

# Fast Approximate Geodesics for Deep Generative Models

Nutan Chen<sup>1</sup>, Francesco Ferroni<sup>2</sup>, Alexej Klushyn<sup>1</sup>,  
Alexandros Paraschos<sup>1</sup>, Justin Bayer<sup>1</sup>, Patrick van der Smagt<sup>1</sup>

Machine Learning Research Lab, Volkswagen Group<sup>1</sup>      Autonomous Intelligent Driving GmbH<sup>2</sup>

## Abstract

*The length of the geodesic between two data points along the Riemannian manifold, induced by a deep generative model, yields a principled measure of similarity. Applications have so far been limited to low-dimensional latent spaces, as the method is computationally demanding: it constitutes to solving a non-convex optimisation problem. Our approach is to tackle a relaxation: finding shortest paths in a finite graph of samples from the aggregate approximate posterior can be solved exactly, at greatly reduced runtime, and without notable loss in quality. The method is hence applicable to high-dimensional problems in the visual domain. We validate the approach empirically on a series of experiments using variational autoencoders applied to image data, tackling the Chair, Faces and FashionMNIST data sets.*

## 1. Introduction

Estimating the similarity between data points often constitutes a central part of any data processing pipeline. In computer vision it is relevant in matching points in from frames [31] or visual place recognition. In the latter case, [27] mention one particular challenge: a place’ visual appearance can change drastically within a short time frame, e.g., through weather conditions or whether a picture was taken during the night or the day time. For a method to be successful, certain invariances have to either be used as an inductive bias or presented through data. This asks for highly expressive models.

Recently, deep learning has made training generative models on large-scale databases, as typically found in computer vision, practical [7, 21]. Such models have been used for similarity estimation from the perspective of Riemannian manifolds in the context of Gaussian process latent variable models [32]. As such non-parametric approaches scale poorly with data set size, several authors [9, 10, 2] proposed the marriage of Riemannian manifold-based metric learning with deep generative models. This results in a principled measure of similarity between two members of

the data distribution by relating it to the shortest path, or, geodesic between two corresponding points in latent space. A downside of this approach is that the geodesic has to be obtained as the solution to a non-linear optimisation problem. Consequently, there is high computational demand and no guarantee for the quality of the solution. The representation of this shortest path is also not obvious, as it is continuous in nature but ultimately has to be represented in a discrete fashion.

In contrast, local and global descriptors [26] do not suffer from said computational challenges. As their representation is a vector over real numbers, it is compactly represented as an array of floating point numbers, paving the way for efficient indexing techniques for nearest neighbour lookup [5, 20].

This work takes a step in making Riemannian manifold based approaches faster, and hence applicable in high-dimensional settings. We will see that spanning the latent space of a latent variable model with a discrete and finite graph allows us to apply a classic search algorithm, A\*. This lets us obtain accurate approximations of the geodesic. These are superior to the previously proposed ODE and neural network based approaches, in terms of computation performance, without loss in quality. Once the graph has been built, estimating the geodesic is  $\mathcal{O}(\text{number of nodes})$  and hence bounded for any pair of points.

We apply the proposed framework to a toy example, that of a visual pendulum, to foster intuition of the approach, as it can be easily visualised. The practical applicability to more challenging data sets is then illustrated for the Chair, Faces, and FashionMNIST data sets.

## 2. Related work

A wide range of approaches have been proposed for estimating the similarity between data points. Typically, distance metrics are assumed to be given, however they often come with certain assumptions about the data. For example instances of the Minkowski distance require the data to be invariant to rotation under the  $L_2$ -norm. The Mahalanobis distance, is a popular choice when the data are multivariate Gaussian distributed, as it is invariant to translation and

scaling.

Transforming the data can further allow to apply known metrics, even when the data does not directly fulfil the assumptions required by the metric. In [34, 17], the authors proposed the use of linear transformations for supervised learning. To enable an accurate measurement of even more complicated data, non-linear transformations based on neural networks were introduced in [30]. Additionally, transformations of time-series data to constant-length spaces have been proposed in [4], which allow to apply similarity measures using recurrent neural networks.

To alleviate the problem of manually specifying a distance metric, learning distances directly from data have been proposed by several approaches [36, 33, 12, 23]. That is especially useful in high-dimensional spaces, where obtaining a meaningful distance metric is challenging. Traditional metrics may not even be qualitatively meaningful, since the ratio of the distances of the nearest and farthest neighbours to a given data point is almost 1 for a wide variety of distributions [1].

In recent work, [32] suggested perceiving the latent space of Gaussian process latent variable models as a Riemannian manifold, where the distance between two data points is given as the shortest path along the data manifold. Treating the latent space as a Riemannian manifold enables to use interpolation [28] and trajectory-generation [11] algorithms between given points, with the advantage that the observable-space trajectory remains sufficiently close to the previously used training data [9, 10, 2, 24]. The geodesics, i.e. the length-minimising curve given the curvature of the manifold, have been approximated by neural networks [9, 10] or represented by ODEs [2].

Beside interpolation, using the geometry of the manifold has been proposed and used in other approaches based on generative models. For instance, in [14] the authors used task manifolds for meta-learning and in [18] the authors developed a constant-curvature manifold prior for adversarial autoencoders used for graph data.

### 3. Methods

#### 3.1. Variational autoencoders

Latent-variable models (LVMs), defined by

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \quad (1)$$

represent observable data  $\mathbf{x} \in \mathbb{R}^{N_x}$  through latent variables  $\mathbf{z} \in \mathbb{R}^{N_z}$ , which are based on hidden characteristics in  $\mathbf{x}$ .

The integral in Eq. (1) is usually intractable and has to be approximated through sampling [19, 15] or variational inference [22, 29]. In the latter case, the problem is rephrased

as the maximisation of the evidence lower bound (ELBO):

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z})} \left[ \ln \frac{p(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q(\mathbf{z})} \right] = \mathcal{L}_{\text{ELBO}}, \quad (2)$$

where  $p(\mathbf{x}|\mathbf{z})$  is the likelihood,  $p(\mathbf{z})$  the prior, and  $q(\mathbf{z})$  approximates the intractable posterior. The distribution parameters of  $q(\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{x}|\mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})$  can be expressed by neural networks—the encoder and the decoder, which are parameterised through  $\phi$  and  $\theta$ , respectively. By this means we obtain a variational autoencoder (VAE) [22, 29].

To overcome the limitations of ordinary VAEs, we use importance-weighted autoencoders (IWAEs) [8] in our experiments. IWAEs achieve a tighter ELBO through importance sampling:

$$\mathcal{L}_{\text{IWAE}} = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[ \ln \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(\mathbf{x}|\mathbf{z}_k) p_\theta(\mathbf{z}_k)}{q_\phi(\mathbf{z}_k|\mathbf{x})} \right], \quad (3)$$

where  $\ln p(\mathbf{x}) \geq \mathcal{L}_{\text{IWAE}} \geq \mathcal{L}_{\text{ELBO}}$  [8].

#### 3.2. Riemannian geometry in variational autoencoders

The Riemannian space is a differentiable manifold  $M$  with an additional metric to describe its geometric properties. This enables assigning an inner product in the tangent space to each point  $\mathbf{z}$  in the latent space through the corresponding metric tensor  $\mathbf{G}$ :

$$\langle \mathbf{z}', \mathbf{z}' \rangle_{\mathbf{z}} := \mathbf{z}'^T \mathbf{G}(\mathbf{z}) \mathbf{z}', \quad (4)$$

with  $\mathbf{z}' \in T_{\mathbf{z}} M$  and  $\mathbf{z} \in M$ .

Treating the latent space of a VAE as a Riemannian manifold allows us to compute the *observation space distance* of latent variables. Assuming we have a trajectory  $\gamma : [0, 1] \rightarrow \mathbb{R}^{N_z}$  in the Riemannian (latent) space that is transformed by a continuous function  $f(\gamma(t))$  (decoder) to an  $N_x$ -dimensional Euclidean (observation) space. The length of this trajectory in the observation space, referred to as the Riemannian distance, is defined by

$$L(\gamma) = \int_0^1 \phi(t) dt, \quad (5)$$

with the Riemannian velocity

$$\phi(t) = \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}}, \quad (6)$$

where  $\dot{\gamma}(t)$  is the time derivative. The metric tensor is defined as  $\mathbf{G} = \mathbf{J}^T \mathbf{J}$ , with  $\mathbf{J}$  as the Jacobian of the decoder. The trajectory which minimises the Riemannian distance  $L(\gamma)$  is referred to as the geodesic.

### 3.3. Graph-based geodesics

Computing the geodesic is a challenging task, since Eq. (6) has to be minimised. The main issue lies in the necessity to use the Hessian of the decoder during the optimisation process. It is a time consuming optimisation procedure, which scales poorly with the dimensionality of the observable and the latent space, and hence is not feasible for a lot of applications. In addition, the Hessian has limitations of the selection of the neural network’s activation functions [2, 9]. To bypass the above-mentioned hurdles we introduce a graph-based approach, where a discrete and finite graph is built in the latent space using a binary tree data structure, k-d tree, with edge weights based on Riemannian distances. Once the graph has been built, geodesics can be approximated by applying a classic search algorithm, A\* [13].

#### 3.3.1 Building the graph

The graph is structured as a k-d tree, a special case of binary space partitioning trees, where each leaf node corresponds to a k-dimensional vector.

The nodes of the graph are obtained by encoding the observable data  $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$  into their latent representations  $\mathbf{z}^{(i)}$ . This is done by using the respective mean values of the approximate posterior  $q_\phi(\mathbf{z}^{(i)}|\mathbf{x}^{(i)})$ . Each node is connected by a undirected edge to its k-nearest neighbours. The edge weights are Riemannian distances  $L(\gamma)$ , where  $\gamma$  is the shortest Euclidean path between the related pair of nodes.

#### 3.3.2 Approximating geodesics

A classic graph-traversing method to obtain the shortest path between nodes is A\* search. It is an iterative algorithm that, given a graph  $\mathcal{G}$ , maintains a sorted list of nodes that can be visited in the current state. The list is typically initialised with the starting node and is being sorted according to the estimated cost of including node  $n$  into the optimal path. The estimated cost is computed by

$$f(n) = g(n) + h(n), \quad (7)$$

where  $g(n)$  is the cost of the path from the starting node  $n_{\text{start}}$  to  $n$  and  $h(n)$  is a heuristic function that estimates the remaining cost from  $n$  to the target node  $n_{\text{target}}$ .

The cost function we use in our approach is the Riemannian distance between two subsequent nodes on the path, whereas the Euclidean distance is used as heuristic. A\*, in order to operate, requires a heuristic function that underestimates the true cost. It can be shown that the proposed heuristic fulfills this requirement. The performance of the algorithm is optimal among any other similar algorithm to the number of nodes that are being expanded. When the target node is reached, the algorithm terminates.

The result is the shortest path through the graph regarding the Riemannian distance. This path approximates the geodesic well as shown in Section 4.

## 4. Results

We present an empirical evaluation of our graph-based approach for approximating geodesics in deep generative models. First, we compare the graph-based approximation to a neuronal network (NN)-based method, that has been proposed in [9]. We proceed by evaluating the our approach in high latent-dimensional spaces on a synthetic dataset of pendulum images, fashion MNIST, chairs, faces and human motion datasets.

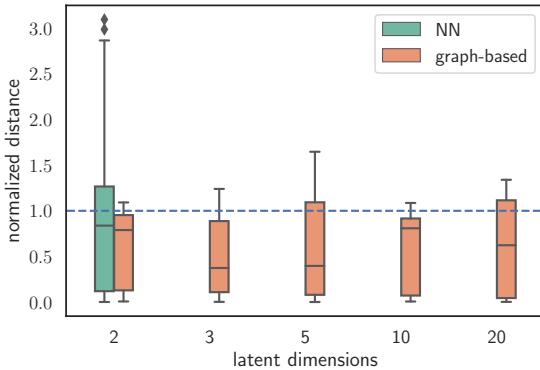
We use the magnification factor (MF) [6] to observe the generative models in 2D latent space and evaluate the approximated geodesic. The  $\text{MF}(\mathbf{z}) =: \sqrt{\det \mathbf{G}(\mathbf{z})}$  can be interpreted as the scaling factor when moving from the Riemannian (latent) to the Euclidean (observation) space, due to the change of variables.

### 4.1. Pendulum

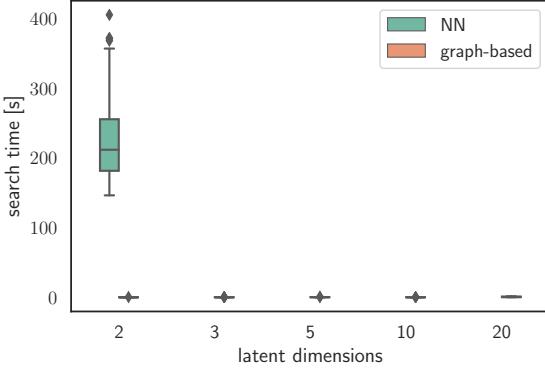
The pendulum dataset contains  $16 \times 16$ -pixel images of a simulated pendulum, used as input to our algorithm. We gathered an image dataset by collecting  $T = 15 \cdot 10^3$  images for two different joint angle ranges,  $R_1 = [0, 150]$  and  $R_2 = [180, 330]$  degrees. Subsequently, we augmented the dataset by adding 0.05 Gaussian noise to each pixel, to avoid overfitting and to improve the coherence of the latent space.

We present our results using 2, 3, 5, 10 and 20 latent dimensions for the IWAE and used 15 samples for the importance weighting step. After training, 1000 points were chosen to build the graph. Each node had four nearest neighbours based on the distance in the latent space. We generated 100 random pairs of data points, as show in Fig. 1, to compute the distances and the searching time. We show that with the increasing latent dimensionality, the search time does not increase, as it is dependent solely to the number of nodes.

Two of the generated geodesic and Euclidean interpolant trajectories of two latent dimensions are illustrated in Fig. 2, with Fig. 3 showing their reconstructions using the decoder. For  $A_1A_2$ , our graph-based search finds a trajectory in the latent space which is significantly longer than a simple Euclidean interpolation. However, this is because at approximately half-way in the Euclidean interpolant path, there is a region of large magnification factor, and consequently large change in the visual output. This is shown in the 9th frame in Fig. 3 where, for the Euclidean version, two different pendulums appear at the same time. On the other hand, for the geodesic trajectory, the reconstruction appears smooth and physically plausible.  $A_3A_4$  moves across a boundary which has no training data. In this case, the geodesic cannot



(a) Normalised distance. Since the NN-based and graph-based  $A^*$  methods require different generative models, the distances are not comparable; therefore, the Riemannian distances of geodesic is divided by the corresponding mean of the 100 distances of the Euclidean interpolation.



(b) Path searching time. The mean of the graph-based  $A^*$  searching time is 0.09 second. Since the variance of  $A^*$  searching time is small, the orange colour is not shown in the figure.

Figure 1: Box plot of distances and searching time of the pendulum dataset. 100 pairs of data points are randomly selected for each model to compute the distances and searching time. The box plot illustrates the median, as well as [25, 75] and [5, 95] percentiles.

avoid a relative high MF, but it still has lower Riemannian velocity than that of Euclidean interpolation.

We compared our proposed method with the NN-based approach. Both the ODE-based [2] and the NN-based [10] methods require second derivatives to approximate the geodesic. The ODE-based method takes more time than the NN-based method, so we only used latter for this method comparison. Fig. 1a shows that the geodesic of both the NN-based and graph-based method are shorter than those of from the Euclidean interpolation. We can see from Fig. 1b that the searching time of the graph-based method is dra-

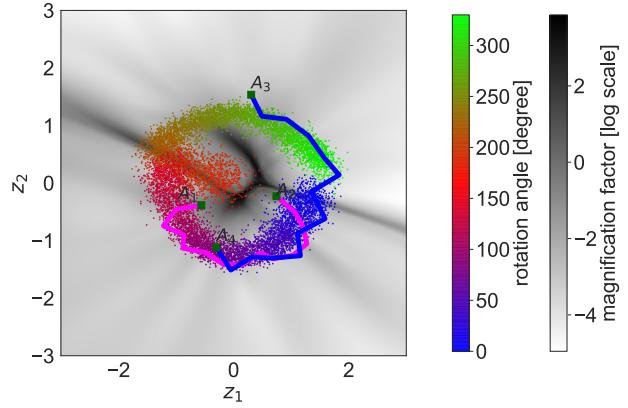


Figure 2: MF of pendulum in two latent dimensions. The blue and magenta lines are the approximate geodesics using the proposed graph-based method.

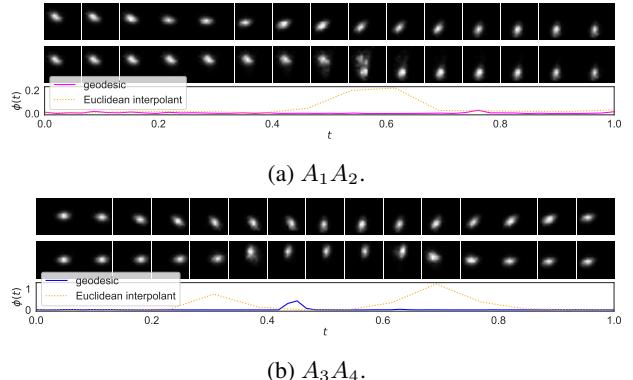


Figure 3: Reconstruction of the pendulum trajectory from the 2D latent space. The  $A^*$  search is used for the geodesic. The upper lines are the geodesic and the middle lines are the Euclidean interpolation.

matically faster than that of the NN-based method. It took about 30 minutes for each graph with 1000 nodes. The graph building time is slightly different with different latent dimensions. We used the same model of 10 hidden layers with tanh activations for the decoder and sigmoid for the output layer. The only difference of the architecture is that the NN-based model requires a radial basis function (RBF) layer for the variance of the decoder. NN-based was evaluated using only two latent dimensions, as the computation time is large and the time increases significantly with increasing the number of latent dimensions.

## 4.2. Fashion MNIST

For the Fashion MNIST [35] evaluation, we used the standard training set, i.e.  $28 \times 28$  pixel images, for fitting an IWAE consisting of eight 128-neuron layers with ReLU activation [16]. We use 2, 3, 5 and 10 latent dimen-

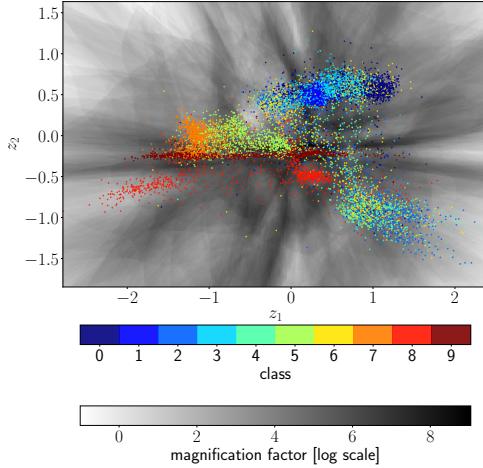


Figure 4: Magnification factor (greyscale) of Fashion MNIST in 2D latent space. Points are sampled validation data in latent space, with the colour defining the class.

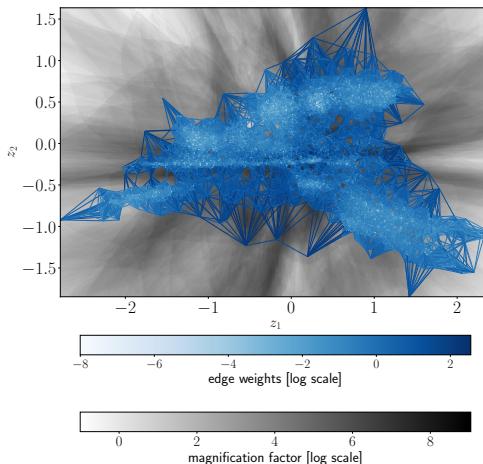


Figure 5: Graph of Fashion MNIST in 2D latent space. The edges are weighted by the geodesic distance, where darker blue signifies a transition with a higher integral of MF. For better observing the edge weights, the nodes are not shown.

sions for our evaluations. Additionally, standard augmentation strategies were used, e.g., horizontal flipping and jitter,

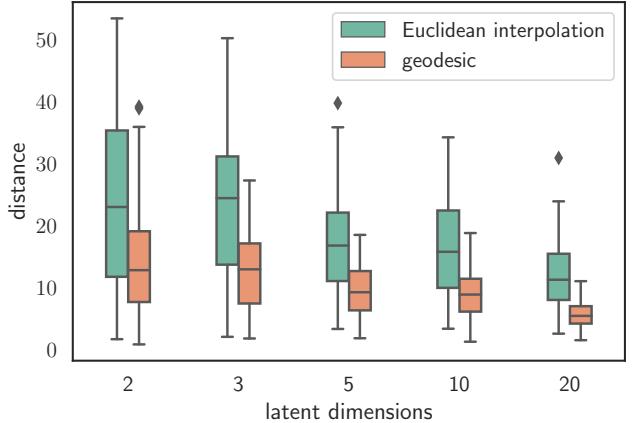


Figure 6: The distribution of distances in Fashion MNIST from 2000 randomly sampled trajectories based on Euclidean interpolation or geodesic, for multiple latent dimensions.

and 20% dropout, to avoid overfitting.

To generate the nodes of the geodesic graph, we used the validation dataset, encoded into the latent space. For each node, we selected the twentieth nearest data samples, based on a Euclidean distance. For each edge we calculated the velocity and geodesic of the trajectory, using fifteen interpolation points.

In Fig. 4 we randomly sampled 2000 trajectories between data points and calculated the geodesic and a distance based on Euclidean interpolation. The Euclidean-based trajectories result in consistently higher MF values, for all latent dimensions used. Therefore, the Euclidean-based trajectories cross more areas with high MF, resulting in fewer smooth transitions in the observational space. However, following the geodesic, smoothness is not guaranteed. There are cases where it is unavoidable for certain trajectories to cross an area of high magnification factor. Rather, following the geodesic can be interpreted as the minimisation of the overall Riemannian distance for a trajectory and it is expected that the MF will be lower than a simple Euclidean interpolation. Fig. 5 demonstrated this property on a 2-D manifold, where the edges between data samples are lighter for lower magnification factors. The areas where the MF is high, the edges are darker even if the samples are adjacent in the observational space. In such situations, the graph-based approach will therefore produce a more complex trajectory, but with a lower MF.

Fig. 7 shows visual examples of such trajectories for Fashion MNIST. Image reconstructions are produced from the decoder of the VAE by moving through the latent space along the trajectory specified by either the geodesic or Euclidean interpolation. Transitions are smoother for the geodesic, with classes being blended more gradually both

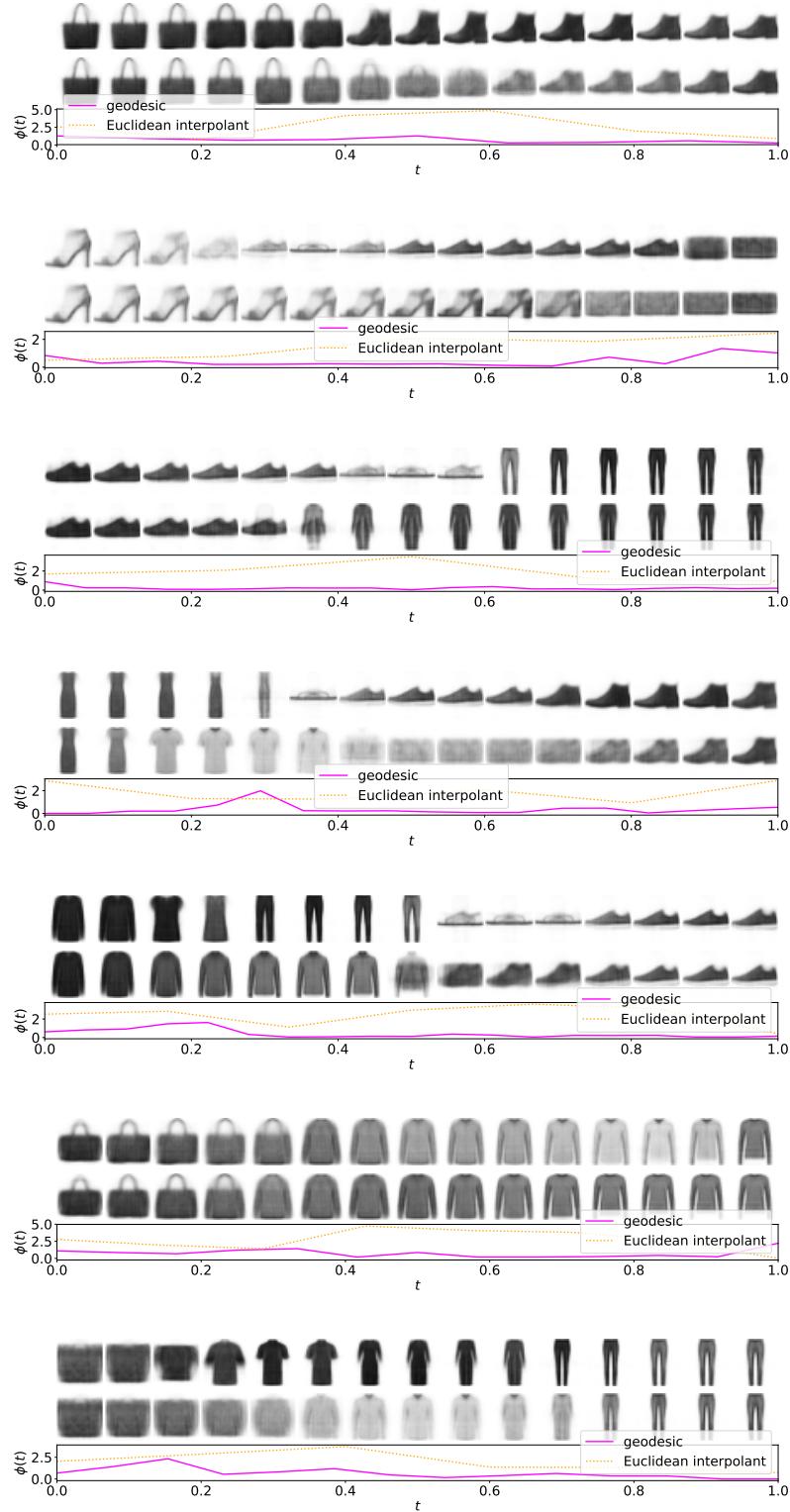


Figure 7: Reconstruction of Fashion MNIST with 20 latent dimensions. The upper lines are the geodesic and the middle lines are the Euclidean interpolation. The geodesic outperforms Euclidean interpolation by producing smoother interpolations that visually stay on the manifold.

in terms of shape and greyscale values.

### 4.3. Chairs

For the chairs dataset [3], we split chair sequences 80/20 for training and validation, with 74400 and 11966 images respectively. We used a zoom factor of 1.3 and rescaled to  $64 \times 64$  pixels.

We generated the geodesic graph in exactly the same way as Fashion MNIST.

In Fig. 9, we can see Euclidean-based trajectories result in consistently higher MF values, for all latent dimensions used (2, 3, 5, 10, 20). Similarly to previous datasets, these were generated by randomly sampling 2000 trajectories.

### 4.4. Celebrity Faces

The CelebFaces Attributes Dataset [25] was separated with a typical 80/20 split for training and validation, rescaling the images to  $64 \times 64$  pixels and grey-scale, using 162079 and 40519 images respectively. We generated the geodesic graph similarly to the Fashion MNIST and Chairs datasets.

In Fig. 11 we present the Euclidean-based trajectories that result in consistently higher MF values, compared to the number of latent dimensions used. The distances were computed using 2000 randomly sampled trajectories, as in the evaluation using the Fashion MNIST dataset. Fig. 10 shows three sampled trajectories.

In Fig. 12 we can see a difference in search time of geodesic paths for this dataset. This is because the graph contains significantly more nodes and edges, since we used the entire validation dataset. To minimise the computation time to build the graph and search, one can choose to subsample.

### 4.5. Human Motions

In addition to the evaluation using image data, we evaluated our approach on the CMU human motion dataset <sup>1</sup>. The dataset includes various movements, where walking (subject 35), jogging (subject 35), balancing (subject 49), punching (subject 143) and waving (subject 143) were selected. The input data was a 50-dimensional vector of the joint angles. In Fig. 13 we present the results with three latent dimensions. We observe that using the Euclidean interpolation the algorithm would generate trajectories crossing areas where the MF is high. However, using the geodesic the proposed approach is able to find similar gestures between the classes compared with the Euclidean interpolation. Interpolating between different classes of movement, the geodesic also generates abrupt movements as during training no data traversing between classes were provided.

---

<sup>1</sup><http://mocap.cs.cmu.edu/>

## 5. Conclusion and future work

We have seen how the major computational demand of applying Riemannian geometry to deep generative models, that of a non-convex optimisation problem, can be sidestepped by solving a related graph-based shortest path problem instead. After probing the method with a toy data set of which the true latent space is known, we have continued to a set of more extensive problems. Although the approach is only approximate, our experiments on a wide range of data sets show little loss in quality of interpolations while linear paths are consistently outperformed. The machinery paves the way towards the application of Riemannian geometry to more applied problems requiring the expressivity of deep generative models and the robustness of distance based approaches.

Further research in this topic can be how to efficiently choose the nodes such as using sigma points of unscented transform.

## Acknowledgment

We are very grateful to Maximilian Soelch for valuable suggestions concerning this work.

## References

- [1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *The International Conference on Database Theory (ICDT)*, volume 1, pages 420–434. Springer, 2001. [2](#)
- [2] 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. [1](#), [2](#), [3](#), [4](#)
- [3] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of cad models. In *CVPR*, 2014. [7](#)
- [4] J. Bayer, C. Osendorfer, and P. van der Smagt. Learning sequence neighbourhood metrics. In *Artificial Neural Networks and Machine Learning (ICANN)*, pages 531–538, 2012. [2](#)
- [5] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proceedings of the Twenty-Third International Conference Machine Learning (ICML)*, pages 97–104, 2006. [1](#)
- [6] C. M. Bishop, M. Svens’ en, and C. K. Williams. Magnification factors for the SOM and GTM algorithms. In *Proceedings Workshop on Self-Organizing Maps*, 1997. [3](#)
- [7] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. [1](#)
- [8] Y. Burda, R. B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015. [2](#)
- [9] N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer, and P. van der Smagt. Metrics for deep generative models. In *In-*

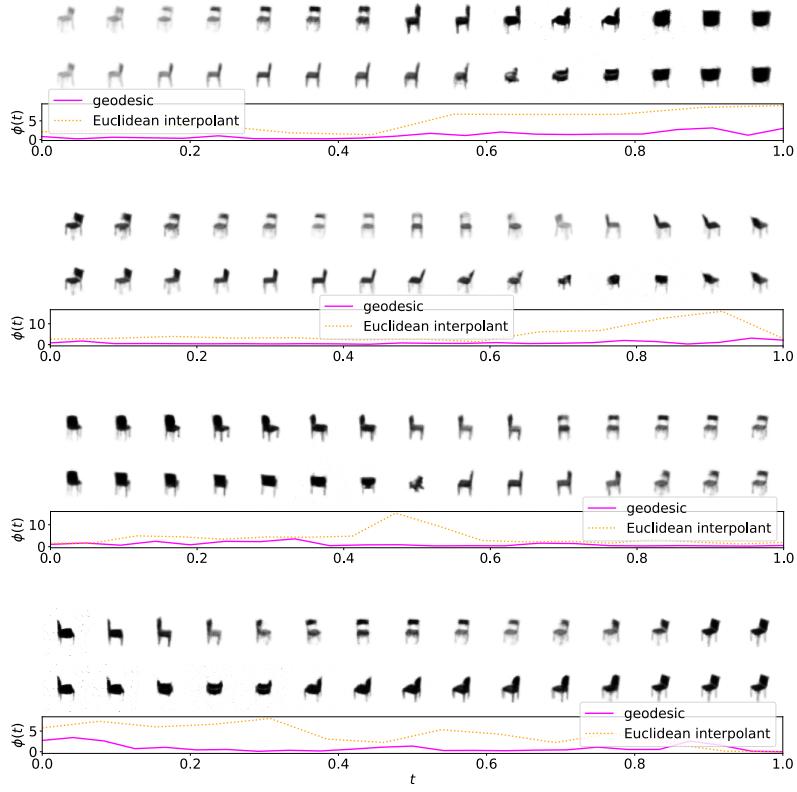


Figure 8: Reconstruction of chairs with 20 latent dimensions. The upper lines are the geodesic and the middle lines are the Euclidean interpolation.

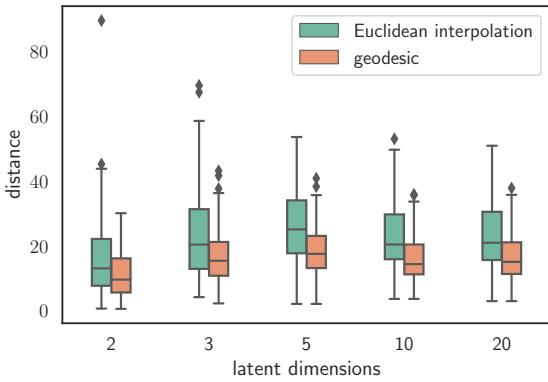


Figure 9: The distribution of distances in the Chairs dataset from 2000 randomly sampled trajectories based on Euclidean interpolation or geodesic, for multiple latent dimensions.

- ternational Conference on Artificial Intelligence and Statistics*, pages 1540–1550, 2018. 1, 2, 3
- [10] N. Chen, A. Klushyn, A. Paraschos, D. Benbouzid, and

P. van der Smagt. Active learning based on data uncertainty and model sensitivity. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018. 1, 2, 4

- [11] P. Crouch and F. S. Leite. The dynamic interpolation problem: on Riemannian manifolds, Lie groups, and symmetric spaces. *Journal of Dynamical and control systems*, 1(2):177–202, 1995. 2
- [12] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning (ICML)*, pages 209–216, 2007. 2
- [13] J. E. Doran and D. Michie. Experiments with the graph traverser program. *Proc. R. Soc. Lond. A*, 294(1437):235–259, 1966. 3
- [14] S. Flennerhag, P. G. Moreno, N. D. Lawrence, and A. Damianou. Transferring knowledge across learning processes. *arXiv preprint arXiv:1812.01054*, 2018. 2
- [15] A. E. Gelfand and A. F. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990. 2
- [16] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011. 5

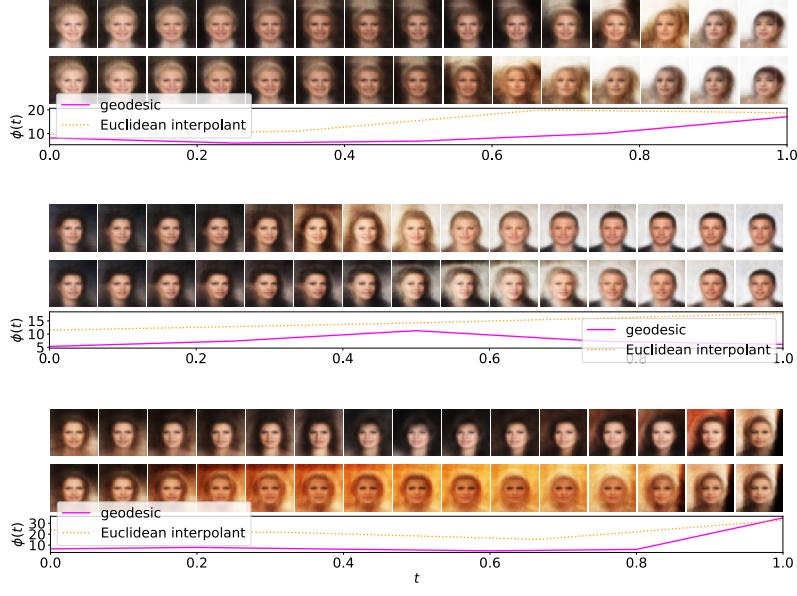


Figure 10: Reconstruction of faces with 25 latent dimensions. The upper lines are the geodesic and the middle lines are the Euclidean interpolation.

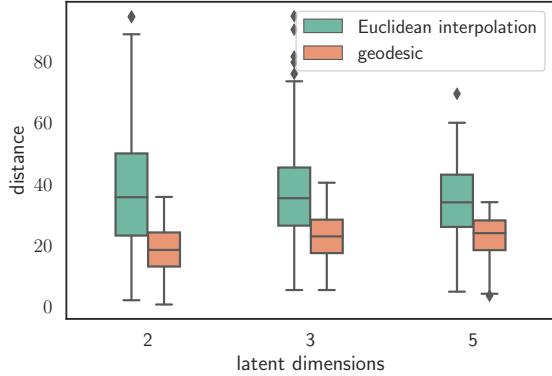


Figure 11: The distribution of distances in the Celebrity Faces dataset from 2000 randomly sampled trajectories based on Euclidean interpolation or geodesic, for multiple latent dimensions.

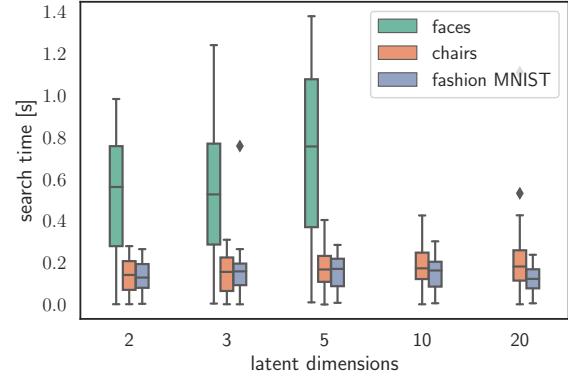


Figure 12: Search time distribution for 2000 randomly sampled trajectories, for multiple datasets and latent dimensions. The times are substantially larger for the Faces dataset due to the size of the graph that the A\* algorithm needs to traverse. However, this is still orders of magnitude faster than other proposed methods.

- [17] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 513–520, 2004. 2
- [18] D. Grattarola, D. Zambon, C. Alippi, and L. Livi. Learning graph embeddings on constant-curvature manifolds for change detection in graph streams. *arXiv preprint arXiv:1805.06299*, 2018. 2
- [19] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. 2
- [20] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 618–625, 1997. 1
- [21] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 1
- [22] D. P. Kingma and M. Welling. Auto-encoding variational

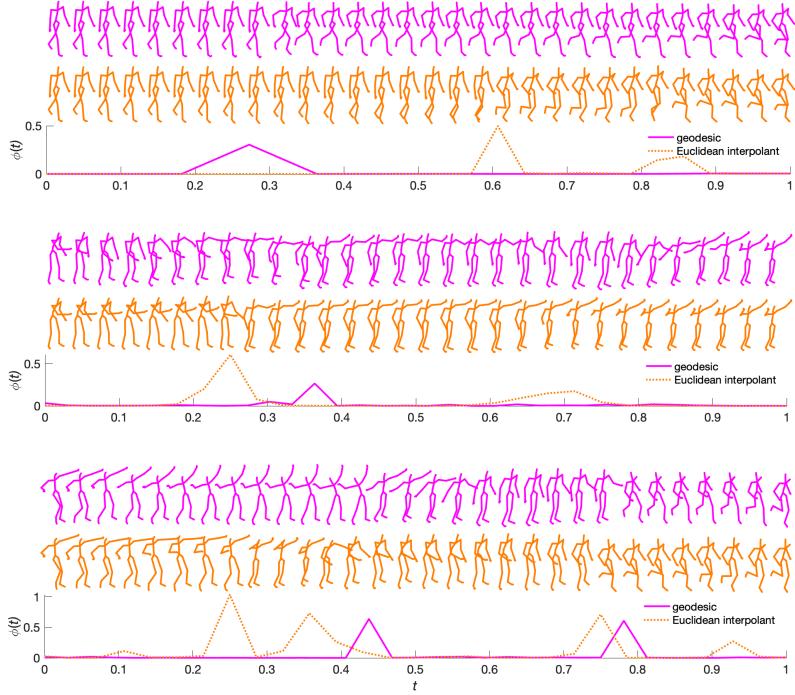


Figure 13: Interpolation of human motions with three latent dimensions.

- Bayes. *CoRR*, abs/1312.6114, 2013. 2
- [23] B. Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013. 2
- [24] A. Kumar, P. Sattigeri, and T. Fletcher. Semi-supervised learning with gans: manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pages 5534–5544, 2017. 2
- [25] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 7
- [26] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 1
- [27] S. M. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. D. Cox, P. I. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Trans. Robotics*, 32(1):1–19, 2016. 1
- [28] L. Noakes, G. Heinzinger, and B. Paden. Cubic splines on curved spaces. *IMA Journal of Mathematical Control and Information*, 6(4):465–473, 1989. 2
- [29] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. pages 1278–1286, 2014. 2
- [30] R. Salakhutdinov and G. E. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 412–419, 2007. 2
- [31] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 1
- [32] A. Tosi, S. Hauberg, A. Vellido, and N. D. Lawrence. Metrics for probabilistic geometries. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI, 2014*, pages 800–808, 2014. 1, 2
- [33] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems (NIPS)*, pages 1473–1480, 2006. 2
- [34] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 2
- [35] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. 4
- [36] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems (NIPS)*, pages 521–528, 2003. 2