklund and Hauberg, 2019). Here, we observe that the metric increases when the generator becomes uncertain due to the second term. This constitutes a desired behavior, as the metric informs us to avoid regions of $\mathcal{Z}$ where there are no latent codes, which directly implies that these regions do not correspond to parts of the data manifold in $\mathcal{X}$. In some sense, we model the topology of $\mathcal{M}$ as well (Hauberg, 2018). In contrast, this is not the case when the uncertainty of $g(\cdot)$ is not taken into account, since then, the behavior of the metric is arbitrary as moving away from the latent codes (Chen et al., 2018; Shao et al., 2017)

Consequently, deterministic generators as the Auto-Encoder (AE) and the GAN capture poorly the structure of $\mathcal{M}$ in $\mathcal{Z}$ because the second term in Eq. 7 does not exist. The reason is that these models are trained with likelihood $p(\mathbf{x} \mid \mathbf{z}) = \delta(\mathbf{x} - g(\mathbf{z}))$ and the uncertainty is not quantified. Of course, for the AE one potential *heuristic* solution is to use the latent codes of the training data to fit *post-hoc* a meaningful variance estimator under the Gaussian likelihood and the maximum likelihood principle as $\theta^* = \text{argmax}_\theta \prod_{n=1}^N \mathcal{N}(\mathbf{x}_n \mid g(e(\mathbf{x}_n)),\ \mathbb{I}_D \cdot \sigma^2_\theta(e(\mathbf{x}_n)))$, with encoder $e : \mathcal{X} \to \mathcal{Z}$. In principle, we could follow the same procedure for the GAN by learning an encoder (Donahue et al., 2016; Dumoulin et al., 2016). However, it is still unclear if the encoder for a GAN learns meaningful representations or if the powerful generator ignores the inferred latent codes (Arora et al., 2018).

Therefore, to properly capture the structure of $\mathcal{M}$ in $\mathcal{Z}$ we mainly rely on stochastic generators with increasing uncertainty as we move further from the latent codes. Even if the RBF based approach is a meaningful way to get the desired behavior, in general, uncertainty quantification with parametric models is still considered as an open problem (MacKay, 1992; Gal and Ghahramani, 2015; Lakshminarayanan et al., 2017; Detlefsen et al., 2019; Arvanitidis et al., 2018). Nevertheless, Eklund and Hauberg (2019) showed that the expected Riemannian metric in Eq. 7 is a reasonable approximation to use in practice. Obviously, when $g(\cdot)$ is deterministic, like the GAN, the second term in Eq. 7 disappears, since these models do not quantify the uncertainty of the generative process. This directly means that deterministic generators are not able, by construction, to properly capture the geometric structure of $\mathcal{M}$ in the latent space, and hence, exhibit a misleading bias (Hauberg, 2018).

4 Enriching the latent space with geometric information

Here, we unify the approaches presented in Sec. 3.1 and Sec. 3.2, in order to provide additional geometric structure in the latent space of a generative model. This is the first time that these two fundamentally different Riemannian views are combined. Their main difference is that the metric induced by $g(\cdot)$ merely tries to capture the intrinsic geometry of the given data manifold $\mathcal{M}$, while $\mathbf{M}_\mathcal{X}(\cdot)$ allows to directly encode high-level information in $\mathcal{X}$ based for instance on domain knowledge. Moreover, we provide in the stochastic case a relaxation for efficient computation of the expected metric. While in the deterministic case, we combine a carefully designed ambient metric with a new architecture for $g(\cdot)$ to extrapolate meaningfully, which is one way to ensure well-behaved shortest paths that respect the structure of the data manifold $\mathcal{M}$.

Stochastic generators. Assuming that an ambient $\mathbf{M}_\mathcal{X}(\cdot)$ is given (see Sec. 3.1), we learn a VAE with Gaussian likelihood, so the stochastic mapping is $g(\mathbf{z}) = \mu(\mathbf{z}) + \text{diag}(\varepsilon) \cdot \sigma(\mathbf{z})$, while using a positive RBF for meaningful quantification of the uncertainty $\sigma(\cdot)$. Therefore, we can apply Eq. 3 to derive the new stochastic more informative pull-back metric in the latent space $\mathcal{Z}$, which is equal to

$$\begin{aligned}\mathbf{M}_\varepsilon(\mathbf{z}) &= \mathbf{J}_{g_\varepsilon}(\mathbf{z})^\intercal \mathbf{M}_\mathcal{X}(\mu(\mathbf{z}) + \text{diag}(\varepsilon) \cdot \sigma(\mathbf{z})) \mathbf{J}_{g_\varepsilon}(\mathbf{z}) \\ &\text{with} \quad \mathbf{J}_{g_\varepsilon}(\mathbf{z}) = \mathbf{J}_\mu(\mathbf{z}) + \text{diag}(\varepsilon) \cdot \mathbf{J}_\sigma(\mathbf{z}). \end{aligned} \tag{8}$$

This is a random quantity and we can compute its expectation directly by sampling $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_D)$, so that $\mathbf{M}(\mathbf{z}) = \mathbb{E}_{\varepsilon \sim p(\varepsilon)}[\mathbf{M}_\varepsilon(\mathbf{z})]$. However, in practice this expectation will increase dramatically the computational

cost, especially, since we need to evaluate the expected metric many times when computing a shortest path. For this reason, we relax the problem and consider only the $\mathbb{E}_{\varepsilon\sim p(\varepsilon)}[g_\varepsilon(\mathbf{z})] = \mu(\mathbf{z})$ for the evaluation of the ambient metric $\mathbf{M}_\mathcal{X}(\cdot)$ in Eq. 8, which simplifies the corresponding expected Riemannian metric to

$$\mathbf{M}(\mathbf{z}) \triangleq \mathbf{J}_\mu(\mathbf{z})^\intercal \mathbf{M}_\mathcal{X}\left(\mu(\mathbf{z})\right) \mathbf{J}_\mu(\mathbf{z}) + \mathbf{J}_\sigma(\mathbf{z})^\intercal \mathbf{M}_\mathcal{X}\left(\mu(\mathbf{z})\right) \mathbf{J}_\sigma(\mathbf{z}). \quad (9)$$

Essentially, the realistic underlying assumption is that near the latent codes $\sigma(\mathbf{z}) \to \mathbf{0}$ so $g(\mathbf{z}) \to \mu(\mathbf{z})$ and the first term dominates. But, as we move in regions of $\mathcal{Z}$ with no codes the $\sigma(\mathbf{z}) \gg 0$, and for this reason, we need the second term in the equation. In particular, $\mathbf{J}_\sigma(\cdot)$ dominates when moving further from the latent codes, and hence, the behavior of $\mathbf{M}_\mathcal{X}(\cdot)$ will be less important there. Thus, we are allowed to consider this relaxation, for which the meaningful uncertainty estimation is still necessary. We further analyze and check empirically this relaxation in Appendix 4.

Deterministic generators. For deterministic $g(\cdot)$, we propose one simple yet efficient way to ensure well-behaved shortest paths, which respect the structure of the given data manifold $\mathcal{M}$. First, we learn an ambient Riemannian metric $\mathbf{M}_\mathcal{X}(\cdot)$ in $\mathcal{X}$ that only roughly represents the structure of $\mathcal{M}$, for instance, by using an RBF or a GMM (Eq. 6). Thus, this metric informs us how close the generated $\mathcal{M}_\mathcal{Z}$ is to the data. Hence, we additionally need $g(\cdot)$ to extrapolate meaningfully. Therefore, $g(\cdot)$ should learn to generate well the given data from a prior $p(\mathbf{z})$, but as we move further from the support of $p(\mathbf{z})$, the generated $\mathcal{M}_\mathcal{Z}$ should also move further from the given data in $\mathcal{X}$. Consequently, since $\mathbf{M}_\mathcal{X}(\cdot)$ is designed to increase far from the given data, the induced Riemannian metric in $\mathcal{Z}$ captures properly the structure of the data manifold.

One of the simplest deterministic generators with this desirable behavior is the probabilistic Principal Component Analysis (pPCA) (Tipping and Bishop, 1999). This basic model has a Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0},\, \mathbb{I}_d)$ and a generator that is simply a linear map. The generator is constructed by the top d eigenvectors of the empirical covariance matrix, scaled by their eigenvalues. Inspired by this simple model, we propose for the deterministic generator the following architecture

$$g(\mathbf{z}) = f(\mathbf{z}) + \mathbf{U} \cdot \mathrm{diag}([\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_d}]) \cdot \mathbf{z} + \mathbf{b}, \quad (10)$$

where $f : \mathcal{Z} \to \mathcal{X}$ is a deep neural network, $\mathbf{U} \in \mathbb{R}^{D\times d}$ the top d eigenvectors with their corresponding eigenvalues λ_d computed from the empirical data covariance, and $\mathbf{b} \in \mathbb{R}^D$ is the data mean. This interpretable model can be seen as a residual network (ResNet) (He et al., 2015). In particular, the desired behavior is that as we move further from the support of $p(\mathbf{z})$ the linear part of Eq. 10 becomes the dominant one, especially, when bounded activation functions are utilized for $f(\cdot)$. Hence, the $\mathcal{M}_\mathcal{Z}$ will extrapolate meaningfully as we move further from the support of the prior. However, we also need the generated $\mathcal{M}_\mathcal{Z}$ to be a valid immersion. For further discussion see Appendix 2.

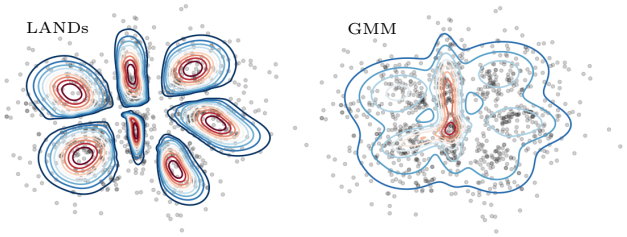

Figure 3: Using the samples in $\mathcal{Z}$ generated from the proposed $q(\mathbf{z})$ we fit a mixture of LANDs and a GMM. Thus, we can capture the true underlying structure of the data manifold in the latent space of a GAN.

5 Experiments

5.1 Deterministic generators

Synthetic data. Usually, in GANs the $g(\cdot)$ is a continuous function, so we expect some generated points to fall off the given data support. Mainly, when the data lie near a disconnected $\mathcal{M}$ and irrespective of training optimallity. This is known as the distribution mismatch problem. We generate a synthetic dataset (Fig. 4) and we train a Wasserstein GAN (WGAN) (Arjovsky et al., 2017) with a latent space $\mathcal{Z} = \mathbb{R}^2$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{0},\, \mathbb{I}_2)$. For the ambient metric we used a positive RBF (Eq. 6). Further details in Appendix 5.

Then, we define a density function in $\mathcal{Z}$ using the induced Riemannian metric as $q(\mathbf{z}) \propto \tilde{p}_r(\mathbf{z}) \cdot (1 + \sqrt{|\mathbf{M}(\mathbf{z})|})^{-1}$ (Lebanon, 2002; Le and Cuturi, 2015), where $\tilde{p}_r(\mathbf{z})$ is a uniform density within a ball of radius r. The behavior of $q(\mathbf{z})$ is interpretable, since the density is high wherever $\mathbf{M}(\cdot)$ is small and that happens only in the regions of $\mathcal{Z}$ that $g(\cdot)$ learns to map near the given data manifold in $\mathcal{X}$. Also, the $\tilde{p}(\mathbf{z})$ simply ensures that the support of $q(\mathbf{z})$ is within the region where the $g(\cdot)$ is trained. Intuitively, we can think of $q(\mathbf{z})$ as an approximate aggregated posterior, but without training an encoder. Thus, we compare samples generated from $q(\mathbf{z})$ using Markov Chain Monte Carlo (MCMC) and the prior $p(\mathbf{z})$. We clearly see in Fig. 4 that our samples align better with the given data manifold. Furthermore, we used the precision and recall metrics (Kynkäänniemi et al., 2019) with K=10, which shows that we generate more accurate samples (high precision), while our coverage is sufficiently good

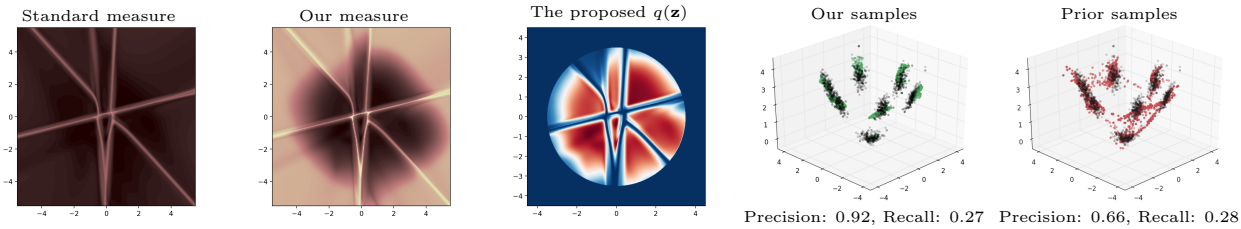

Figure 4: Our measure shows that the ambient metric helps to properly capture the data manifold structure in the latent space. Also, we generate more accurate samples using the proposed $q(\mathbf{z})$ compared to the prior $p(\mathbf{z})$.

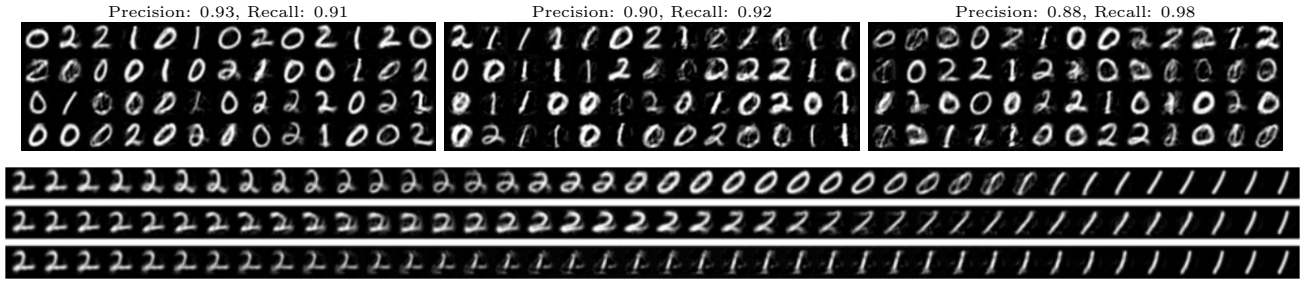

Figure 5: *Top* panels: Samples from $q(\mathbf{z})$ using the $\mathbf{M}_{\mathcal{X}'}(\cdot)$ are more accurate (*left*), even if the ambient metric (*middle*) is not used, compared to the samples from $p(\mathbf{z})$ (*right*). *Bottom* panels: our shortest path (*top*) respects the data manifold structure avoiding off-the-manifold "shortcuts" (*middle*), while the linear (*bottom*) is arbitrary.

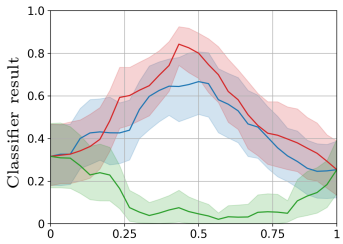

Figure 6: Classification of the paths in $\mathcal{X}$ over time. Clearly, we see that only our shortest paths (—) respect the ambient geometry, while both the straight line (—) and the naive shortest path (—), traverse regions which are classified as blond.

(high recall). Moreover, using the sampled latent representations we fit a mixture of LANDs (Fig. 3), which are adaptive normal distributions on Riemannian manifolds (Arvanitidis et al., 2016). Hence, we can sample from each component individually, without training a conditional GAN (Mirza and Osindero, 2014).

MNIST data. Similarly, we perform an experiment with the MNIST digits 0,1,2 and $\mathcal{Z} = \mathbb{R}^5$. This is a high dimensional dataset so the RBF used for the $\mathbf{M}_{\mathcal{X}}(\cdot)$ fits poorly. Also, the Euclidean distance for images is not meaningful. Thus, after training the WGAN, we use PCA to project linearly the data in a d'-dimensional subspace $D > d' > d$ with $d' = 10$, where we construct the ambient $\mathbf{M}_{\mathcal{X}'}(\cdot)$. This removes the non-informative dimensions from the data, while the global structure of the data manifold is approximately preserved. The results in Fig. 5 show that, indeed, the $\mathbf{M}_{\mathcal{X}'}(\cdot)$ improves the sampling since we achieve higher precision, but lower recall, while our path traverse better the data manifold. We discuss the linear projection step in Appendix 2, and provide further implementation details and results in Appendix 5.

Pre-trained generator. We use for $g(\cdot)$ a pre-trained Progressive GAN on the CelebA dataset (Karras et al., 2018). We also train a classifier that distinguishes the blond people. As before, we use a linear projection to $d' = 1000$, where we define a cost based ambient metric $\mathbf{M}_{\mathcal{X}'}(\cdot)$ using a positive RBF (Eq. 6). This metric is designed to penalize regions in $\mathcal{X}$ that correspond to blonds. Our goal is to avoid these regions when interpolating in $\mathcal{Z}$. As discussed in Sec. 3, it is not guaranteed that this deterministic $g(\cdot)$ properly captures the structure of the data manifold in $\mathcal{Z}$, but even so, we test our ability to control the paths.

As we observe in Fig. 7 only our path that utilizes the informative $\mathbf{M}_{\mathcal{X}'}(\cdot)$ successfully avoids crossing regions with blond hair. In particular, it corresponds to the optimal path on the generated manifold, while taking into account the high-level semantic information. In contrast, the shortest path without the $\mathbf{M}_{\mathcal{X}'}(\cdot)$ passes through the high cost region in $\mathcal{X}$, as it merely minimizes the distance on the manifold and does not utilize the additional information. Also, we show in Fig. 6 the classifier prediction along several interpolants. For further details and interpolation results see Appendix 5.

Figure 7: Our interpolant (*top*) successfully avoids the high cost regions on the data manifold (blond hair), since it utilizes the high-level semantic information that is encoded into the ambient metric. The standard shortest path (*middle*) provides a smoother interpolation than the linear case (*bottom*).

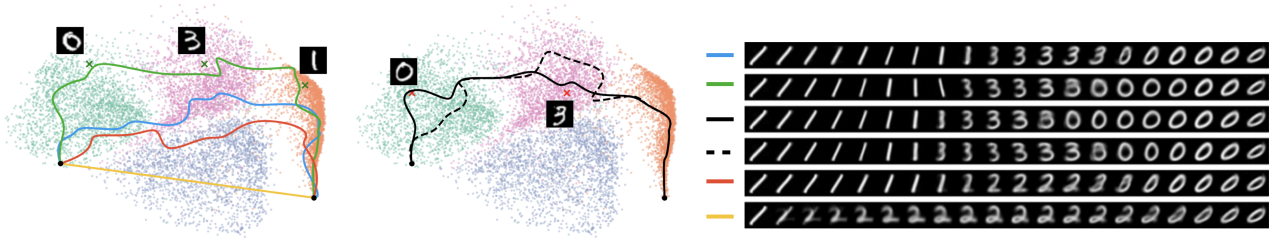

Figure 8: Comparing interpolants in the latent space under different ambient metrics and their generated images. We can control effectively the shortest paths to follow high-level information incorporated in $\mathbf{M}_{\mathcal{X}'}(\cdot)$.

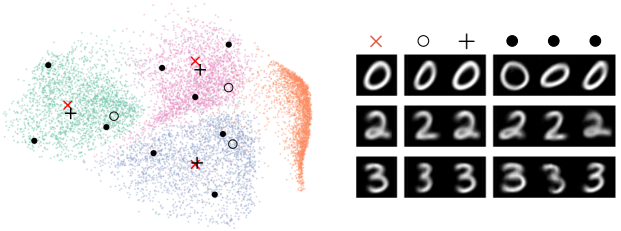

Figure 9: Compare means using the standard shortest path (+) and our approach (○) that respects high-level information i.e. it avoids the area around (×) in $\mathcal{X}'$.

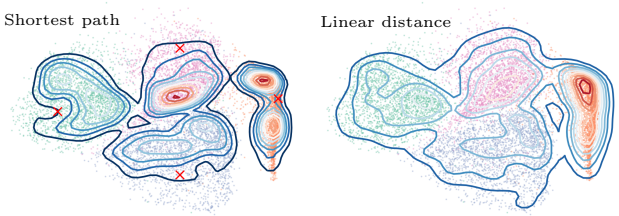

Figure 10: The KDE using the shortest path and the straight line as distance measure.

5.2 Stochastic Generators

Controlling Shortest Paths. In Fig. 8 we compare the effect of several interpretable ambient metrics on the shortest paths in the latent space of a VAE trained with $\mathcal{Z} = \mathbb{R}^2$ on the MNIST digits 0,1,2,3. As before, we project linearly the data in $d' = 10$ to construct there the $\mathbf{M}_{\mathcal{X}'}(\cdot)$. Under the Euclidean metric in $\mathcal{X}$ the path (—) merely follows the structure of the generated $\mathcal{M}_{\mathcal{Z}}$ since it avoids regions with no data. At first, we construct an LDA metric in $\mathcal{X}'$ (Eq. 4) by considering the digits 0,1,3 to be in the same class. Hence, the resulting path (—) avoids crossing the regions in $\mathcal{Z}$ that correspond to digit 2, while simultaneously respects the geometry of $\mathcal{M}_{\mathcal{Z}}$. Also, we select 3 data points (×) and using their 100 nearest neighbors in $\mathcal{X}'$ we construct the local covariance based metric (Eq. 5), so that the path (—) moves closer to the selected points.

Moreover, we linearly combine these two metrics to enforce the path (—) to pass through 0,1,3 while moving closer to the selected points (×). Finally, we include to this linear combination a cost related metric (Eq. 6) based on a positive RBF that increases near the points (×) in $\mathcal{X}'$. Therefore, the resulting path (- -) avoids these neighborhoods, while respecting the other ambient metrics and the geometry of $\mathcal{M}_{\mathcal{Z}}$.

Compare means. In Fig. 9 we compute the mean values for three sets of points (•), comparing the standard shortest path (+) with our approach (○) using a cost based $\mathbf{M}_{\mathcal{X}'}(\cdot)$ relying on a positive RBF with centers (×) in $\mathcal{X}'$. Our method allows to use domain knowledge to get prototypes that respect some desired properties. In such a way, we can use generative models for applications as in natural sciences, where domain experts provide information through the metric.

Density Estimation. We show in Fig. 10 the kernel density estimation (KDE) in $\mathcal{Z}$ comparing the straight line to the shortest path, under the same bandwidth. For $\mathbf{M}_{\mathcal{X}'}(\cdot)$ we linearly combine an LDA metric, where

each digit is a separate class, and a cost based metric with centers ($\times$) in $\mathcal{X}'$. The $\mathbf{M}(\cdot)$ distinguishes better the classes due to the LDA metric, while the density is reduced near the high cost regions. Further discussion for all the experiments can be found in Appendix 5.

6 Conclusion

We considered the ambient space of generative models as a Riemannian manifold. This allows to encode high-level information through the metric and we proposed an easy way to construct suitable metrics. In order to capture the geometry into the latent space, proper uncertainty estimation is essential in stochastic generators, while in the deterministic case one way is through the proposed meaningful extrapolation. Thus, we get interpretable shortest paths that respect the ambient space geometry, while moving optimally on the learned manifold. In the future, it will be interesting to investigate if and how we can use the ambient space geometry during the learning phase of a generative model.

Acknowledgments

SH was supported by a research grant (15334) from VILLUM FONDEN. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n^{o} 757360).

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Arora, S., Risteski, A., and Zhang, Y. (2018). Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations (ICLR)*.

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

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2017). Maximum likelihood estimation of riemannian metrics from euclidean data. In *Geometric Science of Information (GSI)*.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2018). Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations (ICLR)*.

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

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and Smagt, P. (2018). Metrics for Deep Generative Models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders.

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

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real NVP. *CoRR*, abs/1605.08803.

do Carmo, M. (1992). *Riemannian Geometry*. Mathematics (Boston, Mass.). Birkhäuser.

Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Adversarial feature learning.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2016). Adversarially learned inference.

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

Gal, Y. and Ghahramani, Z. (2015). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *arXiv:1506.02142*.

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

Hastie, T. and Tibshirani, R. (1994). Discriminant adaptive nearest neighbor classification.

Hauberg, S. (2018). Only bayes should learn a manifold.

Hauberg, S., Freifeld, O., and Black, M. (2012). A Geometric Take on Metric Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Hennig, P. and Hauberg, S. (2014). Probabilistic Solutions to Differential Equations and their Application to Riemannian Statistics. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*

Le, T. and Cuturi, M. (2015). Unsupervised riemannian metric learning for histograms using aitchison transformations. In *International Conference on International Conference on Machine Learning (ICML)*.

Lebanon, G. (2002). Learning riemannian metrics. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI)*.

MacKay, D. J. C. (1992). A practical bayesian framework for backpropagation networks. *Neural Comput.*

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Peyré, G., Péchaud, M., Keriven, R., and Cohen, L. D. (2010). Geodesic methods in computer vision and graphics. *Found. Trends. Comput. Graph. Vis.*

Que, Q. and Belkin, M. (2016). Back to the future: Radial basis function networks revisited. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Cadiz, Spain.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*.

Shao, H., Kumar, A., and Fletcher, P. T. (2017). The Riemannian Geometry of Deep Generative Models. In *arXiv*.

Suárez, J. L., García, S., and Herrera, F. (2018). A tutorial on distance metric learning: Mathematical foundations, algorithms and experiments.

Tipping, M. E. and Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*.

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

Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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