Abstract

Variational autoencoders (VAEs) have become one of the most popular deep learning approaches to unsupervised learning and data generation. However, traditional VAEs suffer from the constraint that the latent space must distributionally match a simple prior (e.g. normal, uniform), independent of the initial data distribution. This leads to a number of issues around modeling manifold data, as there is no function with a bounded Jacobian that maps a normal distribution to certain manifolds (e.g. a hypersphere). Similarly, there are not many theoretical guarantees on the encoder and decoder created by the VAE. In this work, we propose a variational autoencoder that maps manifold valued data to its diffusion map coordinates in the latent space, resamples in a neighborhood around a given point in the latent space, and learns a decoder that maps the newly resampled points back to the manifold. The framework is built off of SpectralNet [Shaham et al., 2018a], and is capable of learning this data dependent latent space without computing the eigenfunction of the Laplacian explicitly. We prove that our method is capable of learning a locally bi-Lipschitz map between the manifold and the latent space, and that our resampling method around a point in the latent space maps points back to the manifold around the point specifically into a neighborhood on the tangent space at the point on the manifold. We also provide empirical evidence of the benefits of using a diffusion map latent space on manifold data.

1 Introduction

Recent developments in generative models such as variational autoencoders (VAEs) [Kingma and Welling, 2013] and generative adversarial networks (GANs) [Goodfellow et al., 2014] have made it possible to generate remarkably realistic examples from complex high dimensional distributions. VAEs in particular are notable for their ability to jointly learn both an algorithm for generating samples and a latent space that concisely describes them. They consist of two networks, an encoder that takes the input and transforms it to a latent representation, and a decoder that reconstructs the input. The VAE aims to minimize the reconstruction error while regularizing the statistics of the latent representation so as to resemble some parametric prior. On the other hand, GANs directly constrain the latent representation to be some parametric prior. Although VAEs and GANs are powerful tools, their representational power is limited by the set of distributions that these parametric priors can map to (or specifically the set of mappings that neural network functions can feasibly learn). The plethora of domains in which the two methods have been successfully leveraged is testament to the richness of this set.

But they are also surprisingly weak at modeling some very simple distributions. While much effort has been put into developing methods that generate the distribution of a given data set, less emphasis
has been placed on preserving its geometric structure, though there are some works in this direction. 
Lindenbaum et al. [2018] use the geometry of the data to generate points such that the data is 
uniformly sampled along a manifold. In Cloninger et al. [2017], the authors defined an optimization 
scheme for mapping points from intrinsic diffusion coordinates back to the data manifold. 
The modeling ability of VAEs is closely tied to the geometric relationship between the prior manifold 
and the data manifold. Two recent works, Davidson et al. [2018] and Xu and Durrett [2018], 
demonstrate that traditional VAEs fail to represent data that lie on some hyperspherical manifold. 
To solve this issue, the authors suggest to use the von Mises-Fisher (vMF) distribution, which leads 
the hypersphere as the latent prior. Rey et al. [2019] then extend the prior to arbitrary manifolds, 
and demonstrate empirically that certain manifolds are well suited for certain data distributions. 
The fundamental motivation behind these new priors is to emphasize latent distributions that better 
represent the geometric structure of the data. Data distributions that are topologically torus-like will 
be better captured by torus priors, hyperspherical better captured by hyperspherical priors, gaussian 
by gaussian, and so on. However, picking the appropriate prior in these methods requires some very 
exact knowledge about the data manifold, and moreover very exact knowledge of the prior itself. This 
dramatically reduces the set of tractable priors. 

But clearly, the geometry of the data manifold is present in the data itself. Moreover, it has already 
been shown to be learnable, through traditional manifold learning techniques (e.g. Tenenbaum et al. 
[2000], Roweis and Saul [2000], Coifman and Lafon [2006]). In this paper we leverage these approaches 
to motivate a variational Bayesian method with a class of adaptive geometric priors. Specifically, we 
consider the family of priors described by the diffusion embedding \( \psi(X) \) of the data \( X \) with respect 
to a kernel \( k \) in the manner of Coifman and Lafon [2006]. 

Our contributions are as follows: 1) We apply this prior to the stochastic gradient variational Bayes 
(SGVB) framework proposed by [Kingma and Welling 2013], which naturally leads to a generative 
model that retains the efficient stochastic maximum likelihood estimation, posterior inference, and 
marginal inference properties of VAEs. 2) We re-interpret the derivation of the evidence lower bound 
optimization (ELBO) framework in this geometric setting, and propose a variant of it specifically for 
the case when the prior is the diffusion embedding. 3) We propose a random walk based sampling 
procedure that arises from the random walk specified by the diffusion kernel. 4) We prove function 
approximation type results for the encoder and decoder learned by our algorithm, and provide 
evidence that suggests it approximates both the diffusion map \( \psi \) and its inverse mapping \( \psi^{-1} \). 5) We 
demonstrate in experiments that the proposed method performs superior to traditional VAE and GAN 
over several geometrically structured distributions.

2 Background

We first introduce some notation. Let \( X \subseteq \mathbb{R}^m \) be our data set and \( p \) be its distribution. We assume 
that \( X \) is some subset of the entire data manifold, i.e. \( X \subseteq \mathcal{M}_X = \text{support}(p) \). Note that this 
assumption is not at all restrictive and made for notational convenience, as \( \mathbb{R}^m \) is trivially a manifold. 
Now let \( \mathbb{R}^D \) be the latent space, and \( \psi : \mathbb{R}^m \rightarrow \mathbb{R}^D \) be the diffusion map. We can now similarly 
define the latent \( d \)-dimensional manifold as \( \mathcal{M}_Z = \psi(\mathcal{M}_X) \), where \( d \leq D \), and the data set as 
\( Z \subseteq \mathcal{M}_Z \).
2.1 Variational Autoencoders

The VAE is a popular latent variable model for approximating complex distributions. VAEs assume and exploit a latent structure in the data generation process: that the observed variable $x$ is conditionally distributed given an unobserved latent variable $z$. By marginalizing the learned conditional distribution, as in

$$p_θ(x) = \int p_θ(x|z)p_φ(z)dz,$$

we have the likelihood that $x$ could have been drawn from $p_θ$, the model’s approximation of the true distribution $p$. Directly optimizing (1) is intractable so the value is instead estimated by the evidence lower bound

$$\log p_θ(x) = D_{KL}(q_φ(z|x)||p_θ(z)) - D_{KL}(q_φ(z|x)||p_θ(z)) + E_{z \sim q_φ(z|x)}[\log p_θ(x|z)]$$

$$\geq -D_{KL}(q_φ(z|x)||p_θ(z)) + E_{z \sim q_φ(z|x)}[\log p_θ(x|z)],$$

where $q_φ(z|x)$ is some approximation of the true posterior $p_θ(z|x)$. This leads to the empirical loss function

$$\mathcal{L}_{VAE} = -D_{KL}(q_φ(z|x)||p_θ(z)) + \log p_θ(x|z),$$

where $z_i = g_c(x, \epsilon_i), \epsilon_i \sim N(0, I)$, and $g_c$ is a deterministic, differentiable function that generates $q_φ(z|x)$ by the reparameterization trick [Kingma et al., 2015]. As a result of this trick and the differentiability of the remaining components, all parameters of the method can be learned by regular gradient-based optimization techniques such as SGD. Maximizing (2) (or the stochastic version of it, (3)) is known as evidence lower bound optimization (ELBO). One noteworthy extension of this method is [Sohn et al., 2015] in the case where $x$ depends on another random variable $y$ and the conditional likelihood of $x$ given $y$ can be bounded as $\log p_θ(x|y) \geq -D_{KL}(q_φ(z|x,y)||p_θ(z|y)) + E_{z \sim q_φ(z|x,y)}[\log p_θ(x|z, y)].$

Though the prior $p(z)$ in (1) is posed as a distribution freely chosen by the user, it is in reality harshly restricted to a small subset of all possible distributions — with minimum requirement of being tractably computed and sampled from, and recommended requirement of being computed in closed form. The most commonly used priors are the normal and uniform distributions, in large part because they satisfy these requirements.

In theory, VAEs (and variational Bayesian methods at large) are invariant to this choice of prior $p(z)$, so long as there exists some mapping from $M_Z$ to $M_X$ and back. In practice, many highly complex distributions (e.g. images, text, audio) can be convincingly generated by VAEs with such priors. However, it is not difficult to see that the choice of $p(z)$ given $p(x)$ will have repercussions on the resulting model. At least, it will affect how “violent” the resulting mappings will be between the spaces.$^1$ We would like to have mappings that are more well-behaved, as they are often easier for a neural network to learn.

2.2 Diffusion Maps

Proposed by [Coifman and Lafon, 2006], diffusion maps (DM) are kernel methods that perform non linear dimensionality reduction on a data set $X \subseteq M_X$. Given a symmetric and positive kernel $k : M_X \times M_X \to \mathbb{R}$, DM considers the induced random walk on the graph $M_X$, where the transition probabilities between points $x, y \in M_X$ are proportional to the value of $k(x, y)$. Namely, if we let $\mu$ be the probability distribution of points on $M_X$, and define $d(x) = \int_{M_X} k(x, y)\,d\mu(y)$ to be the “weighted degree” of $x$ (this interpretation is precise in the case of finite $X$), and further define $p(x, y) = \frac{k(x, y)}{d(x)}$, then we have the conservation property $\int_{M_X} p(x, y)\,d\mu(y) = 1$, and $p(x, \cdot)$ is the transition probability of the induced random walk at $x$.

Moreover, if we define the diffusion distance as $D_l(x, y) = \int_{M_X} (p_l(x, u) - p_l(y, u))^2 \,du/u$ and the diffusion map as $\psi(x) \doteq [\lambda_1 f_1(x), \lambda_2 f_2(x), \ldots, \lambda_D f_D(x)]$, where $\{f_i\}_{1 \leq i \leq D}$ and $\{\lambda_i\}_{1 \leq i \leq D}$ are the first $D$ eigenfunctions and eigenvalues of $p$, then [Coifman and Lafon, 2006] show that the

$^1$ Violent in the sense that the Jacobian of the mapping will be very large. Consider, for example, the continuous function mapping a two dimensional unit normal distribution to a circle centered at the origin. Such a function would have to map all points in $\mathbb{R}^2$ to a topologically one dimensional manifold. This is a very extreme mapping of points.
diffusion map $\psi$ embeds the data $X \in \mathbb{R}^m$ into the Euclidean space $\mathbb{R}^D$ so that diffusion distances are approximated by Euclidean distances (up to relative accuracy $\frac{k}{\lambda^2}$). This is a powerful property, as it allows the arbitrarily complex random walk induced by $k$ on $M_X$ to become an isotropic Gaussian random walk in $\psi(M_X)$.

In practice, a popular choice of $k$ that efficiently captures the geometry of the data is the Gaussian kernel, defined as

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \text{ for } x, y \in M_X,$$

where $\sigma$ is a user specified bandwidth. In practice, if $X = \{x_1, \ldots, x_N\}$ is a finite set of observations from $M_X$, then $k(x_i, x_j) \equiv K_{ij}$ is an $N \times N$ kernel matrix.

Until recently, the diffusion map $\psi_k$ was tractable only via the eigendecomposition of $K_{ij}$, and could therefore only be evaluated at a small set of points $\{x_i\}_{i=1}^N$. However, Shaham et al. [2018a] propose approximations of the function $\psi$ itself in the case that the kernel $k$ is symmetric. In particular, we will leverage the algorithm in Shaham et al. [2018a] to enforce our diffusion embedding prior.

2.3 Locally bi-Lipschitz Coordinates by Laplacian eigenfunctions

The construction of local coordinates of Riemannian manifolds by Laplacian eigenfunctions and diffusion map coordinates has been analyzed in Jones et al. [2008]. The result there establishes the existence of a set of $d$ spectral coordinates which maps from a neighborhood $U(x)$ of each point $x$ on the manifold to $\mathbb{R}^d$ and the mapping is guaranteed to be bi-Lipschitz on $U(x)$. With smooth compact manifold and smooth Riemannian metric, the neighborhood $U(x)$ can be chosen to be geodesic ball with the radius being a constant multiple of the inradius (the largest possible radius until intersecting with the manifold boundary), where the constant is uniform for all $x$. $U(x)$ can be charted by the exponential map at $x$. The indices of the $d$ spectral coordinates as well as the local bi-Lipschitz constants may depend on $x$. Specifically, the Lipschitz constants involve inverse of the inradius at $x$ multiplied again by some global constants. For completeness we give a simplified statement of the Jones et al. [2008] result in the supplementary material.

Making use of the compactness of the manifold, one can always cover the manifold with $m$ many neighborhoods (geodesic balls) on which the bi-Lipschitz property guaranteed by the theory in Jones et al. [2008] holds. As a result, there are totally $D$ many spectral coordinates, $D \leq md$ (in practice $D$ is much smaller than this upper bound, since the selected spectral coordinates in the proof of Jones et al. [2008] tend to be low-frequency ones, and thus the selection on different neighborhoods tend to overlap), such that on each of the $m$ neighborhoods, there exists a subset of $d$ spectral coordinates out of the $D$ ones which are bi-Lipschitz on the neighborhood, and the Lipschitz constants can be bounded uniformly from below and above.

3 Method

In this section we propose a variational Bayesian method that, given data $X$ and a kernel $k$, models the geometric structure of $X$ by approximating a random walk over $M_X$. The model is trained by maximizing the conditional log-likelihood of each data point given its random walk neighborhood. Points are generated from the trained model by sampling from $\pi$, the stationary distribution of the diffusion random walk.

Central to our method is an approximation of the random walk over $M_X$ given $k$. This is made possible by the fact that diffusion distances on $M_X$ are approximated by Euclidean distances in $\psi(M_X)$. Starting from some point $x \in X$, we can roughly describe one step of the walk as the composition of three functions:

1. approximate $z = \psi(x)$ by a forward map $\tilde{\psi} : M_X \rightarrow M_Z$,
2. take one step of an isotropic Gaussian random walk on $M_Z$ from the point $z$ to reach $z'$,
3. approximate $x' = \psi^{-1}(z')$ by a reverse map $\tilde{\psi}^{-1} : M_Z \rightarrow M_X$.

We would like to have that $\psi, \psi^{-1}$ approximates the diffusion map and its inverse. Formally, for each $x \in X$, we define the local evidence of $x$ as $\log p_\theta(x' | x)$, where $x' \in U_x := B_d(x, \delta) \cap M_X$ and $\theta$ is a set of hyperparameters.

Small enough to compute eigenvectors of $K$. 

4
$B_d(x, \delta)$ is the $\delta$-ball around $x$ with respect to $d$, the diffusion distance on $\mathcal{M}_d$. The resulting local evidence lower bound is:

$$\log p_\theta(x'|x) \geq -D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) + \mathbb{E}_{z' \sim q_\phi(z'|x)} \log p_\theta(x'|z'),$$

which produces the empirical loss function:

$$\hat{L}_{\text{DVAE}} = -D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) + \log p_\theta(x'|z'),$$

where $z' = g_\phi(\psi(x), \epsilon), \epsilon \sim \mathcal{N}(0, I), g_\phi$ is the deterministic, differentiable function that, when composed with $\psi$ generates $q_\phi$ by the reparameterization trick, and $x'$ is drawn from the random walk neighborhood of $x$, i.e.,

$$x' = \{\psi^{-1}(z') : z' \sim p_\theta(z'|x)\}$$

$$\approx \arg\min_y \{||\psi(y) - z'|^2 : z' \sim p_\theta(z'|x), y \in A\},$$

where $A \subseteq X$ is the training batch and $d$ is again the diffusion distance. This leads to Algorithm 1. Moreover, by recursively applying the approximate step of the random walk to an initial point $x_0$, we can eventually sample from $\pi$, the stationary distribution of the random walk. This gives Algorithm 2. In the remainder of this section, we derive the algorithms stated above from concepts in variational Bayesian methods and diffusion maps.

**Algorithm 1** Training the variational random walk autoencoder

- $\phi, \theta \leftarrow$ Initialize parameters
- $\psi \leftarrow$ Obtain $\tilde{\psi}$, the approximate diffusion map of data $X$ and $k$ (e.g., by [Shaham et al., 2018a])

while not converged do

- $X \leftarrow$ Random batch from $X$
- $\epsilon \leftarrow$ Random batch from noise distribution
- $Z \leftarrow \psi(X)$
- $Z' \sim g_\phi(Z, \epsilon)$
- $X' \leftarrow \arg\min_x \{|\tilde{\psi}(x') - z'|^2 : x' \in A\}$
- $g \leftarrow \nabla_{\phi, \theta} \hat{L}_{\text{DVAE}}(\phi, \theta, X, X')$
- Update $\phi, \theta$ using $g$

- $\triangleright$ Compute diffusion embedding of minibatch
- $\triangleright$ Generate neighborhoods of $\psi(X)$
- $\triangleright$ Find approximate neighborhoods
- $\triangleright$ Compute gradients of empirical loss, i.e. (6)

**Algorithm 2** Sampling from the variational random walk autoencoder

- $X_0 \leftarrow$ Initialize with points $X_0 \subset X$
- $t \leftarrow 0$

while $p(X_0) \not\approx \pi$ do

- $\epsilon \leftarrow$ Random batch from noise distribution
- $Z_t \leftarrow \psi(X_t)$
- $Z_{t+1} \leftarrow g_\phi(Z_t, \epsilon)$
- $X_{t+1} \sim p_\theta(X|Z_{t+1})$
- $t \leftarrow t + 1$

- $\triangleright$ Obtain diffusion embedding of $X_t$
- $\triangleright$ Take one step of the diffusion random walk
- $\triangleright$ Map back into input space

### 3.1 The lower bound

We begin with the same goal as in a variational Bayesian method: to learn model parameters that maximize the log likelihood of a random variable under the model distribution. As usual, this maximum likelihood estimation is performed by lower bounding the log-likelihood, then optimizing the bound. However, we deviate from the derivation in the following three crucial ways:

1. We assume that we have an approximation $\tilde{\psi}$ of the diffusion map $\psi$ and that $p(z)$ is the distribution of points on the diffusion embedding, i.e. $p(z) = p(\psi(x))$. 
2. We assume that the posterior distribution \( p(z'|x) \) is the distribution of the single-step random walk on \( \mathcal{M}_Z \), induced by our choice of kernel \( k \), starting at \( z = \psi(x) \).

3. We consider maximizing the likelihood: \( \log p_\theta(x'|x) \), for \( x' \in B_d(x, \delta) \cap X \), namely the likelihood of the neighborhood given the point. In contrast, the usual way is to maximize \( \log p_\theta(x) \) for \( x \in X \), that is, simply the likelihood of the point. The proposed maximization can be computed from sampling the random walk on \( \mathcal{M}_Z \).

Given this framework, we propose the following local evidence lower bound. For \( x \in X \):

\[
\log p_\theta(x'|x) \geq -D_{KL}(q_\phi(z'|x,x')||p_\theta(z'|x)) + \mathbb{E}_{z' \sim q_\phi(z'|x,x')} \log p_\theta(x'|z'), \quad (9)
\]

(Derivation equivalent to that in Sohn et al. [2015].) Furthermore, if we make the simplifying assumption that \( q_\phi(z'|x,x') = q_\phi(z'|x) \) and \( p_\theta(x'|z'), z = p_\theta(x'|z') \), then we obtain the form as expressed in (5).

To make the relationship between \( z, z' \), and \( x' \) explicit, we can rewrite the functions in the bound entirely in terms of \( z, z' \in \mathcal{M}_Z \) and the inverse diffusion embedding operator \( \psi^{-1} : \mathcal{M}_Z \to \mathcal{M}_X \):

\[
p_\theta(x'|z') = p_\theta(\psi^{-1}(z')|\psi(z)), \quad q_\phi(z'|x) = q_\phi(z'|\psi^{-1}(z)) \seteq n \seteq M \seteq X.
\]

Recall that that the transition probability of a diffusion random walk on a space \( \mathcal{M}_X \) is \( p(y'|y) \). Thus \( q_\phi(z'|x) \) (and \( p_\theta(x'|z') \), the true probability that it approximates) is a composition of the steps 1 and 2 defined at the start of this section while \( p_\theta(x'|z) \) is step 3. This shows how the variational Bayesian method defined in this section can be seen as a random walk on \( \mathcal{M}_Z \).

3.2 The sampling procedure

Note that allowing the prior to be the diffusion embedding of \( X \) comes at a cost: sampling from an arbitrary manifold prior is no longer a single-step procedure as it is with simpler priors. But we claim that the cost is not too high, since we have access to \( \psi \). Recall that, by the diffusion map framework, \( k \) defined on \( \mathcal{M}_Z \) generates a random walk on \( \mathcal{M}_X \).

Composing \( q_\phi(z'|x)(\approx p_\theta(z'|x)) \) with \( p_\theta(x'|z') \) gives us an approximation of \( p_\theta(x'|z) \). Then the simple, parallelizable, and fast random walk based sampling procedure naturally arises: initialize with an arbitrary point on the manifold \( x_0 \in \mathcal{M}_X \), then pick suitably large \( N \) and for \( n = 1, \ldots, N \) draw \( x_n \sim p(x|x_{n-1}) \). See Section 5.2 for examples of points drawn from this procedure.

3.3 A practical implementation

We now introduce a practical implementation of Algorithms 1 and 2, considering the case where \( k \) is the Gaussian kernel as defined in (4), and \( \psi(x) \), \( q_\phi(z'|x) \) and \( p_\theta(x'|z') \) are neural network functions, as they are in VAEs and SpectralNet, respectively.

We first consider Algorithm 1. A practical implementation of this algorithm simply requires a practical estimator of the lower bound (5). For all \( x', x \in \mathcal{M}_X \), we can think of (5) as the relationship between \( \log p_\theta(x'|x) \) and two other terms: a **divergence of random walk distributions** and a **neighborhood reconstruction error**. We rely crucially on two advantages of our latent space: a) that it is well-defined (given the first \( D \) eigenvalues of \( k \) are distinct) and well-approximated (given SpectralNet) and b) that Euclidean distances in \( \mathcal{M}_Z \) approximate single-step random walk distances on \( \mathcal{M}_X \) (see Section 2.2 and Coifman and Lafon [2006]). Thus the transition probabilities induced by \( k \) can be approximated by Gaussian kernels \( \phi \) in \( \mathcal{M}_Z \).

The **neighborhood reconstruction error** \( \mathbb{E}_{x' \sim q_\phi(z'|x)} \log p_\theta(x'|z') \) should be differentiated from the **self reconstruction error** in VAEs, i.e. reconstructing \( x' \) vs \( x \). Since \( q_\phi(z'|x) \) models the neighborhood of \( \psi(x) \), we may sample \( q_\phi \) to obtain \( z' \) (the neighbor of \( x \) in the latent space). Assuming \( \psi^{-1} \) exists, we have (7). To make this practical, we can approximate \( x' \) by finding the closest data point to \( x' \) in random walk distance (due to the aforementioned advantages of the latent space). This gives (8).

- \( \psi^{-1} \) cannot generally be assumed to exist. However, we show in Section 4 that under certain conditions (that we empirically observe our model to satisfy) \( \psi^{-1} \) does exist and our model does approximate it.

- Importantly, note the choice of a Gaussian kernel in the latent space is **not** dependent on the choice of \( k \). We have this invariance due to the aforementioned property of diffusion embeddings.
On the other hand, the **divergence of random walk distributions** $-D_{KL}(q_\phi(z'|x)||p_\theta(z'|x))$ can be modeled simply as the divergence of two Gaussian kernels defined on $\mathcal{M}_Z$. Though $p_\theta(z'|x)$ is intractable, the diffusion map $\psi$ gives us the diffusion embedding $Z$, which is an approximation of the true distribution of $p_\theta(z'|x)$ in a neighborhood around $z = \psi(x)$. We estimate the first and second moments of this distribution in $\mathbb{R}^D$ by computing the local Mahalanobis distance of points in the neighborhood. Then, by minimizing the KL divergence between $q_\phi(z'|x)$ and the one implied by this Mahalanobis distance, we obtain the loss:

$$-D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) = -\log \frac{\alpha \Sigma_\phi}{|\Sigma|} + d - \text{tr}\{(\alpha \Sigma_\phi)^{-1} \Sigma\}, \quad (10)$$

where $\Sigma_\phi(x)$ is a neural network function, $\Sigma_\phi(x) = \text{Cov}(B_d(\psi(x), \delta) \cap Z)$ is the covariance of the points in a neighborhood of $z = \psi(x) \in Z$, and $\alpha$ is a scaling parameter. Note that $\Sigma_\phi(x)$ does not have to be diagonal, and we do not assume it to be so. Empirically, we observe that these neighborhoods need not be very small to obtain good approximations (see A.4). Combining (8) and (10) we obtain a practical implementation of Algorithm 1.

Now we consider Algorithm 2. Since we use neural networks to approximate $q_\phi(z'|x)$ and $p_\theta(x'|z')$, the generation procedure is highly parallelizable. We empirically observe the random walk enjoys rapid mixing properties — it does not take many iterations of the random walk to sample from all of $\mathcal{M}_Z$.

### 4 Theory

In this section, we theoretically prove that the desired inverse map $\psi^{-1}$ from spectral coordinate codes back to the manifold can be approximated by a decoder network, where the network complexity is bounded by quantities related to the intrinsic geometry of the manifold.

The theory for the capacity of the encoder to map $\mathcal{M}$ to the diffusion map space $\psi(\mathcal{M})$ has already been considered in [Shaham et al., 2018b] and [Mishne et al., 2017]. We instead focus on the decoder, which requires a different treatment. The following theorem is proved in Supplementary Material, based upon the result in [Jones et al., 2008].

**Theorem 4.1.** Let $\mathcal{M}_X \subset \mathbb{R}^m$ be a smooth $d$-dimensional manifold, $\psi(\mathcal{M}_X) \subset \mathbb{R}^D$ be the diffusion map for $D \geq d$ large enough to have a subset of coordinates that are locally bi-Lipschitz. Let $X = [X_1, \ldots, X_m]$ be the set of all $m$ extrinsic coordinates of the manifold. Then there exists a sparsely-connected ReLU network $f_N$, with $4DCM_X$ nodes in the first layer, $6dmN$ nodes in the second layer, and $2mN$ nodes in the third layer, and $m$ nodes in the output layer, such that

$$\|X(\psi(x)) - f_N(\psi(x))\|_{L_2(\psi(\mathcal{M})))} \leq \sqrt{m}C_\psi/\sqrt{N} \quad (11)$$

where the norm is interpreted as $\|F\|_{L_2(\psi(\mathcal{M})))}^2 := \int \|F(\psi(x))\|^2 d\psi(x)$. Here $C_\psi$ depends on how sparsely $X(\psi(x))$ can be represented in terms of the ReLU wavelet frame on each neighborhood $U_i$, and $C_{M_X}$ on the curvature and dimension of the manifold $M_X$.

We also wish to discuss the connections between the distribution each point in diffusion map space, $q_\phi(z|x)$, and the result of this distribution after being decoded through the decoder network $f_N(z)$ for $z \sim q_\phi(z|X)$. Similar to [Singer and Coifman, 2008], we begin by characterizing the covariance matrix $\text{Cov}(f_N(z)) := \mathbb{E}_{z \sim q_\phi(z|z)} [f_N(z)f_N(z)^T]$. The following theorem is proved in Supplementary Material.

**Theorem 4.2.** Let $f_N$ be a neural network approximation to $X$ as in Theorem 4.1, such that it approximates the extrinsic manifold coordinates. Let $C \in \mathbb{R}^{m \times m}$ be the covariance matrix $C = \mathbb{E}_{z \sim q_\phi(z|z)} [f_N(z)f_N(z)^T]$. Let $q_\phi(z|x) \sim N(\psi(x), \Sigma)$ with small enough $\Sigma$ that there exists a patch $U_{z_0} \subset \mathcal{M}$ around $z_0$ satisfying the bi-Lipschitz property of $\psi$ for $\Sigma$, and such that $\text{Pr}(\psi(z \sim q_\phi(z|x) \neq \psi(U_{z_0})) < \epsilon$. Then the number of eigenvalues of $C$ greater than $\epsilon$ is at most $d$, and $C = J_{z_0}\Sigma J_{z_0}^T + O(\epsilon)$ where $J_{z_0}$ is the $m \times D$ Jacobian matrix at $z_0$.

---

3For all experiments in Section 5 the number of steps required to draw approximately from $\pi$ is less than 50.
5 Experimental Results

5.1 Rotating MNIST Digit

In this section we compare our method to VAE and a Wasserstein GAN (Gulrajani et al. [2017]) (with a bi-lipchitz constraint on the critic). We create an artificial low-dimensional circular manifold by rotating an image of a handwritten digit from MNIST. We rotate an example of ‘6’ by randomly drawing \( N = 300 \) angles from a uniform distribution in the range \( [0, 2\pi] \), we denote this dataset as \( X_{\text{PR}300} \). Next, we evaluate the capabilities of a VAE, GAN and our method for generating new images based on \( X \). For the VAE, we use a two dimensional uniform prior \( p_{\theta} z \), such that \( z_i \sim U(0, 1), i = 1, 2 \). The noise injected to the GAN is also drawn from the same two dimensional uniform distribution \( p_{\theta} z \). We use the same architecture for all networks which consists of one hidden layer with 128 neurons, activation function for all networks are tanh. In Fig. 1, we present 300 generated samples, by displaying them on a scatter plot with coordinates corresponding to their latent dimensions \( z_1 \) and \( z_2 \). Next, to evaluate the variability of the generated images (compared to the original images), we compute the distance to nearest point between each point \( x_i \) to all generated points \( \tilde{x}_i \). The histogram of the closest distance between generated points and original points is presented in Fig. 2.

5.2 Data generation from uniformly sampled manifolds

In the following experiments, we evaluate the random walk algorithm described in [Algorithm 2]. This involves selecting an initial seed point, and resampling new points a large number of times to simulate a random walk on the manifold. In Fig. [A.1] we plot the principal components of the learned covariance of \( q_\phi \) in the latent space. In Fig. [A.2] we highlight the location of the initial seed point, take 50 steps of the random walk, and display the resulting generated points on three learned manifolds. Clearly after a large number of resampling iterations, the algorithm continues to generate points on the manifold, and the distribution of sampled points converges to a steady state distribution on the manifold. In Fig. [A.2] we also display a large number of points sampled from the same initial seed, in order to display the resulting distribution \( p_{\theta}(x|z) \). In Fig. [A.2] we show how the distribution of sampled points from the random walk changes with step number. Finally, in Fig. [A.2] we display the histograms of the distortion between \( \|\psi(x) - \psi(y)\|_2 / \|x - y\|_2 \) in a neighborhood of \( x \). This demonstrates that our learned manifold representation is attained the local bi-Lipschitz property described in [Jones et al., 2008] and used in Theorem 4.1.
5.3 Robustness of method to batch and neighborhood size

In the following experiment, we evaluate the our method’s ability to generate points that follow the structure of a low dimensional manifold. We let $X \subset \mathbb{R}^2$ be a two dimensional circle, and compare the distribution of $X$ to the reconstructed $\tilde{X}$. We run 20 simulations for different batch sizes, and compute both the Maximum Mean Discrepancy (MMD) \cite{Gretton2012} and Gromov-Wasserstein (GW) \cite{Memoli2011} distance between original samples and the generated samples. As evident from Fig. A.3, our method (as well as alternatives) seems robust to the batch size. Furthermore, the data generated by the proposed method is closer in distribution to the original points (at least based on the MMD and GW). Finally, we evaluate the effect of the neighborhood size, defined by the parameter alpha on the proposed method. As depicted by Fig. A.4 the proposed method is stable for a wide range of neighborhood sizes.

Figure 4: Histograms of the local Lipschitz ratio over three distributions: a loop \((a)\), sphere \((b)\), and the Stanford bunny \((c)\). The local Lipschitz ratio is defined as the ratio between the upper and lower Lipschitz constants over a local neighborhood of a point.

References

Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. \textit{arXiv preprint arXiv:1801.01587}, 2018a.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. \textit{arXiv preprint arXiv:1312.6114}, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In \textit{Advances in neural information processing systems (NIPS)}, volume 27, pages 2672–2680, 2014.

Ofir Lindenbaum, Jay Stanley, Guy Wolf, and Smita Krishnaswamy. Geometry based data generation. In \textit{Advances in Neural Information Processing Systems}, pages 1400–1411, 2018.

Alexander Cloninger, Wojciech Czaja, and Timothy Doster. The pre-image problem for laplacian eigenmaps utilizing 1 l regularization with applications to data fusion. \textit{Inverse Problems}, 33(7):074006, 2017.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. \textit{arXiv preprint arXiv:1804.00891}, 2018.

Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. \textit{arXiv preprint arXiv:1808.10805}, 2018.

Luis A. Pérez Rey, Vlado Menkovski, and Jacobus W. Portegies. Diffusion variational autoencoders. \textit{CoRR}, abs/1901.08991, 2019. URL \url{http://arxiv.org/abs/1901.08991}.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. \textit{science}, 290(5500):2319–2323, 2000.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. \textit{science}, 290(5500):2323–2326, 2000.
Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

Peter W Jones, Mauro Maggioni, and Raanan Schul. Manifold parametrizations by eigenfunctions of the laplacian and heat kernels. *Proceedings of the National Academy of Sciences*, 105(6):1803–1808, 2008.

Uri Shaham, Alexander Cloninger, and Ronald R Coifman. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018b.

Gal Mishne, Uri Shaham, Alexander Cloninger, and Israel Cohen. Diffusion nets. *Applied and Computational Harmonic Analysis*, 2017.

Amit Singer and Ronald R Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

Roy R Lederman and Ronen Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536, 2018.
A Results in [Jones et al., 2008]

To state the result in [Jones et al., 2008], we need the following set-up:

(C1) $\mathcal{M}$ is a $d$-dimensional smooth compact manifold, possibly having boundary, equipped with a smooth (at least $C^2$) Riemannian metric $g$.

We denote the geodesic distance by $d$. Let $\psi$ be the eigenfunction associated with $\lambda_j$, $\{\psi_j\}$ form an orthonormal bases of $L^2(M, \mu)$. The last condition is

(C3) The heat kernel (defined by the heat equation on $M$) has the spectral representation as

$$K_t(x, y) = \sum_{j=0}^{\infty} e^{-t\lambda_j} \psi_j(x)\psi_j(y).$$

Theorem A.1 (Theorem 2 Jones et al. [2008], simplified version). Under the above setting and assume (C1)-(C2), there are positive constants $c_1, c_2, c_3$ which only depend on $\mathcal{M}$ and $g$, s.t. for any $x \in \mathcal{M}$, $r_M(x)$ being the inradius, there are $d$ eigenfunctions of $\triangle_M$, $\psi_{j_1}, \cdots, \psi_{j_d}$, which collectively give a mapping $\Psi : \mathcal{M} \rightarrow \mathbb{R}^d$ by

$$\Psi_x(x) = (\psi_{j_1}(x), \cdots, \psi_{j_d}(x))$$

satisfying that $\forall y, y' \in B(x, c_1r_M(x))$,

$$c_2r_M(z)^{-1}d_M(y, y') \leq \|\Psi_x(y) - \Psi_x(y')\| \leq c_3r_M(z)^{-1-d/2}d_M(y, y').$$

That is, $\Psi$ is bi-Lipschitz on the neighborhood $B(x, c_1r_M(x))$ with the Lipschitz constants indicated as above. The subscript $x$ in $\Psi_x$ emphasizes that the indices $j_1, \cdots, j_d$ may depend on $x$.

B Proofs

Proof of Theorem 4.1. The proof of Theorem 4.1 is actually a simple extension of the following theorem, Theorem B.1, which needs to be proved for each individual extrinsic coordinate $X_k$, hence the additional factor of $\sqrt{m}$ coming from the $L^2$ norm of $m$ functions.

Theorem B.1. Let $\mathcal{M} \subset \mathbb{R}^m$ be a smooth $d$-dimensional manifold, $\psi(\mathcal{M}) \subset \mathbb{R}^D$ be the diffusion map for $D \geq d$ large enough to have a subset of coordinates that are locally bi-Lipschitz. Let one of the $m$ extrinsic coordinates of the manifold be denoted $X(\psi(x))$ for $x \in \mathcal{M}$. Then there exists a sparsely-connected ReLU network $f_{\mathcal{N}}$, with $4DC_M$ nodes in the first layer, $8dN$ nodes in the second layer, and $2N$ nodes in the third layer, such that

$$\|X - f_{\mathcal{N}}\|_{L^2(\psi(\mathcal{M}))} \leq \frac{C_\psi}{\sqrt{N}}$$

(A.1)

where $C_\psi$ depends on how sparsely $X(\psi(x))|_{U_i}$ can be represented in terms of the ReLU wavelet frame on each neighborhood $U_i$, and $C_M$ on the curvature and dimension of the manifold $\mathcal{M}$.

Proof of Theorem 4.1. The proof borrows from the main theorem of [Shaham et al., 2018b]. We adopt this notation and summarize the changes in the proof here. For a full description of the
theory and guarantees for neural networks on manifolds, see [Shaham et al., 2018b]. Let $C_M$ be the number of neighborhoods $U_i = B(x_i, \delta) \cap M$ needed to cover $M$ such that for $x, y \in U_i$, $(1 - \epsilon)|x - y| \leq d_M(x, y) \leq (1 + \epsilon)|x - y|$. Here, we choose $\delta = \min(\delta_M, \kappa^{-1} \rho)$ where $\delta_M$ is the largest $\delta$ that preserves locally Euclidean neighborhoods and $\kappa^{-1} \rho$ is the smallest value from [Jones et al., 2008] such that every neighborhood $U_i$ has a bi-Lipschitz set of diffusion coordinates.

Because of the locally bi-Lipschitz guarantee from [Jones et al., 2008], we know for each $U_i$ there exists an equivalent neighborhood $\psi(U_i)$ in the diffusion map space, where $\psi(x) = [\psi_1(x), \ldots, \psi_d(x)]$. Note that the choice of these $d$ coordinates depends on the neighborhood $U_i$. Moreover, we know the Euclidean distance on $\psi(U_i)$ is locally bi-Lipschitz w.r.t. $d_M(\cdot, \cdot)$ on $U_i$.

First, we note that as in [Shaham et al., 2018b], the first layer of a neural network is capable of using $4D$ units to select the subset of $d$ coordinates $\tilde{\psi}(x)$ from $\psi(x)$ for $x \in U_i$ and zeroing out the other $D - d$ coordinates with ReLU bump functions. Then we can define $X(\tilde{\psi}(x)) = X(x)$ on $x \in U_i$.

Now to apply the theorem from [Shaham et al., 2018b], we must establish that $X|_{U_i}: \tilde{\psi}(U_i) \to \mathbb{R}$ can be written efficiently in terms of RELU functions. Because of the manifold and diffusion metrics being bi-Lipschitz, we know at a minimum that $\psi$ is invertible on $\tilde{\psi}(U_i)$. Because of this invertibility, we will slightly abuse notation and refer to $X(\psi(x)) = X(x)$, where this is understood to be the extrinsic coordinate of the manifold at the point $x$ that corresponds to $\psi(x)$. We also know that $\forall x, y \in U_i$,

$$|X(\tilde{\psi}(x)) - X(\tilde{\psi}(y))| = |X(x) - X(y)| \leq \max_{z \in U_i} \|\nabla X(z)\|d(x, y) \leq \max_{z \in U_i} \|\nabla X(z)\| \frac{1 - \epsilon}{1 - \epsilon} \|\tilde{\psi}(x) - \tilde{\psi}(y)\|,$$

where $\nabla X(z)$ is understood to be the gradient of $X(z)$ at the point $z \in M$. This means $X(\tilde{\psi}(x))$ is a Lipschitz function w.r.t. $\tilde{\psi}(x)$. Because $X(\tilde{\psi}(x))$ Lipschitz continuous, it can be approximated by step functions on a ball of radius $2^{-\ell}$ to an error that is at most $\frac{\max_{z \in U_i} \|\nabla X(z)\|}{1 - \epsilon} (2^{-\ell} + 2^{-\ell + 1})$. This means the maximum ReLU wavelet coefficient is less than $\frac{\max_{z \in U_i} \|\nabla X(z)\|}{1 - \epsilon} (2^{-\ell} + 2^{-\ell + 1})$. This fact, along with the fact that $\tilde{\psi}(U_i)$ is compact, gives the fact that on $\tilde{\psi}(U_i)$, set of ReLU wavelet coefficients is in $\ell^1$. And from [Shaham et al., 2018b], if on a local patch the function is expressible in terms of ReLU wavelet coefficients in $\ell^1$, then there is an approximation rate of $\frac{1}{\sqrt{N}}$ for $N$ ReLU wavelet terms.

**Proof of Theorem 4.2** We borrow from [Singer and Coifman, 2008] to prove the following result. Given that the bulk of the distribution $q$ lies inside $\psi(U_{z_0})$, we can consider only the action of $f_N$ on $\psi(U_{z_0})$ rather than on the whole space. Because the geodesic on $U$ is bi-Lipschitz w.r.t. the Euclidean distance on the diffusion coordinates (the metric on the input space), we can use the results from [Singer and Coifman, 2008] and say that on $\psi(U_{z_0})$ the output covariance matrix is characterized by the Jacobian of the function $f_N$ mapping from Euclidean space (on the diffusion coordinates) to the output space, at the point $z_0$. So the covariance of the data lying inside $\psi(U_{z_0})$ is $J_{z_0} \Sigma J_{z_0}^T$, with an $O(\epsilon)$ perturbation for the fact that $\epsilon$ fraction of the data lies outside $\psi(U_{z_0})$.

The effective rank of $C$ being at most $d$ comes from the locally bi-Lipschitz property. We know $X(\psi(x))$ only depends on the $d$ coordinates $\psi(x)$ as in the proof of Theorem 4.1, which implies $f_N(\psi(x))$ satisfies a similarly property if $f_N$ fully learned $X(\psi(x))$. Thus, while $J \in \mathbb{R}^{m \times D}$, it is at most rank $d$, which means $J \Sigma J^T$ is at most rank $d$ as well.
Figure A.1: An example of the covariances $\Sigma_\theta$ learned by our method. We plot the principal component of $\Sigma_\theta(x)$, centered at each point $\psi(x)$ in the latent space, for all $x \in X$. The distributions $X$ are a loop (left), sphere (middle), and the Stanford bunny (right). Note that the manifolds are all topologically spherical, and this is reflected in the visualizations of the covariances.

Figure A.2: An example of the mixing rate of the random walk. Starting from a single point, we loop through Algorithm 2 except generating 8 new random walk points for each pre-existing point (instead of 1). Plotted (top to bottom, left to right) are the points at times $t = \{1, 2, 3, 4, 5, 50\}$.

Figure A.3: An empirical comparison of input distribution $p(x)$ and generated distribution $p(x')$ for the circular manifold. Evaluation of the generated distribution using VAE, GAN and proposed based on the Maximum Mean Discrepancy (MMD)\[1\] and Gromov-Wasserstein distance\[6\].
Figure A.4: Stability of reconstruction for the circular manifold as a function of the neighborhood size. Evaluation of the proposed method using the MMD and the Gromov-Wasserstein distances between the input distribution $p(x)$ and the network output distribution $p(x')$.

Figure A.5: An example of the generated circular embedding based on VAE (a), GAN (b) and proposed (c).

C Additional Figures

D Additional experiments

D.1 The spinning Yoda

In this section we provide an additional comparison between our method, VAE and a Wasserstein GAN [Gulrajani et al. 2017] (with a bi-lipchitz constraint on the critic). We use a video of a spinning 'Yoda' doll, recorded by the authors of Lederman and Talmon [2018]. The original video consists of two spinning dolls, in this experiment we focus on the spinning 'Yoda' doll. The data consists of 200 frames with $80 \times 30$ pixels. We denote this dataset as $X \sim \mathbb{R}^{200 \times 2400}$. Next, we evaluate the capabilities of a VAE, GAN and our method for generating new images based on $X$. For the VAE, we use a two dimensional uniform prior $p_{\theta}(z)$, such that $z_i \sim U(0, 1), i = 1, 2$. The noise injected to the GAN is also drawn from the same two dimensional uniform distribution $p_{\theta}(z)$. We use the same architecture for all networks: a single hidden layer with 128 neurons, and tanh activation functions. In Fig. A.6 we plot 200 generated samples. The $x, y$ coordinates of the plot correspond to $z_1, z_2$ latent dimensions of each example, and we overlay the generated image from each plotted point. To evaluate the variability of the generated images (compared to the original images), we compute the distance to the closest generated example, for each $x_j \in X$. The histogram of these closest distances is presented in Fig. A.7. One key advantage of our method for this example is that the latent space provides a meaningful representation of the latent coordinate in the 'Yoda' video.
Figure A.6: Generated points based on the spinning 'Yoda' video, plotted in the latent space of VAE (left) and GAN (middle) compared to our method (right).

Figure A.7: Nearest neighbors distribution between original points and generated ones ('Yoda' video) by VAE (left), GAN (middle) and our method (right). Observe that even though the manifold is well-represented, this distance is large, which implies that our method is able to generate novel points.