Improving Autoencoder Image Interpolation via Dynamic Optimal Transport

Xue Feng
University of California, Davis
xffeng@ucdavis.edu

Thomas Strohmer
University of California, Davis
strohmer@math.ucdavis.edu

Abstract

Autoencoders are important generative models that, among others, have the ability to interpolate image sequences. However, interpolated images are usually not semantically meaningful. In this paper, motivated by dynamic optimal transport, we consider image interpolation as a mass transfer problem and propose a novel regularization term to penalize non-smooth and unrealistic changes in the interpolation result. Specifically, we define the path energy function for each path connecting the source and target images. The autoencoder is trained to generate the L^2 optimal transport geodesic path when decoding a linear interpolation of their latent codes. With a simple extension, this model can handle complicated environments, such as allowing mass transfer between obstacles and unbalanced optimal transport. A key feature of the proposed method is that it is physics-driven and can generate robust and realistic interpretation results even when only very limited training data are available.

1. Introduction

Autoencoders are known to uncover the underlying structure of a dataset by learning to encode data points to lower-dimensional latent codes which can then be decoded to reconstruct the input data. In more recent studies, evidence has emerged demonstrating the robust capabilities of autoencoders in the domain of generative modeling [12, 15]. They also have been used successfully to interpolate between data points by decoding a convex combination of their latent codes. This is useful for many applications, such as image sequence interpolation, also known as frame interpolation.

However, this process of image interpolation by decoding a convex combination of their latent codes often leads to artifacts or yields unrealistic outcomes. See Figure 1a for an example. There are two prevalent strategies to improve the quality of the interpolated result. The first approach is to introduce a regularization loss term to penalize unrealistic results, such as an adversarial term [4, 15, 20]. In the adversarial regularization method, a *critic network* is trained to evaluate certain criteria, such as the similarity between the interpolated image and the training data, while an autoencoder is concurrently trained to fool this critic network. A second approach consists of shaping the latent representation to follow a manifold consistent with the training images [6, 12, 16, 19, 21]. The idea is that the incongruities in the interpolation result are probably caused by the fact that such straightforwardly interpolated latent vectors stray from the data manifold. So shaping the latent space may be benifical.

In this paper, we propose a regularization term that aligns with the first strategy, using a robust physics model to enhance the interpolated images produced by decoding a linear combination of latent codes. While there are numerous possibilities to define some notion of *data sequence interpolation*, it is natural to consider the image interpolation problem as a mass transfer problem. For example, in a grayscale image, the value of the pixels represents the mass at that location. In this context, image interpolation is the process of transferring the mass from the source image to the target image. There are infinitely many possible paths to transfer the mass. We define the best path as the one with the least transportation cost.

In the past decade, optimal transport (OT) has developed into a highly-regarded research field in machine learning areas such as generative modeling [1], domain adaption[9], and image interpolation. The traditional optimal transport defines a family of metrics, known as the Wasserstein distance, between probability distributions. We can solve the optimal transport plan and then use it to push the source image to get the interpolation path. Nevertheless, the metric itself cannot directly assess the quality of the interpolated images.

Our motivation comes from the dynamic OT proposed by Benamou and Brenier [3]. Dynamic OT considers the mass transfer problem in a fluid mechanics framework and associates each path with a kinetic energy cost. The path with minimal energy is the so-called geodesic path according to the Wasserstein metric. The continuity equation constrain ensures that the path will follow the law of physics. Meanwhile, due to its fluid mechanics nature, dynamic OT can

handle complex scenarios and model real-world problems, such as those involving obstacles or varying transport costs over time [3, 17].

Inspired by this, we define a path energy term by eliminating the momentum variable in the original dynamic OT model, and add it to the autoencoder as a regularization term to measure the performance of the generated path. In this way, the interpolated path is trained to be consistent with physical principles. Our contribution is summarized as follows:

- Our regularization term is mathematically explainable and encourages the autoencoder to generate geodesic paths between images, even in an environment with obstacles. This also brings a continuous and smooth morphing along the interpolated images.
- We study the properties of the path energy term. It has a closed form in a discrete setting. Gradient properties are given to facilitate training.
- We evaluate the effectiveness of our approach in a variety of scenarios. It is noteworthy that traditional deep learning techniques usually require a considerable amount of training data to accurately learn features. Additionally, conventional optimization techniques are usually restricted to interpolating between individual pairs of images. On the contrary, our method produces robust and smooth interpolation results in all cases.
- Our method can be regarded as a variant of dynamic OT, where the path variable is parameterized by the weight parameters of the autoencoder network. As we know, this is the first successful attempt to apply optimal transport with a large deep learning training set on interpolation problem.

Generative AI models, such as Stable Diffusion, DALL-E and ChatGPT, have been the focus of much attention in recent years. Despite their impressive results, these models require a large amount of training data. Unfortunately, in many cases, obtaining such data can be difficult, costly, or even impossible due to privacy issues. Our method is a type of physics-driven deep learning[7], which is becoming popular in the fields of machine learning and computational physics recently. It can produce more reliable and realistic results, particularly in situations where data is scarce.

In section 2, we will discuss some related research. Section 3 will present our proposed approach, and section 4 will provide numerical results.

1.1. Notation

We briefly introduce some important notation of our work. A standard autoencoder consists of an encoder $z = \mathbb{E}(x)$ where z is the latent code and a decoder $\hat{x} = \mathbb{D}(z)$. In this context, z represents the *latent code*, a lower-dimensional representation of the input data x. The encoder and decoder are trained simultaneously to recover the input data, most commonly by minimizing a loss function $||x - \hat{x}||^2$.

In this paper, we focus on image interpolation using an autoencoder by decoding a linear interpolation of two latent codes. Specifically, for a pair of images x_i, x_j with the corresponding latent codes z_i and z_j, we define a new latent code

$$z_{i\to j}(t) = (1-t)z_i + tz_j, \quad t \in [0,1].$$

Then, the output $\mathbb{D}(z_{i\to j}(t))$ by decoding $z_{i\to j}(t)$ describes a path showing how pixels from x_i move to x_j when t increases from 0 to 1.

2. Related Work

2.1. Classic Optimal Transport and Wasserstein Distance

Optimal transport (OT) is a well-developed mathematical theory that defines a family of metrics between probability distributions, with a wide range of potential applications and extensions [18, 22]. In this work, we only consider the problem of two-dimensional image interpolation. Thus, without loss of generality, we define two density functions $\rho_0(s) \geq 0$ and $\rho_T(s) \geq 0$ with $s \in [0,1]^2$, which we assume to be bounded by total mass one

$$\int_{[0,1]^2} \rho_0(s)ds = \int_{[0,1]^2} \rho_T(s)ds = 1.$$

The associated $L^p(p \geq 1)$ Wasserstein (or Kantorovich-Rubinstein) distance between ρ_0 and ρ_T is defined by:

$$d_p(\rho_0, \rho_T)^p = \inf_M \int |M(s) - s|^p \rho_0(s)ds,$$

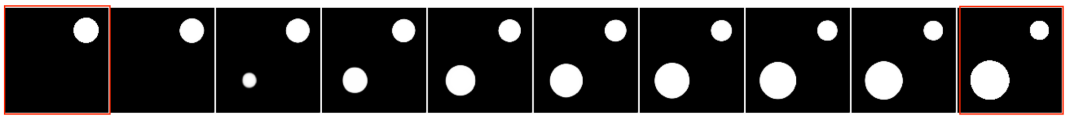
where the infimum is taken over all maps $M : [0,1]^2 \to [0,1]^2$ that push forward the measure $\rho_0 ds$ to $\rho_T ds$. The map M that minimizes this objective, denoted as M^*, is known as the optimal transport map. To interpolate between images, we can compute the optimal transport map and push forward the source image to the target image according to this map [5]:

$$\rho(t,s) = \rho_0\left(M_t(s)\right)\left|\det\left(\partial M_t(s)\right)\right|,$$

where $M_t = (1-t)\,\mathrm{I} + tM^*$. Here, $\rho(t,s)$ is the geodesic path between ρ_0 and ρ_T. In the context of image interpolation, $\rho(t)$ represents the interpolated image at any given time t.

2.2. Dymamic Optimal Transport

Dynamic optimal transport, initially proposed in [3], resets the mass transfer problem into a continuum mechanics framework. The path $\rho(t)$ is described by advecting the measure using a vector field $\mathbf{v}(t)$, with $\mathbf{v}$ and ρ satisfying the continuity equation, AKA, the conservation of mass formula. Dynamic optimal transport finds the geodesic path



(a) Interpolation results using a standard (vanilla) autoencoder. The changes between images are localized.

(b) Interpolation results using our method. The result shows a smooth transition between images that is consistent with the laws of physics.

Figure 1. Comparative analysis of image sequence interpolation. Given the images marked in red, the interpolated images are generated by decoding a linear combination of their latent codes after training.

$\rho(t,s)$ between $\rho_0(s)$ and $\rho_T(s)$ by minimizing the kinetic energy along the path:

$$\begin{aligned} \min_{\rho,\mathbf{v}} \quad & \frac{1}{2}\int_0^T \int_{[0,1]^2} \rho(t,s)|\mathbf{v}(t,s)|^2 dsdt, \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, , \\ & \rho(0,\cdot) = \rho_0, \quad \rho(T,\cdot) = \rho_T, \end{aligned} \tag{1}$$

where the first constraint is the continuity equation, and the remaining are the initial and final conditions Proper boundary conditions in the velocity field should be considered. In our work, we consider the Dirichlet boundary condition, specifically zero at the boundary positions. Extension to more complex fluid mechanics models and combining it with other metrics such as L^2 distance [3, 17] is flexible, by changing the objective function or constraints.

After introducing a new variable, momentum $\mathbf{m} = \rho\mathbf{v}$, the above problem can be reformulated as a convex problem as follows:

$$\begin{aligned} \min_{\rho,\mathbf{m}} \quad & \frac{1}{2}\int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t,s)|^2}{\rho(t,s)} dsdt, \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\mathbf{m}) = 0, \\ & \rho(0,\cdot) = \rho_0, \quad \rho(T,\cdot) = \rho_T. \end{aligned} \tag{2}$$

This problem remains challenging to solve since the objective function is nonsmooth. After discretization, there are several ways to solve it, such as the Douglas-Rachford algorithm and the primal-dual method in [17], as well as the alternating direction multiplier method (ADMM) in [3]. The drawback is that it is computationally expensive.

The square of the L^2-Wasserstein distance is equal to $2T$ times the infimum of dynamic optimal transport defined in (1) [3]. Dynamic OT provides the geodesic point of view of OT and allows us to explore the path space. In the context of image sequence interpolation, the optimal solution $\rho(t)$ gives the interpolated sequence directly.

2.3. Adversarial Regularizer

The adversarial regularizer-based method is an important branch of autoencoder image interpolation. This technique utilizes generative adversarial networks (GANs) [11] to produce high-quality interpolated images. The generative model, here an autoencoder, produces the interpolated images, while the discriminative model, referred to as the critic, evaluates the generated images with certain criteria such as the similarity between the interpolated image and the training data. The output of the critic network is added as a regularization term for the autoencoder loss function, helping it to refine the interpolated images further through joint training. Key research in this area includes works by [2, 4, 14, 15]. Despite the promising results these methods have achieved in refining image interpolation techniques, their data-driven approach significantly differs from our method which leverages robust mathematical models to ease the need for large amount of training data.

3. Path Energy

In this section, we explain the motivation to define path energy and how to apply it to the autoencoder, encouraging the interpolated results to be realistic.

As discussed earlier, given two images x_i and x_j, the autoencoder has the ability to generate a path $\mathbb{D}(z_{i\to j}(t)), t \in [0,1]$ by decoding a linear combination of their latent codes. However, these generated images are usually not as realistic as expected. See Figure 1a for an example. Thus, we need an evaluation metric to penalize some "bad" paths.

The image interpolation problem can be interpreted as a mass transfer problem. For a gray image, the value of the pixel represents the mass at that position. Thus, dynamic OT can be adapted to the image interpolation problem. The problem (2) minimizes over two variables $\rho, \mathbf{m}$, with ρ representing a path connecting x_i and x_j and $\mathbf{m}$ describing the momentum of fluid movement. A natural idea is that for any given ρ, we can optimize through $\mathbf{m}$ to find the least motion momentum and use the kinetic energy as the path

energy value associated with ρ.

Formally, for any given $\rho(t,s)$ satisfying $\rho(0,\cdot) = \rho_0, \rho(T,\cdot) = \rho_T$, we can define the path energy over $\rho(t,s)$ based on normal optimal transport as

$$J(\rho) = \min_{\mathbf{m}} \quad \frac{1}{2}\int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t,s)|^2}{\rho(t,s)} dsdt \\ \text{s.t.} \quad \partial_t \rho + \nabla \cdot (\mathbf{m}) = 0. \tag{3}$$

Note that $J(\rho)$ is convex as (2) is a bivariate convex problem.

We consider a variation in this paper in which there are obstacles in the environment. [17] proposed a generalized cost function with spatially varying weights, which can be accomplished by adding a constraint to make the momentum zero at all obstacle positions:

$$m(t,s) = 0, \quad \forall s \in C,$$

where C is the set of obstacle positions.

Another variation we consider in this paper is unbalanced OT to address the case when the source and target images have different masses. A general methodology consists of introducing a source term $\mathfrak{s}$ into the continuity equation and penalizing it in the same way as the momentum [8]. Formally, the path energy over $\rho(t,s)$ based on unbalanced OT is

$$J(\rho) = \min_{\mathbf{m}} \quad \frac{1}{2}\int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t,s)|^2}{\rho(t,s)} + \tau \frac{|\mathfrak{s}(t,s)|^2}{\rho(t,s)} dsdt \\ \text{s.t.} \quad \partial_t \rho + \nabla \cdot (\mathbf{m}) = \mathfrak{s}, \tag{4}$$

where τ is the source term weight.

3.1. Discretization and Solver

Here, we discretize the time to $\{0,1,2,..,T\}$, and assume that the space is uniformly discreted at the points $(i/n, j/n) \in [0,1]^2$. Source and target densities are represented as $\rho_0, \rho_t \in \mathbb{R}^{n,n}$ and a path $\rho \in \mathbb{R}^{T+1,n,n}$. We adopt a staggered grid discretization scheme, which is commonly used in fluid dynamics [17]. The momentum vector $\mathbf{m}$ and its corresponding weight vector $\mathbf{w}$ will be defined at half-grid points in each direction of space at time t. We denote $\mathbf{m}_t = (\mathbf{m}_t^1, \mathbf{m}_t^2)$ where $\mathbf{m}_t^1 \in \mathbb{R}^{n+1,n}$ and $\mathbf{m}_t^2 \in \mathbb{R}^{n,n+1}$ and $\mathbf{w}_t = (\mathbf{w}_t^1, \mathbf{w}_t^2)$ where $\mathbf{w}_t^1 \in \mathbb{R}^{n+1,n}$ and $\mathbf{w}_t^2 \in \mathbb{R}^{n,n+1}$. Given a path ρ, $\mathbf{w}_t$ is deterministicaly defined as follows

$$\mathbf{w}_{t,i,j}^1 = \frac{2}{\rho_{t,i,j} + \rho_{t,i+1,j}}, \qquad \mathbf{w}_{t,i,j}^2 = \frac{2}{\rho_{t,i,j} + \rho_{t,i,j+1}}.$$

Using the staggered grid discretization scheme, divergence operator associated with a vector field $\mathbf{m}_t$ in the linear constraint of problem (3) is defined as:

$$(\nabla \cdot \mathbf{m}_t)_{i,j} = \mathbf{m}_{t,i+1,j}^1 - \mathbf{m}_{t,i,j}^1 + \mathbf{m}_{t,i,j+1}^2 - \mathbf{m}_{t,i,j}^2.$$

After flattening the vectors $\mathbf{m}$ and $\mathbf{w}$, problem (3) becomes:

$$J(\rho) = \min_{\mathbf{m}} \sum_{t=0,1,2,..T-1} \mathbf{m}_t^T \operatorname{Diag}(\mathbf{w}_t)\mathbf{m}_t \\ \text{s.t.} \quad \nabla \cdot (\mathbf{m}_t) = b_t, t = 0,1,2,...T-1$$

where $b_t = -\partial_t \rho = \rho_t - \rho_{t+1}$.

This is a quadratic problem with linear constraint, and its optimality condition (KKT condition) is given by

$$\begin{bmatrix} \operatorname{Diag}(\mathbf{w}_t) & \nabla \cdot^\top \\ \nabla \cdot & 0 \end{bmatrix} \begin{bmatrix} \mathbf{m}_t \\ \lambda_t \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b_t} \end{bmatrix}, t = 0,1,2,...,T-1, \tag{5}$$

where λ_t is the Lagrange multiplier. After solving for $\mathbf{m}$ in the above linear system, we get a closed formula for the path energy defined in (3):

$$J(\rho) = \sum_{t=0}^{T-1} b_t^T \left(\nabla \cdot \operatorname{Diag}(\mathbf{w_t})^{-1} \nabla \cdot^\top\right)^{-1} b_t,$$

by assuming $\mathbf{w}_t$ are positive. The extensions to the obstacle case and unbalanced OT case are similar and can be found in Section 6 in the supplementary material.

3.2. Gradient Computation

First, we denote $\nabla \cdot \operatorname{Diag}(\mathbf{w}_t)^{-1} \nabla \cdot^\top$ as A_t, which is a reweighted poison operator. There are T sparse linear systems to solve in $J(\rho)$:

$$A_t y = b_t, \quad t = 0,1,2,...T-1. \tag{6}$$

We show the gradient property of our path energy term in the following theorem, which is important in the backpropagation of neural network training.

Theorem 1. *The first-order gradient of the path energy defined in (3) at time $t = 0,1,2,...T-1$ is given by*

$$(\frac{\partial J}{\partial \rho_t})_{i,j} = -\frac{1}{4} \sum_{(k,l) \in \mathcal{O}_{i,j}} (y_{t,k,l} - y_{t,i,j})^2 + 2y_{t,i,j},$$

where $\mathcal{O}_{i,j}$ is the connected neighbor of (i,j), i.e., $\{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$ and $y_t \in \mathbb{R}^{n \times n}$ is the solution of equation (6).

Proof. At time t, let $\mathbf{u}$ represent the elementwise inverse of $\mathbf{w}$, that is, each element of $\mathbf{u}$ is the reciprocal of the corresponding element in $\mathbf{w}$. Then, we compute

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{u}} &= \frac{\partial}{\partial \mathbf{u}} b_t^\top A^{-1} b_t \\ &= b_t^\top (\frac{\partial}{\partial \mathbf{u}} A_t^{-1}) b_t \\ &= -(y_t^\top \nabla \cdot) \frac{\partial}{\partial \mathbf{u}} \operatorname{Diag}(\mathbf{u})(\nabla \cdot^\top y_t) \\ &= -(\nabla y_t) \odot (\nabla y_t), \end{aligned}$$

where $\odot$ is the Hadamard (entrywise) product. Note that the transpose of a discrete divergence operator is the gradient operator.

Secondly, $\frac{\partial J_t}{\partial b_t} = (A_t^{-1} + (A_t^{-1})^T)b_t = 2A_t^{-1}b_t = 2y_t$. Using the chain rule, the result is straightforward. □

3.3. Proposed Algorithm

The goal of this work is to improve the autoencoder image interpolation result by $\mathbb{D}(z_{i\to j}(t))$, which decodes a linear combination of latent codes of image x_i and x_j. To that end, we propose a new approach, which adds the path energy defined in (3) as a regularization term when reconstructing the input using an autoencoder. This encourages $\mathbb{D}(z_{i\to j}(t))$ to reach the geodesic path between x_i and x_j, and also brings about a smooth transportation effect, which will be shown in the experiments. Here x_i and x_j can be any pair of data in the training dataset.

However, there are some issues with the path $\mathbb{D}(z_{i\to j}(t))$ generated by the autoencoder. First, the generated path does not necessarily meet the mass-preserving properties, which are required when adapting the image interpolation problem to a mass transfer problem in the normal dynamic OT setting. The initial dimensions of the pictures may be too big to be plugged into the path energy term. To address this, downsampling the path $\mathbb{D}(z_{i\to j}(t))$ would be beneficial. Furthermore, to ensure that the KKT system (5) has a solution, the weight $\mathbf{w}_t$ must be positive. Thus, some preprocessing on $\mathbb{D}(z_{i\to j}(t))$ is necessary.

The full procedure of our algorithm is presented in Algorithm 1. The Sigmoid activation at the last layer of the autoencoder guarantees a non-negative output. Therefore, we measure the reconstruction loss by computing the binary cross entropy $\mathrm{BCE}(x_i, \hat{x}_i)$ between the input data x_i and the autoencoder output $\hat{x}_i$. J is the path energy defined in (3), $D_{i\to j}$ refers to the path $\mathbb{D}(z_{i\to j}(t)), t \in [0,1]$ after preprocessing, and

$$\begin{aligned}\mathrm{MassLoss}_{i\to j} = \sum_{t=1}^{T-1} |\,&\mathrm{Mass}(\mathbb{D}(z_{i\to j}(t))) \\ &- (t\,\mathrm{Mass}(x_i) + (1-t)\,\mathrm{Mass}(x_j))|,\end{aligned}$$

with $\mathrm{Mass}(x) = \sum_{i,j} |x_{i,j}|$. The term $\mathrm{MassLoss}_{i\to j}$ is used to constrain the mass of the generated path. For computational efficiency, the choice of the indices i, j can be a random subset of the entire data set. For motion-dominated interpolation problems, we use the path energy based on normal optimal transport, and for shape-dominated interpolation problems, we use the path energy based on unbalanced optimal transport (see Section 7.3 in supplementary materials for more explanation).

After training, we generate interpolation images between x_i and x_j by decoding a convex combination of their latent codes.

Algorithm 1 Our proposed method for autoencoder image interpolation

Input: training image dataset X, an autoencoder with Sigmoid activation at the last layer

Step 1: training process: for each epoch:

1. generate $\mathbb{D}(z_{i\to j}(t)), t = 1, 2, ..., T-1$ for the chosen i, j pairs;
2. (optional) downsample the $\mathbb{D}(z_{i\to j}(t))$
3. threshold the $\mathbb{D}(z_{i\to j}(t))$ with positive value ϵ;
4. (optional) normalize $\mathbb{D}(z_{i\to j}(t))$ with a standard mass;
5. compute the loss function as follows and backpropagate:

$$\sum_i \mathrm{BCE}(x_i, \hat{x}_i) + \alpha \sum_{i,j} J(D_{i\to j}) + \beta \sum_{i,j} \mathrm{MassLoss}_{i\to j} \quad (7)$$

Step 2: interpolation process: generate interpolated images between x_i and x_j by decoding a convex combination of their latent codes

$$\mathbb{D}((1-t)z_i + tz_j), \quad 0 < t < 1.$$

Note: For more detailed explanations of the steps and the used functions, please refer to Section 3.3.

4. Numerical Experiments

In this section, we evaluate the effectiveness of our approach in a variety of scenarios, from the most extreme (two training samples) to the moderate (a few training samples) to the large (MNIST). It is noteworthy that traditional deep learning techniques usually require a considerable amount of training data to accurately learn features. Additionally, conventional optimization techniques are usually restricted to interpolating between individual pairs of images. In contrast, our method, based on the principles of physics-driven deep learning, produces robust and smooth interpolation results in all cases.

The experimental setup is as follows, unless otherwise specified. For all experiments, we adopt an autoencoder architecture similar to that of ACAI [4]. The encoder consists of 3 blocks of two 3×3 convolutional layers followed by a 2×2 average pooling. The number of channels is doubled before each average pooling layer. This is followed by two more 3×3 convolutional layers, and the final output is used as the latent code whose dimension is $16 \times 4 \times 4$. All of the convolution layers, except the final one, use a leaky ReLU activation. The decoder has 3 similar blocks with average pooling replaced by upsampling layer and channel halved. Two more convolutional layers are then performed, the last having a sigmoid activation. We implemented our codes in Pytorch and use the package scipy.sparse.spsolve to solve

linear systems in the path energy term. The parameters are optimized with Adam with the default parameters.

The threshold value ϵ is set to $1e-5$. We adjust the values of the path energy weight α, mass weight β, and source weight τ to obtain the best performance. All experiments use a downsampling size of 32×32, and the time discretization number T is chosen between 10 to 30. Section 7.2 in supplementary materials discussed how to choose the right T.

We quantify the quality of the interpolated images by using the Structural Similarity Index (SSIM) score in some experiments. This score evaluates the similarity between two images in terms of structure, brightness, and contrast, providing a more perceptually relevant assessment of the image quality. It is worth noting that if the SSIM score is equal to one, it means that the two images are identical. Since we have a sequence of images, we report the mean of the SSIM score of any two adjacent images.

4.1. Ablation Study

This ablation study aims to show the improvements brought by our regularization term, particularly when the training dataset is limited to only two data points. We compare our algorithm to a standard baseline autoencoder across different image types: gray-scale, binary, and RGB. We use normal optimal transport based path energy for the first two examples and unbalanced optimal transport based path energy for the last one. The results of these comparisons are presented in Figure 2.

Our method produces significantly better results than the baseline autoencoder, which typically yields local alterations, resulting in a cross-dissolving effect in the images. In the grayscale image, our technique effectively shifts the pixel density from one corner to the other. In the binary image scenario, our approach redistributes the mass from a central large circle into two distinct circles at opposite corners. For RGB images, our method achieves a smooth gradation in color transitions, as seen in the cartoon face. We also provided the SSIM score of the interpolated images as shown in the figure caption. Even though our method does not always achieve a higher SSIM score, the visual improvement still demonstrates the effectiveness of our regularization term in generating realistic interpolated images that follow the laws of physics.

4.2. Interpolation with Obstacle in the Environment

In this example, we introduce obstacles into the environment. Figure 3 displays a labyrinth map with a circle moving from the bottom left corner to the right upper corner. The walls of the labyrinth are colored pink and the mass cannot pass across the wall. Subfigure (a) shows the result of our interpolation, which is very smooth, and the circle is able to squeeze through the narrow paths. To compare, we also conducted an experiment using the proximal splitting method to solve the original dynamic OT [17]. The result of this experiment is shown in subfigure (b). The mass transfer follows the geodesic path; however, the mass splits and goes to different paths, resulting in a visually messy interpolation. Additionally, the pixels carrying mass in the interpolated result have values 15 times higher than the original value, leading to a smaller object region. Therefore, the result of our method is much smoother than the one from the optimization method.

4.3. Barycenter Problem

In this section, we evaluate our approach in situations where data is limited, particularly focusing on the barycenter problem, which has recently become a major topic in optimal transport theory. A barycenter, or Wasserstein barycenter, in this context is a distribution that minimizes the weighted sum of Wasserstein distances to a set of given distributions. In image processing, this concept allows us to treat each image as a distribution. Traditional optimization techniques calculate the transportation plan to obtain the barycenter, a process that is computationally intensive. However, using an autoencoder, the barycenter can be efficiently obtained by decoding a convex combination of the latent codes after training. For example, with just four training images represented by latent codes z_1, z_2, z_3, z_4, the barycenter at the i-th row and j-th column of a 6×6 grid can be generated by decoding:

$$\mathbb{D}\left((1-\frac{i}{6})(1-\frac{j}{6})z_1 + \frac{i}{6}(1-\frac{j}{6})z_2 + (1-\frac{i}{6})\frac{j}{6}z_3 + \frac{i}{6}\frac{j}{6}z_4\right).$$

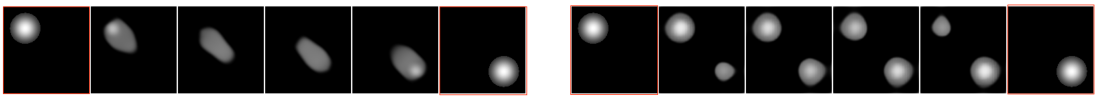
The core is to ensure the reliability of the trained autoencoder in producing meaningful results.

We tested our method on two different barycenter problems, each using four training data points that represented different distributions. The first problem was related to shape analysis and involved four distinct shape images. Thus we used a path energy term based on unbalanced optimal transport. The results, as seen in Figure 4, showed sharply defined yet smooth barycenters. The second problem focused on position changes, using images of a circle at various locations. We used a path energy term based on normal optimal transport. As shown in Figure 5, our method was able to effectively maintain the circle's shape while generating it at different positions. Further details on the results obtained with other methods can be found in Appendix 7.4. To conclude, our approach to the barycenter problem successfully combines sharpness with smooth transitions, thus demonstrating its effectiveness.

4.4. MNIST Dataset

The purpose of this experiment is to show the interpolation performance of our method for a large real-world dataset benchmark. The images were padded to size 32×32 for

(a) Interpolation results on binary images. Left: Our method (SSIM score: 0.87). Right: Baseline autoencoder (SSIM score: 0.91).



(b) Interpolation results on gray-scale images. Left: Our method (SSIM score: 0.89). Right: Baseline autoencoder (SSIM score: 0.93).

(c) Interpolation results on RGB images. Left: Our method (SSIM score: 0.91). Right: Baseline autoencoder (SSIM score: 0.88).

Figure 2. Comparison of our proposed method and the baseline autoencoder method across different image types.

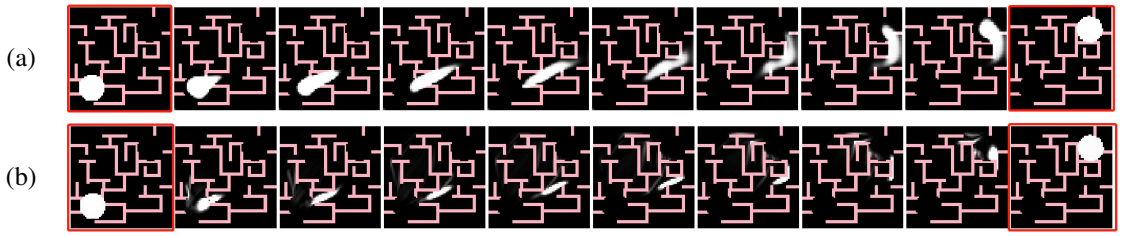


Figure 3. An example of image interpolation when obstacles in the environment (marked pink) are present. (a) The result of our proposed method. (b) The result of the proximal splitting method with a $64 \times 64 \times 120$ space-time discretization grid [17].

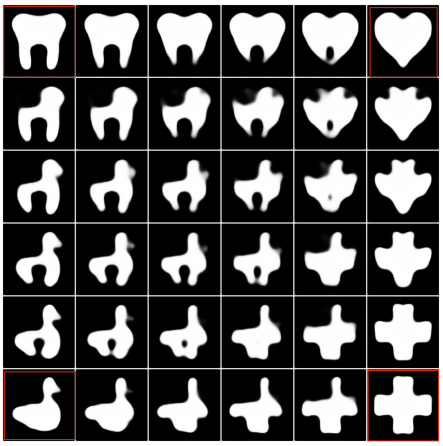


Figure 4. Barycenter example 1. There are only four images (marked red above) in the training dataset. Each remaining image was created by decoding a convex combination of their corresponding latent codes.

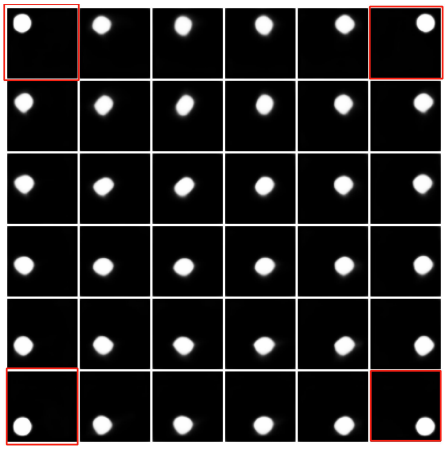


Figure 5. Barycenter example 2. There are only four images (marked red above) in the training dataset. Each remaining image was created by decoding a convex combination of their corresponding latent codes.

training. We employ a batch size of 256 and randomly select 9 pairs of paths to penalize their energy term in the loss function instead of all pairs of paths to reduce computational cost. We used a path energy term based on unbalanced optimal transport. The results are shown in Figure 7. We can see that the morphing between these handwritten digits follows an optimal transportation path.

We compare our method to other state-of-the-art autoencoders, such as the Variational Autoencoder (VAE) [13], Adversarily Constrained Autoencoder Interpolation (ACAI) [4], and a baseline autoencoder, as seen in Figure 6. The baseline model is a basic autoencoder with a BCE loss func-

(a) Ours (SSIM score mean: 0.96, std: 0.01).

(b) ACAI (SSIM score mean: 0.87, std: 0.10). Note that the transition is abrupt in the middle.

(c) VAE (SSIM score mean: 0.9, std 0.10).

(d) Baseline Autoencoder(SSIM score mean: 0.9, std: 0.11). Note that the interpolation results are blurry.

Figure 6. Comparison of interpolation results on the MNIST dataset using four different autoencoder methods: Ours, ACAI, VAE, and a baseline autoencoder.

tion. Our method and VAE show the most satisfactory visual interpolation results. Notably, our method has a higher SSIM score and a much lower standard deviation. The interpolation using the ACAI method has abrupt transitions, likely due to its generated images being trained to follow the training data. In contrast, the baseline autoencoder usually produces blurry interpolations. Therefore, our method has a significant improvement in the interpolation task among autoencoder-based methods.

We also investigate how the latent space of an autoencoder is impacted by the regularization term. We analyze three distinct methods: the Kullback-Leibler (KL) divergence in VAE, the adversarial term in ACAI, and the path energy term in our proposed method. We draw inspiration from [4] and conduct the same classification and clustering experiments on the MNIST dataset using the latent spaces generated by these different autoencoders after training. The results of the experiments are presented in Tables 1 and 2, which demonstrate that ACAI outperforms the other methods in both classification and clustering tasks. Our method exhibits moderate performance, surpassing the Baseline. We believe that this result is reasonable. Unlike the KL divergence in VAEs, which enforces prior assumptions on the latent space, our method's path energy term works directly on the pixel space of reconstructed images, rather than on the latent codes. Consequently, while our method guarantees smooth interpolation and follows the principle of least path energy, it does not necessarily disentangle essential representations within the dataset, such as class identity.

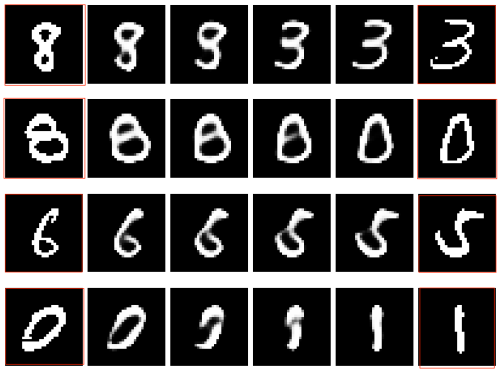


Figure 7. The interpolation results on MNIST dataset using our proposed method.

Baseline	VAE	ACAI	ours
94%	99%	99%	97%

Table 1. Single-layer classifier accuracy achieved by different autoencoders on MNIST dataset.

Baseline	VAE	ACAI	ours
54%	83%	96%	80%

Table 2. Clustering accuracy for using K-Means on the latent space of different autoencoders on MNIST dataset

5. Conclusion

Recently, there has been a great deal of interest in generative AI models such as ChatGPT and Stable Diffusion. However, these models require a large amount of training data. This paper focuses on generative AI for image generation, where training data is scarce. To address this issue, we proposed a path energy term to regularize the autoencoder, resulting in smooth and realistic interpolation results. Our experiments show that this idea has the potential to combine a robust mathematical model with neural network training. The main challenge in our training is solving linear systems in the path energy term. This could be alleviated by using other advanced techniques such as normalizing flow.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223. PMLR, 2017. 1

[2] Christopher Beckham, Sina Honari, Vikas Verma, Alex M Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, and Chris Pal. On adversarial mixup resynthesis. *Advances in neural information processing systems*, 32, 2019. 3

[3] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. 1, 2, 3

[4] David Berthelot*, Colin Raffel*, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019. 1, 3, 5, 7, 8

[5] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using Lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–12, 2011. 2

[6] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1

[7] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021. 2

[8] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018. 4

[9] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 274–289. Springer, 2014. 1

[10] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1

[13] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second international conference on learning representations, ICLR*, page 121, 2014. 7

[14] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016. 3

[15] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016. 1, 3

[16] Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder image interpolation by shaping the latent space. *arXiv preprint arXiv:2008.01487*, 2020. 1

[17] Nicolas Papadakis, Gabriel Peyré, and Edouard Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, 2014. 2, 3, 4, 6, 7

[18] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 2

[19] 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. 1

[20] Jörg Sander, Bob D de Vos, and Ivana Išgum. Autoencoding low-resolution MRI for semantically smooth interpolation of anisotropic MRI. *Medical Image Analysis*, 78:102393, 2022. 1

[21] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European conference on computer vision (ECCV)*, pages 650–665, 2018. 1

[22] Cédric Villani et al. *Optimal transport: old and new*. Springer, 2009. 2

[23] Bohan Zhou and Matthew Parno. Efficient and exact multimarginal optimal transport with pairwise costs. *arXiv preprint arXiv:2208.03025*, 2022. 1

Improving Autoencoder Image Interpolation via Dynamic Optimal Transport

Supplementary Material

6. the Path Energy Term

In this section, we present the specifics of the path energy term when dealing with obstacles and unbalanced optimal transport.

For the path energy term based on original optima transport, we have a closed form as blow

$$J(\rho) = \sum_{t=0}^{T-1} b_t^T \left(\nabla \cdot \mathrm{Diag}(\mathbf{w_t})^{-1} \nabla \cdot^{\top} \right)^{-1} b_t.$$

Note that there is linear system to solve at each time scale.

When there are obstacles in the environment, we delete the momentum at that positions and solve reduced dimensional linear systems.

For the unbalanced optimal transport case, after solving the KKT system, we have

$$J(\rho) = \sum_{t=0}^{T-1} b_t^T \left([\nabla \cdot \ \mathbf{I}] \, \mathrm{Diag}(\tilde{\mathbf{w}}_\mathbf{t})^{-1} [\nabla \cdot \ \mathbf{I}]^{\top} \right)^{-1} b_t,$$

where $\mathbf{I}$ is the identity matrix, $\tilde{\mathbf{w}}_t$ is a extended weight including the weight $\mathbf{w}_t$ for momentum and the weight for source term defined as follows:

$$\mathbf{w}_t^c = \frac{\tau}{\rho_{t,i,j}}.$$

7. Additional expriements

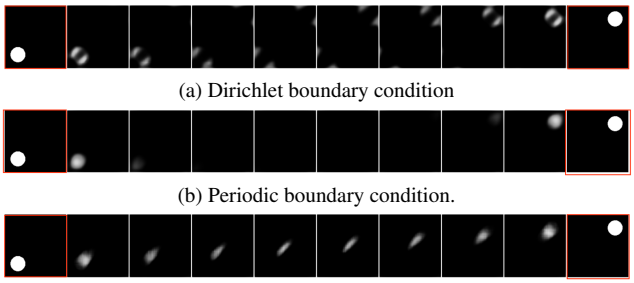
In this section, we present the results of experiments which are not the primary focus of our paper.

7.1. Boundary Condition

In the dynamic optimal transport model, the continuity equation, a PDE constraint, is of great importance. The boundary condition of the divergence operator is a key factor in the solution. Figure 8 shows a comparison of the results obtained using different boundary conditions. It is evident that the Dirichlet boundary condition yields the best result, while the mass vanishes to the boundaries under other conditions. Therefore, in this paper, we have chosen to use the Dirichlet boundary condition as default.

7.2. Choice of T

The discretion of time is of great importance to the interpolation result. The selection of T should be chosen depending on the maximum distance the pixels traverse. For example, if the pixels move a maximum of 10 pixels distance, then T should be set to 10.



(a) Dirichlet boundary condition

(b) Periodic boundary condition.

(c) Neumann boundary condition

Figure 8. The comparison result when trained with different boundary conditions

7.3. the Comparison between OT and Unbalanced OT

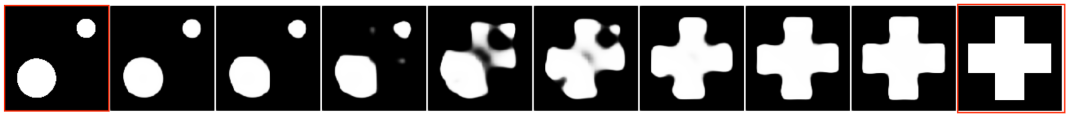
As we discussed earlier, unbalanced OT is proposed to address the unbalanced mass between the source image and the target image. It also makes it more suitable for shape-change dominated interpolation problems. We show a comparison in Figure 9. Our method based on the path energy term of UOT yields the best result, while OT and baseline autoencoders produce unsmooth or unrealistic transportation.

7.4. Barycenter Problem

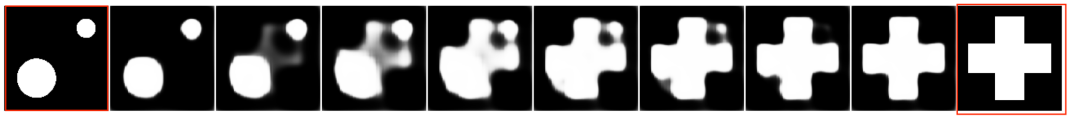
In this section, we show the Barycenter results using other state of art methods, including MMOT[23] and convolutional Wasserstein barycenters in POT package[10]. Our result is shown in Figure 4 in section 4. We can see that the result from POT is blurred, and the result of MMOT is very sharp but complicated. Among them, our barycenter result is very fluid and smooth.

7.5. Auxiliary Training Data

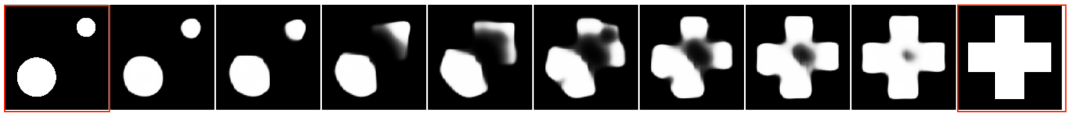
One challenge in solving dynamic OT is the computation cost, and this section explores whether the neural network learning nature of our method can bring benefits to finding the geodesic path. To investigate this question, we design a simple but instructive comparison experiment. Specifically, we use the image of a circle in the upper left corner as the source image x_1 and use the image of a circle in the lower right corner as the target image x_2, shown in Figure 11. The aim is to generate the geodesic path between x_1 and x_2. We also create images with a center circle at every point in time, with a total of 1936 images. The first experiment uses x_1 and x_2 as the only training data, while the second experiment uses all training data with $1, 2$ as the only choice of



(a) Baseline autoencoder.

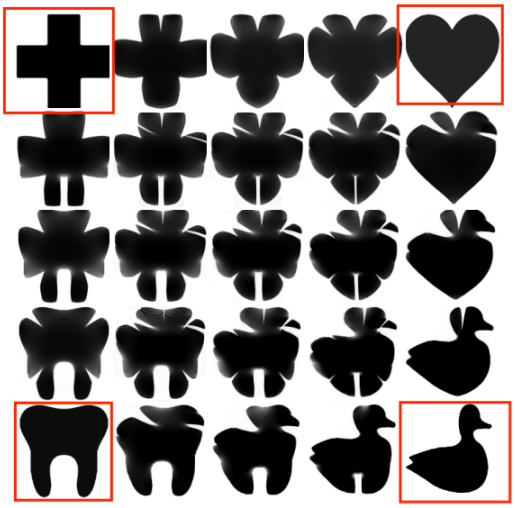


(b) Our method with path energy term based on OT.



(c) Our method with path energy term based on UOT.

Figure 9. Comparative analysis of image sequence interpolation. Given the images marked in red, the interpolated images are generated by decoding a linear combination of their latent codes after training.

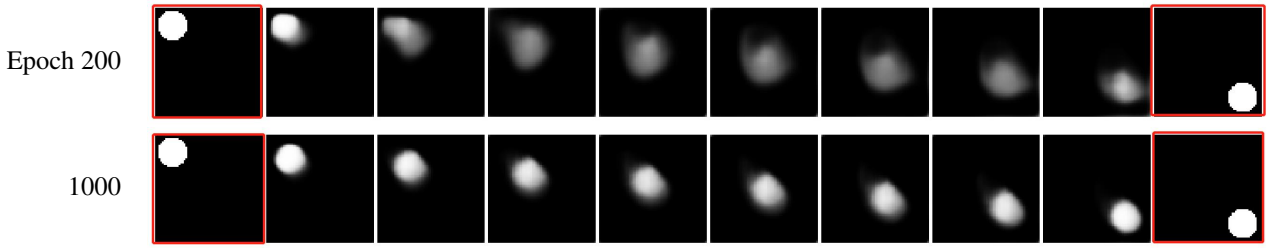


(a) MMOT

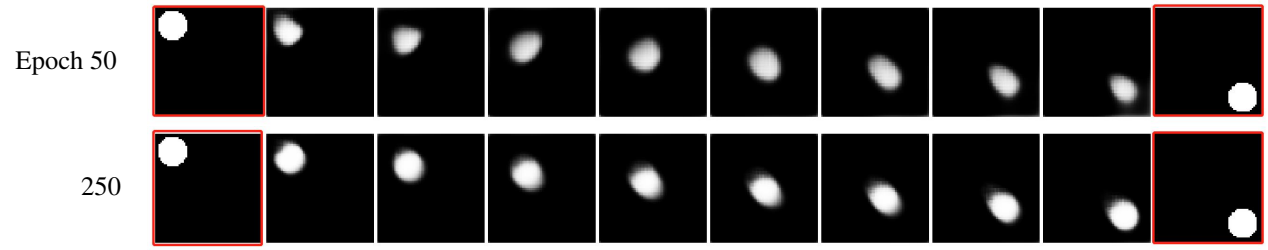
(b) POT

Figure 10. Barycenter results using other methods.

i, j pair in the regularization term. Thus, the second experiment has 1934 auxiliary data. The batch size is 500. In both examples, the blocks in both encoder and decoder are repeated 4 times, resulting in a smaller latent dimension $16 \times 2 \times 2$. The results (Figure 11) indicate that the auxiliary data significantly improve the interpolation quality. Without auxiliary data, the interpolation is blurrier at epoch 200 compared to the result at epoch 50 with auxiliary data. Additionally, the final interpolation result is less accurate without auxiliary data. Therefore, the inclusion of auxiliary data improves the autoencoder's efficiency and accuracy in generating the geodesic path.



(a)



(b)

Figure 11. Comparison of interpolation results at different training epochs. (a) Results with only two training data points at epochs 200 and 1000. (b) Results with auxiliary data at epochs 50 and 250. Detailed descriptions of the training can be found in Section 7.5.