Variational Diffusion Autoencoders with Random Walk Sampling

Henry Li[1]*, Ofir Lindenbaum[1]*, Xiuyuan Cheng[2], and Alexander Cloninger[3]

[1] Applied Mathematics, Yale University, New Haven, CT 06520, USA
{henry.li,ofir.lindenbaum}@yale.edu
[2] Department of Mathematics, Duke University, Durham, NC 27708, USA
xiuyuan.cheng@duke.edu
[3] Department of Mathematics and Halicioğlu Data Science Institute, University of California San Diego, La Jolla, CA 92093, USA
acloninger@ucsd.edu

Abstract. Variational autoencoders (VAEs) and generative adversarial networks (GANs) enjoy an intuitive connection to manifold learning: in training the decoder/generator is optimized to approximate a homeomorphism between the data distribution and the sampling space. This is a construction that strives to define the data manifold. A major obstacle to VAEs and GANs, however, is choosing a suitable prior that matches the data topology. Well-known consequences of poorly picked priors are posterior and mode collapse. To our knowledge, no existing method sidesteps this user choice. Conversely, *diffusion maps* automatically infer the data topology and enjoy a rigorous connection to manifold learning, but do not scale easily or provide the inverse homeomorphism (i.e. decoder/generator). We propose a method [4] that combines these approaches into a generative model that inherits the asymptotic guarantees of *diffusion maps* while preserving the scalability of deep models. We prove approximation theoretic results for the dimension dependence of our proposed method. Finally, we demonstrate the effectiveness of our method with various real and synthetic datasets.

Keywords: deep learning, variational inference, manifold learning, image and video synthesis, generative models, unsupervised learning

1 Introduction

Generative models such as variational autoencoders (VAEs, [19]) and generative adversarial networks (GANs, [10]) have made it possible to sample remarkably realistic points from complex high dimensional distributions at low computational cost. While the theoretical framework behind the two methods are different — one is derived from variational inference and the other from game theory — they both involve learning smooth mappings from a user-defined prior $p(z)$ to the data $p(x)$.

[*] Equal contribution.

[4] https://github.com/lihenryhfl/SpectralVAEGAN

When $p(z)$ is supported on a Euclidean space (e.g. $p(z)$ is Gaussian or uniform) and the $p(x)$ is supported on a manifold (i.e. the Manifold Hypothesis, see [30,8]), VAEs and GANs become manifold learning methods, as manifolds themselves are defined as sets that are locally homeomorphic to Euclidean space. Thus the learning of such homeomorphisms may shed light on the success of VAEs and GANs in modeling complex distributions.

This connection to manifold learning also offers a reason why these generative models fail — when they do fail. Known as *posterior collapse* in VAEs [1,48,14,33] and *mode collapse* in GANs [11], both describe cases where the learned mapping collapses large parts of the input to a single point in the output. This violates the bijective requirement of a homeomorphism. It also results in degenerate latent spaces and poor generative performance.

A major cause of such failings is when the geometries of the prior and target data do not agree. We explore this issue of *prior mismatch* and previous treatments of it in Section 3. Given their connection to manifolds, it is natural to draw from classical approaches in manifold learning to improve deep generative models. One of the most principled methods is kernel-based manifold learning [38,36,4]. This involves embedding data drawn from a manifold $X \subset \mathcal{M}_X$ into a space spanned by the leading eigenfunctions of a kernel on $\mathcal{M}_X$. We focus specifically on *diffusion maps*, where [6] show that normalizations of the kernel define a diffusion process that has a uniform stationary distribution over the data manifold. Therefore, drawing from this stationary distribution samples uniformly from the data manifold. This property was used in [24] to smoothly interpolate between missing parts of the manifold. However, despite its strong theoretical guarantees, *diffusion maps* are poorly equipped for large scale generative modeling as they do not scale well with dataset size. Moreover, acquiring the inverse mapping from the embedding space — a crucial component of a generative model — is traditionally a very expensive procedure [5,21,28].

In this paper we address issues in variational inference and manifold learning by combining ideas from both. The theory in manifold learning allows us to recognize and correct *prior mismatch*, whereas variational inference provides a method to construct a generative model, which also offers an efficient approximation to the inverse *diffusion map*.

Our contributions: 1) We introduce the locally bi-Lipschitz property, a necessary condition of a homeomorphism, for measuring the stability of a mapping between latent and data distributions. **2**) We introduce variational diffusion autoencoders (VDAEs), a class of variational autoencoders that, instead of directly reconstructing the input, have an encoder-decoder that approximates one discretized time-step of the diffusion process on the data manifold (with respect to a user defined kernel k). **3**) We prove approximation theoretic bounds for deep neural networks learning such diffusion processes, and show that these networks define random walks with certain desirable properties, including well-defined transition and stationary distributions. **4**) Finally, we demonstrate the utility of the VDAE framework on a set of real and synthetic datasets, and show that they have superior performance and satisfy the locally bi-Lipschitz property.

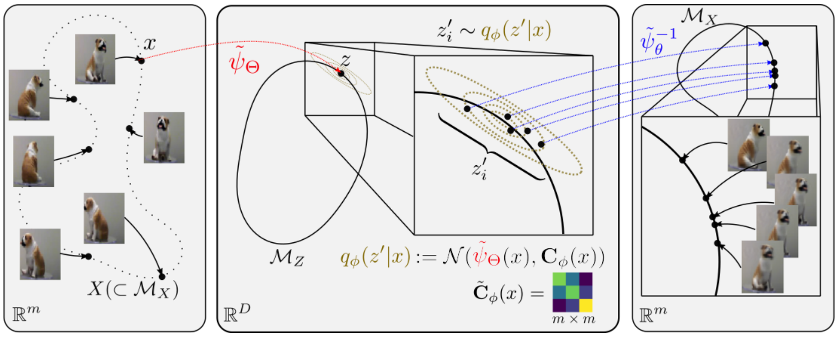

x
$\tilde{\psi}_\Theta$
z
$z_i' \sim q_\phi(z'|x)$
$\mathcal{M}_X$
$\tilde{\psi}_\theta^{-1}$
z_i'
$\mathcal{M}_Z$
$q_\phi(z'|x) := \mathcal{N}(\tilde{\psi}_\Theta(x), \mathbf{C}_\phi(x))$
$\tilde{\mathbf{C}}_\phi(x) =$
$m \times m$
$X(\subset \mathcal{M}_X)$
$\mathbb{R}^m$
$\mathbb{R}^D$
$\mathbb{R}^m$

Fig. 1: A diagram depicting one step of the diffusion process modeled by the variational diffusion autoencoder (VDAE). The *diffusion* and *inverse diffusion maps* ψ, ψ^{-1}, as well as the covariance $\mathbf{C}$ of the random walk on $\mathcal{M}_Z$, are all approximated by neural networks. Images on the leftmost panel are actually generated by our method.

2 Background

Variational inference (VI, [18,46]) combines Bayesian statistics and latent variable models to approximate some probability density $p(x)$. VI exploits a latent variable structure in the assumed data generation process, that the observations $x \sim p(x)$ are conditionally distributed given unobserved latent variables z. By modeling the conditional distribution, then marginalizing over z, as in

$$p_\theta(x) = \int_z p_\theta(x|z)p(z)dz, \tag{1}$$

we obtain the model evidence, or likelihood that x could have been drawn from $p_\theta(x)$. Maximizing the model evidence (Eq. 1) leads to an algorithm for finding likely approximations of $p(x)$. The cost of computing this integral scales exponentially with the dimension of z and thus becomes intractable with high latent dimensions. Therefore we replace the model evidence (Eq. 1) with the evidence lower bound (ELBO):

$$\log p_\theta(x) \geq -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{z \sim q(z|x)}[\log p_\theta(x|z)], \tag{2}$$

where $q(z|x)$ is usually an approximation of $p_\theta(z|x)$. Maximizing the ELBO is sped up by taking stochastic gradients [16], and further accelerated by learning a global function approximator q_ϕ in an autoencoding structure [19].

Diffusion maps [6] refer to a class of kernel methods that perform non-linear dimensionality reduction on a set of observations $X \subseteq \mathcal{M}_X$, where $\mathcal{M}_X$ is the assumed data manifold equipped with measure μ. Let $x, y \in X$; given a symmetric and non-negative kernel k, *diffusion maps* involve analyzing the induced random walk on the graph of X, where the transition probabilities

$P(y|x)$ are captured by the probability kernel $p(x,y) = k(x,y)/d(x)$, where $d(x) = \int_X k(x,y)d\mu(y)$ is the weighted degree of x. The diffusion map itself is defined as $\psi_D(x) := [\lambda_1 f_1(x), \lambda_2 f_2(x), ..., \lambda_D f_D(x)]$, 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. An important construction in *diffusion maps* is the *diffusion distance*:

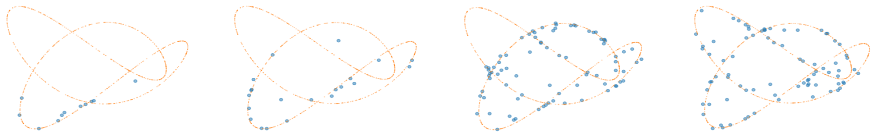
$$D(x,y)^2 = \int (p(x,u) - p(y,u))^2 \frac{d\mu(u)}{\pi(u)}, \tag{3}$$

where $\pi(u) = d(u)/\sum_{z\in X} d(z)$ is the stationary distribution of u. Intuitively, $D(x,y)$ measures the difference between the diffusion processes emanating from x and y. A key property of ψ_D is that it embeds the data $X \in \mathbb{R}^m$ into the Euclidean space $\mathbb{R}^D$ so that the diffusion distance is approximated by Euclidean distance (up to relative accuracy $\frac{\lambda_D}{\lambda_1}$). Therefore, the arbitrarily complex random walk induced by k on $\mathcal{M}_X$ becomes an isotropic Gaussian random walk on $\psi(\mathcal{M}_X)$.

SpectralNet [40] is a neural network approximation of the *diffusion map* ψ_D that enjoys a major computational speedup. The eigenfunctions $f_1, f_2, \ldots, f_D$ that compose ψ_D are learned by optimizing a custom loss function that stochastically maximizes the Rayleigh quotient for each f_i while enforcing the orthogonality of all $f_i \in \{f_n\}_{n=1}^D$ via a custom orthogonalization layer. As a result, the training and computation of ψ is linear in dataset and model size (as opposed to $O(n^3)$). We will use this algorithm to obtain our diffusion embedding prior.

Locally bi-Lipschitz coordinates by kernel eigenfunctions. The construction of local coordinates of Riemannian manifolds $\mathcal{M}_X$ by eigenfunctions of the diffusion kernel is analyzed in [17]. They establish, for all $x \in \mathcal{M}_X$, the existence of some neighborhood $U(x)$ and d spectral coordinates given $U(x)$ that define a bi-Lipschitz mapping from $U(x)$ to $\mathbb{R}^d$. With a smooth compact Riemannian manifold, we can let $U(x) = B(x, \delta r_{\text{in}})$, where δ is some constant and the *inradius* r_{in} is the radius of the largest ball around x still contained in $\mathcal{M}_X$. Note that δ is uniform for all x, whereas the indices of the d spectral coordinates as well as the local bi-Lipschitz constants may depend on x and are order $O(r_{\text{in}}^{-1})$. For completeness we give a simplified statement of the [17] result in the Appendix.

Using the compactness of the manifold, one can always cover the manifold with m many neighborhoods (geodesic balls) on which the bi-Lipschitz property in [17] holds. As a result, there are a total of D spectral coordinates, $D \leq md$ (in practice D is much smaller than md, since the selected spectral coordinates in the proof of [17] 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. We observe empirically that the bi-Lipschitz constants can be bounded uniformly from below and above (see Section 6.4).



(a) $\mathtt{t} = 1$ (b) $\mathtt{t} = 2$ (c) $\mathtt{t} = 4$ (d) $\mathtt{t} = 8$

Fig. 2: An example of the diffusion random walk simulated by our method on a 3D loop dataset. `t` is the number of steps taken in the random walk.

3 Motivation and related work

In this section we justify the key idea of our method: diagnosing and correcting *prior mismatch*, a failure case of VAE and GAN training when $p(z)$ and $p(x)$ are not topologically isomorphic. Intuitively, we would like the latent distribution to have three nice properties: (1) ***realizability***, that every point in the data distribution can be realized as a point in the latent distribution; (2) ***validity***, that every point in the latent distribution maps to a unique valid point in the data distribution (even if it is not in the training set); and (3) ***smoothness***, that points in the latent distribution vary in the intrinsic coordinate system in some smooth and coherent way.

These properties are precisely those enjoyed by a latent distribution that is homeomorphic to the data distribution. *Validity* implies injectivity, *realizability* implies surjectivity, *smoothness* implies continuity; and a mapping between topological spaces that is injective, surjective, and continuous is a homeomorphism. Therefore, studying algorithms that encourage approximations of homeomorphisms is of fundamental interest.

Though the Gaussian distribution for $p(z)$ is mathematically elegant and computationally expedient, there are many datasets for which it is ill-suited. Spherical distributions are known to be superior for modeling directional data [9,26], which can be found in fields as diverse as bioinformatics [13], geology [32], materials science [20], natural image processing [3], and many preprocessed datasets[5]. For data supported on more complex manifolds, the literature is sparse, even though it is well-known that data often lie on such manifolds [30,8]. In general, any manifold-supported distribution that is not globally homeomorphic to Euclidean space will not satisfy conditions (1-3) above.

Previous research on alleviating *prior mismatch* exists in various forms, and has focused on increasing the family of tractable latent distributions for generative models. [7,47] consider VAEs with the von-Mises Fisher distribution, a geometrically hyperspherical prior, and [43] consider mixtures of priors. [35] propose a method that, like our method, also samples from the prior via a diffusion

[5] Any dataset where the data points have been normalized to be unit length becomes a subset of a hypersphere.

process over a manifold. However, their method requires very explicit knowledge of the manifold (including its projection map, scalar curvature constant, and volume), and give up an exact estimation of the KL divergence. [34] avoids mode collapse by lower bounding the KL-divergence term away from zero to avoid overfitting. Similarly, [25] focuses on avoiding mode collapse by using class-conditional generative models, however it requires label supervision and does not provide any guarantees that the latent space generated is homeomorphic to the data space. Finally, [15] propose the re-scaling of various terms in the ELBO to augment the latent space — often to surprisingly great effect on latent feature discovery — but are restricted to the case where the latent features are independent.

While these methods expand the repertoire of feasible priors, they all require explicit user knowledge of the data topology. On the other hand, our method allows the user to be agnostic to this choice of topology; they only need to specify an affinity kernel k for local pairwise similarities. We achieve this by employing ideas from both *diffusion maps* and variational inference, resulting in a fully data-driven approach to latent distribution selection in deep generative models.

4 Method

In this section we propose the variational diffusion autoencoder (VDAE), a class of generative models built from ideas in *variational inference* and *diffusion maps*. Given the data manifold $\mathcal{M}_X$, observations $X \subset \mathcal{M}_X$, and a kernel k, VDAEs model the geometry of X by approximating a random walk over the latent diffusion manifold $\mathcal{M}_Z := \psi(\mathcal{M}_X)$. The model is trained by maximizing the *local evidence*: the evidence (i.e. log-likelihood) of each point given its random walk neighborhood. Points are generated from the trained model by sampling from π, the stationary distribution of the resulting random walk.

Starting from some point $x \in X$, we can think of one step of the walk as the composition of three functions: **1**) the approximate diffusion map $\widetilde{\psi}_\omega : \mathcal{M}_X \to \mathcal{M}_Z$ parameterized by ω, **2**) the stochastic function that samples from the diffusion random walk $z' \sim q_\phi(z'|x) = \mathcal{N}(\widetilde{\psi}_\omega(x), \widetilde{\mathbf{C}}_\phi(x))$ on $\mathcal{M}_Z$, and **3**) the approximate inverse diffusion map $\widetilde{\psi}_\theta^{-1} : \mathcal{M}_Z \to \mathcal{M}_X$ that generates $x' \sim p(x'|z') = \mathcal{N}(\widetilde{\psi}_\theta^{-1}(z'), cI)$ where c is a fixed, user-defined hyperparameter usually set to 1.

Note that Euclidean distances in $\mathcal{M}_Z$ approximate single-step random walk distances on $\mathcal{M}_X$ due to properties of the diffusion map embedding (see Section 2 and [6]). These properties are inherited by our method via the SpectralNet algorithm, since $\widetilde{\psi}_\omega|_{\mathcal{M}_X} : \mathcal{M}_X \to \mathcal{M}_Z$ satisfies the *locally bi-Lipschitz property*. This bi-Lipschitz property also reduces the need for regularization, and leads to guarantees of the ability of the VDAE to avoid posterior and mode collapse (see Section 5).

In short, to model a diffusion random walk over $\mathcal{M}_Z$, we must learn the functions $\widetilde{\psi}_\omega, \widetilde{\psi}_\theta^{-1}$, and $\widetilde{\mathbf{C}}_\phi$ that approximate the diffusion map, the inverse diffusion map, and the covariance of the random walk on $\mathcal{M}_Z$, at all points

$z \in \mathcal{M}_Z$. SpectralNet gives us $\widetilde{\psi}_\omega$. To learn $\widetilde{\psi}_\theta^{-1}$ and $\widetilde{\mathbf{C}}_\phi$, we use variational inference.

4.1 The lower bound

Formally, let us define $U_x := B_d(x, \delta) \cap \mathcal{M}_X$, where $B_d(x, \delta)$ is the δ-ball around x with respect to $d(\cdot, \cdot)$, the diffusion distance on $\mathcal{M}_Z$. For each $x \in X$ we define the *local evidence* of x as

$$\mathbb{E}_{x' \sim p(x'|x)|_{U_x}} \log p_\theta(x'|x), \tag{4}$$

where $p(x'|x)|_{U_x}$ restricts $p(x'|x)$ to U_x. This gives the *local evidence lower bound*

$$\log p_\theta(x'|x) \geq \underbrace{-D_{KL}(q_\phi(z'|x)||p_\theta(z'|x))}_{\text{divergence from true diffusion probabilities}} + \underbrace{\mathbb{E}_{z' \sim q_\phi(z'|x)} \log p_\theta(x'|z')}_{\text{neighborhood reconstruction error}}, \tag{5}$$

which produces the empirical loss function $\tilde{\mathcal{L}}_{\text{VDAE}} = -D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) + \log p_\theta(x'|z_i')$, where $z_i' = g_{\phi,\Theta}(x, \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, I)$. The function $g_{\phi,\Theta}$ is deterministic and differentiable, depending on $\widetilde{\psi}_\omega$ and $\widetilde{\mathbf{C}}_\phi$, that generates q_ϕ by the reparameterization trick[6].

4.2 The sampling procedure

Composing $q_\phi(z'|x)(\approx p_\theta(z'|x))$ with $p_\theta(x'|z')$ gives us an approximation of $p_\theta(x'|x)$. Then the simple, parallelizable, and fast random walk based sampling procedure naturally arises: initialize with an arbitrary point on the manifold $x_0 \in \mathcal{M}_X$ (e.g. from the dataset X), pick suitably large N, and for $n = 1, \ldots, N$ draw $x_n \sim p(x|x_{n-1})$. See Section 6.2 for examples of points drawn from this procedure.

4.3 A practical implementation

We now introduce a practical implementation, considering the case where $\widetilde{\psi}_\omega(x)$, $q_\phi(z'|x)$ and $p_\theta(x'|z')$ are neural network functions.

The **neighborhood reconstruction error** $\mathbb{E}_{z' \sim q_\phi(z'|x)} \log p_\theta(x'|z')$ 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 $\widetilde{\psi}_\omega(x)$, we may sample q_ϕ to obtain z' (the neighbor of x in the latent space). Assuming ψ^{-1} exists, we have $x' \sim p_\theta(x'|x)(\approx \widetilde{\psi}_\theta^{-1}(q_\phi(z'|x)))$. To make this practical, we can approximate x' by finding the closest data point to x' in random walk distance (due to the

[6] Though q depends on ϕ and ω, we will use $q_\phi := q_{\phi,\omega}$ to be consistent with existing VAE notation and to indicate that ω is not learned by variational inference.

aforementioned advantages of the latent space). In other words, we approximate empirically by

$$x' \approx \operatorname*{arg\,min}_{y \in A} |\widetilde{\psi}_\omega(y) - z'|_d^2 \,, \quad z' \sim q_\phi(z'|x), \tag{6}$$

where $A \subseteq X$ is the training batch.

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 ψ 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:

$$-D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) = -\log \frac{|\alpha \varSigma_*|}{|\widetilde{\mathbf{C}}_\phi|} + d - tr\{(\alpha \varSigma_*)^{-1} \widetilde{\mathbf{C}}_\phi\}, \tag{7}$$

where $\widetilde{\mathbf{C}}_\phi(x)$ is a neural network function, $\varSigma_*(x) = \mathrm{Cov}(B_d(\psi(x), \delta) \cap Z)$ is the covariance of the points in a neighborhood of $z = \psi(x) \in Z$, and α is a scaling parameter controlling the random walk step size. Note that the covariance $\widetilde{\mathbf{C}}_\phi(x)$ does not have to be diagonal, and in fact is most likely not. Combining Eqs. 6 and 7 we obtain Algorithm 1.

Since we use neural networks to approximate the random walk induced by the composition of $q_\phi(z'|x)$ and $p_\theta(x'|z')$, the generation procedure is highly parallelizable. This leads naturally to a sampling procedure for this random walk (Algorithm 2). We observe that the random walk enjoys rapid mixing properties — it only takes several iterations of the random walk to sample from all of $\mathcal{M}_Z$ [7].

Finally, we describe a practical method for computing the local bi-Lipschitz property. (In Section 6.4 we then perform comparisons with this method.) Let Z and X be the latent and generated data distributions of our model f (i.e. $f : Z \to X$). We define, for each $z \in Z$ and $k \in \mathbb{N}$, the function $\texttt{bilip}_k(z)$:

$$\texttt{bilip}_k(z) = \min\{K : \frac{1}{K} \leq \frac{d_x(f(z), f(z'))}{d_Z(z, z')} \leq K\},$$

for all $z' \in U_{z,k} \cap Z$, where d_X and d_Z are metrics on X and Z, and $U_{z,k}$ is the k-nearest neighborhood of z. Intuitively, increasing values of K characterize an increasing tendency to *stretch* or *compress* regions of space. By analyzing statistics of the local bi-Lipschitz measure at all points in a latent space Z, we gain insight into how well-behaved a mapping f is.

[7] For all experiments in Section 6, the number of steps required to draw from π is less than 10.

```
$\omega, \phi, \theta \leftarrow$ Initialize parameters
Obtain parameters $\omega$ for the approximate diffusion map $\widetilde{\psi}_\omega$ via SpectralNet [40]
while not converged do
  $A \leftarrow$ Random batch from $X$
  for $x \in A$ do
    $z \leftarrow p_\phi(z'|\widetilde{\psi}_\omega(x))$ {Random walk step}
    $x' \leftarrow \arg\min_{y \in A \setminus \{x\}} |\widetilde{\psi}_\omega(y) - z'|_d^2$ {Find batch neighbors}
    $g \leftarrow g + \frac{1}{|A|} \nabla_{\phi,\theta} \log p_\theta(x'|x)$ {Compute Eq. (5)}
  end for
  Update $\phi, \theta$ using $g$
end while
```

Algorithm 1: VDAE training

```
$X_0 \leftarrow$ Initialize with points $X_0 \subset X$; $t \leftarrow 0$
while $p(X_0) \not\approx \pi$ do
  for $x_t \in X$ do
    $z_{t+1} \sim p_\phi(z'|\widetilde{\psi}_\omega(x_t))$ {Random walk step}
    $x_{t+1} \sim p_\theta(x|z_{t+1})$ {Map back to input space}
  end for
  $t \leftarrow t + 1$
end while
```

Algorithm 2: VDAE sampling

4.4 Comparison to variational inference (VI)

Traditional VI involves maximizing the joint log-evidence of each data point x_i in a given dataset via the ELBO (see 2). Our method differs in both the training and evaluation steps.

In training, our setup is the same as above, except our likelihood is a conditional likelihood $p(x'|x)|_{U_x}$, where x' is in the diffusion neighborhood of x. Thus we maximize the local log-evidence of each data point $\mathbb{E}_{x' \sim p(x'|x_i)} \log p_\theta(x'|x_i)$, which can be lower bounded by Eq. (5). Thus our prior is $p(z'|x)$ and our posterior is $p(z'|x', x) = p(x', z'|x)/p(x'|x)$, and we train an approximate posterior $q_\phi(z'|x)$ and a recognition model $p_\theta(x'|z')$.

In evaluation, we draw from the stationary distribution $p(z')$ of the diffusion random walk on the latent manifold $\mathcal{M}_z = \psi(\mathcal{M}_x)$. We then leverage the latent variable structure of our model to draw a sample $x = p_\theta(x'|z')p(z')$, where $p_\theta(x'|x_i)$ is the recognition model.

5 Theory

In this section, we show that the desired diffusion and inverse diffusion maps $\psi : \mathcal{M}_X \to \mathcal{M}_Z$ and $\psi^{-1} : \mathcal{M}_Z \to \mathcal{M}_X$ can be approximated by neural networks, where the network complexity is bounded by quantities related to the intrinsic geometry of the manifold.

The capacity of the encoder $\widetilde{\psi}$ has already been considered in [39] and [29]. Thus we focus on the capacity of the decoder $\widetilde{\psi}^{-1}$. The following theorem is proved in Appendix A.3, based on the result in [17].

Theorem 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 $\mathbf{X} = \left[X_1, ..., X_m\right]$ be the set of all m extrinsic coordinates of the manifold. Then there exists a sparsely-connected ReLU network f_N, with $4DC_{\mathcal{M}_X}$ nodes in the first layer, $8dmN$ nodes in the second layer, and $2mN$ nodes in the third layer, and m nodes in the output layer, such that*

$$\|\mathbf{X}(\psi(x)) - f_N(\psi(x))\|_{L^2(\psi(\mathcal{M}_X))} \leq \sqrt{m} C_\psi / \sqrt{N}, \qquad (8)$$

where the norm is interpreted as $\|F\|^2_{L^2(\psi(\mathcal{M}))} := \int \|F(\psi(x))\|_2^2 d\psi(x)$. Here C_ψ depends on how sparsely $X(\psi(x))\big|_{U_i}$ can be represented in terms of the ReLU wavelet frame on each neighborhood U_i, and $C_{\mathcal{M}_X}$ on the curvature and dimension of the manifold $\mathcal{M}_X$.

Thm 1 guarantees the existence and size of a decoder network for learning a manifold. Together with the main theorem in [39], we obtain guarantees for both the encoder and decoder on manifold-valued data. The proof is built on two properties of ReLU neural networks: 1) their ability to split curved domains into small, almost Euclidean patches, 2) their ability to build differences of bump functions on each patch, which allows one to borrow approximation results from the theory of wavelets on spaces of homogeneous type. The proof also crucially uses the bi-Lipschitz property of the diffusion embedding [17]. The key insight of Thm 1 is that, because of the bi-Lipschitz property, the coordinates of the manifold in the ambient space $\mathbb{R}^m$ can be thought of as functions of the diffusion coordinates. We show that because each coordinate function X_i is Lipschitz, the ReLU wavelet coefficients of X_i are necessarily ℓ^1. This allows us to use the existing guarantees of [39] to complete the desired bound.

We also discuss the connections between the distribution at 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 [41], we characterize the covariance matrix $Cov(f_N(z)) := \mathbb{E}_{z \in q_\phi(z|x)}[f_N(z) f_N(z)^T]$. The following theorem is proved in Appendix A.3.

Theorem 2 *Let f_N be a neural network approximation to $\mathbf{X}$ as in Thm 1, such that it approximates the extrinsic manifold coordinates. Let $C \in \mathbb{R}^{m \times m}$ be the covariance matrix $C = \mathbb{E}_{z \in q_\phi(z|x)}[f_N(z) f_N(z)^T]$. Let $q_\phi(z|x) \sim N(\psi(x), \Sigma)$ with small enough Σ that there exists a patch $U_{z_0} \subset \mathcal{M}$ around z_0 satisfying the bi-Lipschitz property of [17], and such that $Pr(z \sim q_\phi(z|x) \notin \psi(U_{z_0})) < \epsilon$. Then the number of eigenvalues of C greater than ϵ 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.*

Thm 2 establishes the relationship between the covariance matrices used in the sampling procedure and their image under the decoder f_N to approximate

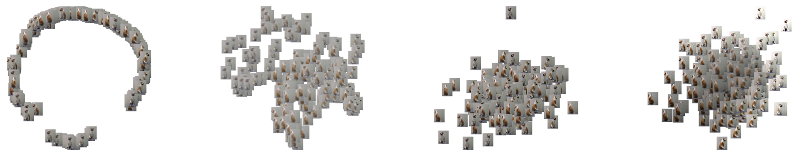


Fig. 3: We consider the rotating bulldog example. Images are drawn from the latent distribution and plotted in terms of the 2D latent space of each model. From left to right: VDAE, SVAE, β-VAE, WGAN.

ψ^{-1}. Similar to [41], we are able to sample according to a multivariate normal distribution in the latent space. Thus, the resulting cloud in the data space is distorted (to first order) by the local Jacobian of the map f_N. The key insight of Thm 2 is from combining this idea with the observation of [17]: that ψ^{-1} depends locally only on d of the coordinates in the D dimensional latent space.

6 Experimental results

In this section we explore various properties of the VDAE and compare it against several deep generative methods on a selection of real and synthetic datasets. Unless otherwise noted, all comparisons are against the Wasserstein GAN (WGAN), β-VAE, and hyperspherical VAE (SVAE). Each model is trained with the same architecture across all experiments (see Section A.6).

6.1 Video generation with rigid-body motion

We first consider the task of generating new frames from videos of rigid-body motion, and examine the latent spaces of videos with known topological structure to demonstrate the homeomorphic properties of the VDAE. We consider two examples, the rotating bulldog example [23] and the COIL-20 dataset. [31].

The rotating bulldog example consists of 200 frames of a color video (each frame is $100 \times 80 \times 3$) of a spinning figurine. The rotation of the bulldog and the fixed background create a data manifold that is topologically circular, corresponding to the single degree of variation (the rotation angle parameter) in the dataset. For all methods we consider a 2 dimensional latent space. In Fig. 3 we present 300 generated samples by displaying them on a scatter plot with coordinates corresponding to their latent dimensions z_1 and z_2. In the Appendix table A.1, we evaluate the quality of the generated images using the Frechet inception distance (FID).

The COIL-20 data set consists of 360 images of five different rotating objects displayed against on a black background (each frame is $448 \times 416 \times 1$). This yields several low dimensional manifolds, one for each object, and results in a difficult data set for traditional generative models given its small size and the complex

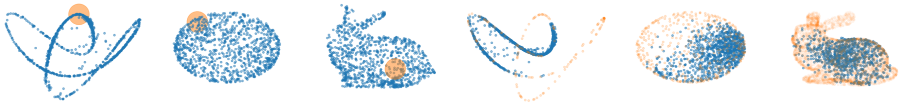


Fig. 4: From left to right, the first three scatterplots show examples of distributions reconstructed from a random walk on $\mathcal{M}_Z$ (via Algorithm 2) given a single seed point drawn from X. The next three are examples of a single burst drawn from $p_\theta(x|z)$. The distributions are a loop (a, d), sphere (b, e), and the Stanford bunny (c, f).

geometric structure. For all comparisons, we use 10 dimensional latent space. The resulting images are embedded with tSNE and plotted in Fig. A.3. Note that, while other methods generate images that topologically mimic the fixed latent distribution of the model (e.g. $\mathcal{N}(0, I_d)$, Uniform$(0,1)^d$), our method generates images that remain true to the actual topological structure of the dataset.

6.2 Data generation from uniformly sampled manifolds

In the next experiment, we visualize the results of the sampling procedure in Algorithm 2 on three synthetic manifolds. As discussed in 4.2, we randomly select an initial seed point, then recursively sample from $p_\theta(x'|x)$ to simulate a random walk on the manifold.

In fig. 4 (a-c) for three different manifolds, the location of the initial seed point is highlighted, then 20 steps of the random walk are taken, and the resulting generated points are displayed. The generated points remain on the manifold even after this large number of resampling iterations, and the distribution of sampled points converges to a uniform stationary distribution on the manifold. Moreover, we observe that this stationary distribution is reached quickly, within 5-10 iterations. In (d-f) of the same Fig. 4, we show $p_\theta(x'|x)$ by drawing a large number of points from a single-step random walk starting from the same seed point. As can be seen, a single step of $p_\theta(x'|x)$ covers a large part of the latent space.

6.3 Cluster conditional data generation

In this section, we deal with the problem of generating samples from data with multiple clusters in an unsupervised fashion (i.e. no a priori knowledge of the cluster structure). Clustered data creates a problem for many generative models, as the topology of the latent space (i.e. normal distribution) differs from the topology of the data space with multiple clusters.

First we show that our method is capable of generating new points from a particular cluster given an input point from that cluster. This generation is done in an unsupervised fashion, which is a different setting from the approach of

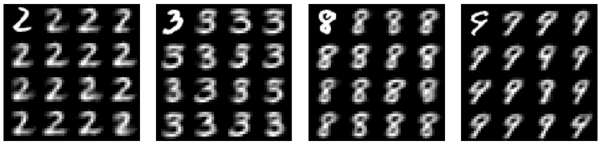


Fig. 5: An example of cluster conditional sampling with our method, given a seed point (top left of each image grid). The VDAE is able to produce examples via the random walk that stay approximately within the cluster of the seed point, without any supervised knowledge of the cluster.

conditional VAEs [42] that require training labels. We demonstrate this property on MNIST [22] in Figure 5, and show that the newly generated points after a short diffusion time remain in the same class as the seeded image.

The problem of addressing differing topologies between the data and the latent space of a generative model has been acknowledged in recent works on rejection sampling [2,45]. Rejection sampling of neural networks consists of generating a large collection of samples using a standard GAN, and then designing a probabilistic algorithm to decide in a *post-hoc* fashion whether the points were truly in the support of the data distribution $p(x)$.

In the following experiment, we compare to a standard example in the literature for rejection sampling in generative models (see [2]). The data consists of nine bounded spherical densities with significant minimal separation, lying on a 5×5 grid. A GAN struggles to avoid generating points in the gaps between these densities, and thus requires the post-sampling rejection analysis described in [2]. Conversely, our model creates a latent space that separates each of these clusters into their own coordinates and generates only points that in the neighborhood of the support of $p(x)$. Figure 6 shows that this results in significantly fewer points generated in the gaps between clusters. Our VDAE architecture is described in A.6, GAN and DRS-GAN architectures are as described in [2].

6.4 Quantitative comparisons of generative models

For this comparison, we consider seven datasets: three synthetic (circle, torus, Stanford bunny [44]) four involving natural images (rotating bulldog, Frey faces, MNIST, COIL-20). The β parameter in the β-VAE is optimized via a cross validation procedure. see Appendix for a complete description of the datasets. We report the mean and standard deviation of the Gromov-Wasserstein distance [27] and median bi-Lipschitz over 5 runs in Table 1. We further evaluate the results using kernel Maximum Mean Discrepancy [12], see Table A.2 in the Appendix.

By constraining our latent space to be the diffusion embedding of the data, our method finds a mapping that automatically enjoys the homeomorphic properties of an ideal mapping, and this is reflected in the low values of the local bi-Lipschitz constant. Conversely, other methods do not consider the topology of the data in the prior distribution. This is especially apparent in the β-VAE and SVAE,

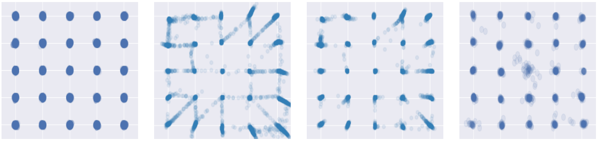


Fig. 6: Comparison of samples from our method against several others on a 5 × 5 Gaussian grid. Left-right are original data, GAN, DRS-GAN, and VDAE (our method). GAN and DRS-GAN samples taken from [2].

which must generate from the entirety of the input distribution X because they minimize a reconstruction loss. Interestingly, the mode collapse tendency of GANs alleviate the pathology of the bi-Lipschitz constant by allowing the GAN to focus on a subset of the distribution — but this comes at the cost of collapse to a few modes of the dataset. Our method is able to reconstruct the entirety of X while simultaneously maintaining a low local bi-Lipschitz constant.

7 Discussion

In this work, we have shown that VDAEs provide an intuitive, effective, and mathematically rigorous solution to *prior mismatch*, which is a common cause for posterior collapse in latent variable models. Unlike prior works, we do not require user specification of the prior — our method infers the prior geometry directly from the data, and we observe that it achieves state-of-the-art results on several real and synthetic datasets. Finally, our work points to several directions for future research: (1) can we leverage recent architectural advances to VAEs to further improve VDAE performance, and (2) can we leverage manifold learning techniques to improve latent representations in other methods?

Acknowledgements

This work was supported by NSF DMS grants 1819222 and 1818945. AC is also partially supported by NSF (DMS-2012266), and Russell Sage Foundation (grant 2196). XC is also partially supported by NSF (DMS-1820827), NIH (Grant R01GM131642), and the Alfred P. Sloan Foundation.

G-W	WGAN	β-VAE	SVAE	VDAE
Circle	14.9(6.8)	46.1(9.7)	7.9(2.2)	**2.6(1.3)**
Torus	6.4 (1.9)	11.7(1.6)	23.4(2.8)	**4.9 (0.5)**
Bunny	11.4(3.9)	32.8(5.9)	14.3(5.5)	**2.9(1.1)**
Bulldog	117.3(8.4)	61.3(9.7)	53.9(7.6)	**15.3(1.7)**
Frey	18.1(2.9)	19.8(4.6)	13.4(3.6)	**9.7(3.3)**
MNIST	**3.6(0.9)**	10.2(3.3)	15.2(4.9)	14.4(3.5)
COIL-20	16.5(2.4)	23.8(5.9)	32.1(4.9)	**11.8(2.1)**

biLip	WGAN	β-VAE	SVAE	VDAE
Circle	4.6	3.7	3.6	**3.1**
Torus	**3.3**	7.9	9.5	4.8
Bunny	5.6	34.4	35.6	**5.5**
Bulldog	17.4	7.6	12.9	**6.8**
Frey	37	33.3	39.4	**29.7**
MNIST	1.9	**1.6**	6.7	8.4
COIL-20	4.7	3.8	8.4	**3.1**

Table 1: Left: means and standard deviations of the Gromov-Wasserstein (G-W) distance between original and generated samples. Right: medians of the bi-Lipschitz measure.

References

1. Alemi, A.A., Poole, B., Fischer, I., Dillon, J.V., Saurous, R.A., Murphy, K.: An information-theoretic analysis of deep latent-variable models. CoRR **abs/1711.00464** (2017), `http://arxiv.org/abs/1711.00464`
2. Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., Odena, A.: Discriminator rejection sampling. arXiv preprint arXiv:1810.06758 (2018)
3. Bahlmann, C.: Directional features in online handwriting recognition. Pattern Recognition **39**(1), 115–125 (2006)
4. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in neural information processing systems. pp. 585–591 (2002)
5. Cloninger, A., Czaja, W., Doster, T.: The pre-image problem for laplacian eigenmaps utilizing l 1 regularization with applications to data fusion. Inverse Problems **33**(7), 074006 (2017)
6. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and computational harmonic analysis **21**(1), 5–30 (2006)
7. Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., Tomczak, J.M.: Hyperspherical variational auto-encoders. Uncertainty in Artificial Intelligence (UAI) (2018)
8. Fefferman, C., Mitter, S., Narayanan, H.: Testing the manifold hypothesis. Journal of the American Mathematical Society **29**(4), 983–1049 (2016)
9. Fisher, N.I., Lewis, T., Embleton, B.J.: Statistical analysis of spherical data. Cambridge university press (1993)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems (NIPS). vol. 27, pp. 2672–2680 (2014)
11. Goodfellow, I.J.: NIPS 2016 tutorial: Generative adversarial networks. CoRR **abs/1701.00160** (2017), `http://arxiv.org/abs/1701.00160`
12. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research **13**(Mar), 723–773 (2012)
13. Hamelryck, T., Kent, J.T., Krogh, A.: Sampling realistic protein conformations using local structural bias. PLoS Computational Biology **2**(9), e131 (2006)
14. He, J., Spokoyny, D., Neubig, G., Berg-Kirkpatrick, T.: Lagging inference networks and posterior collapse in variational autoencoders. CoRR **abs/1901.05534** (2019), `http://arxiv.org/abs/1901.05534`
15. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. Iclr **2**(5), 6 (2017)
16. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
17. Jones, P.W., Maggioni, M., Schul, R.: Manifold parametrizations by eigenfunctions of the laplacian and heat kernels. Proceedings of the National Academy of Sciences **105**(6), 1803–1808 (2008)
18. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine learning **37**(2), 183–233 (1999)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Krieger Lassen, N., Juul Jensen, D., Conradsen, K.: On the statistical analysis of orientation data. Acta Crystallographica Section A: Foundations of Crystallography **50**(6), 741–748 (1994)

21. Kwok, J.Y., Tsang, I.H.: The pre-image problem in kernel methods. IEEE transactions on neural networks **15**(6), 1517–1525 (2004)
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
23. Lederman, R.R., Talmon, R.: Learning the geometry of common latent variables using alternating-diffusion. Applied and Computational Harmonic Analysis **44**(3), 509–536 (2018)
24. Lindenbaum, O., Stanley, J., Wolf, G., Krishnaswamy, S.: Geometry based data generation. In: Advances in Neural Information Processing Systems. pp. 1400–1411 (2018)
25. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1429–1437 (2019)
26. Mardia, K.V.: Statistics of directional data. Academic press (2014)
27. Mémoli, F.: Gromov–wasserstein distances and the metric approach to object matching. Foundations of computational mathematics **11**(4), 417–487 (2011)
28. Mika, S., Schölkopf, B., Smola, A.J., Müller, K.R., Scholz, M., Rätsch, G.: Kernel pca and de-noising in feature spaces. In: Advances in neural information processing systems. pp. 536–542 (1999)
29. Mishne, G., Shaham, U., Cloninger, A., Cohen, I.: Diffusion nets. Applied and Computational Harmonic Analysis (2017)
30. Narayanan, H., Mitter, S.: Sample complexity of testing the manifold hypothesis. In: Advances in Neural Information Processing Systems. pp. 1786–1794 (2010)
31. Nene, S.A., Nayar, S.K., Murase, H., et al.: Columbia object image library (coil-20) (1996)
32. Peel, D., Whiten, W.J., McLachlan, G.J.: Fitting mixtures of kent distributions to aid in joint set identification. Journal of the American Statistical Association **96**(453), 56–63 (2001)
33. Razavi, A., van den Oord, A., Poole, B., Vinyals, O.: Preventing posterior collapse with delta-vaes. CoRR **abs/1901.03416** (2019), `http://arxiv.org/abs/1901.03416`
34. Razavi, A., Oord, A.v.d., Poole, B., Vinyals, O.: Preventing posterior collapse with delta-vaes. arXiv preprint arXiv:1901.03416 (2019)
35. Rey, L.A.P., Menkovski, V., Portegies, J.W.: Diffusion variational autoencoders. CoRR **abs/1901.08991** (2019), `http://arxiv.org/abs/1901.08991`
36. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
37. Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. In: Advances in Neural Information Processing Systems. pp. 5228–5237 (2018)
38. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural computation **10**(5), 1299–1319 (1998)
39. Shaham, U., Cloninger, A., Coifman, R.R.: Provable approximation properties for deep neural networks. Applied and Computational Harmonic Analysis **44**(3), 537–557 (2018)
40. Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y.: Spectralnet: Spectral clustering using deep neural networks. arXiv preprint arXiv:1801.01587 (2018)
41. Singer, A., Coifman, R.R.: Non-linear independent component analysis with diffusion maps. Applied and Computational Harmonic Analysis **25**(2), 226–239 (2008)

42. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in neural information processing systems. pp. 3483–3491 (2015)
43. Tomczak, J.M., Welling, M.: Vae with a vampprior. arXiv preprint arXiv:1705.07120 (2017)
44. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. pp. 311–318 (1994)
45. Turner, R., Hung, J., Saatci, Y., Yosinski, J.: Metropolis-hastings generative adversarial networks. arXiv preprint arXiv:1811.11357 (2018)
46. Wainwright, M.J., Jordan, M.I., et al.: Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning **1**(1–2), 1–305 (2008)
47. Xu, J., Durrett, G.: Spherical latent spaces for stable variational autoencoders. arXiv preprint arXiv:1808.10805 (2018)
48. Zhao, S., Song, J., Ermon, S.: Infovae: Information maximizing variational autoencoders. CoRR **abs/1706.02262** (2017), `http://arxiv.org/abs/1706.02262`

A Appendix

A.1 Derivation of Local Evidence Lower Bound (Eq. 5)

We begin with taking the log of the random walk transition likelihood,

$$
\begin{aligned}
\log p_\theta(x'|x) &= \log \int_z^{\prime} p_\theta(x', z'|x) dz' && \text{(A.1)}\\
&= \log \int_z p_\theta(x'|z', x) p(z'|x) \frac{q(z')}{q(z')} dz' && \text{(A.2)}\\
&= \log \mathbb{E}_{z' \sim q(z')} \left[p_\theta(x'|z', x) \frac{p(z'|x)}{q(z')} \right] && \text{(A.3)}\\
&\geq \mathbb{E}_{z' \sim q(z')} \left[\log p_\theta(x'|z', x) \right] + \mathbb{E}_{z' \sim q(z')} \left[\log \frac{p(z'|x)}{q(z')} \right] && \text{(A.4)}\\
&\geq \mathbb{E}_{z' \sim q(z')} \left[\log p_\theta(x'|z', x) \right] + D_{KL}[q(z')||p(z'|x)] && \text{(A.5)}
\end{aligned}
$$

where $q(z')$ is an arbitrary distribution. We let $q(z')$ to be the conditional distribution $q(z'|x)$. Furthermore, if we make the simplifying assumption that $p_\theta(x'|z', z) = p_\theta(x'|z')$, then we obtain Eq. 5

$$
\log p_\theta(x'|x) \geq -D_{KL}(q_\phi(z'|x)||p_\theta(z'|x)) + \mathbb{E}_{z' \sim q_\phi(z'|x)} \log p_\theta(x'|z'). \qquad \text{(A.6)}
$$

A.2 Results in [17]

To state the result in [17], 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 $d_{\mathcal{M}}$, and the geodesic ball centering at x with radius r by $B_{\mathcal{M}}(x, r)$. Under (C1), for each point $x \in \mathcal{M}$, there exists $r_{\mathcal{M}}(x)$ which is the inradius, that is, r is the largest number s.t. $B_{\mathcal{M}}(x, r)$ is contained $\mathcal{M}$.

Let $\triangle_{\mathcal{M}}$ be the Laplacian-Beltrami operator on $\mathcal{M}$ with Neumann boundary condition, which is self-adjoint on $L^2(M, \mu)$, μ being the Riemannian volume given by g. Suppose that $\mathcal{M}$ is re-scaled to have volume 1. The next condition we need concerns the spectrum of the manifold Laplacian

(C2) $\triangle_{\mathcal{M}}$ has discrete spectrum, and the eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots$ satisfy the Weyl's estimate, i.e. exists constant C which only depends on $\mathcal{M}$ s.t.

$$
|\{j : \ \lambda_j \leq T\}| \leq C T^{d/2}.
$$

Let ψ_j be the eigenfunction associated with λ_j, $\{\psi_j\}_j$ form an orthonormal bases of $L^2(M, \mu)$. The last condition is

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

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

Theorem 3 (Thm 2 [17], simplified version) *Under the above setting and assume (C1)-(C2), then there are positive constants c_1, c_2, c_3 which only depend on $\mathcal{M}$ and g, s.t. for any $x \in \mathcal{M}$, $r_{\mathcal{M}}(x)$ being the inradius, there are d eigenfunctions of $\triangle_{\mathcal{M}}$, $\psi_{j_1}, \cdots, \psi_{j_d}$, which collectively give a mapping $\Psi : \mathcal{M} \to \mathbb{R}^d$ by*

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

satisfying that $\forall y, y' \in B(x, c_1 r_{\mathcal{M}}(x))$,

$$c_2 r_{\mathcal{M}}(z)^{-1} d_{\mathcal{M}}(y, y') \leq \|\Psi_x(y) - \Psi_x(y')\| \leq c_3 r_{\mathcal{M}}(z)^{-1-d/2} d_{\mathcal{M}}(y, y').$$

That is, Ψ is bi-Lipschitz on the neighborhood $B(x, c_1 r_{\mathcal{M}}(x))$ with the Lipschitz constants indicated as above. The subscript x in Ψ_x emphasizes that the indices $j_1, \cdots, j_d$ may depend on x.

A.3 Proofs

Proof (of Thm 1). The proof of Thm 1 is actually a simple extension of the following Thm, Thm 4, which needs to be proved for each individual extrinsic coordinate X_k, hence the additional factor of m coming from the $L2$ norm of m functions.

Theorem 4 *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_N, with $4DC_{\mathcal{M}}$ nodes in the first layer, $8dN$ nodes in the second layer, and $2N$ nodes in the third layer, such that*

$$\|X - f_N\|_{L^2(\psi(\mathcal{M}))} \leq \frac{C_\psi}{\sqrt{N}} \tag{A.7}$$

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

Proof (of Thm 4).

The proof borrows from the main theorem of [39]. 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 [39]. Let $C_{\mathcal{M}}$ be the number of neighborhoods $U_i = B(x_i, \delta) \cap \mathcal{M}$ needed to cover $\mathcal{M}$ such that $\forall x, y \in U_i$, $(1-\epsilon)\|x - y\| \leq d_{\mathcal{M}}(x, y) \leq (1+\epsilon)\|x - y\|$. Here, we choose $\delta = \min(\delta_{\mathcal{M}}, \kappa^{-1}\rho)$ where $\delta_{\mathcal{M}}$ is the largest δ that preserves locally Euclidean neighborhoods and $\kappa^{-1}\rho$ is the smallest value from [17] such that every neighborhood U_i has a bi-Lipschitz set of diffusion coordinates.

Because of the locally bi-Lipschitz guarantee from [17], we know for each U_i there exists an equivalent neighborhood $\widetilde{\psi}(U_i)$ in the diffusion map space, where

$\widetilde{\psi}(x) = \left[\psi_{i_1}(x), ..., \psi_{i_d}(x)\right]$. 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. $d_{\mathcal{M}}(\cdot,\cdot)$ on U_i.

First, we note that as in [39], the first layer of a neural network is capable of using $4D$ units to select the subset of d coordinates $\widetilde{\psi}(x)$ from $\psi(x)$ for $x \in U_i$ and zeroing out the other $D-d$ coordinates with ReLU bump functions. Then we can define $X(\widetilde{\psi}(x)) = X(\psi(x))$ on $x \in U_i$.

Now to apply the theorem from [39], we must establish that $X\big|_{U_i} : \widetilde{\psi}(U_i) \to \mathbb{R}$ 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 $\widetilde{\psi}$ is invertible on $\widetilde{\psi}(U_i)$. 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 cooresponds to $\psi(x)$. we also know that $\forall x, y \in U_i$,

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

where $\nabla X(z)$ is understood to be the gradient of $X(z)$ at the point $z \in \mathcal{M}$. This means $X(\widetilde{\psi}(x))$ is a Lipschitz function w.r.t. $\widetilde{\psi}(x)$. Because $X(\widetilde{\psi}(x))$ Lipschitz continuous, it can be approximated by step functions on a ball of radius $2^{-\ell}$ to an error that is at most $\frac{\max_{z\in U_i}\|\nabla X(z)\|}{1-\epsilon} 2^{-\ell}$. 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 $\widetilde{\psi}(U_i)$ is compact, gives the fact that on $\widetilde{\psi}(U_i)$, set of ReLU wavelet coefficients is in ℓ^1. And from [39], if on a local patch the function is expressible in terms of ReLU wavelet coefficients in ℓ^1, then there is an approximation rate of $\frac{1}{\sqrt{N}}$ for N ReLU wavelet terms.

Proof (of Thm 2). We borrow from [41] 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 [41] 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 insize $\psi(U_{z_0})$ is $J_{z_0} \Sigma J_{z_0}^T$, with an $O(\epsilon)$ perturbation for the fact that ϵ 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 $\widetilde{\psi}(x)$ as in the proof of Thm 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.

A.4 Spectral Net

A.5 Additional Experimental Result

To evaluate the quality of the generated images in the Bulldog dataset, we use the Frechet inception distance (FID). We train the different generative models 5 times and compute the FID between source and generated images. In table A.1 we present the mean and standard deviations of the FID.

FID	GAN	VAE	SVAE	VDAE
Bulldog	264.4(18.4)	245.7(14.7)	400.6 (6.2)	144.3(12.6)

Table A.1: Frechet inception distance (FID) on the Bulldog dataset, mean and standard deviation.

MMD	GAN	VAE	SVAE	VDAE
Circle	9.3(11.1)	8.3(4.4)	8.1 (4.2)	7.3(4.3)
Torus	12.3 (4.7)	63.3 (12.9)	84.5(11.7)	41.9 (4.1)
Bunny	175.6(68.6)	725.8(3.8)	601.7(41.1)	3.6(0.3)
Bulldog	741.8(88)	167.3(16.4)	213.7(13.1)	9.68(3.44)
Frey	34.9(5.1)	39.3(6.1)	29.4	47.0
MNIST	3.5(0.6)	27.9(1)	20.6(1.2)	5.79(0.3)
COIL-20	3.3(0.9)	39.2(9.6)	55.7(4.7)	7.4(1.07)

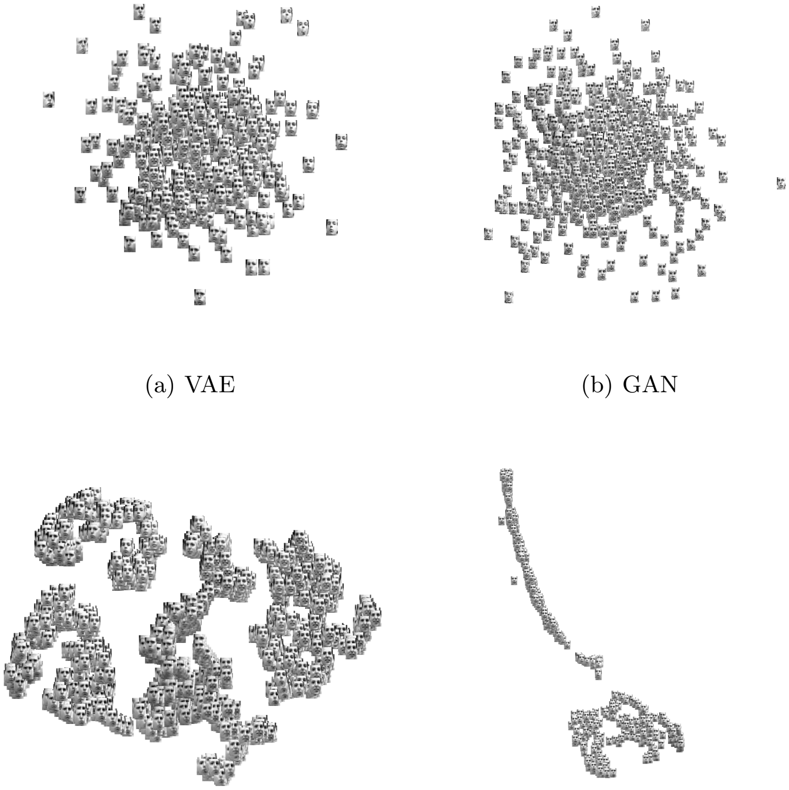
Table A.2: Measures of similarity between training data and generated data using Maximum Mean Discrepancy. Comparisons are across a variety of synthetic and real data sets

A.6 Experimental Architectures

For the circle, torus, Stanford bunny, Frey faces [8], and the 5x5 spherical density datasets, we used a single 500-unit hidden layer network for all models used in the paper (i.e. decoder, encoder, generator, discriminator, for the VAE, Wasserstein GAN, hyperspherical VAE, and our method).

As higher dimensional datasets, we used a slightly larger architecture for the MNIST, COIL-20, and rotating bulldog datasets: two hidden-layer decoder/generators of width 1024 and 2048, and two hidden-layer encoder/discriminators of width 2048 and 1024. All activations are still ReLU.

[8] https://cs.nyu.edu/ roweis/data.html



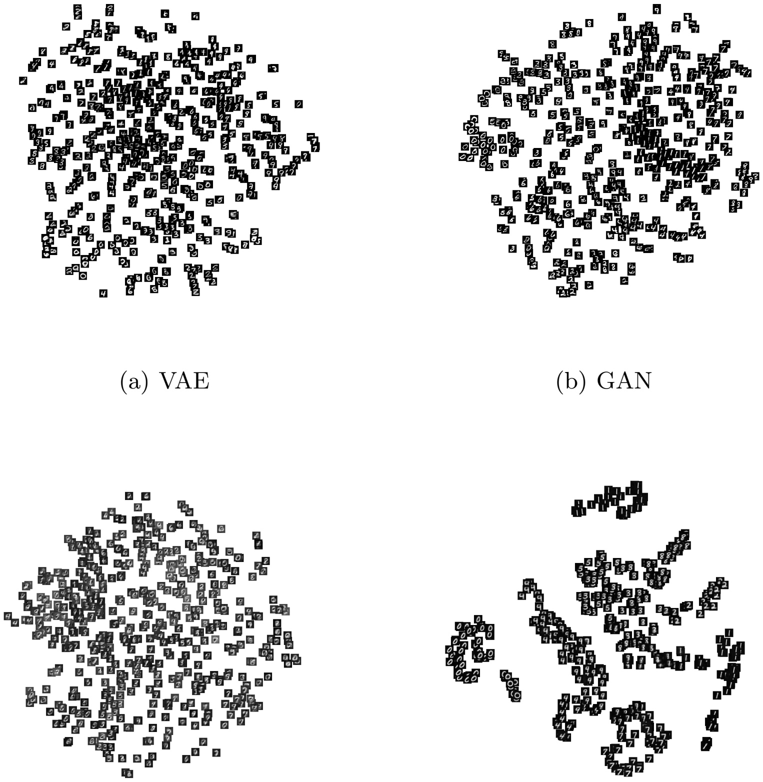
(a) VAE

(b) GAN

(c) SVAE

(d) VDAE

Fig. A.1: A tSNE plot of generated images from Frey data set. While the images from the VAE and GAN are compelling, they do not capture the geometric structure of the Frey faces dataset. This structure is much more apparent in the images generated by SVAE and VDAE. In particular, the VDAE has captured a linear structure in the data, which reflects the fact that the dataset was created from a video.



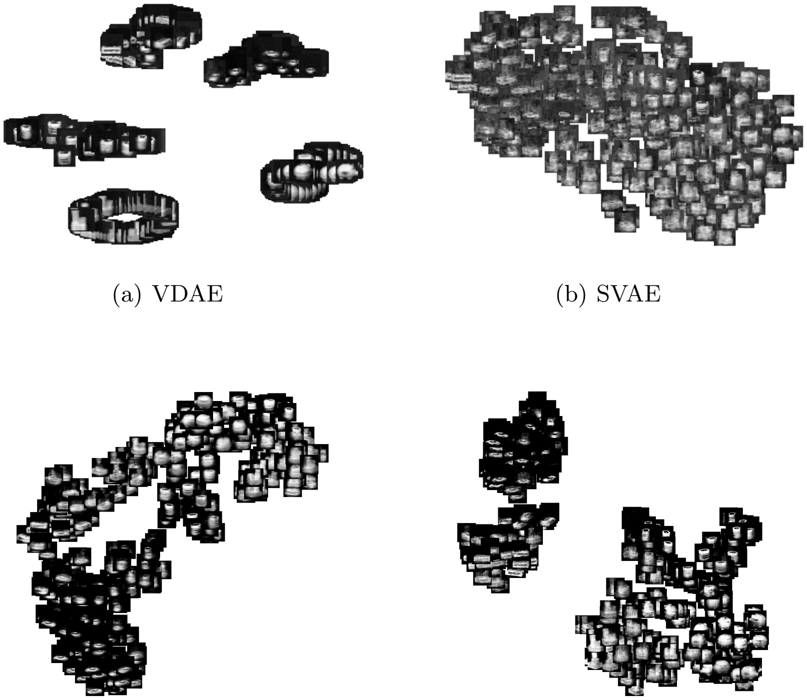
(a) VAE

(b) GAN

(c) SVAE

(d) VDAE

Fig. A.2: A tSNE plot of generated images from Frey data set. Like with Fig. A.1 (Frey faces), the images generated by VAE, GAN, and SVAE have a unimodal distribution that does not capture the clustered structure of the MNIST dataset. VDAE, on the other hand, organizes the digits into clear clusters, and does not generate from regions where there is low support in the training distribution.



(a) VDAE

(b) SVAE

(c) β-VAE

(d) WGAN

Fig. A.3: A tSNE embedding of 360 generated images from COIL-20 data set.