Variational Autoencoder with Learned Latent Structure

Marissa Connor **Gregory Canal** **Christopher Rozell**

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
(marissa.connor,gregory.canal,crozell)@gatech.edu

Abstract

The manifold hypothesis states that high-dimensional data can be modeled as lying on or near a low-dimensional, nonlinear manifold. Variational Autoencoders (VAEs) approximate this manifold by learning mappings from low-dimensional latent vectors to high-dimensional data while encouraging a global structure in the latent space through the use of a specified prior distribution. When this prior does not match the structure of the true data manifold, it can lead to a less accurate model of the data. To resolve this mismatch, we introduce the Variational Autoencoder with Learned Latent Structure (VAELLS) which incorporates a learnable manifold model into the latent space of a VAE. This enables us to learn the nonlinear manifold structure from the data and use that structure to define a prior in the latent space. The integration of a latent manifold model not only ensures that our prior is well-matched to the data, but also allows us to define generative transformation paths in the latent space and describe class manifolds with transformations stemming from examples of each class. We validate our model on examples with known latent structure and also demonstrate its capabilities on a real-world dataset.[1]

---

1 INTRODUCTION

Generative models represent complex data distributions by defining generator functions that map low-dimensional latent vectors to high-dimensional data outputs. In particular, generative models such as variational autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014) and generative adversarial networks (Goodfellow et al., 2014) sample from a latent space with a specified prior distribution in order to generate new, realistic samples. VAEs have the additional benefit of an encoder that maps data inputs into a latent space. This latent space embedding of data points makes VAEs an effective tool for generating and understanding variations in natural data.

According to the manifold hypothesis, high-dimensional data can often be modeled as lying on or near a low-dimensional, nonlinear manifold (Fefferman et al., 2016). There are many manifold learning techniques that compute embeddings of high-dimensional data but very few of them have the ability to generate new points on the manifold (Tenenbaum et al., 2000; Roweis and Saul, 2000; Dollár et al., 2007; Bengio and Monperrus, 2005). Meanwhile, VAEs can both embed high-dimensional data in a low-dimensional space and generate outputs from that space, making them a convenient model for representing low-dimensional data manifolds.

However, there are several aspects of the traditional VAE framework that prevent it from faithfully representing complex natural variations on manifolds associated with separate data classes. First, VAEs enforce a global structure in the latent space through the use of a prior distribution, and that prior may not match the true data manifold; this model mismatch can result in a less accurate generative model of the data. Second, natural paths, which interpolate or extrapolate natural data variations, are poorly defined in the latent space of traditional VAEs. In many cases, the data transformations are defined using linear paths in a Euclidean latent space (Radford et al., 2016). These simple paths

can diverge from the true data manifold, leading to interpolated points that result in unrealistic decoded image outputs. Finally, traditional VAEs encourage points from all data classes to cluster around the origin in the latent space (Kingma and Welling, 2013). Without adequate class separation, traversing the latent space can easily result in a change of class, making it difficult to learn identity-preserving transformations and subsequently use them to understand within-class relationships.

In this paper we incorporate a generative manifold model known as *transport operators* (Culpepper and Olshausen, 2009) into the latent space of a VAE, enabling us to learn the manifold structure from the data and use that structure to define an appropriate prior in the latent space. This approach not only ensures that the prior is well-matched to the data, but it also allows us to define nonlinear transformation paths in the latent space and describe class manifolds with transformations stemming from examples of each class. Our model, named Variational Autoencoder with Learned Latent Structure (VAELLS), more effectively represents natural data manifolds than existing techniques, leads to more accurate generative data outputs, and results in a richer understanding of data variations.

2 METHODS

2.1 Transport Operators

The exact structure of natural data manifolds is typically unknown and manifold learning techniques have been introduced to discover the low dimensional structure of data. While there are a variety of manifold learning techniques (Tenenbaum et al., 2000; Roweis and Saul, 2000; Dollár et al., 2007; Bengio and Monperrus, 2005), we desire a manifold model that allows us to learn the data structure, generate points outside of the training set, and map out smooth manifold paths through the latent space. The transport operator manifold model (Culpepper and Olshausen, 2009) satisfies these requirements by defining a manifold using learned operators that traverse the low-dimensional manifold surface.

Specifically, the basis of the transport operator approach is the linear dynamical system model $\dot{z} = Az$, which defines the dynamics of point $z \in \mathbb{R}^d$ through $A \in \mathbb{R}^{d \times d}$. The solution to this differential equation defines a temporal path given by $z_t = \text{expm}(At)z_0$, where expm is the matrix exponential. This path definition can be generalized to define the transformation between any two points $z_0, z_1 \in \mathbb{R}^d$ on a low-dimensional natural data manifold without an explicit time component by defining $z_1 = \text{expm}(A)z_0 + n$, where n is

white Gaussian noise. To allow for different geometrical characteristics at various points on the manifold, our model should have the flexibility to define a different dynamics matrix A between each pair of points. The transport operator technique achieves this property by defining a dynamics matrix that can be decomposed as a weighted sum of M transport operator dictionary elements ($\Psi_m \in \mathbb{R}^{d \times d}$):

$$A = \sum_{m=1}^{M} \Psi_m c_m. \tag{1}$$

The transport operators $\{\Psi_m\}$ constitute a set of primitives that describe local characteristics over the entire manifold, while for each pair of points (i.e., at each manifold location) the geometry is governed by a small subset of operators through coefficients $c \in \mathbb{R}^M$ specific to each pair.

The final generative manifold model is developed by incorporating the decomposable transformation matrix into the linear dynamical system model and imposing a sparsity-inducing Laplace prior on the coefficients. This Laplace prior encourages the manifold geometry between any given pair of points to be represented by only a small subset of dictionary elements.

$$z_1 = \text{expm}\left(\sum_{m=1}^{M} \Psi_m c_m\right) z_0 + n$$
$$n \sim \mathcal{N}(0, I) \quad c_m \sim \text{Laplace}(0, b). \tag{2}$$

Following the unsupervised algorithm in Culpepper and Olshausen (2009), the transport operators can be learned from pairs of points on the same manifold using descent techniques that alternate between inferring the coefficients and updating the transport operators. The manifold structure can be represented by a combination of these learned transport operators and samples that are known to exist on the manifold. The transport operators constrain motion to only paths along the learned manifold and labeled samples provide starting points from which the operators can generate new samples on the manifold. In this paper we incorporate this structured model into a VAE to encourage the latent space structure to adapt to the true data manifold.

2.2 Variational Autoencoder

The VAE model learns a low-dimensional latent representation by defining a generator function $g : \mathcal{Z} \to \mathcal{X}$ that maps latent points $z \in \mathbb{R}^d$ to high-dimensional data points $x \in \mathbb{R}^D$. The desired objective for training a VAE is maximizing the log-likelihood of a dataset $X = \{x_1, \ldots, x_N\}$ given by $\frac{1}{N} \log p(X) = \frac{1}{N} \sum_{i=1}^{N} \log \int p(x_i, z) dz$. However, this objective is difficult to maximize, especially when parameterized by a

neural network. To address this complication, VAEs instead maximize the Evidence Lower Bound (ELBO) of the marginal likelihood of each datapoint x_i:

$$\log p(x_i) \geq \mathcal{L}(x_i) = \mathbb{E}_{z \sim q_\phi(z|x_i)}[-\log q_\phi(z \mid x_i) + \log p_\theta(x_i, z)], \quad (3)$$

where $q_\phi(z \mid x)$ is a variational approximation of the true posterior, parameterized by ϕ. In the VAE neural network model, ϕ represents the weights of an encoder network $f_\phi(x)$.

Kingma and Welling (2013) developed an efficient method to approximate the ELBO by introducing the *reparameterization trick* that enables the stochastic latent variable z to be represented by a deterministic function $z = h_\phi(x, \varepsilon)$, where ε is an auxiliary random variable with a parameter-free distribution. In the traditional VAE framework, the variational posterior is selected to be a multivariate Gaussian distribution, meaning that z is reparameterized around the encoded point $z = f_\phi(x) + \sigma\varepsilon$ where $\varepsilon \sim \mathcal{N}(0, I)$. Additionally, the prior $p_\theta(z)$ is modeled as a zero-mean isotropic normal distribution that encourages the clustering of latent points around the origin.

2.3 Variational Autoencoder with Learned Latent Structure

In VAELLS, we fuse the versatile manifold learning capabilities of transport operators with the powerful generative modeling of VAEs. Specifically, we integrate transport operators into both the VAE variational posterior distribution and the prior in order to learn a latent probabilistic model that is adapted directly from the data manifold.

We start with the expanded ELBO from Kingma and Welling (2013):

$$\mathcal{L}(x) = \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\phi(z \mid x) \right]. \quad (4)$$

For the likelihood $p_\theta(x \mid z)$, we follow prior work and choose an isotropic normal distribution with mean defined by the decoder network $g_\theta(z)$ and fixed variance σ^2, which has worked well in practice.

Our first key contribution lies in the selection of the variational posterior, which we choose as the family of manifold distributions parameterized by learned transport operators described in Section 2.1. Intuitively, this posterior measures the probability of vector z lying on the manifold in the local neighborhood of the latent encoding of x where the structure of the manifold is defined by learned transport operators. We encode the latent coordinates of x with a neural network $f_\phi(\cdot)$ and then draw a sample from $q_\phi(z \mid x)$.

To approximate (4) with sampling, first let $L_x(z) \equiv \log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\phi(z \mid x)$ and note that by marginalizing over transport operator coefficients c we have $\mathbb{E}_{z \sim q_\phi(z|x)}[L_x(z)] = \mathbb{E}_{z,c \sim q_\phi(z,c|x)}[L_x(z)]$ which allows us to estimate (4) by sampling from $q_\phi(z, c \mid x)$. We draw a sample from $q_\phi(z, c \mid x)$ in two steps: first, as defined in the generative model in (2), we sample a set of coefficients $\widehat{c}$ from a factorized Laplace distribution $q(c)$, and then sample z from $q_\phi(z \mid \widehat{c}, x)$. Both of these sampling steps can be achieved with deterministic mappings on parameter-free random variates, allowing for the use of the reparameterization trick. Specifically,

$$\widehat{c} = l(u; b) \quad u \sim \text{Unif}\left(-\frac{1}{2}, \frac{1}{2}\right)^M$$
$$z = T_\Psi(\widehat{c}) f_\phi(x) + \gamma\varepsilon \quad \varepsilon \sim \mathcal{N}(0, I), \quad (5)$$

where $l(u; b)$ is a mapping described in Appendix A with Laplace scale parameter b and $T_\Psi(c) = \text{expm}\left(\sum_{m=1}^M \Psi_m c_m\right)$.

Fig. 1a shows this sampling process where the data x is encoded to a mean location $\mu = f_\phi(x)$. The latent vector z is the result of transforming μ by $T_\Psi(\widehat{c})$, which moves the vector on random paths along the manifold, and adding Gaussian noise specified by γ.

The resulting transport operator variational posterior follows as

$$q(c) = \left(\frac{1}{2b}\right)^M \prod_{m=1}^M \exp(-\frac{|c_m|}{b})$$
$$q_\phi(z \mid c, x) \sim \mathcal{N}\left(T_\Psi(c) f_\phi(x), \gamma^2 I\right)$$
$$q_\phi(z \mid x) = \int_c q_\phi(z \mid c, x) q(c) dc \quad (6)$$
$$\approx \max_c q_\phi(z \mid c, x) q(c), \quad (7)$$

where the approximation in (7) is motivated by the fact that the sparsity-inducing Laplace prior on c typically results in joint distributions with z that are tightly peaked in coefficient space, as described in Olshausen and Field (1997) (see Appendix A for details). This approximation is widely used in sparse dictionary learning for computational efficiency. The inferred coefficients c^* that maximize $q_\phi(z \mid c, x) q(c)$ define the estimated transformation between the encoded latent coordinates and the sampled point. The dotted green line in Fig. 1b shows a visualization of the inferred transformation path between $f_\phi(x)$ (the black dot) and z (the green dot).

Our next key contribution lies in the construction of a prior distribution learned directly from the underlying data manifold using transport operators. To gain intuition about this prior, imagine a set of N_a *anchor*
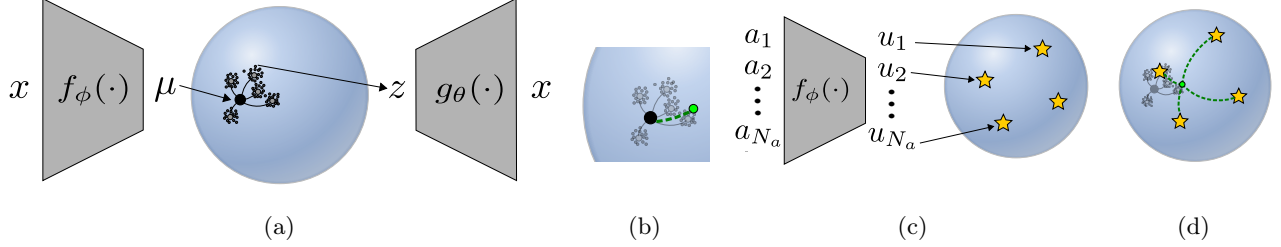
Figure 1: Visualizations of the VAELLS model: (a) Posterior sampling process using transport operator-generated paths and Gaussian noise. (b) Transformation path (green dotted line) inferred between $f_\phi(x)$ (black dot) and z (green dot) when computing the posterior. (c) Encoding of anchor points into the latent space. (d) Transformation paths (green dotted lines) inferred between $f_\phi(a_i)$ (yellow stars) and z (green dot) when computing the prior.

points in the data space that correspond to samples on the desired manifold. The data manifold structure is represented by a combination of these anchor points, encoded in the latent space, and the learned transport operators that can extrapolate the manifold structure in the latent neighborhood of each of them. Fig. 1c shows a set of anchor points encoded into the latent space which represent the scaffold off of which the manifold structure is built. The prior is defined by the same probabilistic manifold model used in the variational posterior, but starting at each anchor point a_i rather than x. This prior requires that paths be inferred between z and each encoded anchor point $u_i = f_\phi(a_i)$, visualized in Fig. 1d as green dotted lines. The overall prior density for z is then defined as

$$p_\theta(z) = \frac{1}{N_a} \sum_{i=1}^{N_a} q_\phi(z \mid a_i). \qquad (8)$$

The introduction of anchor points into the definition of the prior provides a unique opportunity for the user to define the classes within which they want to learn natural, identity-preserving transformations. If desired, the user can choose a set of anchor points, independent of class labels, to define the full data manifold and learn transformations throughout the entire dataset. If the user wants to learn identity-preserving transformations on specific class manifolds, they can separate the anchor points into classes and define the prior with respect to class-specific anchor points. Even when the prior is defined with respect to anchor points in each class individually, the set of learned transport operators is shared over all classes. Finally, if the user desires to learn a manifold representing specific transformations of individual samples, they can select anchor points that are sampled from transformation paths of specific example data points. In practice, the anchor points are initialized by either uniformly sampling the manifold space or through manual selection by a practitioner (e.g., by selecting several anchors per data class) and they can be updated throughout training.

In Tomczak and Welling (2018), a prior is adopted that is similarly composed of a sum of variational posterior terms, and is motivated as an approximation to an optimal prior that maximizes the standard ELBO. However, our motivation for the prior in (8) as a direct sampling of the data manifold is novel, and our prior more directly aligns with the data manifold because it is constructed from operators that traverse the manifold itself.

The final addition to the VAELLS objective is a Frobenius-norm regularizer on the dictionary magnitudes as used in the original transport operator objective (Culpepper and Olshausen, 2009). This prevents the magnitudes of the dictionaries from increasing without bound and helps identify how many transport operators are necessary to represent the manifold by reducing the magnitude of operators that are not used to generate paths between samples. We define the loss function to be minimized as the approximate negative ELBO in (4), restated here for convenience:

$$L_x(z) \equiv \log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\phi(z \mid x)$$
$$\mathcal{L}_{VAELLS}(x) = -\mathbb{E}_{u,\varepsilon}\left[L_x(T_\Psi(l(u;b))f_\phi(x) + \gamma\varepsilon)\right]$$
$$+ \frac{\eta}{2} \sum_{m=1}^{M} \|\Psi_m\|_F^2, \qquad (9)$$

We optimize this loss simultaneously over encoder-decoder networks f_ϕ and g_θ, anchor points $\{a_i\}_{1:N_a}$, and transport operators Ψ. The implementation details of our ELBO and its optimization are described in more detail in Appendix A.

3 RELATED WORK

There are currently many adaptations of the original VAE that handle a subset of the limitations addressed by VAELLS, such as learning the prior from the data, defining continuous paths in the latent space, and separating the individual class manifolds. Table 1 provides

a comparison of techniques which we describe in detail below.

Traditionally, the latent space prior is defined as a Gaussian distribution for model simplicity (Kingma and Welling, 2013). However this prior encourages all points to cluster around the origin which may not occur in the natural data manifold. A mismatch between the latent prior distribution of a VAE and the data manifold structure can lead to over regularization and poor data representation. Other models incorporate more complex latent structures such as hyperspheres (Davidson et al., 2018), tori (Rey et al., 2019; Falorsi et al., 2019), and hyperboloids (Mathieu et al., 2019; Nagano et al., 2019; Skopek et al., 2020). These models have demonstrated their suitability for certain datasets by choosing a prior that is the best match out of a predefined set of candidates.

However, these methods are only capable of modeling a limited number of structured priors and are not able to adapt the prior to match the data manifold itself. This is a serious drawback since, in most practical cases, the latent structure of data is unlikely to easily fit a predefined prior. The variational diffusion autoencoder (VDAE) and the $\mathcal{R}$-VAE define the latent space prior directly from the data using Brownian motion on a Riemannian manifold (Li et al., 2020; Kalatzis et al., 2020). The VampPrior is defined using a sum of variational posterior distributions with learnable hyperparameters (Tomczak and Welling, 2018). As noted in discussion around (8), this definition of a prior incorporating variational posterior terms is the same as what is used in VAELLS. However, the form of the variational posteriors differ between VampPrior and VAELLS with VAELLS using the structured manifold model in (6). This affords the VAELLS model the additional benefits noted below.

In addition to differences in how latent space structure is represented, there are various approaches for defining natural paths in this space. In the simplest case, paths in VAEs with a Euclidean latent spaces are modeled as linear paths. Some methods define geodesic transformation paths in the latent space. This can be done by incorporating a structured latent prior, like a hypersphere, on which geodesic paths are natural to compute (Davidson et al., 2018). Interpolated geodesic paths can also be estimated in a Euclidean latent space using an estimated Riemannian metric (Kalatzis et al., 2020; Arvanitidis et al., 2018; Chen et al., 2018; Shao et al., 2018). However, these methods are limited to defining extrapolated paths by random walks in the latent space rather than structured paths. Finally, there are other methods that lack straightforward definitions for how to compute continuous paths from one point to another (Tomczak and Welling, 2018; Rey et al., 2019).

Another limitation of most VAE models is that they do not encourage class separation and therefore generated paths often do not represent natural identity-preserving transformations on separate class data manifolds. This makes it difficult to understand the within-class relationships in the data. Some techniques encourage class separation through the choice of a prior that does not encourage data clustering (Davidson et al., 2018) but they do not explicitly define separate class manifolds. By defining the prior structure with respect to anchor points on specified data manifolds, VAELLS has the flexibility to define which manifolds it wants to learn transformations on. This results in a latent space structure where transformations correspond to identity-preserving variations in the data.

Two models that have notable similarities to ours are the Lie VAE (Falorsi et al., 2019) and the Manifold Autoencoder (Connor and Rozell, 2020). Both models also use Lie group representations of transformations in the latent space. The Lie VAE model encodes the data into latent variables that are elements in a Lie group which represent transformations of a reference object. This model requires the type of Lie group transformations that the network will represent (e.g., $SO(3)$) to be specified prior to training which may result in a model mismatch. The Manifold Autoencoder (Connor and Rozell, 2020) also represents data variation in a latent space using the transport operator model, but it only defines these variations in the context of a deterministic autoencoder mapping. This approach shares the motivation of representing the structured data manifold in the latent space but it lacks the fully probabilistic generative framework modeled in VAELLS. Notably the development of the probabilistic framework resulted in the introduction of anchor points which are fundamental to defining the structure of the latent manifold model. The introduction of anchor points eliminates the need to jointly select point pairs during training (as described in Section 2.1) and allows VAELLS to be trained from batches of individual training points.

4 EXPERIMENTS

Our experiments highlight the strengths of VAELLS: the ability to adapt the prior to the true data manifold structure, the ability to define nonlinear paths in the latent space, and the ability to separate classes by learning identity-preserving transformations within classes specified by anchor points. First, we begin with two simple datasets with known ground truth latent structures in order to validate the ability of our model to learn the true latent structure. Next, we apply VAELLS to rotated and naturally varying MNIST digits to show that our prior can adapt to represent rotations of individual digits through a learned

Table 1: Comparison of VAE Techniques

Model	Adaptive Prior	Defines Paths	Class Separation
VAE (Kingma and Welling, 2013)	No	Linear	No
Hyperspherical VAE (Davidson et al., 2018)	No	Geodesic	No
ΔVAE (Rey et al., 2019)	No	No	No
VDAE (Li et al., 2020)	**Yes**	Linear	No
VAE with VampPrior (Tomczak and Welling, 2018)	**Yes**	No	No
$\mathcal{R}$-VAE (Kalatzis et al., 2020)	**Yes**	**Nonlinear**	No
Lie VAE (Falorsi et al., 2019)	No	**Nonlinear**	No
VAELLS (our approach)	**Yes**	**Nonlinear**	**Yes**

operator and extend to real-world data with natural transformations.

The unique characteristics of the VAELLS variational posterior and prior lead to specific training considerations. First, as shown in (7), to compute both the variational posterior and prior distributions we must infer transformation coefficients that maximize $q_\phi(z \mid c, x)q(c)$ and $q_\phi(z \mid c, a_i)q(c)$ respectively. This involves coefficient inference between each sampled point z and its neural network encoding $f_\phi(x)$ as well as between z and all encoded anchor points $f_\phi(a_i)$. The coefficient inference is performed using a conjugate gradient descent optimization solver.

For training the networks weights we use the Adam algorithm (Kingma and Ba, 2015). To add stability and improve efficiency of training, we alternate between steps where we update the network and anchor points while keeping the transport operators fixed, and steps where we update the transport operators while keeping the network weights and anchor points fixed.

The selection of anchor points is important for learning identity-preserving transport operators. Training VAELLS on a dataset with multiple classes requires class labels for both the anchor points and training points in order to compare each training sample with only anchor points from the same class. Anchor points are initialized by selecting training samples from each class in the input space. While we do allow for updates to the anchor points, in practice they only vary by a negligible amount during the entire training process. Details of the network architectures and training parameters for each experiment are available in the Appendix as well as an algorithmic view of the training procedure (Algorithm 1).

Swiss Roll: We begin by applying VAELLS to a dataset composed of 20-dimensional vector inputs that are mapped from a 2D ground truth latent space with a swiss roll structure (Fig. 2a). We selected this classic manifold test structure because many VAE techniques that incorporate specific structured priors into the la-

tent space have not demonstrated the ability to adapt to this specific geometry. The latent space is two-dimensional and the VAELLS prior uses four anchor points that are spread out along the swiss roll (shown as black x's in Fig. 3b). Because this experiment involves only one class, the same four anchor points are used for each training sample. In the swiss roll test we learn a single operator.

Fig. 2 shows the latent space embedding for several VAE techniques. The traditional VAE with a Gaussian prior in the latent space loses the latent structure of the true data manifold because it encourages all points to cluster around the origin (Fig. 2b). The hyperspherical VAE similarly loses the true data structure because it distributes the latent points on a hypersphere (Fig. 2c). The VAE with VampPrior is able to estimate the spiraling characteristic of the swiss roll structure (Fig. 2d), but it is not a smooth representation of the true data manifold. By contrast, the encoded points in VAELLS (Fig. 2e) clearly adapt to the swiss roll structure of the data.

We also utilize the swiss roll dataset to provide an intuitive understanding of how the prior in our method is formed as a combination of the learned transport operators and the encoded anchor points. Fig. 3a contains the encoded latent points overlaid with the orbit of the operator learned by VAELLS. Specifically, the colored line shows how the transport operator trajectory for our learned operator evolves over time when applied to a single starting point. This trajectory can be generated for any dictionary element Ψ_m: $z_t = \text{expm}(\Psi_m \frac{t}{T})z_0$, $t = 0, ..., T$. Fig. 3b contains latent points sampled from the prior using the sampling detailed in (5) with increased b and γ parameters to make the sampling easier to visualize. This shows how the prior has been well-adapted to the swiss roll structure. Finally, we demonstrate how transport operators can be used to define nonlinear paths in the latent space. To generate paths between pairs of points with our learned operators, we first infer the coefficients c^* between each pair. We then interpolate the path from the starting point

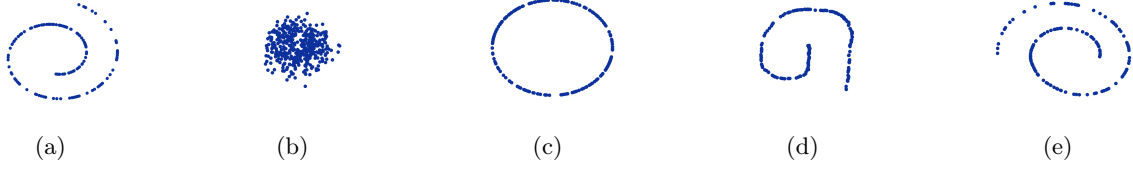Marissa Connor, Gregory Canal, Christopher Rozell

Figure 2: Embedding of swiss roll inputs in VAE latent spaces. (a) Ground truth latent structure. (b) VAE. (c) Hyperspherical VAE. (d) VAE with VampPrior. (e) VAELLS.
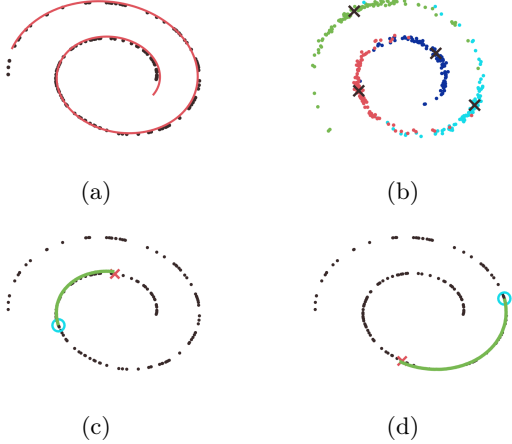

Figure 3: (a) The orbit of the transport operator learned on the swiss roll dataset plotted on top of encoded latent points. (b) Sampling of the latent space prior from each of the anchor points (labeled with black x's); each color indicates a separate anchor point of origin. (c-d) Transport operator paths inferred between pairs of points on the swiss roll manifold in the latent space. The cyan circle is the path starting point and the red x is the desired path ending point

z_0 as follows: $z_t = \text{expm}\left(\sum_{m=1}^{M} \Psi_m c_m^* t\right) z_0$. Fig. 3(c-d) show two example inferred paths between points encoded on the swiss roll manifold. This experiment highlights three beneficial characteristics of VAELLS - learning a specific latent space manifold structure, sampling points from that manifold, and generating paths directly on the manifold surface.

Concentric Circle: Next we apply VAELLS to a dataset composed of 20-dimensional data points that are mapped from a 2D ground truth latent space with two concentric circles (Fig. 4a). As in the previous example, our network maps these inputs into a two-dimensional latent space. This experiment has two classes (inner circle and outer circle) so we select three anchor points per concentric circle with the anchor points evenly spaced on each circle. During training, the training points from each circle are compared against only those anchor points on the same circle

manifold. In the concentric circle test we learn four operators.

This dataset in particular is well-suited for assessing how well each method is able to discriminate between the two concentric circle manifolds once points are mapped into the latent space. Fig. 4 shows the encoded latent points for several different VAE approaches. All three comparison techniques (Fig. 4(b-d)) lose the class separation between the ground truth concentric circle manifolds. Additionally, as in Fig. 2, the Gaussian prior of the traditional VAE distorts the true data structure (Fig. 4b), and the VAE with VampPrior encodes a latent structure with similar characteristics to the ground truth manifold but fails to model the exact shape (Fig. 4d). By contrast, the encoded points in the VAELLS latent space maintain the class separation while simultaneously encoding the true circular structure. This verifies two characteristics of our approach that improve upon the traditional VAE model – learning the prior from the data and class separation.

Rotated MNIST Digits: The rotated MNIST dataset (LeCun et al., 1998) is a natural choice for demonstrating VAELLS because it consists of real images in which we have an intuitive understanding of what the rotational transformations should look like. To define the rotated digit manifold, we specify anchor points as rotated versions of training inputs and aim to learn a transport operator that induces a latent space transformation corresponding to digit rotation. In practice this means that for each training sample we select several rotated versions of that digit as anchor points.

First, to highlight that we can learn a transformation model that is adapted to the rotated digit manifold, we show the result of generating data from input points in the test set using the sampling procedure described in (5). Fig. 5 shows the decoded outputs of latent vectors sampled from the posterior for two example test points using four different VAE models. In each example, the center image (enclosed in a green box) is the decoded version of the input test sample. The images surrounding each center are decoded outputs of latent posterior samples. In order to visualize no-
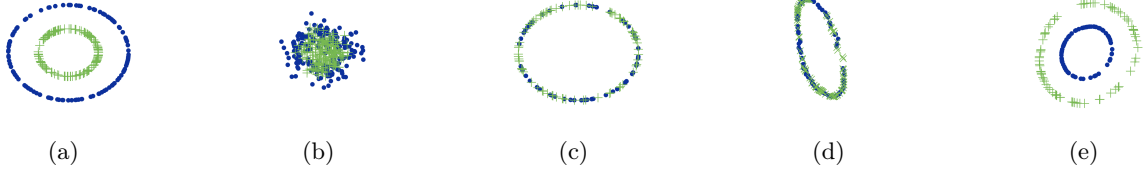
Figure 4: Embedding of concentric circle inputs in VAE latent spaces. (a) Ground truth latent structure. (b) VAE. (c) Hyperspherical VAE. (d) VAE with VampPrior. (e) VAELLS.
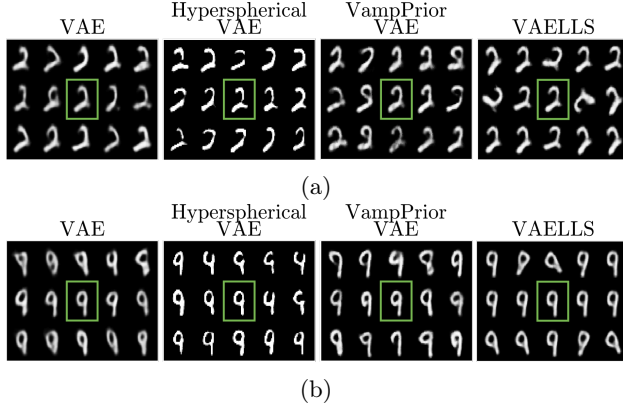

(a)

(b)

Figure 5: Examples of images decoded from latent vectors sampled from the posterior of models trained on rotated MNIST digits. In each example, the center digit (in the green box) is the decoded version of the input digit and the surrounding digits are images decoded from the sampled latent vectors. Sampling in the VAELLS latent space results in rotations in the sampled outputs.

ticeable sampling variations, we increase the standard deviation and scale of the sampling noise in each of these models. The key result is that the VAELLS sampling procedure using the learned transport operator leads to latent space transformations that correspond to rotations in decoded outputs. This verifies that the transport operator corresponds to movement on a learned rotated digit manifold, unlike comparison techniques which only capture natural digit variations and not specifically rotation.

In Fig. 6 we show how we can extrapolate rotational paths in the latent space using our learned transport operator. To generate this figure, we randomly select example MNIST digits with zero degrees of rotation and then encode those digits to get each starting point z_0. The decoded versions of these initial points are shown in the middle columns (enclosed by a green box) in each figure. We then apply the learned operator with both positive and negative coefficients to z_0 and decode the outputs. The images to the left of center show the path generated with negative coefficients and the images to the right of center show the path generated with

positive coefficients. This shows how we can generate rotated paths using the learned transport operator. It also highlights the ability for VAELLS to define identity-preserving transformations with respect to selected input points. In these examples, the class identity of the digit is qualitatively preserved for about 180 degrees of rotation.

$$-c \longleftarrow z_0 \longrightarrow +c \quad -c \longleftarrow z_0 \longrightarrow +c$$
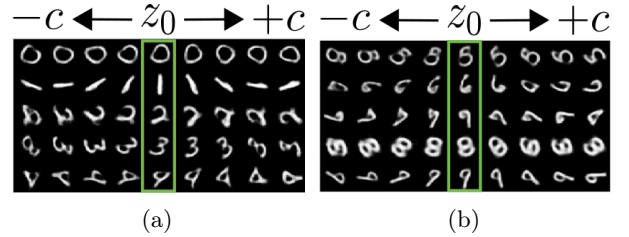

(a)   (b)

Figure 6: Extrapolated rotation paths in the VAELLS latent space. The center digit in each row is a decoded version of an input digit and the digits to the left and right of center show decoded outputs from latent paths extrapolated using the learned transport operator with negative and positive coefficients respectively.

Natural MNIST Digits: In our final experiment, we highlight our ability to learn the natural manifold structure in MNIST digits. The anchor points are initialized by randomly selecting training examples from each digit class without any special consideration given to the specific image selection. Without a priori knowledge of the manifold structure, we have flexibility in how to parameterize our model; two parameters of specific interest are the number of transport operator dictionary elements M used to define latent space transformations and the number of anchor points per class N_a. These parameters impact the two components of the prior definition: the learned transport operator model and the anchor points. Table 2 shows how varying these parameters impacts the quantitative performance of VAELLS as measured by estimated log-likelihood (LL) and mean-squared error (MSE) between input and reconstructed images. Our estimated log-likelihood is based off of Burda et al. (2015) and the details of its computation with our model are given in Appendix D. The number of dictionary elements M has a large impact on the estimated log-likelihood.

Table 2: Evaluation Metrics of VAELLS Trained on MNIST

M	N_a	LL	MSE
2	8	-4.026×10^6	0.0315
4	8	-1.169×10^6	0.0232
8	8	-731.91	0.0244
4	12	-9.248×10^5	0.0221
4	16	-7.523×10^5	0.0232

As the number of dictionary elements increases, the value of $-\log\left[q_\phi(z \mid c, x)\right]$ decreases significantly which increases the prior term in our log-likelihood computation. This indicates that more dictionary elements enable the model to more accurately estimate paths between sampled latent points and the encoded anchor point locations. The number of anchor points does not have a clear impact on the log-likelihood and the MSE stays fairly constant as we vary the M and N_a parameters. For the rest of the results we use $M = 8$, $N_a = 8$.

Fig. 7 shows the result of sampling the variational posterior in a similar manner to Fig. 5. Note that sampling in the VAELLS latent space leads to natural digit transformations in the decoded outputs that maintain the class of the original digit. By contrast, the sampling in the latent space of the comparison techniques can lead to changes in class. Appendix I contains paths generated by each of the learned transport operators that highlight how they represent natural transformation paths. This experiment demonstrates the strengths of VAELLS in cases where the data manifold is unknown. By training on points associated with anchor points of the same class, we are able to define the prior using learned identity-preserving transformations, sample from the class manifold, and generate continuous paths on the latent space manifold.

5   CONCLUSION

In this paper we developed a model that has the flexibility to learn a structured VAE prior from training data by incorporating manifold transport operators into the latent space. This adaptable prior allows us to define a generative model with a latent structure that is a better fit to the data manifold. It also enables us to both interpolate and extrapolate nonlinear transformation paths in the latent space and to explicitly incorporate class separation by learning identity-preserving transformations. We verified the performance of this model on datasets with known latent structure and then extended it to real-world data to learn natural transformations. VAELLS can be used to not only develop more realistic generative models of data but it
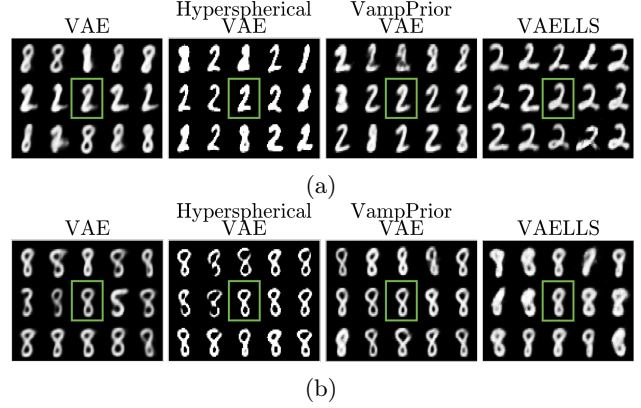

(a)

(b)

Figure 7:   Examples of images decoded from latent vectors sampled from the posterior of a model trained on MNIST digits. In each example, the center digit (in the green box) is the encoded digit input and the surrounding digits are from the sampled latent vectors.

can also be used to more effectively understand natural variations occurring in complex data.

References

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2018). Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*.

Bengio, Y. and Monperrus, M. (2005). Non-local manifold tangent learning. *Advances in Neural Information Processing Systems*, 17:129–136.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and Smagt, P. (2018). Metrics for deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1550. PMLR.

Connor, M. and Rozell, C. (2020). Representing closed transformation paths in encoded network latent space. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 3666–3675.

Culpepper, B. J. and Olshausen, B. A. (2009). Learning transport operators for image manifolds. In *Advances in Neural Information Processing Systems*, pages 423–431.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.

Dollár, P., Rabaud, V., and Belongie, S. J. (2007). Learning to traverse image manifolds. In *Advances in Neural Information Processing Systems*, pages 361–368.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. (2019). Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3244–3253. PMLR.

Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Kalatzis, D., Eklund, D., Arvanitidis, G., and Hauberg, S. (2020). Variational autoencoders with riemannian brownian motion priors. In *International Conference on Machine Learning*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. In *International Conference on Learning Representations*.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, H., Lindenbaum, O., Cheng, X., and Cloninger, A. (2020). Variational diffusion autoencoders with random walk sampling. In *European Conference on Computer Vision*, pages 362–378. Springer.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems*, volume 32.

Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. (2019). A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*, pages 4693–4702.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*.

Rey, L. A. P., Menkovski, V., and Portegies, J. W. (2019). Diffusion variational autoencoders. *arXiv preprint arXiv:1901.08991*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Shao, H., Kumar, A., and Thomas Fletcher, P. (2018). The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 315–323.

Skopek, O., Ganea, O.-E., and Bécigneul, G. (2020). Mixed-curvature variational autoencoders. In *International Conference for Learning Representation*.

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

Tomczak, J. and Welling, M. (2018). Vae with a vampprior. In Storkey, A. and Perez-Cruz, F., editors, *International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1214–1223. PMLR.