ENCODED PRIOR SLICED WASSERSTEIN AUTOENCODER FOR LEARNING LATENT MANIFOLD REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

While variational autoencoders have been successful generative models for a variety of tasks, the use of conventional Gaussian or Gaussian mixture priors are limited in their ability to capture topological or geometric properties of data in the latent representation. In this work, we introduce an Encoded Prior Sliced Wasserstein AutoEncoder (EPSWAE) wherein an additional prior-encoder network learns an unconstrained prior to match the encoded data manifold. The autoencoder and prior-encoder networks are iteratively trained using the Sliced Wasserstein Distance (SWD), which efficiently measures the distance between two *arbitrary* sampleable distributions without being constrained to a specific form as in the KL divergence, and without requiring expensive adversarial training. Additionally, we enhance the conventional SWD by introducing a nonlinear shearing, i.e., averaging over random *nonlinear* transformations, to better capture differences between two distributions. The prior is further encouraged to encode the data manifold by use of a structural consistency term that encourages isometry between feature space and latent space. Lastly, interpolation along *geodesics* on the latent space representation of the data manifold generates samples that lie on the manifold and hence is advantageous compared with standard Euclidean interpolation. To this end, we introduce a graph-based algorithm for identifying network-geodesics in latent space from samples of the prior that maximize the density of samples along the path while minimizing total energy. We apply our framework to 3D-spiral, MNIST, and CelebA datasets, and show that its latent representations and interpolations are comparable to the state of the art on equivalent architectures.

1 INTRODUCTION

Generative models have the potential to capture rich representations of data and use them to generate realistic outputs. In particular, Variational AutoEncoders (VAEs) (Kingma & Welling, 2014) can capture important properties of high-dimensional data in their latent embeddings, and sample from a prior distribution to generate realistic images. VAEs have been very successful in a variety of tasks such as learning visual features (Hou et al., 2017; Higgins et al., 2016), learning molecular structure (Kusner et al., 2017), speech recognition (Hsu et al., 2017) etc. However, the use of a simplistic standard normal prior is known to cause problems such as under-fitting and over-regularization, and fails to use the network's entire modeling capacity (Burda et al., 2016). Gaussian or Gaussian mixture model (GMM) priors are also limited in their ability to represent the underlying data manifold. High-dimensional data can typically be modeled as lying on or near an embedded low-dimensional, nonlinear manifold (Fefferman et al., 2016). Learning the shape of the encoded true data manifold in latent space is a natural way to encode features and is expected to generate realistic images as sampling occurs along the manifold. Thus, a more flexible prior may be desirable for learning the manifold embedding.

Conventional variational inference uses Kullback-Leibler (KL) divergence as a measure of distance between the posterior and the prior, restricting the prior distribution to cases that have tractable approximations of the KL divergence. Many works such as Guo et al. (2020); Tomczak & Welling (2018); Rezende & Mohamed (2015) etc. have investigated the use of more complicated priors (notably GMMs) which lead to improved latent representation and generation compared to a single

Gaussion prior. Alternate approaches such as adversarial training learn arbitrary priors by using a discriminator network to compute a divergence (Wang et al., 2020), however they have been reported to be harder to train and are computationally expensive.

In this work, we introduce the Encoded Prior Sliced Wasserstein AutoEncoder (EPSWAE), which consists of a conventional autoencoder architecture and an additional prior-encoder network that learns an unconstrained prior distribution that matches *any* data manifold topology. We use the Sliced Wasserstein (SW) distance (Bonnotte, 2013; Bonneel et al., 2015), a concept from optimal transport theory that is a simple and convenient alternative to the KL divergence or adversarial training for any sampleable distributions. The SW distance is defined as the average Wasserstein distances (described in (Villani, 2003)) of 1D projections, and can be efficiently computed. The Wasserstein Autoencoder has been shown to be a variant of adversarial autoencoders (Tolstikhin et al., 2018). A Sliced Wasserstein AutoEncoder (SWAE) that regularizes an autoencoder using SW distance was proposed in Kolouri et al. (2018a). In contrast with their linear projections of SW distance, in this work we introduce a Nonlinear Sliced Wasserstein (NSW) distance which measures average Wasserstein distances of 1D projections of random *nonlinear* transformations, and enhances the SW measure by better capturing differences between distributions with fewer projections.

In EPSWAE, the prior-encoder and autoencoder networks are iteratively trained using the NSW distance. Additionally, we introduce a structural consistency term that encourages the latent space to be isometric to the feature space, which is typically measured at the output of the convolutional layers of the data encoder. Variants of this penalty have previously been used to encourage isometry between the latent space and data space (Yu et al., 2013; Benaim & Wolf, 2017; Sainburg et al., 2018). The structural consistency term further encourages the prior to match the encoded data manifold, which in turn leads to improved latent representations.

Conventionally, VAEs use Euclidean interpolation between two points in latent space. However, since manifolds typically have curvature, this is an unintuitive distance metric that can lead to unrealistic intermediate points. Instead, it is natural to interpolate along the manifold geodesics in latent space. Several works such as Shao et al. (2018); Miolane & Holmes (2020b) endow the latent space with a Riemannian geometry and measure corresponding distances, however these are difficult and involve explicitly solving expensive ordinary differential equations. In this work, we introduce 'network-geodesics', a graph-based method for interpolating along a manifold in latent space, that maximizes sample density along paths while minimizing total energy. This involves first generating a distance graph between samples from the prior. Then this network is non-uniformly thresholded such that the set of allowable paths from a given sample traverse high density regions through short hops. Lastly, we use a shortest path algorithm like Dijkstra's algorithm (Dijkstra, 1959) to identify the lowest 'energy' path between two samples through the allowable paths. Since the prior is trained to learn the data manifold, the resulting network-geodesic curves give a notion of distance on the manifold and can be used to generate realistic interpolation points with few prior samples.

The novel contributions of this work are threefold:

- We introduce a novel architecture EPSWAE that consists of a prior-encoder network that is efficiently trained (without expensive adversarial methods) to generate a prior that matches any arbitrary data manifold.
- We present a Nonlinear Sliced Wasserstein (NSW) distance to regularize the autoencoder that enhances the conventional SW distance with nonlinear transforms, capturing more differences between distributions with fewer samples. Additionally, we employ a structural consistency term in the loss that encourages isometry between feature space and latent space, leading to improved latent representations.
- We introduce a novel graph-based method for interpolating along network-geodesics in latent space through maximizing sample density while minimizing total energy. We show that it generates natural interpolations through realistic images.

2 Background and Related Work

The standard VAE (Kingma & Welling, 2014) is a generative model that uses principles of variational inference to approximate the data distribution while maintaining continuity in latent space. It

minimizes the Evidence Lower Bound (ELBO) comprised of the reconstruction error and the KL divergence between the posterior and prior distributions which are typically assumed to be Gaussians (for which there exists a closed-form expression for the KL divergence).

Several works have attempted to increase the complexity of the prior in order to obtain better latent representations. Dilokthanakul et al. (2016) presents improved unsupervised clustering of latent space by using GMM priors. Takahashi et al. (2019) uses the density ratio trick to calculate the KL divergence using implicit priors, however this trick doesn't work well in high dimensions, and hence is approximated by a low dimensional density ratio. Some other representative works are normalizing flows Rezende & Mohamed (2015) which maps the Gaussian prior by a series of simple transformations, Guo et al. (2020) which learns a GMM prior using an approximate ELBO, Goyal et al. (2017) which uses a hierarchical Bayesian framework, and Tomczak & Welling (2018) which learns a two-layer hierarchical GMM from aggregated posteriors.

The KL divergence is tractable only for Gaussian distributions, however the Wasserstein distance of 1D projections (also called Sliced Wasserstein (SW) distance) has a closed-form for any arbitrary distribution (Kolouri et al., 2018b). Wasserstein distances Villani (2003) have been shown to have several of the same properties as KL divergence, and lead to better sampling (Gulrajani et al., 2017; Tolstikhin et al., 2018), and have been used in several machine learning applications such asArjovsky et al. (2017); Tolstikhin et al. (2018); Kolouri et al. (2018b).

Alternatively, many works such as Wang et al. (2020); Makhzani et al. (2015); Arjovsky et al. (2017); Sainburg et al. (2018) replace the KL divergence with an adversarial loss, however adversarial methods tend to be significantly more expensive and difficult to optimize. In higher dimensions, using a discriminator network as in adversarial approaches is a natural way of implicitly computing an equivalent of the Wasserstein-1 distance (Arjovsky et al., 2017; Tolstikhin et al., 2018). However, using the SW distance is much simpler and more efficient. Adversarial training for interpolation in Sainburg et al. (2018) uses a structural consistency term that encourages relative distances in data space to be preserved in latent space. Similar distance-preserving terms have also been used successfully in Yu et al. (2013); Benaim & Wolf (2017), however Euclidean distances in data space can be a poor measure of the natural geometry of the data. In our work, we preserve relative distances in latent space and *feature* space, where features are extracted at the output of the last convolutional layer in the data encoder.

Several manifold learning techniques such as Dollár et al. (2007); Wei et al. (2016) compute embeddings of high-dimensional data but lack generative or interpolative abilities. The hyperspherical VAE in Davidson et al. (2018) outperforms the standard VAE for data residing on a hyperspherical manifold, suggesting that it is advantageous to capture the underlying manifold. Miolane & Holmes (2020a) formulates a Riemannian VAE with an intrinsic generative model of manifold-valued data, however computing its ELBO is challenging. Along similar lines, Arvanitidis et al. (2017) shows that under certain conditions, a Riemannian metric is naturally induced in the latent space, and uses it to compute geodesics. In Chen et al. (2020a), a flat manifold is approximated by penalizing curvature in latent space, and geodesics are defined through Euclidean distances on the flat manifold.

There exists limited work on integrating graphical structures with generative models (Kipf & Welling, 2016). Hadjeres et al. (2017) studies monophonic music using a latent space geodesic regularization that allows interpolations in the latent space to be more meaningful, giving a notion of geodesic distance. Several approaches such as Tenenbaum et al. (2000); Bernstein et al. (2000); Klein & Zachmann (2004); Mémoli & Sapiro (2005); Luo & Hu (2020) approximate geodesics on point clouds by, for instance, building a nearest-neighbor network on the manifold and applying a shortest path algorithm, however, these often require many samples in order generate reasonable approximations and aren't robust to noise (Sober et al., 2020). Inspired by this, we introduce an energy-based network algorithm to identify network-geodesics in latent space with relatively few, noisy samples.

3 Nonlinear Sliced Wasserstein Distance

Wasserstein distances provide a natural metric for measuring distances between probability distributions based on optimal transport, i.e., the cost of deforming one probability distribution into another. For a measurable cost function $c(x, y)$ for $x, y \in \mathbb{R}^d$, and probability distributions μ and ν on $\mathbb{R}^d$,

we define the p-Wasserstein distance between the distributions as

$$d_{W,p}(\mu,\nu) = \inf_{\Gamma\in\Pi(\mu,\nu)} \left(\int_{\mathbb{R}^d\times\mathbb{R}^d} c^p(x,y)\mathrm{d}\Gamma(x,y) \right)^{1/p}, \tag{1}$$

where Π is the set of all joint distributions with marginals μ and ν.

This Wasserstein distance between probability distributions is extremely difficult and computationally intensive in dimensions two and higher, i.e., $d \geq 2$. However, in dimension one, there exists a closed-form solution (Villani, 2003). We use the following algorithm given in (Kolouri et al., 2018a): (a) Generate N one dimensional i.i.d samples $x_j \sim \mu$, $y_j \sim \nu$; (b) sort each list $X = [x_1, \ldots, x_N], Y = [y_1, \ldots, y_N]$ into ascending order denoted by $\tilde{X}, \tilde{Y}$ respectively; (c) define the approximation to the 1D p-Wasserstein distance

$$d_{W,p}(\mu,\nu) \approx \left(\frac{1}{N} \sum_{j=1}^{N} c^p(\tilde{x_j}, \tilde{y_j}) \right)^{1/p}, \tag{2}$$

where $\tilde{x}_j \in \tilde{X}, \tilde{y}_j \in \tilde{Y}$. The Sliced Wasserstein (SW) distance defines a metric on probability measures (Bonnotte, 2013) which provides an alternative to Eqn. 1 by exploiting the computational feasibility of the 1D Wasserstein distance in Eqn. 2. It involves averaging over one-dimensional orthogonal projections $\pi^\theta x := (\theta \cdot x)\theta$ as follows:

$$d_{SW}(\mu,\nu) = \left(\fint_{\mathbb{S}^{d-1}} d^p_{W,p}(\pi^\theta_*\mu, \pi^\theta_*\nu) dS(\theta) \right)^{1/p}, \tag{3}$$

where we define the "push-forward" (also known as "random variable transform") as follows: given any function $f : X \to Z$ and probability measure μ, we define $f_*\mu(A) = \mu(f^{-1}(A))$ for all $A \subset Z$ measurable. Most importantly, to generate samples of $z \sim f_*\mu$ one simply generates $x \sim \mu$ and defines $z = f(x)$. The SW distance has seen a variety of implementations (Bonneel et al., 2015; Kolouri et al., 2018b; Lee et al., 2019).

The integral in Eqn. 3 can be easily approximated by sampling M random one-dimensional vectors θ_k uniformly on $\mathbb{S}^{d-1}$ and computing

$$d_{SW}(\mu,\nu) \approx \left(\frac{1}{M} \sum_{k=1}^{M} d^p_{W,p}(\pi^{\theta_k}_*\mu, \pi^{\theta_k}_*\nu) \right)^{1/p}. \tag{4}$$

However, linear projections may be sub-optimal for extracting information about the differences between μ and ν, since a large number of linear projections may be required to get an accurate approximation for d_{SW}. Several works have suggested possible methods for improving the effectiveness of the SW distance (Chen et al., 2020b; Kolouri et al., 2019; Nguyen et al., 2020; Deshpande et al., 2019). In contrast, we introduce a Nonlinear Sliced Wasserstein (NSW) distance, an averaging procedure over (random) *nonlinear* transformations, defined as

$$d_{NSW}(\mu,\nu) = \mathbb{E} d_{SW}(\mathcal{N}^{\zeta,\gamma}_*\mu, \mathcal{N}^{\zeta,\gamma}_*\nu) \approx \frac{1}{L} \sum_{\ell=1}^{L} d_{SW}(\mathcal{N}^{\zeta_\ell,\gamma_\ell}_*\mu, \mathcal{N}^{\zeta_\ell,\gamma_\ell}_*\nu) \tag{5}$$

where L is the number of nonlinear transformations, and ζ, γ are chosen to be normal random variables with mean and variance matching that of μ. The nonlinear transformations are given by

$$\mathcal{N}^{\zeta_\ell,\gamma_\ell}(x) = x + \zeta_\ell \sin(\gamma_\ell \cdot x). \tag{6}$$

In principle, other ensembles of nonlinear transformations could be used.

4 EPSWAE

In the Encoded Prior Sliced Wasserstein AutoEncoder (EPSWAE) (see Fig. 1 for schematic), let's define the data encoder as Ψ_E, the decoder Ψ_D, and the prior-encoder as Ψ_{PE}, each with parameters $\phi_E, \phi_D, \phi_{PE}$ respectively. Input samples $\mathbf{x}^{(j)} \sim P_X$, where P_X is the probability distribution of the

input data, are passed through Ψ_E to generate posterior samples $\mathbf{z}^{(j)} \sim (\Psi_E)_* P_X$ by setting $\mathbf{z}^{(j)} = \Psi_E(\mathbf{x}^{(j)})$. Similarly, prior-encoder input samples $\xi^{(j)} \sim \mu$, where μ is the probability distribution of the input to the prior-encoder (chosen to be a mixture of Gaussians) from a distribution, are passed through Ψ_{PE} to generate prior samples $\mathbf{y}^{(j)} \sim (\Psi_{PE})_* \mu$ by setting $\mathbf{y}^{(j)} = \Psi_{PE}(\xi^{(j)})$. The prior-encoder network and the autoencoder (data encoder and decoder) network are trained iteratively in a two step process:

1. Given a minibatch, parameters of the autoencoder (ϕ_E, ϕ_D) are trained for k_1 steps while parameters of the prior encoder (ϕ_{PE}) are fixed. The loss function for the autoencoder consists of the reconstruction error, the NSW distance, and a Feature Structural Consistency (FSC) term $\mathcal{L}_{FSC}$ (given in Eqn. 8):

$$\mathcal{L}_{AE} = \alpha \mathbb{E}_{\mathbf{x} \sim P_X} \mathcal{L}_{rec}(\mathbf{x}, \bar{\mathbf{x}}) + \beta d_{NSW}((\Psi_E)_* P_X, (\Psi_{PE})_* \mu) + \kappa \mathcal{L}_{FSC}, \tag{7}$$

   where $\bar{\mathbf{x}} = \Psi_D(\Psi_E(\mathbf{x}))$ is the autoencoder output. Note that d_{NSW} is efficiently computed just from samples of the distributions. In this measure theoretic notation, $(\Psi_E)_* P_X$ is the posterior distribution (denoted as $q_{\phi_E}(z|x)$ in Bayesian literature), and $(\Psi_{PE})_* \mu$ is the prior distribution (denoted as $p_{\phi_{PE}}(z)$ in Bayesian literature). Similar to β-VAE ((Higgins et al., 2016)), the hyperparameters α, β, κ are tuneable; in our experiments they are fixed, but it could be advantageous to have them vary over training epochs.

   $\mathcal{L}_{FSC}$ encourages relative distances in feature space to be preserved in latent space, where features are extracted at the output of the last convolutional layer in the encoder (or at input data if no convolutional layers are present). $\mathcal{L}_{FSC}$ (adapted from Sainburg et al. (2018)) for two point clouds $F = [f_1, \dots f_N]$ in feature space and the corresponding points $Z = [z_1, \dots z_N]$ in latent space in a minibatch of size N is given by:

$$\mathcal{L}_{FSC} = \frac{1}{N^2} \sum_{\ell,j=1}^{N} \left(\log\left(1 + \frac{\|f_j - f_\ell\|^2}{\frac{1}{N^2}\sum_{m,n}\|f_m - f_n\|^2}\right) - \log\left(1 + \frac{\|z_j - z_\ell\|^2}{\frac{1}{N^2}\sum_{m,n}\|z_m - z_n\|^2}\right)\right)^2. \tag{8}$$

2. In the second step of minimization, the parameters of the prior-encoder (Ψ_{PE}) are trained for k_2 steps while parameters of the autoencoder (ϕ_E, ϕ_D) are fixed. The loss function for the prior-encoder consists of the NSW distance between the prior and posterior:

$$\mathcal{L}_{PE}(\mathbf{x}) = d_{NSW}((\Psi_E)_* P_X, (\Psi_{PE})_* \mu). \tag{9}$$

The pseudocode is outlined in Appendix A.

4.1 Interpolation and approximate geodesics

EPSWAE attempts to learn a representation of the embedded manifold geometry that the data lies along. The question arises: how does one make use of this representation? A natural way of interpolating data which lies on a manifold is through geodesics or paths on the manifold which traverse dense regions. Here we present a network algorithm for efficiently approximating network-geodesics with relatively few, noisy samples from the prior, by encouraging connections through high density regions of latent space while minimizing total energy.

1. Gather samples of the posterior $(\Psi_E)_* P_X$, i.e., $\Psi_E(x)$ with x is a minibatch of data. Additional prior samples can be used to supplement this if desired;
2. For each sample j, compute the *average* Euclidean distance c_j of the k-nearest neighbors;
3. Generate a thresholded network: for threshold value t, add an edge with weight $d^h(i,j)$ connecting sample j to i if $d(i,j) < t \cdot c_j$, and h is the energy parameter in the edge weight (chosen to be 1 or 2 in experiments, with $h = 2$ encouraging shorter hops). Sample-specific thresholding using c_j allows each sample to have a different degree and encourages (1) paths through high density regions, i.e., high-degree nodes and (2) through short hops since paths exist only between nearby points.
4. Continue increasing t until the graph is connected, i.e., there exists a path, direct or indirect, from every sample to every other sample. Then, use Dijkstra's algorithm (Dijkstra, 1959) to identify network-geodesics with least total energy through allowable paths on the thresholded network.

5 EXPERIMENTS

We run experiments using EPSWAE on three datasets. First, we use a 3D spiral data, where the latent space can be visualized easily, to demonstrate that the learned prior matches the 3D spiral. Then we present interpolation and generation results on the MNIST (LeCun et al., 2010) and CelebA (Liu et al., 2018) datasets.

5.1 ARCHITECTURE

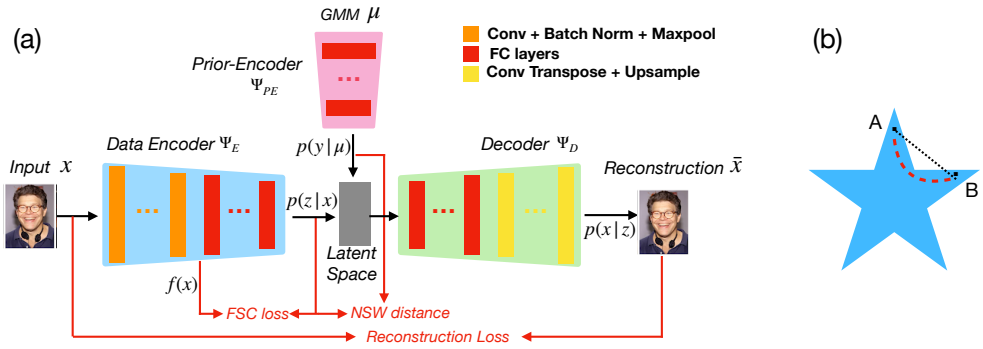

Figure 1: (a) is a schematic of the EPSWAE architecture. The red arrows indicate calculation of the loss terms. The prior-encoder generates a prior in latent space. (b) is simple example of a shape where interpolating between points A and B 'through the manifold' (red dashed line) is desired, since linear interpolation (black line) leads out of the concave hull.

In all of the experiments, the prior-encoder Ψ_{PE} consists of three fully connected hidden layers with ReLU activation. A sampleable distribution with dimension larger than the dimension of latent space is input to the prior-encoder which outputs the prior. In principle, samples from any distribution can be the input μ to the prior-encoder.

In the artificial data set we use a normal distribution, whereas for MNIST and CelebA we use a mixture of 10 Gaussians with random i.i.d means and standard deviation of 2. The schematic of the architecture is presented in Fig. 1(a). Note that the specifics of the layers shown in this schematic are for the image datasets MNIST and CelebA, the 3D Spiral data does not have convolutional layers. Details of the architecture, training, and hyperparameters for each dataset are given in Appendix B. The optimizer Adam (Kingma & Ba, 2014) with a learning rate of 0.001 was used for learning in both networks: the prior-encoder and the autoencoder.

5.2 ARTIFICIAL SPIRAL DATASETS

We consider a 3D spiral randomly embedded to 40D space with noise as follows: (a) The formula for the spiral is given by $(x(t), y(t), z(t)) = (t \cos 7t\pi/4, t \sin 7t\pi/4, 2t)$, (b) a random 40×3 matrix is generated with i.i.d normal entries to map the spiral into $\mathbb{R}^{40}$, and (c) Gaussian noise with standard deviation 0.1 is added. The input to the prior-encoder is a Gaussian in $\mathbb{R}^{40}$, and the latent space is $\mathbb{R}^3$.

Figure 2 ((a) side and (b) top views) show samples of the posterior and prior generated by running EPSWAE for 100 epochs on the high dimensional input. As seen in the figure, after training, the posterior matches the spiral almost exactly and the prior learns a matching noisy representation. A Nonlinear Sliced Wasserstein (NSW) distance and structural consistency term (where latent space is isometric with data space since convolutional layers aren't used) are used in the loss in Eqns. 7 and 9. A comparison highlighting the effects of these terms is shown in Fig. 5 in the Appendix C. We observe that the use of the NSW distance (as opposed to linear SW distance) improves the accuracy of the learned prior. Manifold interpolation (in red in Fig. 2) uses the network-algorithm outlined in section 4.1. The interpolation is seen to have the desired form on the manifold, i.e., it

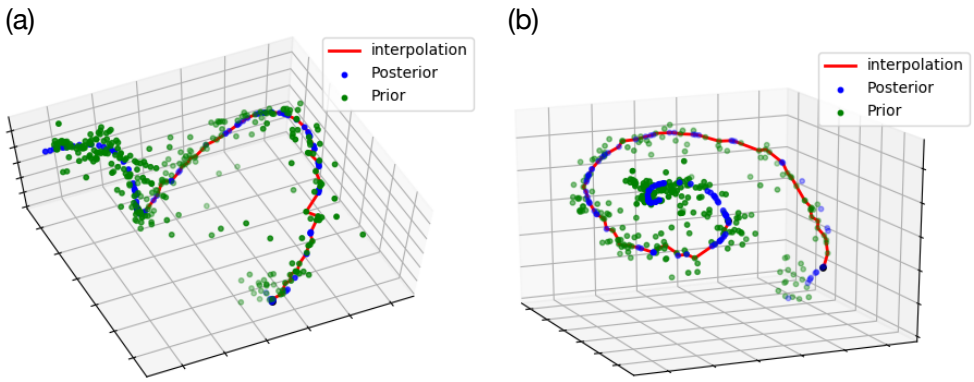

Figure 2: (a) side view and (b) top view of samples of the prior (green) and posterior (blue) in 3D latent space generated by EPSWAE. The red line shows the interpolation along network-geodesics between two prior samples.

approximates geodesics on the manifold. The larger the number of samples of the prior, the smoother the corresponding interpolation.

Comparisons with baselines SWAE (Kolouri et al., 2018a) and VAE (Kingma & Welling, 2014) are shown in Appendix Fig. 6, and EPSWAE is seen to significantly outperform them in learning an improved latent representation.

5.3 Image Generation

Figure 3: Images generated from prior samples in (a) EPSWAE (b) Baseline SWAE.

MNIST results are obtained with $5D$ latent space and CelebA results are obtained with a $128D$ latent space. Generation on MNIST using EPSWAE (and comparisons with baselines SWAE (Kolouri et al., 2018a) and VAE (Kingma & Welling, 2014)) are shown in Appendix D. CelebA raw images are downsized to 64×64 pixels, and a simple architecture is employed. This is sufficient as a proof of principle for a better latent representation and interpolation, without a high computational cost. Naturally, using a more sophisticated network (for instance ResNet, VGG etc.) without downsizing the data would yield higher quality images at a computational cost.

Figure 3 shows generation on the CelebA dataset after 100 epochs with (a) EPSWAE and (b) baseline SWAE (Kolouri et al., 2018a) (right) respectively. Both employ equivalent architectures and take downsized images as input. As seen in the figure, EPSWAE generated images are more realistic. Note that this comparison is for equivalent training epochs and architectural details, and does not make claims about the quality of SWAE images with longer training.

5.4 INTERPOLATION

Figure 4: Instances of EPSWAE interpolations for MNIST (top three panels) and CelebA (bottom three panels). The first and last images are reconstructions of real data, and the interpolations traverse through samples of the prior using the network-geodesic algorithm. Points selected by the network-geodesic algorithm are connected through an intermediate sample corresponding to the midpoint in latent space. A total of 400 samples are used.

Figure 4 shows examples of interpolations on the dataset using the network interpolation algorithm described in 4.1. The start and end points correspond to real images (posterior samples). MNIST interpolations are smooth and intuitive, for instance, in the top panel in Fig. 4, moving the top part of a '7' to the '9' naturally and then transforming the '9' into a '4'. Interpolation along network-geodesics ensures that reconstructions of intermediate samples are realistic. Interpolations on CelebA are smooth and pass through intermediate images that could arguably pass for celebrities these days. The state of the art interpolations along network-geodesics on the manifold indicate that the learned prior does indeed encode the data manifold. For all interpolations shown in Fig. 4, we use value $h = 2$ of the energy parameter. Comparisons between interpolations corresponding to $h = 1$ and $h = 2$ are presented in Appendix 8. In experiments, energy parameter is not found to have a significant effect.

6 CONCLUSION

We introduce the Encoded Prior Sliced Wasserstein AutoEncoder (EPSWAE) that learns improved latent representations through training an encoded prior to approximate an embedding of the data manifold. The learning of an arbitrary shaped prior is facilitated by the use of the Sliced Wasserstein distance, which can be efficiently computed from samples only. We introduce an extension to this, a nonlinear SW distance, that captures differences between two distributions more efficiently by taking random nonlinear transformations. We also employ a feature structural consistency term to improve the latent space representation. Additionally, we introduce an energy-based algorithm to identify network-geodesics in latent space that maximize path density while minimizing total energy. We demonstrate EPSWAE's ability to learn a 3D spiral from noisy samples embedded in 40D space. Additionally, we show that our interpolation results on MNIST and CelebA are comparable to state of the art techniques without requiring additional adversarial training. Our code is publicly available.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *Advances in neural information processing systems*, pp. 752–762, 2017.

Mira Bernstein, Vin De Silva, John C Langford, and Joshua B Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Citeseer, 2000.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Paris 11, 2013.

Yuri Burda, Roger B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2016.

Nutan Chen, Alexej Klushyn, Francesco Ferroni, Justin Bayer, and Patrick van der Smagt. Learning flat latent manifolds with vaes. *arXiv preprint arXiv:2002.04881*, 2020a.

Xiongjie Chen, Yongxin Yang, and Yunpeng Li. Augmented sliced wasserstein distances. *arXiv preprint arXiv:2006.08812*, 2020b.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 10648–10656, 2019.

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1):269–271, 1959.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Piotr Dollár, Vincent Rabaud, and Serge J Belongie. Learning to traverse image manifolds. In *Advances in neural information processing systems*, pp. 361–368, 2007.

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

Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, and Eric P Xing. Nonparametric variational auto-encoders for hierarchical representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5094–5102, 2017.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Chunsheng Guo, Jialuo Zhou, Huahua Chen, Na Ying, Jianwu Zhang, and Di Zhou. Variational autoencoder with optimizing gaussian mixture model priors. *IEEE Access*, 8:43992–44005, 2020.

Gaëtan Hadjeres, Frank Nielsen, and François Pachet. Glsr-vae: Geodesic latent space regularization for variational autoencoder architectures. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7. IEEE, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. 2016.

Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1133–1141. IEEE, 2017.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 16–23. IEEE, 2017.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations.*, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

Jan Klein and Gabriel Zachmann. Point cloud surfaces using geometric proximity graphs. *Computers & Graphics*, 28(6):839–850, 2004.

Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced wasserstein autoencoders. In *International Conference on Learning Representations*, 2018a.

Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffmann. Sliced wasserstein distance for learning gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018b.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. In *Advances in Neural Information Processing Systems*, pp. 261–272, 2019.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10285–10295, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

Shitong Luo and Wei Hu. Differentiable manifold reconstruction for point cloud denoising. *arXiv preprint arXiv:2007.13551*, 2020.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Facundo Mémoli and Guillermo Sapiro. Distance functions and geodesics on submanifolds of \rˆd and point clouds. *SIAM Journal on Applied Mathematics*, 65(4):1227–1260, 2005.

Nina Miolane and Susan Holmes. Learning weighted submanifolds with variational autoencoders and riemannian variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14503–14511, 2020a.

Nina Miolane and Susan Holmes. Learning weighted submanifolds with variational autoencoders and riemannian variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14503–14511, 2020b.

Khai Nguyen, Nhat Ho, Tung Pham, and Hung Bui. Distributional sliced-wasserstein and applications to generative modeling. *arXiv preprint arXiv:2002.07367*, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Tim Sainburg, Marvin Thielk, Brad Theilman, Benjamin Migliori, and Timothy Gentner. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. *arXiv preprint arXiv:1807.06650*, 2018.

Hang Shao, Abhishek Kumar, and P Thomas Fletcher. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 315–323, 2018.

Barak Sober, Ingrid Daubechies, and Robert Ravier. Approximating the riemannian metric from point clouds via manifold moving least squares. *arXiv preprint arXiv:2007.09885*, 2020.

Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5066–5073, 2019.

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

I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. *ArXiv*, abs/1711.01558, 2018.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223, 2018.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Hui-Po Wang, Wen-Hsiao Peng, and Wei-Jan Ko. Learning priors for adversarial autoencoders. *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

Chao Wei, Senlin Luo, Xincheng Ma, Hao Ren, Ji Zhang, and Limin Pan. Locally embedding autoencoders: a semi-supervised manifold learning approach of document representation. *PloS one*, 11(1):e0146672, 2016.

Wenchao Yu, Guangxiang Zeng, Ping Luo, Fuzhen Zhuang, Qing He, and Zhongzhi Shi. Embedding with autoencoder regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 208–223. Springer, 2013.

A PSEUDOCODE

Algorithm 1 Training EPSWAE

while not converged **do**
  Update the autoencoder Ψ_E, Ψ_D:
  for k_1 substeps **do**
    Sample minibatch from data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(j)} \sim P_X$
    Compute feature extractor samples $\mathbf{f}^{(j)} = \Psi_{FE}(\mathbf{x}^{(j)})$
    Compute posterior samples $\mathbf{z}^{(j)} = \Psi_E(\mathbf{x}^{(j)})$
    Compute decoded output $\mathbf{x}^{(j)}_{recon} = \Psi_D((\mathbf{z})^j)$
    Generate samples $\{\xi^{(1)}, ...\xi^{(J)}\}$ with $\xi^{(j)} \sim \mu$
    Compute prior samples $\mathbf{y}^{(j)} = \Psi_{PE}(\xi^{(j)})$
    Compute reconstruction error: $\mathcal{L}_{recon} = \frac{1}{J}\sum_{j=1}^{J} d(\mathbf{x}^{(j)}, \mathbf{x}^{(j)}_{recon})$
    Compute NSW distance: $\mathcal{L}_{NSWdistance} = d_{NSW}(\frac{1}{N}\sum_{j=1}^{N}\delta_{\mathbf{z}^{(j)}}, \frac{1}{N}\sum_{j=1}^{N}\delta_{\mathbf{y}^{(j)}})$
    Compute FSC loss: $\mathcal{L}_{FSC} = d_{FSC}(\{\mathbf{f}^{(1)}, ..., \mathbf{f}^{(J)}\}, \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(J)}\})$
    Compute autoencoder loss $\mathcal{L}_{AE} = \alpha\mathcal{L}_{recon} + \beta\mathcal{L}_{SWdistance} + \kappa\mathcal{L}_{FSC}$
    Compute gradients of $\mathcal{L}_{AE}$ wrt to ϕ_E, ϕ_D
    Update ϕ_E, ϕ_D
  end for
  Update the prior-encoder Ψ_{PE}:
  for k_2 substeps **do**
    Sample minibatch from data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(J)}\}$ with $\mathbf{x}^{(j)} \sim P_X$
    Compute posterior samples $\mathbf{z}^{(j)} = \Psi_E(\mathbf{x}^{(j)})$
    Generate samples $\{\xi^{(1)}, ...\xi^{(J)}\}$ with $\xi^{(j)} \sim \mu$
    Compute prior samples $\mathbf{y}^{(j)} = \Psi_{PE}(\xi^{(j)})$
    Compute prior-encoder loss: $\mathcal{L}_{PE} = d_{SW}(\frac{1}{N}\sum_{j=1}^{N}\delta_{\mathbf{z}^{(j)}}, \frac{1}{N}\sum_{j=1}^{N}\delta_{\mathbf{y}^{(j)}})$
    Compute gradients of $\mathcal{L}_{PE}$ wrt to ϕ_{PE}
    Update ϕ_{PE}
  end for
end while

B ARCHITECTURE AND TRAINING DETAILS

For all datasets, the prior-encoder Ψ_{PE} consists of three fully connected hidden layers and ReLU activations. For all datasets, the autoencoder and prior-encoder losses (given in Eqns 7 and 9) are trained iteratively using the optimizer Adam (Kingma & Welling, 2014) with a learning rate of 0.001. We experimented with both $p = 1$ and $p = 2$ (corresponding to p-Wasserstein) in the SW distance and did not find any significant differences; all results in this paper use $p = 2$. For each calculation of the NSW distance, $L = 5$ random nonlinear transformations were taken followed by $M = 50$ one dimensional projections per transformation. Data-specific model and parameter details are given below.

3D Spiral dataset: The input to the prior-encoder is a 40D Gaussian, and the latent space is 3D. The input to the autoencoder is a 40D embedding of a 3D spiral manifold with 10% noise. The dataset consists of 10000 samples, and a batch size of 100 was used. The prior-encoder, the data encoder, and the decoder consist of three Fully Connected (FC) layers with 40 nodes each and ReLU activations. The reconstruction loss is given by the Mean Square Error and $\alpha = 1, \beta = 0.1, \kappa = 0.01$ in Eqn 7. In the absence of convolutional layers, the FSC term encourages the pairwise distances of the minibatch in latent space to be similar to the pairwise distances of the minibatch in the data space. The prior-encoder is trained $k_1 = 2$ times for each training of the autoencoder $k_1 = 1$. Power of distance $h = 2$ is used to compute edge weights for computing network-geodesics.

MNIST dataset: The input to the prior-encoder is a 40 dimensional *mixture of 10 Gaussians*, and the latent space is 5 dimensional. The data encoder takes MNIST images as input using a batch size of 100, and consists of the following layers Conv (1,10,3) $\rightarrow$ BatchNorm $\rightarrow$ ReLu $\rightarrow$ MaxPool

(2,2) $\rightarrow$ Conv (10,16,3) $\rightarrow$ BatchNorm $\rightarrow$ ReLu followed by two FC layers of 512 and 256 nodes respectively with a leaky ReLu nonlinearity. The encoder outputs a 5 dimensional latent representation. The decoder consists of the reverse, i.e., three fully connected layers of size 256, 512, and 1936 nodes respectively. This is followed by ConvTranspose (16,10,3) $\rightarrow$ LeakyReLu $\rightarrow$ Upsample(2,2) $\rightarrow$ ConvTranspose (10,1,3) $\rightarrow$ Sigmoid. The decoder output has size 28×28. The prior-encoder is trained $k_1 = 1$ times for each training of the autoencoder $k_2 = 1$. The reconstruction loss is given by the Binary Cross Entropy and $\alpha = 1, \beta = 0.1, \kappa = 0.001$ in Eqn. 7. The FSC loss encourages the pairwise distances of the minibatch in latent space to be similar to the pairwise distances of the minibatch in *feature* space computed at the output of the last convolutional layer in the data encoder. Energy parameter $h = 2$ is used to compute edge weights for computing network-geodesics.

CelebA dataset: The input to the prior-encoder is a 186 dimensional *mixture of 10 Gaussians*, and the latent space is 128 dimensional. We ran experiments with different latent dimensions and found that the generation ability didn't vary significantly as a function of latent dimension. We trade-off image quality of the input images for computational speed by downsizing the input and employ a fairly simple network compared to state of the art computer vision architectures that use CelebA. The data encoder takes CelebA images (of size $218 \times 178 \times 3$) and downsizes them to size $64 \times 64 \times 3$. The data encoder consists of the following layers Conv (3,16,3) $\rightarrow$ BatchNorm $\rightarrow$ ReLu $\rightarrow$ MaxPool (2,2) $\rightarrow$ Conv (16,32,3) $\rightarrow$ BatchNorm $\rightarrow$ ReLu $\rightarrow$ MaxPool (2,2) $\rightarrow$ Conv (32,64,3) $\rightarrow$ BatchNorm $\rightarrow$ ReLu followed by two FC layers of 512 and 256 nodes respectively with a leaky ReLu nonlinearity. The encoder outputs a 128 dimensional latent representation. As in the case of MNIST, the decoder consists of the reverse, with convolutions replaced by Convolution Transpose, and MaxPool replaced by Upsample. The output of the decoder is passed through a sigmoid nonlinearity and is of size $64 \times 64 \times 3$. The prior-encoder was trained $k_1 = 1$ times for each training of the autoencoder. The reconstruction loss is given by the Binary Cross Entropy and $\alpha = 500, \beta = 50, \kappa = 0.05$ in Eqn. 7. The FSC loss encourages the pairwise distances of the minibatch in latent space to be similar to the pairwise distances of the minibatch in *feature* space, i.e., computed at the output of the last convolutional layer in the encoder. Energy parameter $h = 2$ is used to compute edge weights for computing network-geodesics.

C SPIRAL BASELINE COMPARISONS

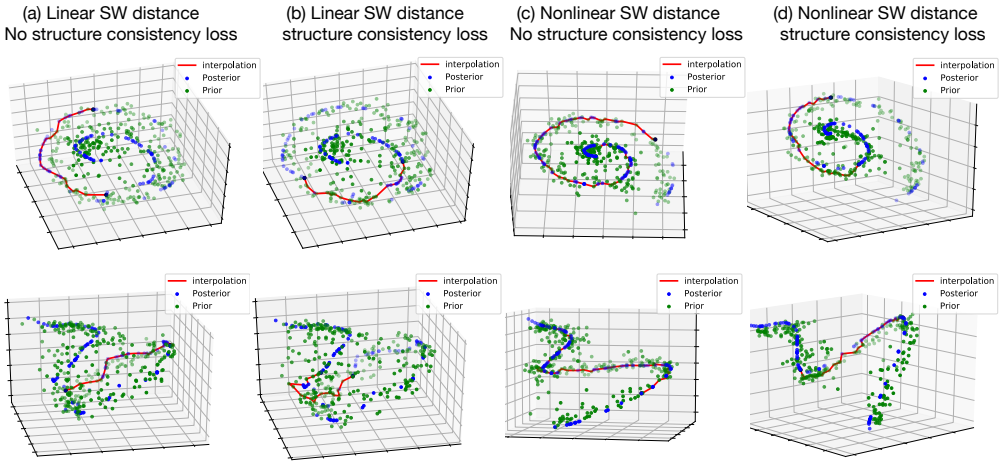

Figure 5: Comparisons of the EPSWAE model with different loss terms. Top panels present top views, and bottom panels present the corresponding side views. The red curves show interpolation between two randomly selected samples using the network-geodesic algorithm. All figures are generated after 100 epochs with a lr=0.01, and batch size =100.$k_1 = 1, k_2 = 2$.

Figure 5 shows the effects of the different loss terms in EPSWAE on the shape of the learned prior and posterior. The position and orientation of the spiral in the 3D plots are random and the views

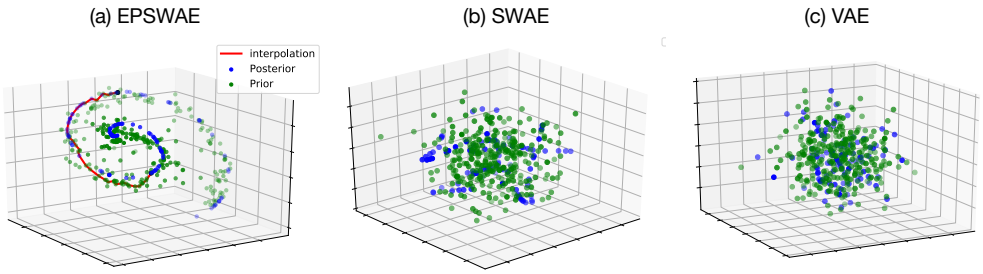

Figure 6: Comparison of EPSWAE with baselines SWAE (Kolouri et al., 2018a) and VAE (Kingma & Welling, 2014). All figures are generated after 100 epochs with a lr=0.01, and batch size =100. For EPSWAE, $k_1 = 1, k_2 = 2$.

in the image are hand-chosen to be equivalent. (a) shows latent space and interpolations (red) using EPSWAE with a linear Sliced Wasserstein distance in the loss, and no structural consistency term. (b) shows that adding a structural consistency term doesn't remarkably improve the quality of the manifold learned, however, consistent with other experiments, it improves interpolation slightly. Note here that since we don't use convolutional layers for the 3D spiral, the stuctural consistency term preserves distance in latent space corresponding to distances in data space. (c) shows that employing the NSW distance term significantly improves the learned structure in latent space. The improvements resulting from incorporation of the NSW and structural consistency terms as seen in these visualizations of the 3D spiral lead us to use both loss terms (as in (d)) on all results in the main paper.

Figure 6 compares the EPSWAE model with baselines VAE and SWAE. The VAE uses KL divergence in the loss, which constrains the prior to be sampled from a Gaussian distribution. While the vanilla SWAE uses the SW distance, the prior remains a Gaussian, leading to an unnatural embedding of the data manifold. In contrast, EPSWAE results in a significantly better learned prior as a consequence of the nonlinear SW distance and the prior-encoder network trained explicitly to improve the latent representation.

D MNIST RESULTS

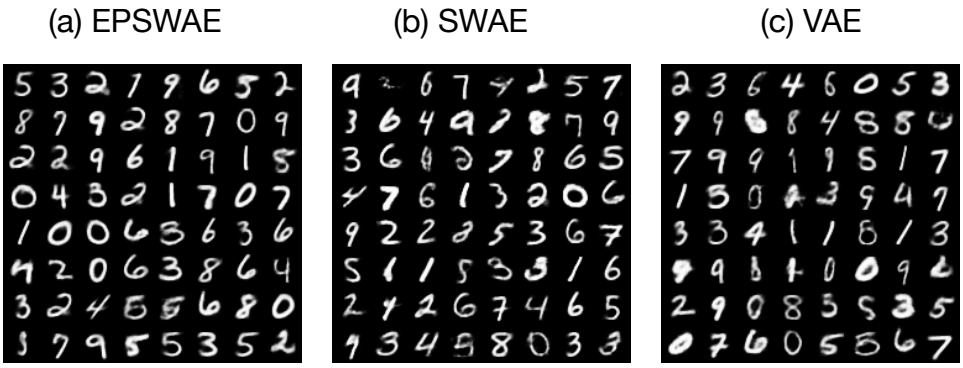

Figure 7: Comparison between (a) EPSWAE (b) SWAE and (c) VAE for generation on MNIST. for all three, batch size = 200, epochs=100, lr = 0.001, for EPSWAE $k_1 = 2$, $k_1 = 1$.

Figures 7 (a,b,c) shows generation on the MNIST dataset after 100 epochs on EPSWAE, baseline SWAE(Kolouri et al., 2018a), and baseline VAE (Kingma & Welling, 2014) respectively. All networks EPSWAE and baselines SWAE and VAE use equivalent architectures (outlined in section of Appendix B), and hyperparameters are optimized individually. As seen in the figure, we observe

that EPSWAE generated samples were consistently found to have a lower fraction of 'false' digits, i.e., digits that are unrealistic.

E EFFECT OF ENERGY PARAMETER ON CELEBA INTERPOLATIONS

Additional CelebA interpolations are shown in Fig. 8. The length of interpolations is automatically selected by the network algorithm. A linearly interpolated point is added between every two samples on the network-geodesic. The figure presents a comparison between energy parameter $h = 1$ and $h = 2$. The energy parameter determines the power of the distance metric used to compute the edge weight. One can think of the higher value ($h = 2$) as corresponding to stronger connections between points, and encouraging shorter hops. In practice, as seen from the figure, there is no clear advantage to choosing a specific value of energy parameter h.

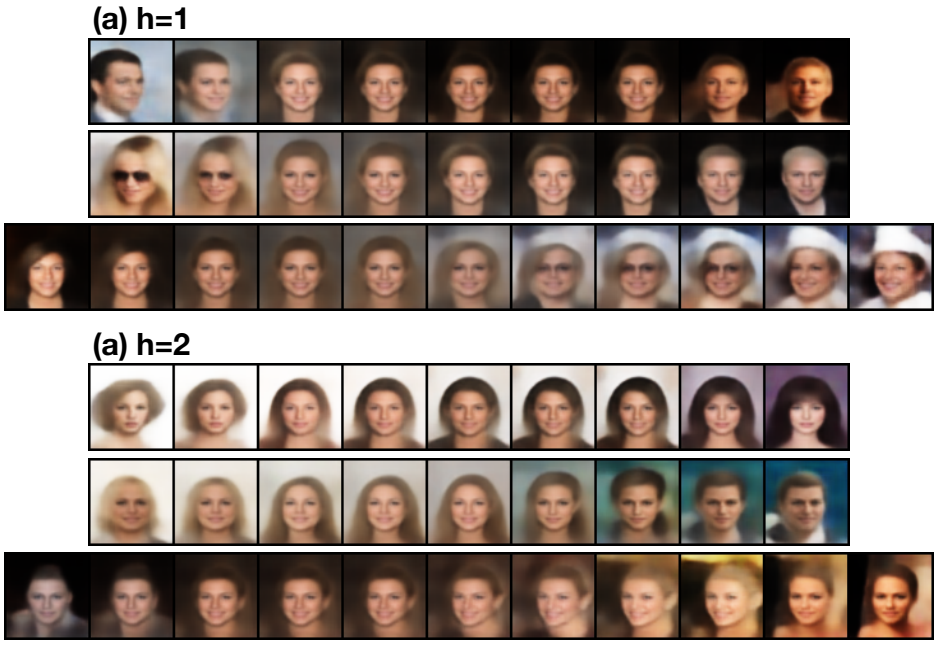

Figure 8: Additional CelebA interpolations. The first and last images are reconstructions of real data, and the interpolations traverse through samples of the prior using the network-geodesic algorithm. (a) Top three panels show interpolations for energy parameter $h = 1$, and (b) bottom three panels show interpolations for energy parameter $h = 2$. Hyperparameters are the same as those used in the paper. A total of a 400 samples in latent space are used.