Uniform Interpolation Constrained Geodesic Learning on Data Manifold

Cong Geng 1  Jia Wang 1  Li Chen 1  Wenbo Bao 1  Chu Chu 1  Zhiyong Gao 1

Abstract

In this paper, we propose a method to learn a minimizing geodesic within a data manifold. Along the learned geodesic, our method is able to generate high-quality interpolations between two given data samples. Specifically, we use an autoencoder network to map data samples into latent space and perform interpolation via an interpolation network. We add prior geometric information to regularize our autoencoder for the convexity of representations so that for any given interpolation approach, the generated interpolations remain within the distribution of the data manifold. Prior to the learning of a geodesic, a proper Riemannian metric should be defined. Therefore, we induce a Riemannian metric by the canonical metric in the Euclidean space which the data manifold is isometrically immersed in. Based on this defined Riemannian metric, we introduce a constant speed loss and a minimizing geodesic loss to regularize the interpolation network to generate uniform interpolations along the learned geodesic on the manifold. We provide a theoretical analysis of our model and use image translation as an example to demonstrate the effectiveness of our method.

1. Introduction

Traditionally, manifold learning methods aim to infer latent representations that can capture the intrinsic geometric structure of data. They are generally applied in dimensionality reduction and data visualization. Classical nonlinear manifold learning algorithms such as Isomap (Tenenbaum et al., 2000), locally linear embedding (Roweis & Saul, 2000) and local tangent space alignment (LTSA) (Zhang & Zha, 2003) preserve local structures in small neighborhoods and derive the intrinsic features of data manifolds. With the success of deep learning on approximating complex non-linear functions, representing data manifolds using neural networks has received considerable interest and been studied from multiple perspectives. Autoencoders (AE) (Kingma & Welling, 2014) and Generative adversarial network (GAN) (Goodfellow et al., 2014) are two notable generative neural network models for learning the geometry structure or the density/distribution of data on the manifold. Recent studies such as adversarial autoencoder (AAE) (Makhzani et al., 2015) and Wasserstein autoencoder (WAE) (Tolstikhin et al., 2018) try to inherit the merits of both AE and GAN while avoiding the disadvantages of them. Other manifold learning methods such as DIMAL (Pai et al., 2019) integrate deep learning into the classical manifold embedding algorithms. These approaches provide the possibility of interpolation on the manifold.

The interpolation between data samples attracts wide attention among the field because it can achieve a smooth transition from one sample to another, such as intermediate face generation and path planning. A path planning algorithm needs to determine the intermediate milestones for a driver who has provided his departure and terminal position. There are several works of literature focusing on the interpolation of data manifold. Agustsson et al. (2019) propose to use distribution matching transport maps to obtain analogous latent space interpolation which preserves the prior distribution of the latent space, while minimally changing the original operation. It can generate interpolated samples with higher quality but it doesn’t have encoder to map the given data samples to latent space. Several other works (Berthelot et al., 2019; Sainburg et al., 2018; Chen et al., 2019b) that combine autoencoders with GANs are proposed to generate interpolated points on data manifold. But limited by the ability of GAN, it’s difficult to generate a convex latent representation when the input manifold has a changing curvature such as swiss-roll and S-curve. To avoid this problem, Arvanitidis et al. (2018) and Chen et al. (2018) all present that the magnification factor (Bishop et al., 1997) can be seen as a measure of the local distortion of the latent space. This helps the generated curves to follow the regions of high data density in the latent space. These methods achieve approximations of the geodesics to make the optimized curve remain within the data manifold. Furthermore, Yang et al. (2018) and Chen et al. (2018) discretize the curve into multiple points and add a penalty term to ensure that the geodesic remain within the data manifold. Chen et al.
(2019a) also find the shortest path in a finite graph of samples as the geodesics on data manifold. This finite graph is built in the latent space using a binary tree data structure with edge weights based on Riemannian distances.

In this paper, we propose a method to capture the geometric structure on a data manifold and find smooth geodesics between data samples on it. Motivated by DIMAL (Pai et al., 2019), we combine our network framework with traditional manifold learning methods. In contrast to existing combinations of autoencoders with GANs, we add some prior geometric information to represent the local geometry of data manifold. The decoding of the latent space is the major obstacle for autoencoders because curvature information of data manifold will be lost in latent representations. We revisit the classical manifold learning algorithms to obtain an approximation of latent embeddings that preserve the local geometric structure of data manifold. Then the encoder and decoder are guided to be optimized towards the expected output with this approximation being the constraint. Using this method, we resolve other algorithms’ problems when the latent distribution is non-convex on some manifolds such as S-curve and swiss-roll. It means our method can unfold the data manifold to flattened latent representations and the new testing points can be mapped to the learned subspace without difficulty.

For manifold interpolation, we propose a geodesic learning method to establish smooth geodesics between data samples on the manifold. We introduce a constant speed loss and minimizing geodesic loss to train a network to interpolate along the geodesics. The Riemannian metric on data manifold can be induced by the canonical metric in the Euclidean space which the manifold is isometrically immersed in. So the geodesics can be obtained by the defined Riemannian metric. Existing methods cannot guarantee that the interpolation along geodesics has a constant speed. We parameterize the generated curve by a parameter $t$. Our constant speed loss achieves uniform interpolation by making $t \in [0, 1]$ be an arc-length parameter of the interpolation curve. According to Riemannian geometry (Carmo, 1992), a curve is a geodesic if and only if its second derivative versus parameter $t$ is orthogonal to the tangent space. Based on this theory, we propose a geodesic loss to force the interpolation network to generate points along the geodesics. Furthermore, geodesic connecting two points may not be unique, such as the geodesics on a sphere. But geodesics are locally unique and the minimizing geodesic is the minimal length of the tangent vector

We propose a framework in which an autoencoder and an interpolation network are introduced to reconstruct the manifold structure. The autoencoder network promotes convex latent representations by adding some prior geometric information to guarantee the generated samples remaining within the distribution of the data manifold.

We propose a constant speed loss and a minimizing geodesic loss to generate geodesic on the manifold given two endpoints. We parameterize each geodesic by an arc-length parameter $t$, such that we can generate points moving along the minimizing geodesic with constant speed and thus fulfill a uniform interpolation.

### 2. Preliminaries

Suppose $M$ is a Riemannian differentiable manifold. First we define tangent vector and tangent space (Carmo, 1992) on $M$. Let $\alpha : (-\epsilon, \epsilon) \rightarrow M$ be a differentiable function called a differentiable curve in $M$. Suppose that $\alpha(0) = p \in M$, let $D$ be the set of functions on $M$ that are differentiable at $p$. Now we can define tangent vectors.

**Definition 2.1** (Tangent vector) The tangent vector to the curve $\alpha$ at $t = 0$ is a function $\alpha'(0) : D \rightarrow \mathbb{R}$ given by

$$\alpha'(0) f = \left. \frac{d(f \circ \alpha)}{dt} \right|_{t=0}, \quad f \in D.$$  

The set of all tangent vectors to $M$ at $p$ will be indicated by tangent space $T_p M$. Let $(U, x)$ be a parametrization of $p$ on manifold $M$, for $\forall q \in M$, suppose $x^{-1}(q) = (z^1(q), z^2(q), \cdots, z^n(q))$. Then the tangent vector at $p$ of the coordinate curve $\frac{\partial}{\partial z^i}|_p$ is defined as

$$\left. \frac{\partial}{\partial z^i} \right|_p (f) = \left. \frac{\partial(f \circ x)}{\partial z^i} \right|_p.$$  

\{ $\frac{\partial}{\partial z^i}|_p$, $1 \leq i \leq n$\} spans a tangent space $T_p M$ at $p$. If the tangent vector is orthonormal with each other, \{ $\frac{\partial}{\partial z^i}|_p$, $1 \leq i \leq n$\} is the orthonormal basis of tangent space.

**Definition 2.2** (Geodesic) A parametrized curve $\gamma : I \rightarrow M$ is a geodesic at $t_0$ if $\frac{D}{dt} \left( \frac{\partial}{\partial t} \right) = 0$ at the point $t_0$; if $\gamma$ is a geodesic at $t$, for all $t \in I$, we say that $\gamma$ is a geodesic.

$\frac{D}{dt}$ is the covariant derivative of a vector field along $\gamma$ (Carmo, 1992). $\frac{\partial}{\partial t}$ is the tangent vector at $t$. If $\gamma : I \rightarrow M$ is a geodesic, then

$$\frac{d}{dt} \left\langle \frac{D}{dt} \frac{d\gamma}{dt}, \frac{D}{dt} \frac{d\gamma}{dt} \right\rangle = 2 \left\langle \frac{D}{dt} \frac{d\gamma}{dt}, \frac{d\gamma}{dt} \right\rangle = 0,$$  

where $\langle \cdot, \cdot \rangle$ is a symmetric,bilinear and positive-definite inner product corresponding to Riemannian metric. That is, the length of the tangent vector $\frac{D}{dt}$ is constant. In the following, we introduce two useful theorems. Chen & Li (2004)
and Carmo (1992) provide the proofs for Theorem 2.1 and Theorem 2.2 respectively.

**Theorem 2.1** Suppose \( f : (M, g) \to \mathbb{R}^n \) is a \( m \)-dimensional Riemannian manifold isometric immersed in euclidean space \( \mathbb{R}^n \). A differentiable curve \( \gamma : I \to M \) is a geodesic on Riemannian manifold \( (M, g) \), if and only if

\[
\left( \frac{d^2 (f \circ \gamma(t))}{dt^2} \right)^\top = 0,
\]

where \( \top \) represents orthogonal projection of tangent space from \( \mathbb{R}^n \) to \( M \).

**Theorem 2.2** If a piece-wise differentiable curve \( \gamma : [a, b] \to M \) (Carmo, 1992), with parameter proportional to arc length, has length less or equal to the length of any other piece-wise differentiable curve joining \( \gamma(a) \) to \( \gamma(b) \) then \( \gamma \) is a geodesic.

A curve which minimizes the action between its endpoints is called a minimizing geodesic. The Hopf-Rinow theorem (Carmo, 1992) guarantees that if the manifold \( M \) is complete, then any two points in \( M \) are joined by at least one minimizing geodesic. Moreover, minimizing geodesic are nonbranching, locally unique and almost everywhere unique.

### 3. Manifold Reconstruction

Original AE and some combinations between AE and GANs can generate high-quality samples from specific latent distributions. But they failed to get convex embeddings on some manifolds with changing curvature such as swiss-roll or S-curve. Some other works (Berthelot et al., 2019; Sainburg et al., 2018; Chen et al., 2019b) were proposed to generate interpolations within the distribution of real data by distinguishing interpolations from real samples. But they also generate unsatisfying samples on the above-mentioned specific manifolds. The encoding results can be seen in Fig. 1. The reason for this problem may be the insufficient ability of GANs and the decoder of the AE. The discriminator of GANs can only distinguish the similarities of distributions between generated samples and real ones. It cannot estimate whether the generated samples are on the data manifold. Furthermore, the autoencoders are prone to put the curvature information of data manifold in latent representations to make the network simple as possible. So the latent embeddings also have a changing curvature to save the curvature information. It may result in the non-convex representations.

In our method, we add some prior geometric information obtained by traditional manifold learning approaches to encoding a convex latent representation. Traditional nonlinear manifold learning approaches such as Isomap, locally linear embedding (LLE), local tangent space alignment (LTSA) are classical algorithms to get a convex embedding by unfolding the manifold. We apply them to our method by adding regularlization to the autoencoder. The loss function of the autoencoder can be written as:

\[
L_{AE} = \left\| D(E(x)) - x \right\| + \left\| E(x) - ML(x) \right\|
\]

\[
+ \left\| D(ML(x)) - x \right\|,
\]

where \( x \) is the input sampled on the data manifold. \( ML(x) \) represents the low-dimensional embedding of input \( x \). Encoder \( E \) and Decoder \( D \) of an autoencoder are trained to minimize the above loss \( L_{AE} \). With \( ML(x) \) as an expected low-dimensional approximation, the encoder is forced to train towards obtaining a convex latent representation while the decoder is forced to learn the lost curvature information on latent embeddings. The original autoencoder loss in \( L_{AE} \) brings input \( x \) back to the original input. Behaviors induced by the \( L_{AE} \) loss can be observed in Fig. 1: the swiss-roll can be flattened on 2-dimensional latent space. For our experiments, we choose local tangent space alignment (LTSA) to compute the approximated low-dimensional embeddings for regularlizing the autoencoder due to its learned local geometry.

### 4. Geodesic Learning

In our model, we denote \( \mathcal{X} \) as a data manifold. The Riemannian metric on \( \mathcal{X} \) can be obtained in the following way. We can establish an identical mapping \( f : \mathcal{X} \to \mathbb{R}^N \) to immerse manifold \( \mathcal{X} \) into a high-dimensional Euclidean space \( \mathbb{R}^N \).

Then the Riemannian metric on manifold \( \mathcal{X} \) can be induced by the canonical metric in the Euclidean space to guarantee
that \( f \) is an isometric immersion. Thus to obtain a geodesic \( \gamma(t) \) on manifold \( \mathcal{X} \), we can use the Riemannian geometry on subspace \( f(\mathcal{X}) \) and the characteristics of isometric immersion \( f \).

4.1. Interpolation Network

The method of interpolating in the latent space can be varied in different situations. The simplest interpolation is linear interpolation as \( z = (1 - t) \cdot z_1 + t \cdot z_2 \). This method enforces the encoder to map manifold into a linear space. For geodesic learning, linear interpolation is not applicable in most situations. Yang et al. (2018) propose to use the restricted class of quadratic functions and Chen et al. (2018) apply a neural network to parameterize the geodesic curves. Considering our manifold reconstruction method can unfold the data manifold, thus we employ polynomial functions similar to Yang et al. (2018)’s approach for our interpolation network. Different from Yang et al. (2018)’s method, we employ cubic functions to parameterize due to the diversity of latent representations. Therefore, a curve has four free parametric vectors. Each vector is \( d \)-dimensional while \( d \) is the dimension of latent coordinates. In practice, we train a geodesic curve \( c(t) \) that connects two pre-specified points \( z_0 \) and \( z_1 \), so the function should be constrained to satisfy \( c(0) = z_0 \) and \( c(1) = z_1 \). We perform the optimization using gradient descent.

4.2. Constant Speed Loss

![Diagram](image)

Figure 2. Framework of our method.

We can produce interpolations along a curve on manifold \( \mathcal{X} \) by decoding the output \( c(t) \) of the interpolation network as \( t \) from 0 to 1. Then the isometric immersion \( f \) maps curve \( G(t) = D(c(t)) \) on manifold \( \mathcal{X} \) to a curve \( f(G(t)) \) on \( \mathbb{R}^N \). Suppose \( y^i \) is the \( i \)-th coordinate in \( \mathbb{R}^N \), \( 1 \leq i \leq N \), then \( \frac{\partial}{\partial y^i} \) is the tangent vector along \( i \)-th coordinate curve in \( \mathbb{R}^N \). Then we get

\[
\frac{df^i(G(t))}{dt} \bigg|_{t=t_0} = \nabla_{t_0} G \cdot \frac{\partial}{\partial y^i},
\]

where \( G'(t_0) \) is the tangent vector along curve \( G(t) \) at \( t_0 \) and \( f_{sG(t_0)} : T_{G(t_0)} \mathcal{X} \rightarrow T_{f(G(t_0))} \mathbb{R}^N \) is a tangent mapping (Chen & Li, 2004). \( f^i \) is the \( i \)-th component of \( f \). The length of tangent vector \( f_{sG(t_0)}(G'(t_0)) \) can be defined as

\[
|f_{sG(t_0)}(G'(t_0))| = \left\| \nabla_{t_0} G \cdot \frac{\partial}{\partial y^i} \right\|, \tag{7}
\]

where \( \langle \cdot \rangle \) is the canonical inner product in Euclidean space. Furthermore, we can deduce that

\[
|f_{sG(t_0)}(G'(t_0))| = \left\| \nabla_{t_0} G \right\|, \tag{8}
\]

where \( \nabla_{t_0} G \) denotes the gradients of networks’ output \( G(t) \) with respect to \( t \) at \( t_0 \). \( \| \cdot \| \) denotes the Frobenius norm of a matrix or 2-norm of a vector. From Section 2 we know if a curve is a geodesic, then the length of the tangent vector along the curve is constant. Thus if \( G(t) \) is a geodesic, because \( f \) is an isometric immersion, we can obtain

\[
|f_{sG(t)}(G'(t))| = C, \quad t \in [0, 1]. \tag{9}
\]

For our constant speed loss, we sample \( n \) points subjecting to uniform distribution on \([0, 1]\), \( 0 = t_0 < t_1 < \cdots < t_n = 1 \).
According to the definition of tangent space in Riemannian geometry, the tangent vector at \( z_0 \) is the local coordinate of \( G(t_0) \). Therefore, our encoder maps a point on the manifold \( X \) to its local coordinates. According to the definition of tangent space in Riemannian geometry, the tangent vector at \( G(t_0) \) along local coordinate curve are the basis of its tangent space \( T_{G(t_0)}X \). Suppose \( z \) is the local coordinate of \( G(t) \), suppose \( z^i \) is the \( i \)-th coordinate of \( z \), the tangent vector at \( G(t_0) \) of coordinate curve can be represented as \( \frac{\partial}{\partial z^i}G(t_0) \). Then we can obtain:

\[
    f_{*G(t_0)}\left( \frac{\partial}{\partial z^i} \right) = \nabla_{z_0} D^i \frac{\partial}{\partial y^i},
\]

where \( z_0 \) is the local coordinate of \( G(t_0) \), \( D^i \) is the \( i \)-th component of the decoder function \( D \). \( \nabla_{z_0} D^i \) denotes the gradients of decoder's output \( D^i(\xi(t)) \) with respect to \( z \) at \( z_0 \). The tangent space \( T_{f(G(t))}f(X) \) can be obtained as \( \{ f_{*G(t_0)}(\frac{\partial}{\partial z^i}), 1 \leq i \leq d \} \). From Theorem 2.1, we can deduce that if \( G(t) \) is a geodesic, then

\[
    \left( \frac{d^2 f(G(t))}{dt^2} \right)_{|t=t_0}, f_{*G(t_0)}\left( \frac{\partial}{\partial z^i} \right) = 0, \quad 1 \leq i \leq d,
\]

which is equivalent to

\[
    \left( \frac{d^2 G(t)}{dt^2} \right)_{|t=t_0}^T \cdot \nabla_{z_0} D = 0,
\]

where \( \frac{d^2 G(t)}{dt^2} \) denotes the second derivative of the decoder’s output \( G(t) \) with respect to \( t \) at \( t_0 \). Now we can redefine our geodesic loss as follows:

\[
    l_{geo} = \left\| [\omega_1, \omega_2, \cdots, \omega_N] \right\|,
\]

where

\[
    \omega_i = \left( \frac{d^2 G(t)}{dt^2} \right)_{|t=t_0}^T \cdot \nabla_{z_0} D.
\]

For \( \frac{d^2 G(t)}{dt^2} \) in practice, we can approximate it by

\[
    \frac{d^2 G(t)}{dt^2} \approx \frac{G(t_0 + \Delta t) + G(t_0 - \Delta t) - 2G(t_0)}{\Delta t^2}.
\]

Geodesic loss and constant speed loss jointly force curve \( G(t) \) to have zero acceleration. That is, \( G(t) \) is a geodesic on data manifold. But geodesic connecting two points may not be unique, such as the geodesic on a sphere. According to Theorem 2.2 in Section 2, the minimizing geodesic is the curve with minimal length connecting two points. Thus our model employs the characteristic of minimal length to guarantee \( G(t) \) is a minimizing geodesic. Different from the summation corresponding to the curve energy by Yang et al. (2018) and the summation of the rate of change at interpolated points by Chen et al. (2018), we just approximate the curve length using the summation of Euclidean distance between adjacent interpolated samples on \( f(X) \) which slacks the choice of \( t \). The minimizing loss to minimize curve length is proposed as:

\[
    l_{min} = \sum_{i=1}^{n} \left| f(G(t_i)) - f(G(t_{i-1})) \right| \left\| G(t_i) - G(t_{i-1}) \right\|,
\]

\[
    || \cdot || \text{ denotes the Frobenius norm of an output matrix or 2-norm of an output vector.}
\]

To summarize this part, the overall loss function of our interpolation network is:

\[
    L_{total} = \lambda_1 l_{conspeed} + \lambda_2 l_{geo} + \lambda_3 l_{min},
\]

where \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) are the weights to balance these three losses. Under this loss constraint, we can generate interpolations moving along the minimizing geodesic with constant speed and thus fulfill a uniform interpolation. The framework of our method is shown in Fig 2.
5. Experimental Results

In this section, we present experiments on the geodesic generation and image translation to demonstrate the effectiveness of our method.

5.1. Geodesic Generation

First, we do experiments on 3-dimensional datasets since their geodesics can be better visualized. We choose the semi-sphere and swiss-roll as our data manifolds.

**Semi-sphere dataset.** We randomly sample 4,956 points subjecting to the uniform distribution on semi-sphere. In Fig. 3, we compare our approach with other interpolation methods, i.e., AAE (Goodfellow et al., 2014), A&H (Arvanitidis et al., 2019) and Nutan Chen’s method (Chen et al., 2018). From Riemannian geometry, we know the geodesic of a sphere under our defined Riemannian metric is a circular arc or part of it. The center of this circular arc is the center of this sphere. AAE cannot guarantee that the curve connecting two points is a geodesic. A&H can find the shortest path connecting two corresponding reconstructed endpoints. But the endpoints are inconsistent with the original inputs due to the uncertainty of their VAE network and their stochastic Riemannian metric. Nutan Chen’s method can generate interpolations along a geodesic, but they cannot fulfill a uniform interpolation. In Fig. 3, we observe that our method can generate uniform interpolation along a fairly accurate geodesic on semi-sphere based on the defined Riemannian metric.

**Swiss-roll dataset.** We choose swiss-roll to demonstrate the effectiveness of our method on manifolds that have large curvature variations. We randomly sample 5,000 points subjecting to the uniform distribution on the swiss-roll manifold. We compare our approach with AAE (Makhzani et al., 2015), ACAI (Berthelot et al., 2019), GAIA (Sainburg et al., 2018) and the method that applies GAN on feature space proposed by Chen et al. (2019b). Fig. 4 shows the experimental results. We observe that except our approach, other methods fail to generate interpolations within the data manifold because they cannot encode efficient latent embeddings for linear interpolations.

We do ablation studies on the semi-sphere dataset to investigate the effect of different losses proposed in our method, including the constant speed loss, geodesic loss and minimizing loss. Fig. 5 presents the results obtained by the combinations of different losses. We observe that the constant speed loss can promote a uniform interpolation. Compared with linear interpolation, our network can generate a shorter
Table 1. The estimated distance of interpolation curve trained with different losses proposed in our method.

| Loss                  | Distance |
|-----------------------|----------|
| Linear                | 1.3110   |
| $l_{\text{conspeed}}$ | 1.4710   |
| $l_{\text{min}}$      | 1.1571   |
| $l_{\text{conspeed}} + l_{\text{min}}$ | 1.1277   |
| $l_{\text{conspeed}} + l_{\text{geo}}$ | 6.0460   |
| $l_{\text{conspeed}} + l_{\text{geo}} + l_{\text{min}}$ | 1.1028   |
| real geodesic         | 1.0776   |

The estimated curve length shown in Table 1 demonstrates our observation. The generated curve trained with all three losses contributes the best result and the estimated curve length 1.1028 is the closest to the length of real geodesic, namely 1.0776.

5.2. Image translation

To further demonstrate our model’s effectiveness in image translation, we choose MNIST (LeCun, 1998) and CelebA (Liu et al., 2015) datasets. MNIST dataset contains handwritten digits with 10 different classes and CelebA dataset contains 202,599 face images. First, we need to employ the LTSA algorithm to compute the approximated low-dimensional embeddings to train our autoencoder. Due to the computing limitation of our device, we randomly select 1,079 images on each dataset to be applied to LTSA. Then we use these computed embeddings to train our autoencoder network. Next, for each dataset, we randomly select two images as the start point and endpoint in the data manifold. We employ different interpolation methods to realize an image translation between these two images. For the MNIST dataset, we do a comparative experiment using linear interpolation and our approach. In Fig. 6, we observe that only the start image and end image can be well reconstructed with linear interpolation. The quality of the interpolated images is bad that sometimes we cannot even tell whether they are digital images or not. Our interpolation method can fulfill a uniform interpolation with high visual quality. The reason for the different results of the two interpolation methods lies in that linear interpolation may make the interpolated points to traverse highly uncertain regions on the manifold. Our interpolation method ensures our generated curves are following the data manifold.

For the CelebA dataset, we also use the methods above to interpolate images between two selected face images and show the results in Fig. 7. Only a linear interpolation fails to generate uniform interpolated face images, i.e., the interpolated images are easy to change suddenly rather than changing gradually. On the contrary, our interpolation method can reconstruct visually pleasing face images with a uniform translation. Experiments of image translation on the two datasets all demonstrate the usefulness and validity of our model.

6. Conclusion

We propose a framework to explore the geometric structure of the data manifold. We apply an autoencoder to generate points on the desired manifold and propose a constant speed loss and a minimizing geodesic loss to generate geodesic on the underlying manifold given two endpoints. Different from existing methods in which geodesic is defined as the shortest path on the graph connecting data points, our model defines geodesic consistent with the definition of a geodesic in Riemannian geometry. For application, we apply our model to image translation to fulfill a uniform interpolation along the minimizing geodesics.
Figure 6. Results of linear interpolation and our interpolation method on Minist Dataset. (a) Results of linear interpolation. (b) Results of our interpolation method.

Figure 7. Results of linear interpolation and our interpolation method on CelebA Dataset. (a) Results of linear interpolation. (b) Results of our interpolation method.
References

Agustsson, E., Sage, A., Timofte, R., and Van Gool, L. Optimal transport maps for distribution preserving operations on latent spaces of Generative Models. In Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent Space Oddity: On the Curvature of Deep Generative Models. In Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018.

Arvanitidis, G., Hauberg, S., Hennig, P., and Schober, M. Fast and Robust Shortest Paths on Manifolds Learned from Data. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 10, 2019.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019.

Bishop, C. M., Svensen, M., and Williams, C. K. Magnification factors for the som and gtm algorithms. In Proceedings 1997 Workshop on Self-Organizing Maps, 1997.

Carmo, M. P. d. Riemannian geometry. Birkhäuser, 1992.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and van der Smagt, P. Metrics for Deep Generative Models. In Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS), 2018.

Chen, N., Ferroni, F., Klushyn, A., Paraschos, A., Bayer, J., and van der Smagt, P. Fast approximate geodesics for deep generative models. In International Conference on Artificial Neural Networks, 2019a.

Chen, W. and Li, X. Introduction to Riemann Geometry: Volume I. Pekking University Press, 2004.

Chen, Y.-C., Xu, X., Tian, Z., and Jia, J. Homomorphic latent space interpolation for unpaired image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2408–2416, 2019b.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Advances in neural information processing systems, pp. 2672–2680, 2014.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), 2014.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), Proceedings of the 17th International Conference on Machine Learning (ICML 2000), pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

LeCun, Y. The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/, 1998.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision (ICCV), pp. 3730–3738, 2015.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

Pai, G., Talmon, R., Bronstein, A., and Kimmel, R. DIMAL: Deep Isometric Manifold Learning Using Sparse Geodesic Sampling. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.

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

Sainburg, T., Thielk, M., Theilman, B., Migliori, B., and Gentner, T. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. arXiv preprint arXiv:1807.06650, 2018.

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

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein Auto-Encoders. In Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018.

Yang, T., Arvanitidis, G., Fu, D., Li, X., and Hauberg, S. Geodesic Clustering in Deep Generative Models. arXiv:1809.04747 [cs, stat], September 2018.

Zhang, Z. and Zha, H. Nonlinear dimension reduction via local tangent space alignment. In International Conference on Intelligent Data Engineering and Automated Learning, pp. 477–481. Springer, 2003.