Diffusion Variational Autoencoders

Luis A. Perez Rey , **Vlado Menkovski** and **Jim Portegies**
Eindhoven University of Technology, Eindhoven, The Netherlands
l.a.perez.rey@tue.nl, v.menkovski@tue.nl, j.w.portegies@tue.nl,

Abstract

A standard Variational Autoencoder, with a Euclidean latent space, is structurally incapable of capturing topological properties of certain datasets. To remove topological obstructions, we introduce Diffusion Variational Autoencoders (ΔVAE) with *arbitrary* (closed) manifolds as a latent space. A Diffusion Variational Autoencoder uses transition kernels of Brownian motion on the manifold. In particular, it uses properties of the Brownian motion to implement the reparametrization trick and fast approximations to the KL divergence. We show that the ΔVAE is indeed capable of capturing topological properties for datasets with a known underlying latent structure derived from generative processes such as rotations and translations.

1 Introduction

A large part of unsupervised learning is devoted to the extraction of meaningful latent factors that explain a certain dataset. The terminology around Variational Autoencoders (VAEs) [Kingma and Welling, 2014; Rezende *et al.*, 2014] suggests that they are a good tool for this task: they encode datapoints in a space that is called *latent space*. Purely based on this terminology, one could be tempted to think of elements in this space as latent variables, but **is this interpretation warranted?**

Part of the interpretation as latent space is warranted by the loss function of the Variational Autoencoder, which stimulates a continuous dependence between the latent variables and the corresponding data points. Close-by points in data space should also be close-by in latent space. This suggests that a Variational Autoencoder could capture topological and geometrical properties of a dataset.

However, a standard Variational Autoencoder (with a Euclidean latent space) is at times structurally incapable of accurately capturing topological properties of a dataset. Take for example the case of a spinning object placed on a turntable and being recorded by a camera from a fixed position. The dataset for this example is the collection of all frames. The true latent factor is the angle of the turntable. However, the space of angles is topologically and geometrically different from Euclidean space. In an extreme example, if we train a Variational Autoencoder with a one-dimensional latent space on the pictures from the object on the turntable, there will be pictures taken from almost the same angle ending up very far away from each other in latent space.

This phenomenon has been called manifold mismatch [Davidson *et al.*, 2018; Falorsi *et al.*, 2018]. To match the latent space with the data structure, Davidson et al. [2018] implemented spheres as latent spaces, whereas Falorsi et al. [2018] implemented the special orthogonal group $SO(3)$.

As further examples of datasets with topologically nontrivial latent factors, we can think of many translations of the same periodic picture, where the translation is the latent variable, or many pictures of the same object which has been rotated arbitrarily. In these cases, there are still clear latent variables, but their topological and geometrical structure is neither that of Euclidean space nor that of a sphere, but rather that of a torus and that of the $SO(3)$ respectively.

To address the problem of manifold mismatch, we developed the Diffusion Variational Autoencoder (ΔVAE) which in principle allows for an *arbitrary* closed manifold as a latent space. Our implementation includes a version of the reparametrization trick, and a fast approximation of the KL divergence in the loss.

We provide empirical results that show that the ΔVAE is

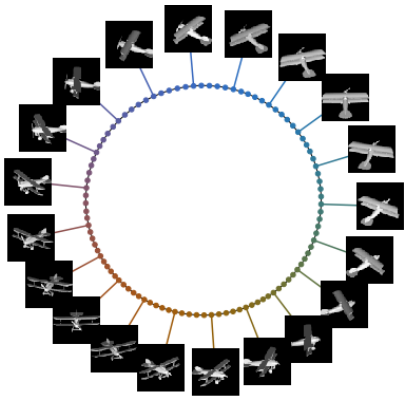


Figure 1: Latent space and reconstruction images for a ΔVAE with S^1 as latent space trained on the rendered images of a 3d model of a rotating airplane. The ΔVAE manages to capture the underlying geometrical structure.

capable of retrieving the underlying geometrical structure of a dataset consisting of rendered images of a spinning 3d model (Fig. 1) and translations of a picture (Fig. 4).

2 Related Work

Our work originated out of the search for algorithms that find semantically meaningful latent factors of data. Certainly, there is a heuristic argument for the use of VAEs for this purpose. The starting point and guiding principle is that of Occam's razor, that such latent factors might arise from the construction of simple, low-complexity models that explain the data [Portegies, 2018]. Strict formalizations of Occam's razor exist, through Kolmogorov complexity and inductive inference [Solomonoff, 1964; Schmidhuber, 1997], but are often computationally intractable. A more practical approach leads to the principle of minimum description length [Rissanen, 1978; Hinton and Zemel, 1994] and variational inference [Honkela and Valpola, 2004].

The use of VAEs and their extensions to this end has mostly taken place in the context of *disentanglement of latent factors* [Higgins *et al.*, 2017; Higgins *et al.*, 2018; Burgess *et al.*, 2018]. Examples of extensions that aim at disentangling latent factors are the β-VAE [Higgins *et al.*, 2017], the factor-VAE [Kim and Mnih, 2018], the β-TCVAE [Chen *et al.*, 2018] and the DIP-VAE [Kumar *et al.*, 2018].

However, the examples in the introduction already show that in some situations, the topological structure of the latent space makes it practically impossible to disentangle latent factors. The latent factors are inherently, topologically entangled: in the case of a 3d rotation of an object, one cannot assign globally linearly independent angles of rotation.

Still, it is exactly global topological properties that we feel a VAE has a chance of capturing. What do we mean by this? One instance of 'capturing' topological structure is when the encoder and decoder of the VAE provide bijective, continuous maps between data and latent space, also called homeomorphic auto-encoding [Falorsi *et al.*, 2018; de Haan and Falorsi, 2018]. This can only be done when the latent space has a particular topological structure, for instance that of a particular manifold.

We can also ask for more, that besides topological structure also geometric structure is captured. In that case, we require that distances in latent space carry some important meaning, for instance that distances in latent space are close to distances in data space, or to distances between ground-truth latent variables in case they are known. Tosi et al. [2014] and Arvanitidis et al. [2018] take a related, but different point of view. They do not consider a standard metric or predetermined metric on latent space, but rather determine a Riemannian (pullback) metric that by construction reflects the distances in data space.

One of the main challenges when implementing a manifold as a latent space is the design of the reparametrization trick. In [Mathieu *et al.*, 2019; Nagano *et al.*, 2019] a VAE with a hyperbolic and in [Davidson *et al.*, 2018] a VAE with a hyperspherical latent space were implemented. To our understanding, in the hyperspherical VAE they implemented a reparametrization function which was discontinuous.

If a manifold has the additional structure of a Lie group, this structure allows for a more straightforward implementation of the reparametrization trick [Falorsi *et al.*, 2018]. In our work, we do not assume the additional structure of a Lie group, but develop a reparametrization trick that works for general submanifolds of Euclidean space, and therefore by the Whitney (respectively Nash) embedding theorem, for general closed (Riemannian) manifolds.

The method that we use has similarities with the approach of Hamiltonian Variational Inference [Salimans *et al.*, 2015]. Moreover, the implementation of a manifold as a latent space can be seen as enabling a particular, informative, prior distribution. In that sense, our work relates to [Dilokthanakul *et al.*, 2016; Tomczak and Welling, 2017]. The prior distribution we implement is degenerate, in that it does not assign mass to points outside of the manifold.

There are also other ways to implement approximate Bayesian inference on Riemannian manifolds. For instance, Liu and Zhu adapted the Stein variational gradient method to enable training on a Riemannian manifold [Liu and Zhu, 2017]. However, this method is computationally expensive.

Finally, while manifold-learning methods such as LLE [Roweis, 2000], isomap [Tenenbaum *et al.*, 2000] and diffusion maps [Coifman and Lafon, 2006] all successfully embed datasets in low-dimensional space while approximately preserving distances, they typically do not allow for direct extraction of latent variables, because one would still need to identify the position of the embedding in ambient space and the manifold structure of the embedding. In addition, many of these methods do not have an explicit inverse map, a decoder, making them unsuitable for data generation.

3 Methods

A VAE has generally the following ingredients:

- a prior probability distribution $\mathbb{P}_Z$ on a latent space Z,
- a family of encoder distributions $\mathbb{Q}_Z^\alpha$ on Z, parametrized by α in a parameter space $\mathcal{A}$,
- an encoder neural network $\boldsymbol{\alpha}$ which maps from data space X to the parameter space $\mathcal{A}$,
- a family of decoder distributions $\mathbb{P}_X^\beta$ on data space X, parametrized by β in a parameter space $\mathcal{B}$,
- a decoder neural network $\boldsymbol{\beta}$ which maps from latent space Z to parameter space $\mathcal{B}$.

The neural network weights are optimized to minimize the negated evidence lower bound (ELBO)

$$\mathcal{L}(x) = -\mathbb{E}_{z\sim\mathbb{Q}_Z^{\boldsymbol{\alpha}(x)}}\left[\log p_X^{\boldsymbol{\beta}(z)}(x)\right] + D_{\mathrm{KL}}\left(\mathbb{Q}_Z^{\boldsymbol{\alpha}(x)}||\mathbb{P}_Z\right).$$

The first term is called *reconstruction error* (RE), the second term is called the KL-loss.

In a very common implementation, both latent space Z and data space X are Euclidean, and the families of decoder and encoder distributions are multivariate Gaussian. The encoder and decoder networks then assign to a datapoint or a latent variable a mean and a variance respectively.

When we implement Z as a Riemannian manifold, we need to find (i) an appropriate prior distribution, for which we will

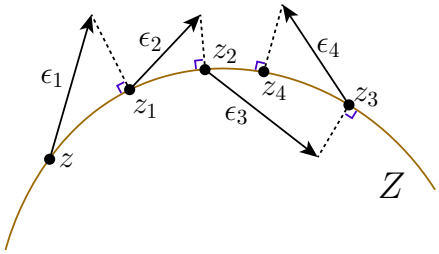


Figure 2: Random walk on a (one-dimensional) submanifold Z of $\mathbb{R}^2$, with time step $\tau = 1$.

choose the normalized Riemannian volume measure, (ii) a family of encoder distributions $\mathbb{Q}_Z^\alpha$, for which we will take transition kernels of Brownian motion, and (iii) an encoder network mapping to the correct parameters.

3.1 Brownian motion on a Riemannian manifold

We will briefly discuss Brownian motion on a Riemannian manifold, recommending lecture notes by Hsu [2008] as a more extensive introduction. In the paper, we always assume that Z is a smooth Riemannian submanifold of Euclidean space, which is closed, i.e. it is compact and has no boundary. There are many different, equivalent definitions of Brownian motion. We present here the definition that is closest to our eventual approximation and implementation.

We will construct Brownian motion out of random walks on a manifold. We first fix a small time step $\tau > 0$. We will imagine a particle, jumping from point to point on the manifold after each time step, see also Fig. 2. It will start off at a point $z \in Z$. We describe the first jump, after which the process just repeats. After time τ, the particle makes a random jump $\sqrt{\tau}\epsilon_1$ from its current position, into the surrounding space, where ϵ_1 is distributed according to a radially symmetric distribution in $\mathbb{R}^n$ with identity covariance matrix. The position of the particle after the jump, $z + \sqrt{\tau}\epsilon_1$, will therefore in general not be on the manifold, so we project the particle back: The particle's new position will be $z_1 = P(z + \sqrt{\tau}\epsilon_1)$ where the closest-point-projection $P : \mathbb{R}^n \to Z$ assigns to every point $x \in \mathbb{R}^n$ the point in Z that is closest to x. After another time $\tau > 0$ the particle makes a new, independent, jump ϵ_2 according to the same radially symmetric distribution, and its new position will be $z_2 = P(P(z + \sqrt{\tau}\epsilon_1) + \sqrt{\tau}\epsilon_2)$. This process just repeats.

Key to this construction, and also to our implementation, is the projection map P. It has nice properties, that follow from general theory of smooth manifolds. In particular, $P(x)$ smoothly depends on x, as long as x is not too far away from Z. The description of the manifolds and projections P used in this work is given in Table 1.

An implementation of a ΔVAE for an arbitrary closed manifold is easily achieved if the manifold is represented in generalized submanifold-coordinates of Euclidean space: If Z is a d-dimensional smooth closed Riemannian submanifold of Euclidean space $\mathbb{R}^n$, one can always find a finite number of open sets $U_i \subset \mathbb{R}^n$ and a finite set of generalized submanifold-coordinates $\phi_i : \mathbb{R}^n \supset U_i \to \mathbb{R}^n$ such that $\phi_i(U_i \cap Z) \subset \{x \in \mathbb{R}^n | \, x_{d+1} = \cdots = x_n = 0\}$ and

$$\phi_i(P(\phi_i^{-1}(x_1, \ldots, x_n)) = (x_1, \ldots, x_d, 0 \ldots, 0)$$

for all $(x_1, \ldots, x_n) \in \phi_i(U_i)$.

This gives an easy expression for the projection P which can be implemented with a partition of unity. Moreover, the scalar curvature can be computed from the functions ϕ_i in a standard way.

For $\tau > 0$ fixed, we have constructed a random walk, a random path on the manifold. We can think of this path as a discretized version of Brownian motion. Let now τ_S be a sequence converging to 0 as $S \to \infty$. For fixed $S \in \mathbb{N}$, we can construct a random walk with time step τ_S, and get a random path $W^S : [0, \infty) \to Z$.

The random paths W^S converge as $S \to \infty$ to a random path W (in distribution). This random path W is called Brownian motion. The convergence statement can be made precise by for instance combining powerful, general results by [Jørgensen, 1975] with standard facts from Riemannian geometry. But, because Riemannian manifolds are locally, i.e. when you zoom in far enough, very similar to Euclidean space, the convergence result essentially comes down to the central limit theorem and its upgraded version, Donsker's invariance theorem.

In fact, W can be interpreted as a Markov process, and even as a diffusion process. If A is a subset of Z, the probability that the Brownian motion $W(t)$ started at z is in the set A at time t is measured by a probability measure $\mathbb{Q}_Z^{t,z}$ applied to the set A. We denote the density of this measure with respect to the standard Riemannian volume measure Vol by $q_Z(t; z, \cdot)$. The function q_Z is sometimes referred to as the heat kernel.

3.2 Riemannian manifold as latent space

A ΔVAE is a VAE with a Riemannian submanifold of Euclidean space as a latent space, and the transition probability measures of Brownian motion $\mathbb{Q}_Z^{t,z}$ as a parametric family of encoder distributions. We propose the uniform distribution for $\mathbb{P}_Z$, which is the normalized standard measure on a Riemannian manifold (although the choice of prior distribution could easily be generalized).

As in the standard VAE, we then implement functions $\mathbf{z} : X \to Z$ and $\mathbf{t} : X \to (0, \infty)$ as neural networks.

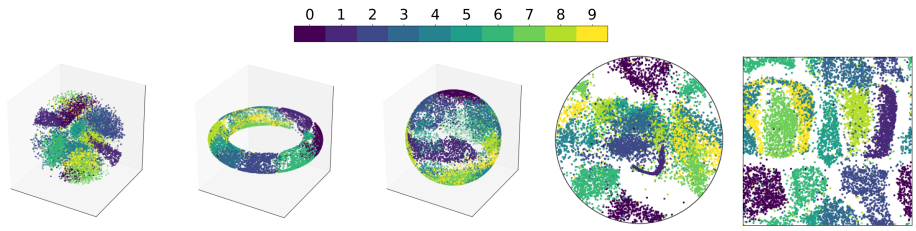
We optimize the weights in the network, aiming to minimize the average loss for the loss function

$$-\mathbb{E}_{z \sim \mathbb{Q}_Z^{\mathbf{t}(x), \mathbf{z}(x)}} \left[\log p_X^{\boldsymbol{\beta}(z)}(x) \right] + D_{\mathrm{KL}} \left(\mathbb{Q}_Z^{\mathbf{t}(x), \mathbf{z}(x)} \| \mathbb{P}_Z \right).$$

The first integral can often only be approached by sampling, and in that case it is often advantageous to perform a change of variables, commonly known as the *reparametrization trick* [Kingma and Welling, 2014].

Approximate Reparametrization by Random Walk

We construct an *approximate* reparametrization map by approximating Brownian motion by a random walk. For a given datapoint x, starting from the estimated location parameter $\boldsymbol{z}(x)$ on the manifold Z, we set a random step in *ambient space* $\mathbb{R}^n$ with properly scaled length. We then project back



(a) $SO(3) \cong \mathbb{RP}^3$ (b) Torus (c) S^2 (d) $\mathbb{RP}^2$ (e) Flat torus

Figure 3: Latent space representation of MNIST for several manifolds, each color represents the digit value. The projective spaces are represented by a 3- and 2-dimensional ball respectively, for which every point on the boundary is identified with its antipode. The effect of this identification can be seen, since the same digits that map close to a point on the boundary also map close to the reflected point.

MANIFOLD	PROJECTION (P)	$P(x)$	SCALAR CURVATURE (SC)
FLAT TORUS	$P : \mathbb{R}^4 \mapsto S^1 \times S^1$	$\frac{Q_4 x}{\|Q_4 x\|} + \frac{(I-Q_4)x}{\|(I-Q_4)x\|}$	0
EMBEDDED TORUS	$P : \mathbb{R}^3 \mapsto S^1 \times S^1$	$a\frac{x-x_c}{\|x-x_c\|} + x_c$	$\frac{2(\|Q_3 x\|-c)}{a^2\|Q_3 x\|}$
S^d	$P : \mathbb{R}^{d+1} \mapsto S^d$	$\frac{x}{\|x\|}$	$d(d-1)$
$SO(3)$	$P : \mathbb{R}^{3\times 3} \mapsto SO(3)$	$\det(U(x)V^T(x))(U(x)V^T(x)))$	3

Table 1: Manifold and the corresponding projection map P, its domain, codomain, and its scalar curvature (Sc). The map $Q_d : \mathbb{R}^d \to \mathbb{R}^2$ denotes the orthogonal projection to the first two coordinates. For the embedded torus, the parameter a and c correspond to the tube radius and the loop radius respectively. Additionally, $x_c = c\frac{Q_3 x}{\|Q_3 x\|}$. The functions $U : \mathbb{R}^{3\times 3} \mapsto \mathbb{R}^{3\times 3}$ and $V^T : \mathbb{R}^{3\times 3} \mapsto \mathbb{R}^{3\times 3}$ correspond to the orthogonal matrices obtained from the singular value decomposition of a 3×3 matrix $x = U\Sigma V^T$.

to the manifold using the projection function P and repeat. In total, we take S steps, see Fig. 2.

We define the function $g : \mathcal{E}^S \times (0, \infty) \times Z \to Z$ by

$$g(\epsilon_1, \ldots, \epsilon_S, \boldsymbol{t}(x), \boldsymbol{z}(x)) =$$
$$P\left(..P\left(P\left(\boldsymbol{z}(x) + \sqrt{\tfrac{\boldsymbol{t}(x)}{S}}\epsilon_1\right) + \sqrt{\tfrac{\boldsymbol{t}(x)}{S}}\epsilon_2\right)..+\sqrt{\tfrac{\boldsymbol{t}(x)}{S}}\epsilon_S\right).$$

If we take $\epsilon_1, \ldots, \epsilon_S$ as i.i.d. random variables, distributed according to a radially symmetric distribution, then $g(\epsilon_1, \ldots, \epsilon_S, \boldsymbol{t}(x), \boldsymbol{z}(x))$ is approximately distributed as a random variable with distribution $\mathbb{Q}_Z^{\boldsymbol{t}(x), \boldsymbol{z}(x)}$. The computational complexity of the sampling is linear in S. Yet the approximation is very accurate for small value of $\boldsymbol{t}(x)$, even for small values of S, if we take $\epsilon_1, \ldots, \epsilon_S$ approximately Gaussian. We have set $S = 10$ throughout the presented results.

Approximation of the KL-divergence

Unlike the standard VAE, or the hyperspherical VAE with the Von-Mises distribution, the KL-term cannot be computed exactly for the ΔVAE. By a parametrix expansion cf. [Berger *et al.*, 1971], we can nonetheless achieve a good approximation.

Proposition 1. *The KL divergence follows the following formal asymptotic expansion, where d is the dimensionality of Z, and Sc is the scalar curvature of the manifold Z in z.*

$$D_{\mathrm{KL}}(\mathbb{Q}^{t,z} \| \mathbb{P}_Z) =$$
$$\int_Z q_Z(t; z, y) \log q_Z(t; z, y) dy + \log \mathrm{Vol}(Z) =$$
$$-\frac{d}{2}\log(2\pi t) - \frac{d}{2} + \log \mathrm{Vol}(Z) + \frac{1}{4}\mathsf{Sc}\, t + o(t).$$

Besides the asymptotic approximation, one may also choose to approximate the heat kernel numerically or use Monte Carlo approximation.

4 Experiments

We have implemented [1] ΔVAEs with latent spaces of d-dimensional spheres, a flat two-dimensional torus, a torus embedded in $\mathbb{R}^3$, the $SO(3)$ and real projective spaces $\mathbb{RP}^d$.

For all our experiments we used multi-layer perceptrons for the encoder and decoder with three and two hidden layers respectively. Recall that the encoder needs to produce both a point z on the manifold and a time t for the transition kernel. These functions share all layers, except for the final step where we project, with the projection map P, from the last hidden layer to the manifold to get z, and use an output layer with a $\tanh$ activation function to obtain $10^{-7} \leq t \leq 10^{-5}$.

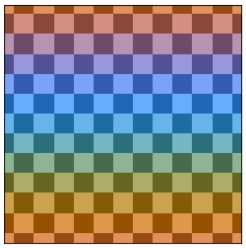
The encoder and decoder are connected by a sampling layer, in which we approximate sampling from the transition kernel of Brownian motion according to the reparametrization trick described in Section 3.2.

4.1 ΔVAEs for MNIST

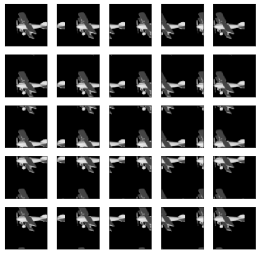
Mainly as a first test of our algorithm, we trained ΔVAEs on binarized MNIST [Salakhutdinov and Murray, 2008]. Fig. 3 shows the encoded digits for different manifolds.

For training the projective spaces, we embedded S^d in $\mathbb{R}^{d+1}$, and make the decoder neural network *even* by construction (i.e. the decoder applied to a point s on the sphere

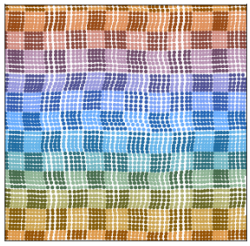
[1] https://github.com/luis-armando-perez-rey/diffusion_vae



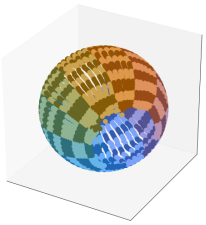
(a) Color scheme true factors



(b) Training data examples



(c) Embeddings in flat torus



(d) Embeddings in S^2

Figure 4: a) Colors used to represent the horizontal and vertical translations of the training dataset. The boundary conditions of the scheme are periodic. b) Examples of airplane images translated vertically and horizontally used during training. c) Image embeddings encoded in a flat torus by a ΔVAE, the boundary is periodic. d) Image embeddings encoded in a sphere S^2 by a ΔVAE.

equals the decoder applied to $-s$). Then, an encoder and decoder to and from the $\mathbb{RP}^3$ are defined implicitly. However, it must be noted that this setup does not allow for a homeomorphic encoding (because $\mathbb{RP}^d$ does not embed in S^d).

The numerically computed ELBO, reconstruction error, KL-divergence and the estimated log-likelihood are shown in Table 2 for a test dataset of the binarized MNIST. We provide a comparison with the values in [Davidson *et al.*, 2018] trained on a spherical latent space S^2 with a uniform prior. Additionally we present the results in [Figurnov *et al.*, 2018] trained on a latent space consisting of two circular independent latent variables with a uniform prior, which can be directly compared to the ΔVAE with a flat torus latent space.

The ΔVAE achieves similar log-likelihood estimates with respect to the results on $\mathcal{S}^2$ from [Davidson *et al.*, 2018]. On the other hand, the results for the ΔVAE trained on a flat torus have a lower log-likelihood compared to the results from [Figurnov *et al.*, 2018] (higher values are better).

Estimation of log-likelihood.
For the evaluation of the proposed methods we have estimated the log-likelihood of the test dataset according to the importance sampling presented in [Burda *et al.*, 2016]. The approximate log-likelihood of a datapoint x is calculated by sampling M latent variables $z^{(1)}, \dots, z^{(M)}$ according to the approximate posterior $\mathbb{Q}_Z^{\mathbf{t}(x),\mathbf{z}(x)}$ and is given by the formula

$$\log p_X^{\boldsymbol{\beta}}(x) \approx \log \left(\frac{1}{M} \sum_{m=1}^{M} \frac{p_X^{\boldsymbol{\beta}(z^{(m)})}(x) p_Z(z^{(m)})}{q_Z(\mathbf{t}(x); \mathbf{z}(x), z^{(m)})} \right).$$

The estimates in Table 2 are obtained with $M = 1000$ samples for each datapoint, averaged over all datapoints.

Distance Distortion Measure
In order to measure how the ΔVAE manages to capture the geometry of the dataset, we will measure the distance distortion of the encodings with respect to the true latent factors that generated the data.

Let z_i^* represent the i-th true latent factor that generated the i-th datapoint x_i from a dataset of N images, for instance the rotation angle. The distortion of distance will be measured by

$$\min_A \frac{2}{N(N-1)} \sum_{i<j} \left(d(z_i^*, z_j^*) - A\, d(\boldsymbol{z}(x_i), \boldsymbol{z}(x_j)) \right)^2.$$

The optimization over A takes into account any possible scaling of the embeddings. The distance function d corresponds to the geodesic distance in the corresponding space for both, the embeddings and the true latent factors. Lower is better.

4.2 Periodic Translation of Pictures

To test whether a ΔVAE can capture topological properties, we trained it on a synthetic dataset consisting of horizontal and vertical translations of a picture of an airplane subject to periodic boundaries. This example illustrates the capabilities of the ΔVAE to extract the underlying toroidal structure in more natural images as shown in Fig. 4.

We show the numerical comparison of the trained ΔVAE for different manifolds in Table 3. Notice that even though the LL, ELBO, KL and RE values are very similar among manifolds, the distance distortion measure between the true factors and the embeddings is an order of magnitude smaller for the ΔVAE trained with a flat torus as latent space compared to Euclidean or a spherical latent space.

The fact that Fig. 4c is practically a reflection and translation of the legend in Fig. 4a, shows that there is an almost isometric correspondence between the translation of the original picture and the encoded latent variable.

We contrast this to when we train the same dataset on a ΔVAE with a sphere as a latent space. Fig. 4d displays typical results, showing that large parts of the sphere are not covered with a discontinuity in the embeddings.

4.3 Object Rotation

We have investigated the capabilities of the ΔVAE in capturing the underlying topological structure for a synthetic dataset consisting of rendered images from a 3d model of an airplane within the ModelNet dataset [Wu *et al.*, 2014] . The images consists of gray-scale renders of 64×64 pixels, showing the 3d model centered in a frame of reference and rotated around the z-axis. The angle of rotation for each image is chosen from a regular partition of the interval $[-\pi, \pi)$.

Fig. 1 shows the latent variables of a ΔVAE trained with S^1 as latent space and 100 rendered images. The color encoding represents the true angles at which each image was generated. It is important to note that the ΔVAE is capable of identifying in an unsupervised manner the topological and geometrical structure associated to the object's orientation.

We present in Table 3 the numerical results for the trained ΔVAE with S^1 as latent space compared to the standard VAE

MANIFOLD	LL	ELBO	KL	RE
S^2	-132.20±0.39	-134.67±0.47	7.23±0.05	-127.44±0.47
EMBEDDED TORUS	-132.79±0.53	-137.37±0.59	9.14±0.18	-128.23±0.67
FLAT TORUS	-131.73±0.69	-139.97±0.78	12.91±0.08	-127.07±0.81
$\mathbb{RP}^3$	-125.27±0.37	-128.17±0.58	9.38±0.12	-118.79±0.60
$\mathbb{RP}^2$	-135.87±0.66	-138.13±0.72	7.02±0.12	-131.11±0.73
$\mathbb{R}^3$	-124.71±0.93	-128.01±1.05	9.12±0.09	-118.89±1.01
$\mathbb{R}^2$	-134.17±0.53	-136.61±0.64	7.05±0.06	-129.56±0.63
S^2 [DAVIDSON *et al.*, 2018]	-132.50±0.83	-133.72±0.85	7.28±0.14	-126.43±0.91
FLAT TORUS[FIGURNOV *et al.*, 2018]	-127.60±0.40	-	-	-

Table 2: Numerical results for ΔVAEs trained on binarized MNIST. The values indicate mean and standard deviation over 10 runs. The columns represent the (data-averaged) log-likelihood estimate (LL), Evidence Lower Bound (ELBO), KL-divergence (KL) and reconstruction error (RE) evaluated on the test data. For comparison we present results for S^2 and flat torus as reported in previous work.

MANIFOLD	LL	ELBO	KL	RE	DISTORTION
PERIODIC TRANSLATION OF PICTURES					
FLAT TORUS	-14765.19 ± 6.72	-15120.95 ± 6.9	8.81 ± 0.2	15112.14 ± 7.04	0.073 ± 0.0022
$\mathbb{R}^2$	-14769.32 ± 7.07	-15129.63 ± 6.96	6.35 ± 0.31	15123.28 ± 7.24	1.0172 ± 0.0811
$\mathbb{R}^4$	-14765.45 ± 5.24	-15126.68 ± 5.14	7.77 ± 0.12	15118.91 ± 5.14	0.3076 ± 0.1117
S^2	-14767.94 ± 2.37	-15120.97 ± 2.46	5.70 ± 0.23	15115.27 ± 2.40	0.5109 ± 0.0151
OBJECT ROTATION					
S^1	-11295.57 ± 0.87	-11297.16 ± 1.81	2.03 ± 0.39	11295.13 ± 2.21	0.0057 ± 0.0041
$\mathbb{R}^2$	-11294.77 ± 0.19	-11299.57 ± 0.25	5.41 ± 0.15	11294.16 ± 0.32	0.8485 ± 0.0511
$\mathbb{R}^3$	-11294.83 ± 0.05	-11299.60 ± 0.07	5.86 ± 0.13	11293.74 ± 0.11	0.7394 ± 0.0498

Table 3: Numerical results for the ΔVAE trained on the periodic translation of picture and on the object rotation datasets. The values indicate the mean and standard deviation over 10 runs. The columns represent the (data-averaged) log-likelihood estimate (LL), Evidence Lower Bound (ELBO), KL-divergence (KL), reconstruction error (RE) and the distance distortion metric.

with Euclidean latent space. Once again the results for the LL, KL and RE do not differ significantly across different manifolds. In contrast, the results for the distance distortion differ by two orders of magnitude for the S^1 with respect to the Euclidean latent space VAEs showing that the latent structure is better preserved by using the appropriate manifold.

We compared to results obtained after projecting embeddings by standard manifold-learning algorithms onto the desired manifold in Table 4. The isomap method achieves lowest distortion, but its success is dependent on the right choice of orientation of the torus in latent space: projecting on a rotated torus gives a much worse result. This illustrates why out-of-the-box manifold-learning algorithms do not directly capture latent variables: the relevant manifold structure in the embedded space still needs to be identified.

TECHNIQUE	TRANSLATION	ROTATION
LLE	0.5670	0.0528
ISOMAP	0.0075	0.0014
DIFFUSION MAP	0.8348	0.0115

Table 4: The distortion metric obtained after projecting the results of Locally Linear Embedding (LLE), isomap and the diffusion maps to the flat torus (for translation) and the circle (for rotation).

5 Conclusion

We developed and implemented Diffusion Variational Autoencoders, which allow for arbitrary manifolds as a latent space. Our original motivation was to investigate to which extent VAEs find semantically meaningful latent variables, and more specifically, whether they can capture topological and geometrical structure in datasets. By allowing for an arbitrary manifold as a latent space, ΔVAEs can remove obstructions to capturing such structure.

Indeed, our experiments with translations of periodic images and rotations of objects show that a simple implementation of a ΔVAE with the appropriate manifold as latent space is capable of capturing topological properties. We see this as important steps towards algorithms that capture semantically meaningful latent variables.

Acknowledgements

This work has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737459 (project Productive 4.0).

References

[Arvanitidis *et al.*, 2018] G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent space oddity: on the curvature of deep generative models. In *ICLR*, 2018.

[Berger *et al.*, 1971] M. Berger, P. Gauduchon, and E. Mazet. *Le spectre d'une variété riemannienne*. Springer, 1971.

[Burda *et al.*, 2016] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *ICLR*, 2016.

[Burgess *et al.*, 2018] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-VAE. arXiv:1804.03599, 2018.

[Chen *et al.*, 2018] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in Variational Autoencoders. In *NeurIPS 31*. 2018.

[Coifman and Lafon, 2006] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, jul 2006.

[Davidson *et al.*, 2018] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence*. 2018.

[de Haan and Falorsi, 2018] P. de Haan and L. Falorsi. Topological constraints on homeomorphic auto-encoding. *arXiv preprint*, arXiv:1812.10783, 2018.

[Dilokthanakul *et al.*, 2016] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with Gaussian mixture Variational Autoencoders. *arXiv preprint*, arXiv:1611.02648, 2016.

[Falorsi *et al.*, 2018] L. Falorsi, P. de Haan, T. R. Davidson, N. De Cao, M. Weiler, P. Forré, and T. S. Cohen. Explorations in homeomorphic variational auto-encoding. arXiv:1807.04689, 2018.

[Figurnov *et al.*, 2018] M. Figurnov, S. Mohamed, and A. Mnih. Implicit Reparameterization Gradients. *NeurIPS*, 2018.

[Higgins *et al.*, 2017] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[Higgins *et al.*, 2018] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. arXiv:1812.02230, 2018.

[Hinton and Zemel, 1994] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *NeurIPS*, 1994.

[Honkela and Valpola, 2004] A. Honkela and H. Valpola. Variational learning and bits-back coding: An information-theoretic view to Bayesian learning. *IEEE Transactions on Neural Networks*, 15(4):800–810, 7 2004.

[Hsu, 2008] E. P. Hsu. *A brief introduction to Brownian motion on a Riemannian manifold*. Lecture notes, 2008.

[Jørgensen, 1975] E. Jørgensen. The central limit problem for geodesic random walks. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(1-2):1–64, 1975.

[Kim and Mnih, 2018] H. Kim and A. Mnih. Disentangling by factorising. arXiv:1802.05983, 2018.

[Kingma and Welling, 2014] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[Kumar *et al.*, 2018] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of distentangled latent concepts from unlabeled observations. *ICLR*, 2018.

[Liu and Zhu, 2017] C. Liu and J. Zhu. Riemannian Stein variational gradient descent for Bayesian inference. arXiv:1711.11216, 2017.

[Mathieu *et al.*, 2019] E. Mathieu, C. L. Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh. Hierarchical Representations with Poincaré Variational Auto-Encoders. *NeurIPS*, 2019.

[Nagano *et al.*, 2019] Y. Nagano, S. Yamaguchi, Y. Fujita, and M. Koyama. A Wrapped Normal Distribution on Hyperbolic Space for Gradient-Based Learning. *ICML*, 2019.

[Portegies, 2018] J. W. Portegies. Ergo learning. *Nieuw Archief voor Wiskunde*, 5(3):199–205, 2018.

[Rezende *et al.*, 2014] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. arXiv:1401.4082, 2014.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 9 1978.

[Roweis, 2000] S. T. Roweis. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, dec 2000.

[Salakhutdinov and Murray, 2008] R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *ICML*, 2008.

[Salimans *et al.*, 2015] T. Salimans, D. Kingma, and M. Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the gap. In *ICML*, 2015.

[Schmidhuber, 1997] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857 – 873, 1997.

[Solomonoff, 1964] R.J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7(1):1–22, 3 1964.

[Tenenbaum *et al.*, 2000] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Tomczak and Welling, 2017] J. M. Tomczak and M. Welling. VAE with a VampPrior. arXiv:1705.07120, 2017.

[Tosi *et al.*, 2014] A. Tosi, S. Hauberg, A. Vellido, and N. D. Lawrence. Metrics for probabilistic geometries. In *UAI*, 2014.

[Wu *et al.*, 2014] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. 2014.