Geometric Data Analysis :
Latent space oddity : On the curvature of deep generative models

Pierre-Louis Benveniste
Master MVA
pierrelouis.benveniste03@gmail.com

Archibald Fraikin
Master MVA
fraikinarchie@gmail.com

1. Introduction

Generative models have gained widespread popularity in recent years due to their ability to generate new data samples that are similar to a training dataset. These models have a wide range of applications, including image generation, language translation, and anomaly detection. At the core of generative models is the latent space, a lower-dimensional representation of the input data that captures the underlying structure and patterns of the data. In this project, we discuss the paper "Latent Space Oddity: On the curvature of deep generative models" [2], which aims to shed light on the geometric structure of latent spaces in generative models and how it can be leveraged to improve model performance.

The latent space is an essential component of generative models, as it encodes important information about the data and how it is being modeled. It allows the network to generate new samples that are similar to the training data, but also provides a way to interpolate between different data points, allowing the model to smoothly transition between different samples and create a continuous range of data.

By understanding the geometric structure of the latent space, we can gain insights into the properties of the generative model and how it is modeling the data. For example, we may be able to identify patterns or structures in the latent space that correspond to particular features or characteristics of the data. In order to analyze the geometry of latent spaces, it is useful to consider Riemannian metrics, as these spaces are most often curved and Euclidean geometry is ill-suited to understand their structure. In light of this, the authors of [2] propose a stochastic Riemannian metric which helps understand the curvature of latent spaces and develop more performant models.

The two main contributions of [2] are:

- A new stochastic Riemannian metric in the latent space of generators.
- A new variance network for the generator.

This Riemannian metric helps us better understand latent space structure and improves the performance of deep generative models, as it captures the geometry (more specifically the curvature) of latent spaces, which Euclidian metrics fail to do.

These contributions are very fundamental and based on theoretical insights rather than specific applications, which makes the potential applications of this work very broad. Most learning task which involve handling variables in a latent space, such as classification or clustering using VAEs, or image translation with GANs [5], would benefit from a more accurate metric to measure distances in latent spaces.

2. Related work

The geometry of latent spaces in generative models has not been studied extensively, which makes [2] particularly exciting.

The authors of [9] develop an algorithm for parallel translation of a tangent vector along a path on the manifold. They find that the curvature of manifolds learned by generative models (applied to real images) is nonlinear but close to zero. They conclude that Euclidean metrics (straight lines) form a sufficient approximation to geodesics in the latent space, which is rather contradictory with the findings of our article [2]. [9] do concede that the dataset of real images they used was very limited.

In [10], the geometry of latent spaces is linked to the level of disentanglement in the generative model's representation of the data. Specifically, the authors find that in more disentangled VAEs, the latent spaces exhibit higher curvature than in traditional VAEs. [10] also finds that using Riemannian metrics in the latent space vastly improves distance measures and interpolation.

[4] independently worked on the same topic as our article [2], and proposed a smooth approximation of geodesics

on the data manifold. They focus their experiments on applications to robotics, specifically path learning.

Finally, [6] study optimal transport [11] in the context of GANS and show that (1) the discriminator of a GAN computes the Wasserstein distance via the Kantorovich potential, and (2) the generator calculates the transportation map. They use these insights to develop an alternative to Wasserstein GANs [1].

After this overview of existing research on the geometry of latent spaces, we present in the following section some background knowledge required to understand [2].

3. Variational Autoencoders and latent spaces

Variational Autoencoders (VAEs) are a type of generative model: they can generate new data samples that are similar to a training dataset. VAEs work by learning a compact, continuous representation of a training dataset in an encoding space, and then using this representation to generate new data samples. This is possible thanks to the "manifold hypothesis", which states that datasets generally lie on a manifold of lower dimension than the input space.

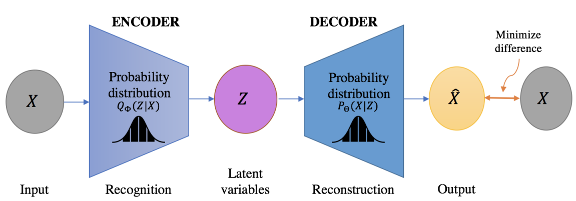

Figure 1: Architecture of Variational Autoencoer

The basic structure of a VAE (see figure above 1) consists of an encoder and a decoder (also called generator). The encoder takes in data samples and maps them to a lower-dimensional encoding (latent variables) in the latent space. The decoder maps these latent variables to the original data space to reconstruct the original data sample. The latent space must respect two properties for the VAE's generation to be possible : continuity (two points close in the latent space must not give completely different outputs) and completeness (a point sampled from a latent space must give meaningful content).

VAEs are trained by minimizing the reconstruction error between the original data samples and their reconstructions, while also maximizing the compactness of the encoding space. The following is the VAE loss function, which is derived using variational inference:

$$\theta^*, \phi = \text{argmax } \mathbb{E}_{q_\phi(z|x)}[log(p_\theta(x|z))] - \text{KL}[q_\phi(z|x)||p(z)]$$

4. The Riemannian metric

In order to compute meaningful distances in the latent space of deep generative models, [2] proposes a stochastic Riemannian metric. The article first introduces a deterministic metric. Considering a smooth latent curve $\gamma_t : [0,1] \rightarrow \mathcal{Z}$, we can map it through a function f to compute lengths in the input space of the VAE. The length of a curve $f(\gamma_t)$ is then equal to:

$$||\frac{\partial f}{\partial z}|_{z=\gamma}\dot{\gamma}|| = ||\mathbf{J}_\gamma \dot{\gamma}|| = \sqrt{\dot{\gamma}^\intercal(\mathbf{J}_\gamma^\intercal \mathbf{J}_\gamma)\dot{\gamma}} = \sqrt{\dot{\gamma}^\intercal \mathbf{M}_\gamma \dot{\gamma}}$$

Here, $\mathbf{M}_\gamma = \mathbf{J}_\gamma^\intercal \mathbf{J}_\gamma$ is a symmetric positive definite matrix which acts similarly to a Mahalanobis distance measure. Mahalanobis distance measures the distance between a point P and a distribution Q [7] . The distance can be written as : $d_M(\vec{x}, Q) = \sqrt{(\vec{x}-\vec{\mu})^\intercal S^{-1}(\vec{x}-\vec{\mu})}$ with $\vec{\mu} = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^\intercal$ the mean of the distribution. If the generator f is sufficiently smooth, then $\mathbf{M}_\gamma$ defines a Riemannian metric.

We must now adapt the metric to stochastic generators (such as a VAE decoder) defined as follows:

$$f(z) = \mu(\mathbf{z}) + \sigma(\mathbf{z}) \cdot \epsilon \quad \mu : (Z) \rightarrow \mathcal{X}, \ \sigma : \mathcal{Z} \rightarrow \mathbb{R}_+^D$$

where $\epsilon \sim \mathcal{N}(0, I_D)$. The theorem below helps develop the stochastic metric for this model.

Theorem 1: If the functions μ and σ are at least twice differentiable, then the expected metric equals:

$$\bar{\mathbf{M}}_z = \mathbb{E}_{p(\epsilon)}[\mathbf{M}_z] = (\mathbf{J}_z^{(\mu)})^\intercal \mathbf{J}_z^{(\mu)} + (\mathbf{J}_z^{(\sigma)})^\intercal \mathbf{J}_z^{(\sigma)} \qquad (1)$$

where $\mathbf{J}_z^{(\mu)}$ and $\mathbf{J}_z^{(\sigma)}$ are the Jacobian matrices of $\mu(\cdot)$ and $\sigma(\cdot)$.

We can thus approximate $\mathbf{M}_z$ using $\bar{\mathbf{M}}_z$. This estimate is more precise when the data dimensionality D is large. One advantage of this metric is that no additional learning is required, it can entirely be derived from an existing generator.

Theorem 1 can be proven as follows. Firstly, we need to compute the Jacobian matrix of the generator:

$$\frac{\partial f(z)}{\partial z} = \mathbf{J}_z = \begin{pmatrix} \frac{\partial f_z^{(1)}}{\partial z_1} & \cdots & \frac{\partial f_z^{(1)}}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_z^{(D)}}{\partial z_1} & \cdots & \frac{\partial f_z^{(D)}}{\partial z_d} \end{pmatrix}$$

$$\iff \mathbf{A} + \mathbf{B} = \begin{pmatrix} \frac{\partial \mu_z^{(1)}}{\partial z_1} & \cdots & \frac{\partial \mu_z^{(1)}}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mu_z^{(D)}}{\partial z_1} & \cdots & \frac{\partial \mu_z^{(D)}}{\partial z_d} \end{pmatrix} + [\mathbf{S}_1\epsilon, \cdots, \mathbf{S}_d\epsilon]_{D\times d}$$

where $\mathbf{S}_i = \begin{pmatrix} \frac{\partial \sigma_z^{(1)}}{\partial z_i} & 0 & \cdots & 0 \\ 0 & \frac{\partial \sigma_z^{(2)}}{\partial z_i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial \sigma_z^{(D)}}{\partial z_i} \end{pmatrix}$ for $i = 1, ..., d$

The new stochastic metric in the latent space is $\mathbf{M}_z = \mathbf{J}_z^\intercal \mathbf{J}_z$: its randomness is due to the random variable ϵ. We can then compute the expected metric :

$$\bar{\mathbf{M}}_z = \mathbb{E}_{p(\epsilon}[\mathbf{M}_z] = \mathbb{E}_{p(\epsilon)}[(\mathbf{A}+\mathbf{B})^\intercal(\mathbf{A}+\mathbf{B})]$$
$$\bar{\mathbf{M}}_z = \mathbb{E}_{p(\epsilon)}[\mathbf{A}^\intercal\mathbf{A} + \mathbf{A}^\intercal\mathbf{B} + \mathbf{B}^\intercal\mathbf{A} + \mathbf{B}^\intercal\mathbf{B})$$

Then:

$$\mathbb{E}_{p(\epsilon)}[\mathbf{A}^\intercal\mathbf{B}] = \mathbb{E}_{p(\epsilon)}[\mathbf{A}^\intercal[\mathbf{S}_1\epsilon, \cdots, \mathbf{S}_d\epsilon]] = \mathbf{A}^\intercal[\mathbf{S}_1\cancel{\mathbb{E}_{p(\epsilon)}[\epsilon]}, ..., 0]$$

Because $\mathbb{E}_{p(\epsilon)}[\epsilon] = 0$ and:

$$\mathbb{E}_{p(\epsilon)}[\mathbf{B}^\intercal\mathbf{B}] = \mathbb{E}_{p(\epsilon)}\left[\begin{pmatrix} \epsilon^\intercal\mathbf{S}_1\mathbf{S}_1\epsilon & \cdots & \epsilon^\intercal\mathbf{S}_1\mathbf{S}_d\epsilon \\ \vdots & \ddots & \vdots \\ \epsilon^\intercal\mathbf{S}_d\mathbf{S}_1\epsilon & \cdots & \epsilon^\intercal\mathbf{S}_d\mathbf{S}_d\epsilon \end{pmatrix}\right]$$

Also, we have that $\mathbb{E}_{p(\epsilon)}[\epsilon^\intercal\mathbf{S}_i\mathbf{S}_j\epsilon]$

$$= \mathbb{E}_{p(\epsilon)}\left[\begin{pmatrix} \epsilon_1\frac{\partial \sigma_z^{(1)}}{\partial z_i} & \cdots & \epsilon_D\frac{\partial \sigma_z^{(D)}}{\partial z_i} \end{pmatrix}\begin{pmatrix} \epsilon_1\frac{\partial \sigma_z^{(1)}}{\partial z_j} \\ \vdots \\ \epsilon_D\frac{\partial \sigma_z^{(D)}}{\partial z_j} \end{pmatrix}\right]$$
$$= diag(\mathbf{S}_i)^\intercal diag(\mathbf{S}_j)$$

since $\mathbb{E}_{p(\epsilon)}[\epsilon^2] = 1$. Finally, with $\mathbf{A} = \mathbf{J}_z^{(\mu)}$ and

$$J_z^{(\sigma)} = \begin{pmatrix} \frac{\partial \sigma_z^{(1)}}{\partial z_1} & \cdots & \frac{\partial \sigma_z^{(1)}}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial \sigma_z^{(D)}}{\partial z_1} & \cdots & \frac{\partial \sigma_z^{(D)}}{\partial z_d} \end{pmatrix}$$

It becomes clear that $\mathbb{E}_{p(\epsilon)}[\mathbf{B}^\intercal\mathbf{B}] = (\mathbf{J}_z^{(\sigma)})^\intercal\mathbf{J}_z^{(\sigma)}$. Therefore, the expectation of the induced Riemannian metric in the latent space by the generator is:

$$\bar{\mathbf{M}}_z = (\mathbf{J}_z^{(\mu)})^\intercal\mathbf{J}_z^{(\mu)} + (\mathbf{J}_z^{(\sigma)})^\intercal\mathbf{J}_z^{(\sigma)}$$

which proves Theorem 1! To establish $\mathbf{M}_z$ as a metric, it must verify the following properties :

- **Non-negativity**: The distance between any two points must be non-negative, i.e., $d(p, q) \geq 0$ for all points p and q on the manifold.
- **Identity of indiscernibles**: The distance between any two points is zero if and only if the points are the same, i.e., $d(p, q) = 0$ if and only if $p = q$.
- **Symmetry**: The distance between two points is the same in either direction, i.e., $d(p, q) = d(q, p)$.
- **Triangle inequality**: The distance between two points is no greater than the sum of the distances between intermediate points, i.e., $d(p, q) \leq d(p, r) + d(r, q)$ for any intermediate point r.

Since $\mathbf{M}_z$ is a symmetric positive semi-definitie matrix, it verifies non-negativity, identity of indiscernables, symmetry and triangle inequality. Therefore, the Riemannian distance measure proposed by [2] is indeed a metric.

The pillar of all the derivations in this section is the assumption that the metric tensor changes smoothly, i.e. that the Jacobians themselves are smooth functions. The authors of [2] state this can be ensured by using activation functions in the generator networks that are C^2 differentiable (tanh($\cdot$), sigmoid($\cdot$), and softplus($\cdot$)). This does exclude common activation functions such as ReLu or Binary Step, which tend to perform well but cannot be used in the generator if one intends to use a Riemannian metric in the latent space.

5. Meaningful Variance Functions

Theorem 1 tells us that the Riemannian metric, and thus the geometry of our generator, depends on both the mean and variance networks. The variance function is a neural network, and unfortunately, neural nets tend do extrapolate very poorly to unseen data. This means that variance estimates of our generator will be of low quality outside the support of the training data distribution.

To overcome this, [2] propose to model the precision (inverse varince) $\beta_\phi(z) = \frac{1}{\sigma_\phi^2(z)}$ with a model that extrapolates towards 0, so that variance in unseen regions tends to infinity. The chosen model is a Radial Basis Function (RBF) network [8], defined as follows:

$$\beta_\phi(z) = W \cdot v(z) + \zeta, \text{ with } v_k(z) = \exp\left(-\lambda_k ||z - c_k||_2^2\right) \tag{2}$$

for $k = 1, ..., K$, with ϕ the parameters of the model. ζ is simply a vector of very small positive constants which guarantees we are never dividing by zero. $W \in \mathbb{R}^{D\times K}$ are the weights of the network (all positive), c_k and λ_k respectively designate the centres and the bandwih of the K radial basis functions. Training the above RBF model results in much more reliable variance estimates for the generator, and thus more precise estimates of the Riemannian metric.

6. Results of the paper

This part of the paper demonstrate the benefits of using their variance network along with a Riemannian metric in the latent space with several experiments:

- **Meaningful distances:**

  Using the MNIST dataset for a clustering task, [2] compares results obtained under both metrics. The Riemannian metric performs a lot better (around 90% accuracy compared to 70% for Euclidean distances)

- **Interpolations:**

  Again, when investigating interpolations, the Riemannian metric gives a smoother transition from zeros to ones (when trained on part of the MNIST dataset).

- **Latent probability distributions:**

  Instead of a classical normal distribution for data sampling in the latent space, we can use the locally adaptive normal distribution (LAND distribution) which benefits from using a Riemannian extension of the Mahalanobis distance. The results are again conclusive on the benefit of using Riemannian distances.

- **Random walk on the data manifold:**

  The Riemannian metric also benefits the model in terms of generation. Indeed, when exploring the latent space with random walk with a Euclidian distance, the model wonders beyond the data support and thus generates poor images. With the Riemannian metric, the randon walk naturally remains in the support of the data distribution. This can be explained by the presence of a "barrier variance" when using the enhances variance network and Riemannian metric.

Overall, these experiments highlight the significant improvements resulting from the use of a Riemannian metric in the latent spaces of deep generative models, as well as a more accurate variance network.

7. Our Experiments

7.1. Evaluation of the proposed VAE architecture

The VAE architecture used in the codebase of [2] is not standard. We first sought to evaluate this new architecture's performance and compare it to that of a standard VAE. The proposed architecture is represented in Figure 2 below:

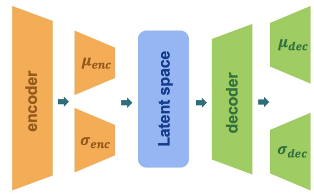

Figure 2: Diagram of the VAE architecture used in [2]

The encoder has a few common layers but then splits into two networks, one to predict the mean of the latent space distributions, and one for the variance. Then, the generator takes in a sample z from the latent space and tries to predict the data distribution $\mathcal{N}(\mu_{dec}(z), \sigma_{dec}(z))$, where a standard VAE would simply predict the reconstruction x. This technique thus assumes that the data follows a Gaussian distribution in the input space. We compare reconstructed images with that of a standard VAE in Figure 3:

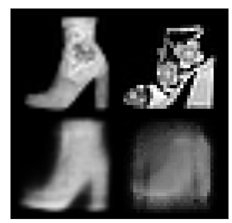

Figure 3: Reconstruction of a shoe from Fashion-MNIST using (left) a standard VAE, and (right) the VAE architecture of [2]. Both networks have the same depth and layer widths where possible, as well as the same activation function, to ensure a fair comparison.

The proposed architecture performs less well on Fashion-MNIST. The reconstructed samples tend to mix different classes of clothes (here, a shoe and a shirt). This effect is reduced when we sample close to the mean of the decoder distribution $\mathcal{N}(\mu_{dec}(z), \sigma_{dec}(z))$. We explain this lower performance as follows:

- Firstly, as we saw in Dr. Feydy's "Geometric Data Analysis" course, the Gaussian assumption for the data distribution is rather unrealistic, it is very unlikely that the manifold on which the data lie in the input space resembles a Gaussian distribution.

- Secondly, most VAEs simply use one network for the generator, which predicts a reconstruction in the input space from a latent space sample. Here, the VAE predicts distribution parameters (mean and variance). Learning distribution parameters is harder than learning to reconstuct a sample, which probably contributes negatively to the performance.

- Finally, this is not something we have backed with theory (there is no literature about it), but we believe the two-network generator could be trained slightly differently. Presently, the generator networks take in a sample from the latent space to predict distribution parameters in the input space. We believe it might be interesting to experiment with feeding the generator the latent distribution parameters instead of a sample, since it is not reconstructing a data point but predicting data distribution parameters.

From a few experiments on the Fashion-MNIST dataset, we observe that the proposed VAE architecture of [2] is less performant than a standard VAE.

7.2. Latent space manifolds

In order to test the strengths and weaknesses of the proposed Riemannian metric (and the reproduciability of the paper's original experiments), we decided to conduct the following experiment. We first train a VAE and apply the RBF variance network on the Fashion-MNIST dataset, to then examine the latent space manifold and its curvature. We chose this dataset as it seemed like the natural next step (in terms of complexity) after the MNIST dataset used in the orginal paper. Also, it is still simple enough to run experiments on our personal computers with a reasonably sized model.

The VAE architecture used is that proposed in [2], as the codebase provided to compute manifolds and geodesics is unfortunately not compatible with other structures. This is one of the limits of this paper, the provided code is not flexible at all. Further, it is not trivial to adapt the manifold and geodesic computation to new architectures: the algorithm used is quite complex, the code is not commented, and the algorithm is only briefly described in [3].

After training the VAE on Fashion-MNIST, the following data manifold is obtained:

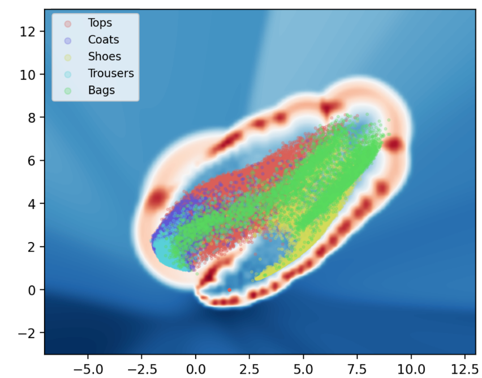

Figure 4: Data manifold for Fashion-MNIST. The colorscale (blue to red) indicates the local curvature of the manifold. The scatter points represent projections of images in the latent space.

The input data is projected as points on this manifold, colored by class (red: tops, purple: coats, green: bags, yellow-green: shoes, sky-blue: trousers). Coloring the data by class shows us why reconstructed images sometimes mix classes: classes overlap on the learned data manifold, so different points of the input space get mapped to the same region in the latent space. We expect a better model (with a different architecture, perhaps deeper/wider or convolutional) to have more distinctively separated classes.

Nonetheless, this manifold still shows very interesting structure in the botoom left of the image: while the curvature of the manifold is mostly homogenous on the support of the data, the area separating the shoe class and the trouser/tops classes exhibits very high curvature (shown in red in Figure 4). This can be interpreted as the model's understanding that a shoe has very different structure to trousers, trousers, or tops. This is then traduced by a much longer geodesic to go from a shoe to a trouser than from a trouser to a t-shirt. Already, this simple observation highlights the importance of the proposed Riemannian metric: it allows to draw semantic meaning from distances and structure in the latent space.

In this example, a Euclidian distance would fail to capture how different a shoe is from a trouser vs. a trouser from a t-shirt. Indeed, ignoring the curvature, taking a straight line between a shoe from the bottom left and a trouser would yield a shorter distance than between a trouser and some t-shirts. This obviously is semantically wrong, highlighting the fact that Euclidian distances are inadequate to describe latent spaces, and the Riemannian metric proposed in [2] constitutes a much better description of the low-dimensional data-manifold.

7.3. Latent space interpolations

To further illustrate the adequacy of the Riemannian metric, we perform the following experience. Taking a point A from the shoe class and a point B from the trouser class, we compute the shortest path between A and B using both Euclidian distance (shown in green in Figure 5) and the Riemannian metric (shown in red).

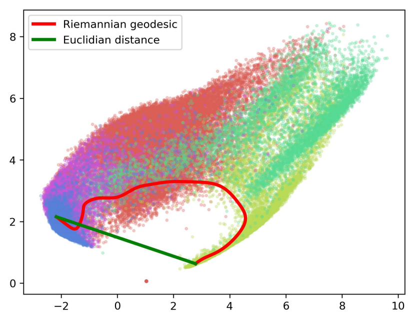

Figure 5: Shortest paths between a shoe and a trouser on the data manifold using Euclidian distance (green) and a Riemannian geodesic (red).

On figure 5 we see that the Euclidian path goes straight to the region of the shoe class which contains the point, while the geodesic follows a path along the edge of the shoe class, transitioning more smoothly to the destination. To verify this, we have performed interpolations in the latent space (from the trouser to the shoe) under both metrics, and obtained the following images:

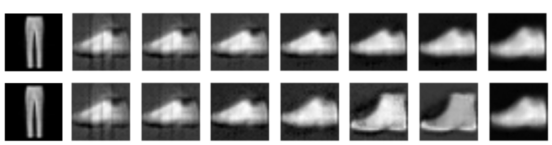

Figure 6: Sample images obtained by interpolating between a trouser and a shoe in the latent space. The top row contains the results for the Euclidian distance, while the bottom row contains results for the Riemannian geodesic.

Figure 6 shows that under both metrics, the interpolation is dominated by the shoe class (the trouser sdisappear after 3 images). Nonetheless, the transition is smoother under the Riemannian geodesic. Indeed, it transitions from a low-top sneaker to an ankle boot to a high-top sneaker, whereas the Euclidian metric shows a low-top sneaker for 6 of the 8 images and abruptly becomes a high-top sneaker in the alst image. This is coherent with what was stated earlier: the Riemnanian geodesic goes along a the edge of the shoe class (thus exploring more of the shoe class), while the Euclidian path goes through a data 'void' and appears at the desired shoe.

We can summarize the take-aways of these experiments as follows. On one hand, the Riemmanian metric offers key insights into the geomtry of latent spaces and without doubt has the potential to improve generative modelling. On the other hand, our experiments show that the VAE structure used by the authors of [2] is suboptimal and hinders the applicability of this paper. Finally, the codebase of [2] needs to be packaged more rigourously, as it is not complete (very little code was available for the expriments we conducted - only the manifold plotting functions); and its flexibility should also be improved to accomodate more model architectures.

8. Limitations of the paper

While this paper provides crucial insights into the geometry of latent spaces and contains detailed theoretical work backed by experimental results, we indentify a few weaknesses in the presented resarch.

- Firstly, while the findings of this paper are intended to be applicable to all generative models, the theoretical derivations are tightly linked to the use of a VAE. This made us wonder if one could find such a neat expression for the stochastic Riemannian metric in a different generative model, such as GANs or score-based models. On the upside, the proposed RBF variance network does seem model-agnostic, so long as the original model contains a variance netowrk.

- As observed in our experiments, the proposed VAE architecture is non-standard and tends to perform less well than a standard VAE. We attribute this to an unrealistic assumption that data is Gaussian distributed, and to the generator's two network structure. We also hypothesize that the VAE training used is not optimal.

- The paper provides a variety of experiments to demonstrate the superiority of the Riemannian metric over Euclidian distances, but these all take place on toy datasets. While the results are conclusive, we believe it is important to test these methods on more complicated challenges, which will highlight if the method scales to larger datasets, higher-dimensional data, and more complicated latent spaces. This is what motivated us to conduct experiments on Fashion-MNIST, a (slightly) more complicated dataset than MNIST. It is especially important to test this paper's contributions on more complicated problems because of the modifications of the standard VAE model. As we highlighted previously, this modified VAE has some issues, which might cause it to perform worse than a standard VAE in real-life applications. This would make the research presented here less useful, as even if the modified VAE with the Riemannian metric performs as well as the standard VAE in the end, one should choose the simpler option (Occam's razor), i.e. use a standard VAE.

- We have not been able to verify this, but the random-walk experiment described in [2] leads us to believe that it is possible that the Riemannian metric sometimes bounds exploration in the latent space too harshly, which might restrict the diversity of the reconstructed images. This might be problematic in applications such as AI in art: we don't want to be too close to the training data, and would like to explore the tails of the data distribution to produce original and 'creative' pieces of art.
- Also, we would like to highlight that the smoothness requirement for the generator is rather vague. There is no quantitative evaluation of smoothness, and we think it would be key to have more detailed insights into how choices of network architecture, activation functions etc. impact the smoothness of the network. This is crucial, as the authors state that insufficient smoothness breaks the metric property of the proposed Riemannian metric.
- Finally, and we believe this is the main weakness of the paper, the research is only partly reproducible, greatly limiting its applicability. The quality of the codebase is poor, we summarize its problems as follows. Firtly, the code is not flexible and can only accomodate one VAE architecture. Secondly, the code required to reproduce experiments of the paper is not readily available: only RBF training and manifold plotting functions are implemented. Thirdly, the code is not documented at all and is very sparsely commented. This is a shame, as packaging this code properly could make the use of Riemannian metrics in latent spaces much more widespread amongst practicioners.

9. Conclusion

In conclusion, the paper "Latent Space Oddity: On the curvature of deep generative models" presents a new stochastic Riemannian metric and variance network to understand the geometry of latent spaces and to improve the performance of deep generative models. The proposed metric allows for more accurate measurement of distances in latent spaces and can be derived from an existing generator without additional learning. The proposed variance network is far more accurate than the standard one, and correctly extrapolates to infity outside the data support. The improvements in generative modelling brought by this research have been confirmed by numerous experiments conducted by the authors of [2] as well as our personal experiments.

However, it is important to note that this work has some limitations. The Riemannian metric proposed in the paper is based on smoothness assumptions which are loosely formulated and may not always hold in practice. Additionally, some of the mathematical derivations and proposed models/algorithms seem specific to VAEs, and not universally applicable to all generative models. Finally, the VAE structure used in this paper is not standard and seems suboptimal based on our experiments.

An interesting line of research would be to try to generalise the results of [2] to more generative models, and not be reliant on a specific VAE architecture. This would greatly broaden the applicability of the presented work. A more pragmatic line of future work is to make this piece of research reproducible. The code needs to be packaged in a more user-friendly way, and must be reformatted in order to be more flexible (it must be able to accomodate more variety in the model architectures) and scalable. This will make the research in [2] usable by machine learning practicioners and more applicable to real-world problems.

Despite these limitations, the proposed metric and variance network offer a promising approach for understanding and improving the performance of deep generative models. Understanding the geometric structure of latent spaces can provide valuable insights into the properties of generative models and how they model the data. The potential applications of this work are broad, as it can be applied to any learning task that involves handling variables in a latent space.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 2

[2] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models, 2017. 1, 2, 3, 4, 5, 6, 7

[3] Georgios Arvanitidis, Soren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1506–1515. PMLR, 2019. 5

[4] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick Smagt. Metrics for deep generative models. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 2018. 1

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and

Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[6] Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng David Gu. A geometric view of optimal transportation and generative model. *Computer Aided Geometric Design*, 68:1–21, 2019. 2

[7] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. page 49–55. Proceedings of the National Institute of Sciences of India, 1936. 2

[8] Qichao Que and Mikhail Belkin. Back to the future: Radial basis function networks revisited. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, 09–11 May 2016. 3

[9] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 1

[10] Ankita Shukla, Shagun Uppal, Sarthak Bhagat, Saket Anand, and Pavan Turaga. Geometry of deep generative models for disentangled representations. ICVGIP 2018. Association for Computing Machinery, 2020. 1

[11] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. 2