

OT-Net: A Reusable Neural Optimal Transport Solver

Zezeng Li^{1,4†}, Shenghao Li^{1†}, Lianbao Jin², Na Lei^{3*},
Zhongxuan Luo¹

¹School of Software, Dalian University of Technology, Dalian, 116620, China.

²School of Mathematical Sciences, Dalian University of Technology, Dalian, 116024, China.

³ DUT-RU ISE, Dalian University of Technology, Dalian, 116620, China.

⁴Beijing Key Laboratory of Light-field Imaging and Digital Geometry,
Capital Normal University, Beijing, 100048, China.

*Corresponding author(s). E-mail(s): nalei@dlut.edu.cn;

Contributing authors: zezeng.lee@gmail.com; lshymfl@126.com;

1099630577@qq.com; zxluo@dlut.edu.cn;

†These authors contributed equally to this work.

Abstract

With the widespread application of optimal transport (OT), its calculation becomes essential, and various algorithms have emerged. However, the existing methods either have low efficiency or cannot represent discontinuous maps. A novel reusable neural OT solver **OT-Net** is thus presented, which first learns Brenier's height representation via the neural network to obtain its potential, and then gained the OT map by computing the gradient of the potential. The algorithm has two merits, 1) it can easily represent discontinuous maps, which allows it to match any target distribution with discontinuous supports and achieve sharp boundaries. This can well eliminate mode collapse in the generated models. 2) The OT map can be calculated straightly by the proposed algorithm when new target samples are added, which greatly improves the efficiency and reusability of the map. Moreover, the theoretical error bound of the algorithm is analyzed, and we have demonstrated the empirical success of our approach in image generation, color transfer, and domain adaptation.

Keywords: Neural-network, OT map, reusability, Brenier's height representation

1 Introduction

The OT map transmits one probability measure to another in the most economical way, which is widely applied in various areas, such as computer vision [1–3], computer graphics[4, 5], and deep learning[6–11]. Consequently, an effective algorithm to solve the OT problem is essential. It can be regarded as a standard linear programming (LP) task, which can be computed by LP tools. However, in practical issues, the dimension of two distributions is usually thousands upon thousands. In this situation, the computational complexity is unaffordable if the problem is solved by LP algorithms. Later, Cuturi et al. [12] propose the Sinkhorn-Knopp algorithm by adding an entropic regularizer into the original OT problem, which can solve it quickly by sacrificing accuracy. Therefore, various improved variants have emerged, such as the iterative Bregman projections algorithm [13], the adaptive primal-dual accelerated algorithm [14], the inexact proximal point method [15], and an accurate algorithm based on Nesterov’s smoothing technique [16].

Although these algorithms can effectively address the OT problem, learning or even approximating such an OT plan is computationally challenging for large and high-dimensional datasets due to the intrinsic curse of dimensionality. Thus, some academics have proposed using neural networks to compute OT map [2, 17–23] to mitigate these problems. For example, Seguy et al. [2] propose a two-step approach. First, they employ a simple dual stochastic gradient algorithm for solving regularized OT, then they estimate a Monge map as a neural network learned by approximating the barycentric projection of the OT plan in the first step. This method requires two optimization processes, which leads to a significant reduction in the efficiency of the algorithm. Makkula et al. [17] present a new framework to estimate the OT map based on input convex neural networks. Although this algorithm improves the model’s performance and certifies the existence of the OT map, it needs to calculate two potential functions simultaneously, which directly doubles the number of parameters and floating-point operations. Furthermore, the above OT maps lack a theoretical guarantee of being globally optimal and unique.

To address the above issue, Gu and Lei et al. [7, 8, 24] propose a novel algorithm of solving OT based on convex geometry and variational principles. When the cost function is quadratic, Brenier theorem [25] indicates that the OT map is given via the gradient of a piece-wise convex function which is called Brenier’s potential. As [7, 8, 24] remark, Brenier’s potential is obtained by optimizing a convex energy, its solution is globally optimal and unique. This means that the OT map has the same characteristics. Whereas, when the number of target domain features changes, it has to re-optimize the convex energy equation to compute OT maps, resulting in low reusability. Consequently, this paper presents a novel neural network-based method to work out the problem. Note that our algorithm is different from other neural network-based methods [2, 18, 22]. While our algorithm learns Brenier’s potential via a single neural network, and the OT map is obtained by computing its gradient which can easily represent discontinuous mappings, this can effectively eliminate mode collapse in the generated models. In addition, the solution exists and is unique with an upper bound on the error.

As afore-mentioned, our algorithm is described as follows: first, computing the height vector through neural-network to obtain Brenier’s potential, and then the OT map is obtained by computing its gradient, which endows our algorithm with the ability to represent discontinuous maps. Second, when new target samples are added, the OT map can be calculated straightforwardly via the learned height representation without retraining or optimization, which greatly reduces the time cost of repeated calculations and improves its reusability.

In summary, our main **contributions** are as follows:

- We propose a novel algorithm to compute the OT map, so-called the reusable neural optimal transport solver, which is a general algorithm framework suitable for various probability transformation scenarios, including image generation, domain adaptation, and color transfer.
- Our algorithm can compute the OT map straightly when adding new target samples without retraining or optimization, which significantly improves its computational efficiency and reusability.
- The algorithm enables represent discontinuous maps, which could nicely avoid mode collapse in the generation.
- We theoretically analyze the error bound of the height vector, and the experimental results show that the algorithm has comparable performance in generating models, color transfer, and domain adaptation.

2 Related Works

With the application of OT in various engineering fields, a large number of algorithms and applications related to OT have arisen. The detailed theory, algorithms, and applications can be found in Ref. [26–28]. In the following, we will briefly review related algorithms for solving OT maps and their application, as well as the semi-discrete optimal transport (SDOT) that is most relevant to this paper.

Algorithms and Applications. With the continuous improvement of OT theory, various OT algorithms have been presented, and its application in the field of machine learning becomes more and more widespread, for example, generative model [8, 9, 18, 22, 29], domain adaptation [30–33], color transfer [34, 35], image super-resolution [21, 36, 37] and so on. One of the most representative entropy-regularization-based algorithms is the Sinkhorn-Knopp algorithm [12]. Although it reduces the computational complexity of the OT problem, the algorithm does not scale well to measures supported on a large number of samples, since each of its iterations has an $\mathcal{O}(n^2)$ complexity. Various improved algorithms thus have emerged to eliminate the above problems. However, due to an intrinsic curse of dimensionality, learning or even approximating such a map is computationally challenging for large and high-dimensional datasets.

Therefore, neural-network-based OT solvers have emerged. These existing methods in two ways. The first is computing the **OT cost** and using it as the loss function [9, 10, 29, 38, 39]. The second is the **OT map** itself can be used as a generative model [17, 18, 22]. Specifically, Daniels et.al [18] proposes a method for solving entropy-regularized OT using neural networks, but it is extremely time-consuming to generate

samples via the Langevin dynamics. Makkuvu et.al [17] employ input convex neural networks to parametrize potentials in the dual problem and recover the OT map by using their gradients. Rout et al. [22] apply OT directly in ambient spaces, such as spaces of high-dimensional images.

According to the OT regularity theory [40], the OT map is discontinuous when the support of the target domain is non-convex. Nevertheless, neural networks can only express continuous mapping. So, transport maps learned by the above algorithms [17, 18, 22] are continuous, while OT maps are discontinuous at singular sets [7], and this intrinsic conflict leads to mode collapse. Therefore, Lei et al. [8] propose a generative model called AE-OT which perfectly avoids mode collapse/mixture. In the following, we provide a brief description of the SDOT algorithm that was proposed in Ref. [8].

Semi-discrete optimal transport. Suppose the source measure μ defined on a convex domain $\Omega \subset \mathbb{R}^d$, the target domain is a discrete set, $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}, \mathbf{y}_i \in \mathbb{R}^d$. The target measure is a Dirac measure $\nu = \sum_{i=1}^n \nu_i \delta(\mathbf{y} - \mathbf{y}_i), i = 1, 2, \dots, n$, with the equal total mass as the source measure, $\mu(\Omega) = \sum_{i=1}^n \nu_i$. A cell decomposition is induced $\Omega = \bigcup_{i=1}^n W_i$ under semi-discrete optimal transport map $T : \Omega \rightarrow Y$, such that every \mathbf{x} in each cell W_i is mapped to the target \mathbf{y}_i , $T : \mathbf{x} \in W_i \mapsto \mathbf{y}_i$. If the μ -volume of each cell W_i equals to the ν -measure of target domain, i.e., $T(W_i) = \mathbf{y}_i$, $\mu(W_i) = \nu_i$, The map T is measure preserving, denoted as $T_\# \mu = \nu$. The cost function is given by $c : \Omega \times Y \rightarrow \mathbb{R}$, where $c(\mathbf{x}, \mathbf{y})$ represents the cost for transporting a unit mass from \mathbf{x} to \mathbf{y} . The total cost of T is given by

$$\int_{\Omega} c(\mathbf{x}, T(\mathbf{x})) d\mu(\mathbf{x}) = \sum_{i=1}^n \int_{W_i} c(\mathbf{x}, \mathbf{y}_i) d\mu(\mathbf{x}).$$

The OT map minimizes the total transport cost as follows,

$$T^* := \arg \min_{T_\# \mu = \nu} \int_{\Omega} c(\mathbf{x}, T(\mathbf{x})) d\mu(\mathbf{x}). \quad (1)$$

As the **Theorem 3.5** of Ref. [41] remarks, the OT map is given by the gradient map of Bernier's potential \mathbf{u} , which is the upper envelope of a collection of hyperplanes. \mathbf{u} can be parametrized uniquely up to an additive constant by a parameter \mathbf{h} , called the height vector [8, 24]. \mathbf{u} is often referred to as \mathbf{u}_h using this parametrization. In such a case, the form of \mathbf{u}_h can be stated as below,

$$\mathbf{u}_h = \max_{i=1}^n \{\pi_{h,i}(\mathbf{x})\} = \max_{i=1}^n \{\mathbf{x}^T \mathbf{y}_i + h_i\}, \mathbf{u}_h : \Omega \rightarrow \mathbb{R}, \quad (2)$$

where $\pi_{h,i}(\mathbf{x}) = \mathbf{x}^T \mathbf{y}_i + h_i$ is the hyperplane corresponding to \mathbf{y}_i , $\mathbf{h} = (h_1, h_2, \dots, h_n)$ is the height vector, h_i denotes the height of the i -th sample.

Based on the above elaboration, Bernier's potential \mathbf{u}_h is deterministic by \mathbf{h} . As a result, given the target measure ν , there exists potential in Eq.(2) whose projected volume of each support plane is equal to the given target measure ν_i . The projection of the graph of \mathbf{u}_h decomposes Ω into cells $W_i(h_i)$, each cell is the projection of the

supporting plane $\pi_{\mathbf{h},i}(\mathbf{x})$. (A detailed illustration can be found in Fig. 3 of Ref. [8]). That is, the key to obtaining the potential energy function is to optimize the height vector \mathbf{h} . Therefore, Gu et al. constructed a convex energy function with respect to the height vector \mathbf{h} by the variational principle (**Theorem 1.2** of Ref. [24]).

That is, under the condition $\sum_i h_i = 0$, the height vector \mathbf{h} is the unique minimum argument of the following convex energy:

$$E(\mathbf{h}) = \int_0^{\mathbf{h}} \sum_{i=1}^n w_i(\eta) d\eta_i - \sum_{i=1}^n h_i \nu_i , \quad (3)$$

where $w_i(\eta)$ is the μ -volume of $W_i(\eta)$. Further, the gradient of Eq. (3) is given by the following:

$$\nabla E(\mathbf{h}) = [w_1(\mathbf{h}) - v_1, w_2(\mathbf{h}) - v_2, \dots, w_n(\mathbf{h}) - v_n]^T . \quad (4)$$

Hence, the convex energy $E(\mathbf{h})$ can be optimized simply by the gradient descent method. Yet this algorithm needs to recalculate or reoptimize the OT map in the case of changes in target domain samples, resulting in low reusability. Consequently, this paper presents a novel neural-network-based algorithm to address OT map, *i.e.*, the reusable neural OT solver. Compared with the AE-OT[8], we provide a more efficient method to calculate the height vector and Brenier's potential.

3 The Proposed Algorithm

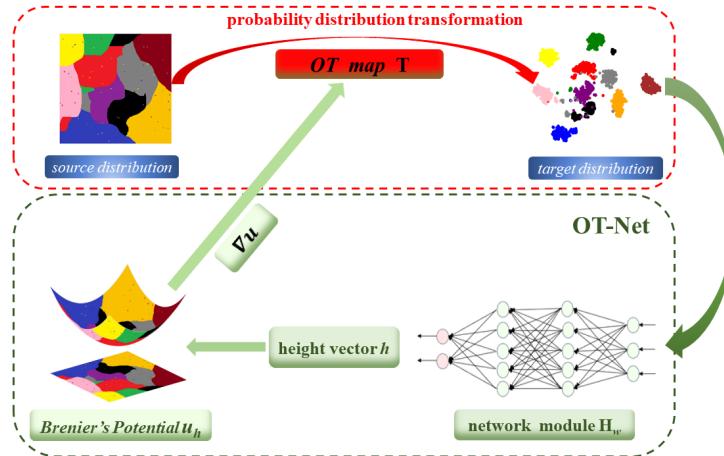


Fig. 1: Architecture of our algorithm. **OT-Net** (green dotted box): First, the height vector \mathbf{h} is obtained by a well-trained Bernier's height representation H_w . Then, Bernier's potential u_h is calculated by Eq. (6). Finally, the OT map is induced by its gradient. The **probability distributions transformation** (red dotted box): learning transformation of source distribution to target distribution via the OT map T .

The core task of deep learning is to learn the manifold structure of data and transform probability distributions. OT has a natural advantage in learning transformations between probability distributions. Hence, this section mainly discusses the implementation of probability distribution transformations by computing the OT map. Inspired by the AE-OT algorithm, a new solving algorithm is proposed, which is a reusable neural OT solver, its pipeline is shown in Fig. 1.

From Fig. 1, the key to our algorithm is learning Brenier’s height representation using the network H_w . Specifically, we can randomly select some samples in the target domain and learn the height representation through H_w . Then, we can use the learned Bernier’s height representation to directly calculate the height vectors of the remaining samples. Even if new samples are added, the height vectors can be calculated directly using H_w , without further training and optimization. Finally, Brenier’s potential is obtained by substituting Eq. (2) whose gradient is the OT map. This greatly improves the computational efficiency and reusability of the OT map. Detailed numerical results are presented in section 4.1.3.

3.1 Learning Bernier’s Height Representation

The core of calculating the OT map is to optimize the convex energy Eq. (3) to acquire the Brenier’s height vector \mathbf{h} . AE-OT calculates the numerical solution of \mathbf{h} under ad-hoc data and does not accept any change of data. That is to say, the height vector is not reusable, and the entire height vector $\mathbf{h} = (h_1, h_2, \dots, h_n)$ needs to be recalculated when new samples are added. To improve the reusability of the OT map and reduce the huge time cost of duplicate computing, we reformulate the support hyperplane $\pi_{h,i}(\mathbf{x})$ as follows,

$$\pi_{H,i}(\mathbf{x}) = \mathbf{x}^T \mathbf{y}_i + H(\mathbf{y}_i), \quad (5)$$

where $H(\mathbf{y})$ is Brenier’s height representation which is a function defined on the target domain \mathbf{Y} . Then, as a surrogate, Eq. (5) is substituted into Eq. (2) to obtain Brenner’s potential \mathbf{u}_h :

$$\mathbf{u}_h = \max_{i=1}^n \{\mathbf{x}^T \mathbf{y}_i + H(\mathbf{y}_i)\}. \quad (6)$$

\mathbf{u}_h is the upper envelope of the hyperplane $\pi_{H,i}(\mathbf{x})$, and its gradient with respect to \mathbf{x} is :

$$\begin{aligned} \nabla_{\mathbf{x}} \mathbf{u}_h &= \nabla_{\mathbf{x}} [\max_{i=1}^n \{\mathbf{x}^T \mathbf{y}_i + H(\mathbf{y}_i)\}] = \nabla_{\mathbf{x}} [\mathbf{x}^T \mathbf{y}_i + H(\mathbf{y}_i)] \\ &= \nabla_{\mathbf{x}} [\mathbf{x}^T \mathbf{y}_i] + \nabla_{\mathbf{x}} [H(\mathbf{y}_i)] = \mathbf{y}_i . \end{aligned} \quad (7)$$

According to Eq. (7), the gradient of Brenier’s potential gives the OT map $T(\mathbf{x}) = \nabla_{\mathbf{x}} \mathbf{u}_h = \mathbf{y}$.

Considering the universal approximation ability of the neural network, we use a neural network to fit height representation $H(\mathbf{y})$. Thanks to the continuous expression ability of neural networks, when new samples arrive in the target domain, we don’t need to recalculate the height vector $H(\mathbf{y})$ of all samples but only need to calculate the height vector of the out-of-sample separately, which greatly improves the reusability of the algorithm.

Algorithm 1 OT-Net

Require: target sample points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{B*b}$, number of noise samples N , number of batch B , batch size b , termination criterion δ , \mathbb{P}_x is the given source distribution.

Ensure: Bernier's Height Representation H_w .

- 1: **while** $\|\nabla E(\mathbf{h})\|_2 > \delta$ **do**
- 2: $\nu = zeros(B * b)$.
- 3: **for** $k = 0; k < B$ **do**
- 4: Extract the k -th batch $\{\mathbf{y}_i\}_{i=k*b}^{(k+1)*b-1}$ from the target sample points set \mathbf{Y} .
- 5: Sample $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^N \sim \mathbb{P}_x$ a batch from source distribution.
- 6: Update \mathbf{h} by $H_w(\mathbf{Y})$.
- 7: $\pi_{\mathbf{h},i}$ is calculated by Eq.(5).
- 8: Calculate the index corresponding to the maximum π , i.e., $index = argmax(\pi_{\mathbf{h},i})$.
- 9: Count the number of occurrences of each index and add it to ν .
- 10: $k = k + 1$.
- 11: **end**
- 12: $\nu = \nu / (B * N)$.
- 13: Calculate Eq.(4) and then use Adam [42] to optimize H_w .
- 14: **end while**
- 15: return H_w .

To be specific, when the new ad-hoc data are joined, the added samples and trained original samples fed to the **OT-Net** can directly acquire the whole height vector. Then, the support planes of \mathbf{u}_h are obtained by Eq. (5), and the potential is gained by taking the upper envelope of all supporting planes according to Eq. (6). Finally, the OT map is given by the gradient map of potential. Unlike standard neural networks that are constrained to be continuous, the OT map from **OT-Net** can match any target distribution with many discontinuous supports and achieve sharp boundaries. **Algorithm 1** summarizes the whole trained procedure of our algorithm.

3.2 Error Analysis

In this section, we analyze the error bound of the presented algorithm. Specifically, we demonstrate that the proposed algorithm is convergent, and the height vector obtained directly using the trained $\mathbf{H}(\mathbf{Y})$ can approximate the exact solution under a given upper error bound.

To optimize the energy function, the μ -volume $w_i(\mathbf{h})$ of each cell $W_i(\mathbf{h})$ is the key step, which can be estimated using Monte Carlo method [43]. N random samples is drew from distribution \mathbb{P}_x , then the μ -volume of each cell is estimated as follows:

$$\tilde{w}_i(\mathbf{h}) = \text{crad}(\{j \in \mathcal{J} | \mathbf{x}_j \in W_i(\mathbf{h})\}) / N, \quad \mathcal{J} \in \{1, 2, \dots, N\}.$$

When N is large enough, $\tilde{w}_i(\mathbf{h})$ converges to $w_i(\mathbf{h})$. Then, to minimize energy Eq. (3), we learn Bernier's height representation $H(\mathbf{y})$ by parameterizing the neural network

to obtain the optimal \mathbf{h} . Based on the properties of \mathbf{h} mentioned in Ref. [24], we propose the following proposition.

Proposition 1. Let $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$ be the feature set of the target samples, $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^N$ be source data sampled from source distribution \mathbb{P}_x . The height vector $\mathbf{h} = (H(\mathbf{y}_1), H(\mathbf{y}_2), \dots, H(\mathbf{y}_n)) \in \mathbb{R}^n$ is obtained which minimizing the convex energy $E(\mathbf{h})$ of Eq. (3) under the condition $\sum_i h_i = 0$ through the **Algorithm 1**. Then, it is generated sequence $\{\mathbf{h}^{(k)}\}$ which satisfy

$$\mathbf{h}^{(k)} \rightarrow \mathbf{h}^*, \text{ if } \lim_{k \rightarrow \infty} \|\nabla E(\mathbf{h}^{(k)})\|_2 = 0. \quad (8)$$

Where $\mathbf{h}^* = (h_1^*, h_2^*, \dots, h_n^*) \in \mathbb{R}^n$ is the exact solution of the energy function, it is existence and uniqueness.

According to the demonstration of Theorem 1.2 in Ref. [24], it is clear that the above proposition holds.

Based on Proposition 1, $\|\mathbf{h} - \mathbf{h}^*\|_2 \leq \epsilon$ if $\|\nabla E(\mathbf{h})\|_2 < \delta$, which is the termination criterion of the algorithm. Then we will further analyze the error bounds for obtaining out-of-sample height vectors using the trained $H(\mathbf{Y})$ when new samples are joined. First, considering one new sample \mathbf{y}_{n+1} is added, when this sample is close to the manifold of the training data, $|H(\mathbf{y}_{n+1}) - h_{n+1}^*| \leq \tau_1$ is held. Thereby the following inequality is yielded

$$\|\mathbf{h} - \mathbf{h}^*\|_2 \leq \sqrt{(h_1 - h_1^*)^2 + \dots + (h_n - h_n^*)^2 + (H(\mathbf{y}_{n+1}) - h_{n+1}^*)^2} = \sqrt{\epsilon^2 + \tau_1^2}, \quad (9)$$

where ϵ and τ_1 can be sufficiently small constants. $\mathbf{h}^* \in \mathbb{R}^{n+1}$ is the exact solution of energy Eq. (3). $\mathbf{h} \in \mathbb{R}^{n+1}$ represents the height vector of $n+1$ samples, which is the result of combining $H(\mathbf{y}_{n+1})$ into the optimal height vector in Proposition 1. Analogously, if m samples are joined and nearly the manifold the target domain, the optimal $\hat{\mathbf{h}} \in \mathbb{R}^m$ of out-of-sample is obtained by the trained $H(\mathbf{Y})$, its corresponding exact solution is $\hat{\mathbf{h}}^* \in \mathbb{R}^m$. We then give the following theorem.

Theorem 2. Let $\mathbf{h} := (\mathbf{h}, \hat{\mathbf{h}}) \in \mathbb{R}^{n+m}$ and $\mathbf{h}^* := (\mathbf{h}^*, \hat{\mathbf{h}}^*) \in \mathbb{R}^{n+m}$ be the height vector of our method and the exact solution of Eq. (3), respectively. Then, the following inequality is valid

$$\|\mathbf{h} - \mathbf{h}^*\|_2 \leq \sqrt{\epsilon^2 + m\tau_1^2}. \quad (10)$$

proof. According to the above derivation and Eq. (9),

$$\begin{aligned} \|\mathbf{h} - \mathbf{h}^*\|_2^2 &\leq (h_1 - h_1^*)^2 + \dots + (h_n - h_n^*)^2 + (\bar{h}_1 - \bar{h}_1^*)^2 + \dots + (\bar{h}_m - \bar{h}_m^*)^2 \\ &\leq \epsilon^2 + m\tau_1^2. \end{aligned}$$

Thus the Eq. (10) holds. \square

Combined with the above theoretical analysis, if the learned Bernier's height representation $H(\mathbf{Y})$ is accurate enough and m samples are adjacent to the manifold of the target domain, then the resulting height vector \mathbf{h} will approximate the exact

solution well within a given upper bound on the error. And it also indicates that the proposed algorithm is convergent, and its solution exists and is unique.

4 Experiments

To evaluate the performance of the **OT-Net**, we experiment with generative models, color transfer, and domain adaptation to show the effectiveness and efficiency of our algorithm. The detailed network architecture of the algorithm is showcased in the Appendix A.

4.1 Application to generative model

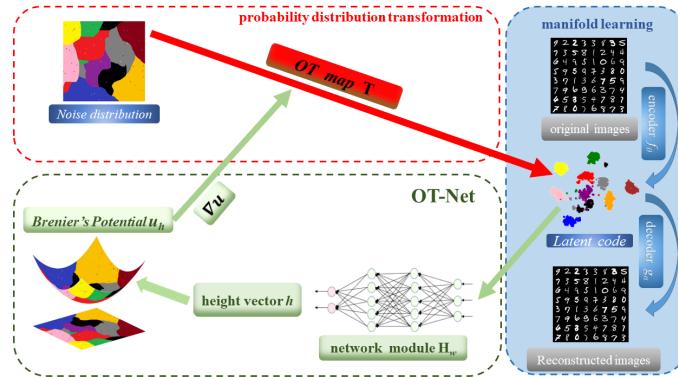


Fig. 2: Framework for embedding **OT-Net** in generative models. **manifold learning** (blue dotted box): f_θ and g_α represent the encoding and decoding map respectively, where θ and α are their network parameters. The illustrations of **OT-Net** (green dotted box) and **probability distributions transformation** (red dotted box) are detailed in the caption of Fig 1.

This section focuses on analyzing the performance of our algorithm when applied to the generative model, the diagram of our model is revealed in Fig 2. Here, we use a network module to learn Brenier’s height representation instead of calculating it through traditional numerical methods. Specifically, the latent codes are fed into the trained height representation to obtain its vector h , which in turn directly calculates the OT map. Finally, the generator of our model, which generates realistic images from random noise samples, is the composition g_α and T .

These experiments divide into three parts, in the first part, we conducted experiments on toy data to demonstrate that the proposed method can also avoid mode collapse/mixture. The second part evaluates the performance of our algorithm in the generative model. The final part confirms that our model can significantly enhance computational efficiency and reusability. Specifically, our algorithm is capable of training Brenier’s height representation on a randomly selected subset of the samples, and

the height vectors of the remaining samples can be predicted by the trained model, ultimately obtaining the OT map. These comparative experiments were conducted on four public datasets, *i.e.*, **MNIST** [44], **MNIST-FANSION** [45], **CIFAR-10** [46] and **CelebA** [47].

4.1.1 Evaluation of Mode Collapse/Mixture in Synthetic Dataset

The experiments focus on a synthetic dataset, as it has explicit distributions and known modes, allowing accurate measurement of mode collapse and the quality of generated samples. We selected the same synthetic dataset and metric indices as in [48]. Details are as follows.

Dataset. The **2D-ring** [49] is a mixture of eight two-dimensional spherical Gaussians with means $(\cos((2\pi/8)i), \sin((2\pi/8)i))$ and variances 10^{-4} in each dimension for $i \in \{1, \dots, 8\}$. The **2D-grid** [49] is a mixture of twenty-five two-dimensional spherical Gaussians with means $(-4 + 2i, -4 + 2j)$ and variances 0.0025 in each dimension for $i, j \in \{0, 1, 2, 3, 4\}$.

Metric. To quantify the mode collapse behavior, we report three metrics: 1) the **number of modes** counts the number of modes captured by samples generated from a generative model; 2) the **percentage of high-quality samples** is the ratio of such samples to the total number of synthetic data samples. If a sample falls within three standard deviations of the nearest mode, we consider it a high-quality sample; 3) the **reverse Kullback-Leibler divergence (reverse KL)** measures the balance between the induced distribution from generated samples and the induced distribution from the real samples.

Table 1: Experiments on **2D-ring** synthetic datasets under standard benchmark settings. The ratio of selected sample features in the target domain is indicated by the different features ratio r (r is 0.9, 0.8, and 0.7 in our experiments).

	2D-ring		
	Modes(Max 8)	high-quality samples	reverse KL
GAN[50]	6.3 ± 0.5	$98.2 \pm 0.2\%$	0.45 ± 0.09
ALI[51]	6.6 ± 0.3	$97.6 \pm 0.4\%$	0.36 ± 0.04
MD[52]	4.3 ± 0.8	$36.6 \pm 8.8\%$	1.93 ± 0.11
PacGAN2[48]	7.9 ± 0.1	$95.6 \pm 2.0\%$	0.07 ± 0.03
PacGAN3[48]	7.8 ± 0.1	$97.7 \pm 0.3\%$	0.10 ± 0.02
PacGAN4[48]	7.8 ± 0.1	$95.9 \pm 1.4\%$	0.07 ± 0.02
AE-OT[8]	8.0 ± 0.0	$99.6 \pm 0.3\%$	0.004 ± 0.001
Ours($r=0.9$)	8.0 ± 0.0	$99.48 \pm 0.07\%$	0.045 ± 0.014
Ours($r=0.8$)	8.0 ± 0.0	$98.72 \pm 0.23\%$	0.239 ± 0.073
Ours($r=0.7$)	8.0 ± 0.0	$98.20 \pm 0.47\%$	0.303 ± 0.049

Lin et al. [48] evaluated GAN, ALI, MD, and PacGAN on synthetic sets with the above three metrics. For the test of AE-OT and our model, there is no need to train an autoencoder because both the source domain and the target domain are two-dimensional. A two-dimensional extended OT map is straightforwardly computed. But

Table 2: Experiments on **2D-grid** synthetic datasets under standard benchmark settings. The ratio of selected sample features in the target domain is indicated by the different features ratio r .

	2D-grid		
	Modes(Max 25)	high-quality samples	reverse KL
GAN[50]	17.3 ± 0.8	$94.8 \pm 0.7\%$	0.70 ± 0.07
ALI[51]	24.1 ± 0.4	$95.7 \pm 0.6\%$	0.14 ± 0.03
MD[52]	23.8 ± 0.5	$79.9 \pm 3.2\%$	0.18 ± 0.03
PacGAN2[48]	23.8 ± 0.7	$91.3 \pm 0.8\%$	0.13 ± 0.04
PacGAN3[48]	24.6 ± 0.4	$94.2 \pm 0.4\%$	0.06 ± 0.02
PacGAN4[48]	24.8 ± 0.2	$93.6 \pm 0.6\%$	0.04 ± 0.01
AE-OT[8]	25.0 ± 0.0	$99.8 \pm 0.2\%$	0.007 ± 0.002
Ours($r=0.9$)	25.0 ± 0.0	$99.62 \pm 0.02\%$	0.0324 ± 0.0004
Ours($r=0.8$)	25.0 ± 0.0	$99.37 \pm 0.02\%$	0.0606 ± 0.0002
Ours($r=0.7$)	25.0 ± 0.0	$99.27 \pm 0.02\%$	0.0932 ± 0.0029

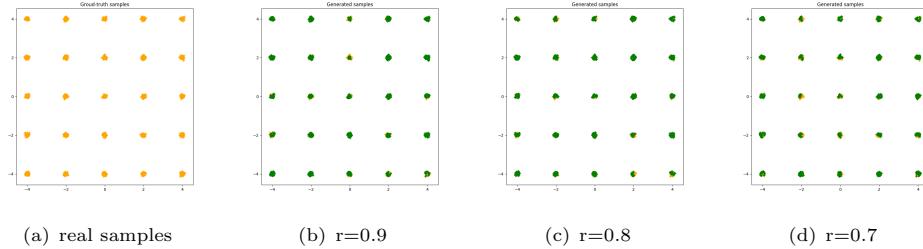


Fig. 3: Mode collapse/mixture comparison on the 2D grid dataset. Scatter plots are shown for 2D samples generated from (a) the true distribution of the 2D grid, and (b), (c), and (d) the generated distributions of the 2D grid using our model with different ratios of selected sample features ($r = 0.9, 0.8$, or 0.7). In the plots, orange marks represent real samples, and green marks represent generated samples. The results show that our model successfully captures all 25 modes.

the way we calculate the OT map is different from AE-OT since the height vector is obtained through **OT-Net**. Our advantage is that we do not need to re-optimize the height representation. In other words, when adding new samples, **OT-Net** can directly provide the height vector of out-of-sample.

To verify the proposed model also avoid mode collapse/mixture, we choose the total number of synthetic data samples with a ratio of r ($0.9, 0.8$, or 0.7) to calculate the entire OT map. The results are reported in Tab. 1 ~ Tab. 2, and results of previous methods are copied from Lin et al. [48] and An et al. [8]. It can be seen that both the AE-OT and the proposed model outperform other models in these three evaluation metrics. Even though the percentage of high-quality samples and the reverse KL decrease as the proportions decrease, the results are still comparable to

other methods, where the mean and standard deviation were obtained through multiple experiments. The results indicate that our method can capture all modes and generate higher-quality samples, even when the number of features is limited. Furthermore, the visual results for 2D-grid data are displayed in Fig. 3, which indicates that our approach can capture all modes without mode mixture when selecting different feature ratios. Based on these results, it can be concluded that the proposed model effectively addresses mode collapse/mixture issues. What's more, this shows the height representation learned with part of the samples can predict the out-of-sample height.

4.1.2 Quality Evaluation of Generative Image

To evaluate the performance of our algorithm used for image generation, we perform quantitative and qualitative experiments. Fréchet Inception Distance (FID) was proposed by Heusel et al. [53] to use quantify the quality of generated samples.

Table 3: Quantitative comparison with FID \downarrow . The best result is shown in bold. MM: manifold matching; NS: non-saturating; LSGAN: least-squares GAN; BEGAN: boundary equilibrium GAN.

Dataset	MM GAN [54]	NS GAN[54]	LSGAN[55]	WGAN[56]	BEGAN[57]	VAE[58]	GLO[59]	AE-OT[8]	Ours
Fashion	29.6	26.5	30.7	21.5	22.9	58.7	57.7	10.98	9.85
Cifar10	72.7	58.5	87.1	55.2	71.4	155.7	65.4	25.98	24.09
CelebA	65.6	55.0	53.9	41.3	38.9	85.7	52.4	38.68	34.83

Depending on its calculation formula, the FID scores are reported in Tab. 3, and statistics of various GANs come from Ref. [8, 60, 61]. The results of the AE-OT and our model are obtained under the same Encoder-Decoder architecture. It reveals that our method achieves the best scores.



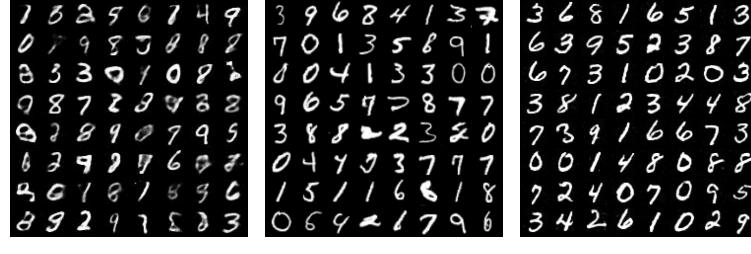
Fig. 4: The visual comparison on Cifar10 dataset.

Furthermore, we provide **qualitative comparisons** between our method and various GAN models. Fig. 4 ~ Fig. 5 depict visual comparisons between random samples generated by our model and the reported results of other models. From the visual results, the generative images produced by our proposed method are of better quality compared to other models. Moreover, the generated face images are not blurred or mixed.



(a) α -GAN [64] (b) VEEGAN [49] (c) WGAN-QC [9] (d) AE-OT [8] (e) OT-Net

Fig. 5: The visual comparison on CelebA dataset.



(a) OT-ICNN[17] (b) OTM[22] (c) OT-Net

Fig. 6: The visual comparison of three neural-network-based algorithms on MNIST.



Fig. 7: Our method for latent space interpolation on CelebA (64×64).

In addition to the aforementioned results, our algorithm is compared with other neural network-based algorithms. Namely, Makkula et al. [17] proposes a new framework called OT-ICNN, which utilizes Input Convex Neural Networks to estimate the gradient of a convex function as an OT map; Rout et al. [22] present an end-to-end algorithm (OTM) for fitting OT map with quadratic cost. Results are shown in Fig. 6, where we can observe that the generated image results of OT-ICNN and OTM exhibit mode mixing, whereas our algorithm does not. Moreover, we display the latent space interpolation between the generated samples in Fig. 7, which also shows our model can avoid mode mixture.

4.1.3 Verifying Reusability of The Proposed Algorithm

In this section, we will highlight the strengths of our algorithm, *i.e.*, we can select some sample points to train the Brenier’s height representation, and then use the trained model to predict the height vectors of the remaining samples, and finally obtain the OT map directly. Compared to the AE-OT algorithm, the **OT-Net** can improve its

Table 4: A comparison was made between the time cost of AE-OT and OT-net for computing OT mapping using the same encoder-decoder architecture. The training feature ratio indicates the proportion of selected samples. The results demonstrate that this method significantly improves computational efficiency.

Trained feature ratio	Fahion		Cifar		CelebA	
	AE-OT (s)	Ours (s)	AE-OT (s)	Ours (s)	AE-OT (s)	Ours (s)
1.0	134.83	37.22	222.82	21.97	417.73	57.31
0.95	—	24.56	—	28.11	—	32.64
0.90	—	26.62	—	19.81	—	36.16
0.85	—	29.30	—	17.27	—	37.33
0.80	—	19.61	—	28.49	—	36.31
0.75	—	23.42	—	24.42	—	32.35
0.70	—	16.35	—	22.90	—	44.15

reusability. Since the AE-OT algorithm needs to recalculate the OT map when the number of features in the target domain changes. Hence, to verify the feasibility and effectiveness of the proposed algorithm, we apply it to the generative model in our experiments.

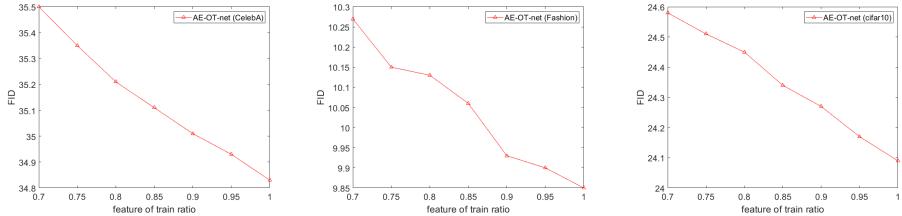


Fig. 8: The curves on the left, middle, and right display the visual of FID scores with different feature ratios selected on the CelebA, MNIST-fashion, and Cifar10, respectively.

Firstly, based on the results reveal in Fig. 8, although the FID score gradually increases as the number of selected features decreases, the image quality evaluation results are still superior to those obtained by AE-OT. As presented in Tab. 3, the network enables the computation of the whole height vector using a randomly selected subset of features for training. In case of new feature additions, the trained network can directly calculate the height vector to obtain OT mapping. The visual effects of the newly generated images, with randomly selected part features, are shown in Fig. 9. Where 0.7 ~ 1.0 indicates the ratio of randomly selected samples from the dataset.

Secondly, in Tab. 4, the time taken to train the OT mapping is reported for different feature ratios. However, the time reduction is not exact as the ratio decreases, due to the discrepancy in the number of iteration steps required for the algorithm to converge under the same error threshold. When all the features are fed into the network for training, the proposed approach only takes 37.22s, 21.97s, and 57.31s to



Fig. 9: The generated images by our model were visually compared under different feature ratios. The ratios of selected sample features on MNIST-fashion and CelebA are 0.7, 0.8, 0.9, and 1.0, from left to right.

train the OT map on the MNIST-fashion, Cifar10, and CelebA datasets, respectively. Compared with the AE-OT model, the time is greatly reduced. To be exact, once new features are added, we can readily calculate the height vector using the trained height representation. When we select 80% of the data samples, calculating the height vector of the remaining samples takes only 0.028s on the MNIST-fashion, in a similar way, Cifar10 consumes 0.078s and CelebA consumes 0.513s. However, when encountering the above problem, the AE-OT model needs to be retrained, which inevitably results in increased computational costs. The time taken for training OT is reported in Tab. 4, which is 134.83s, 222.82s, and 417.73s for the MNIST-fashion, Cifar10, and CelebA datasets, respectively. In summary, the **OT-Net** not only significantly reduces the training time of OT, but also is capable of predicting the height vectors of out-of-sample, thereby directly getting the OT mapping.

Finally, the experimental results show that the algorithm can not only directly predict out-of-sample height vectors using the trained height representation, but also can significantly improve the computational efficiency of the algorithm.

4.2 Application to color transfer

This section will present comparative experiments on color transfer between our algorithm and other classic OT algorithms. The color transfer problem is to modify an input image X_0 so that its colors match the colors of another input image Y_0 . For a fair comparison, we use the solver provided in the POT¹ [66]. Specifically, OT Network Simplex solver [65] for calculating Earth Movers Distance(EMD), and Entropic

¹<https://pythonot.github.io/#>



Fig. 10: Comparison of the visual results of color transfer. The first column is the original image, and the second to fourth columns are the results of the OT Network Simplex solver [65], Sinkhorn algorithm [12], and **OT-Net**, respectively.

regularization OT solver with Sinkhorn-Knopp Algorithm [12]. Furthermore, we experiment on the same hardware platform and environments, the regularized coefficient of the Sinkhorn algorithm is 0.1, and the learning rate of our algorithm is 0.05. We selected two color styles of flowers and oceans for the color transfer task. The visual results are displayed in Fig. 10, which reveals our algorithm generates sharper textures with more realistic and rich colors, especially in the background areas.

Specifically, the first two rows of Fig. 10 show the results of color transfer between the white and purple flowers, where the result of **OT-Net** in the last column reveals that the purple nicely transfers to the white flowers. The last two rows represent the color transfer from the ocean blue sky with white clouds to the red sunset, where the last column shows that **OT-Net** is capable of transforming the colors of both scenes very effectively. The second and third columns display the results of the OT Network Simplex solver and Sinkhorn algorithms, respectively. We found that the image appears dull and the colors are not bright enough after color transfer. This also demonstrates the effectiveness of our algorithm.

4.3 Application to domain adaptation

This part mainly introduces the applications of classical methods and our algorithm in domain adaptation. In the 2D toy dataset setting, we randomly generate 4000 sample points and divide them into 3 classes as source domain (target domain) which are generated by different Gaussian mixture models. Then, three classical algorithms are

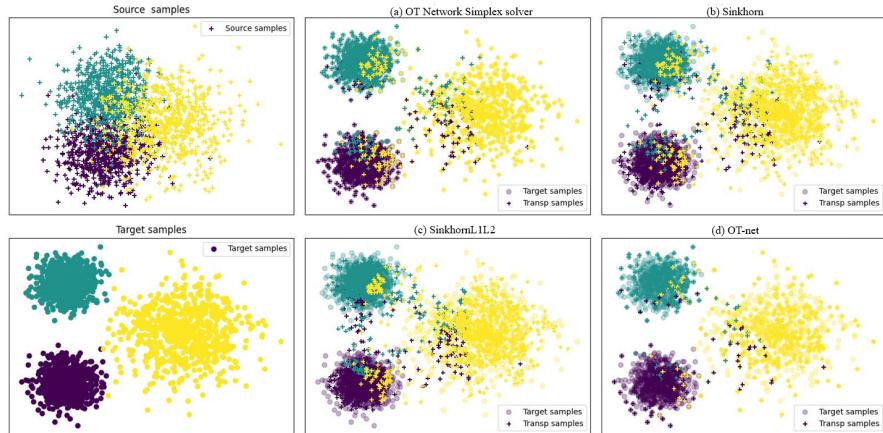


Fig. 11: Comparison of the visual results of domain adaption. The domain adaptation accuracies are (a): OT Network Simplex solver [65], 83.55%, (b): Sinkhorn algorithm [12], 79.49%, (c): SinkhornL1L2 [31], 81.20%, (d): our **OT-Net**, **84.45%**.

selected for application to domain adaptation. The first one is proposed by Flamary et al. [31] which adds a class-based term to the regularized OT for domain adaptation, shorthand for SinkhornL1L2, where L1 and L2 represent the regularization term of OT and class-based, respectively. The coefficients of OT and class-based regularization terms are 1.0 and 0.1. The remaining two are the OT Network Simplex solver [65] for computing EMD, and the Sinkhorn-Knopp algorithm [12] that the regularization coefficient is 0.1. Finally, we evaluate the performance of all those algorithms by calculating the percentage of correctly classified instances across all categories. From the results shown in Fig. 11, the proposed algorithm obtains higher accuracy **84.45%**, this shows that our algorithm is effective.

Table 5: Classification accuracy of domain adaptation. The first row represents the ratio of selected data sample features used for training. The proposed algorithm is used to compute the OT map by randomly selecting the data features. This map is then applied to calculate the classification accuracy of the corresponding feature proportions, the results are shown in the second row. Finally, the learned $H(\mathbf{y})$ is used to compute the height vector of the remaining samples to obtain the whole OT map, which is used to produce the classification result in the third row.

Data-Ratio	70%	75%	80%	85%	90%	95%
Part-data-Acc	83.46%	83.83%	83.87%	84.15%	84.25%	84.61%
All-data-Acc	83.18%	83.55%	83.65%	84.11%	84.18%	84.33%

What's more, to verify one of the strengths of our algorithm. That is, a subset of sample points is selected and used to train Bernier's height representation, while the

remaining samples are directly predicted using the trained model. Tab. 5 displays the classification accuracy of **OT-Net** in domain adaptation. First, a subset of samples is randomly selected for training the height representation. Then, the trained model is used to predict the height vectors of the remaining samples to obtain the OT map. Finally, the classification accuracy is calculated. The results show that the method can predict the out-of-sample height vectors to directly obtain the entire height vector and OT map, and the algorithm is effective when applied to the domain adaptation task.

5 Conclusion

In this paper, a neural network-based new algorithm is presented to solve the OT map, which is a reusable neural optimal transport solver. To be specific, when new samples are joined, we can directly utilize the learned Bernier’s height representation to calculate the height vector of out-of-sample, which can avoid recalculating or retraining the whole OT map. This can greatly improve the computational effectiveness and reusability of the OT map of our algorithm. Moreover, we analyze the convergence of the algorithm and its theoretical error bounds, and experiments demonstrate that our algorithm achieves comparable performance on generative models, color transfer, and domain adaptation.

Appendix A

This section introduces the Encoder-Decoder architecture that is applied to the generative model and height representation network architecture of our algorithm. The autoencoder network structures are presented in Tab. A1 and Tab. A2, and the height representation network structure can be found in Tab. A3, where P_{data} and P_{latent} represent the data distribution and the latent coding distribution, respectively. The Encoder-Decoder architecture was trained using the Adam algorithm with mini-batches of size 512, and learning rates of 2e-4, 1e-4, and 2e-5 in MNIST-FASHION [45], CIFAR-10 [46], and CelebA [47], respectively. The height representation network was also trained using Adam with mini-batches of size 512, and learning rates of 0.004, 0.005, and 0.005 in MNIST-FASHION, CIFAR-10, and CelebA, respectively.

Table A1: The Encoder architecture for CelebA, Fashion and Cifar10.

Input $x \sim P_{data}$	kernel size			stride	padding	BN	activation	number of outputs			
	CelebA	Fahion	Cifar10					CelebA	Fahion	Cifar10	
Layer1	Convolution	4*4	4*4	4*4	2	1	Yes	LeakyReLU	32*32*dim	14*14*dim	16*16*dim
Layer2	Convolution	4*4	4*4	4*4	2	1	Yes	LeakyReLU	16*16*dim*2	7*7*dim*2	8*8*dim*2
Layer3	Convolution	4*4	3*3	4*4	2	1	Yes	LeakyReLU	8*8*dim*4	4*4*dim*4	4*4*dim*4
Layer4	Convolution	4*4	4*4	4*4	2	1	Yes	LeakyReLU	4*4*dim*8	2*2*dim*8	2*2*dim*8
Layer5	Convolution	4*4	2*2	2*2	1	—	—	—	1*1*100	1*1*100	1*1*100

Table A2: The Decoder architecture for CelebA, Fashion and Cifar10.

Input $y \sim P_{latent}$	kernel size			stride	padding	BN	activation	number of outputs			
	CelebA	Fahion	Cifar10					CelebA	Fahion	Cifar10	
Layer1	Transposed Convolution	4*4	2*2	2*2	1	—	—	4*4*dim*8	2*2*dim*8	2*2*dim*8	
Layer2	Transposed Convolution	4*4	4*4	4*4	2	1	Yes	ReLU	8*8*dim*4	4*4*dim*4	4*4*dim*4
Layer3	Transposed Convolution	4*4	3*3	4*4	2	1	Yes	ReLU	16*16*dim*2	7*7*dim*2	8*8*dim*2
Layer4	Transposed Convolution	4*4	4*4	4*4	2	1	Yes	ReLU	32*32*dim	14*14*dim	16*16*dim
Layer5	Transposed Convolution	4*4	4*4	4*4	2	1	—	Tanh	64*64*3	28*28*3	32*32*3

Table A3: The height representation architecture for OT-Net.

Input $y \sim P_{latent}$	BN	activation	number of outputs
Nums*100 (Nums indicates the numbers of latent code features.)			
Layer1	Linear(100, 512)	Yes	ReLU
Layer2	Linear(512, 512)	Yes	ReLU
Layer3	Linear(512, 512)	Yes	ReLU
Layer4	Linear(512, 1)	—	Nums*1

Declarations

Author Contributions. Zezeng Li provided original ideas and code implementation of the proposed algorithm. Shenghao Li was responsible for most of the experimental validation, theoretical derivation and manuscript writing. Lianbao Jin, Na Lei, and Zhongxuan Luo provided constructive ideas for theoretical derivation and experimental setup. All authors participated in the writing of the manuscript, and read and approved the final manuscript.

Funding. This research was supported by the National Key R&D Program of China (2021YFA1003003), and the National Natural Science Foundation of China under Grant (61936002, T2225012).

Availability of data and material. The data/reanalysis that supports the findings of this study are publicly available online at <http://yann.lecun.com/exdb/mnist/>, and <https://github.com/zalandoresearch/fashion-mnist>, and <http://www.cs.toronto.edu/~kriz/cifar.html>, and <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.

Conflicts of interest/Competing interests. The authors declare that they have no conflict of interest.

Ethics approval and Consent to participate. The authors declare that this research did not require Ethics approval or Consent to participate since it does not concern human participants or human or animal datasets.

Consent for publication. The authors of this manuscript consent to its publication.

Code availability. The code can be obtained by contacting Shenghao Li and Zezeng Li.

References

- [1] Alvarez-Melis, D., Jaakkola, T., Jegelka, S.: Structured optimal transport. In: International Conference on Artificial Intelligence and Statistics, pp. 1771–1780 (2018). PMLR
- [2] Seguy, V., Damodaran, B.B., Flamary, R., Courty, N., Rolet, A., Blondel, M.: Large-scale optimal transport and mapping estimation. In: ICLR 2018-International Conference on Learning Representations, pp. 1–15 (2018)
- [3] Wang, W., Xu, H., Wang, G., Wang, W., Carin, L.: Zero-shot recognition via optimal transport. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3471–3481 (2021)
- [4] Lavenant, H., Claici, S., Chien, E., Solomon, J.: Dynamical optimal transport on discrete surfaces. ACM Transactions on Graphics (TOG) **37**(6), 1–16 (2018)
- [5] Cui, L., Qi, X., Wen, C., Lei, N., Li, X., Zhang, M., Gu, X.: Spherical optimal transportation. Computer-Aided Design **115**, 181–193 (2019)
- [6] Rakotomamonjy, A., Flamary, R., Gasso, G., Alaya, M.E., Berar, M., Courty, N.: Optimal transport for conditional domain matching and label shift. Machine Learning, 1–20 (2022)
- [7] Lei, N., An, D., Guo, Y., Su, K., Liu, S., Luo, Z., Yau, S.-T., Gu, X.: A geometric understanding of deep learning. Engineering **6**(3), 361–374 (2020)
- [8] An, D., Guo, Y., Lei, N., Luo, Z., Yau, S.-T., Gu, X.: Ae-ot: A new generative model based on extended semi-discrete optimal transport. ICLR 2020 (2019)
- [9] Liu, H., Gu, X., Samaras, D.: Wasserstein gan with quadratic transport cost. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4832–4841 (2019)
- [10] Li, Z., Wang, W., Lei, N., Wang, R.: Weakly supervised point cloud upsampling via optimal transport. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2564–2568 (2022). IEEE
- [11] Miroshnikov, A., Kotsiopoulos, K., Franks, R., Ravi Kannan, A.: Wasserstein-based fairness interpretability framework for machine learning models. Machine Learning **111**(9), 3307–3357 (2022)
- [12] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)
- [13] Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., Peyré, G.: Iterative bregman projections for regularized transportation problems. SIAM Journal on Scientific Computing **37**(2), 1111–1138 (2015)

- [14] Dvurechensky, P., Gasnikov, A., Kroshnin, A.: Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In: International Conference on Machine Learning, pp. 1367–1376 (2018). PMLR
- [15] Xie, Y., Wang, X., Wang, R., Zha, H.: A fast proximal point method for computing exact wasserstein distance. In: Uncertainty in Artificial Intelligence, pp. 433–453 (2020). PMLR
- [16] An, D., Lei, N., Xu, X., Gu, X.: Efficient optimal transport algorithm by accelerated gradient descent. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 10119–10128 (2022)
- [17] Makkula, A., Taghvaei, A., Oh, S., Lee, J.: Optimal transport mapping via input convex neural networks. In: International Conference on Machine Learning, pp. 6672–6681 (2020). PMLR
- [18] Daniels, M., Maunu, T., Hand, P.: Score-based generative neural networks for large-scale optimal transport. Advances in neural information processing systems **34**, 12955–12965 (2021)
- [19] Fan, J., Liu, S., Ma, S., Chen, Y., Zhou, H.: Scalable computation of monge maps with general costs. arXiv preprint arXiv:2106.03812, 4 (2021)
- [20] Korotin, A., Selikhanovich, D., Burnaev, E.: Neural optimal transport. arXiv preprint arXiv:2201.12220 (2022)
- [21] Gazdieva, M., Rout, L., Korotin, A., Filippov, A., Burnaev, E.: Unpaired image super-resolution with optimal transport maps. arXiv preprint arXiv:2202.01116 (2022)
- [22] Rout, L., Korotin, A., Burnaev, E.: Generative modeling with optimal transport maps. arXiv preprint arXiv:2110.02999 (2021)
- [23] Asadulaev, A., Korotin, A., Egiazarian, V., Burnaev, E.: Neural optimal transport with general cost functionals. arXiv preprint arXiv:2205.15403 (2022)
- [24] Gu, X., Luo, F., Sun, J., Yau, S.-T.: Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations. arXiv preprint arXiv:1302.5472 (2013)
- [25] Brenier, Y.: Polar factorization and monotone rearrangement of vector-valued functions. Communications on pure and applied mathematics **44**(4), 375–417 (1991)
- [26] Villani, C.: Topics in optimal transportation. OR/MS Today **30**(3), 66–67 (2003)
- [27] Villani, C.: Optimal transport: Old and new. **338** (2008)

- [28] Peyré, G., Cuturi, M., *et al.*: Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* **11**(5-6), 355–607 (2019)
- [29] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. *Advances in neural information processing systems* **30** (2017)
- [30] Courty, N., Flamary, R., Habrard, A., Rakotomamonjy, A.: Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems* **30** (2017)
- [31] Flamary, R., Courty, N., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (2016)
- [32] Xu, B., Zeng, Z., Lian, C., Ding, Z.: Few-shot domain adaptation via mixup optimal transport. *IEEE Transactions on Image Processing* **31**, 2518–2528 (2022)
- [33] El Hamri, M., Bennani, Y., Falih, I.: Hierarchical optimal transport for unsupervised domain adaptation. *Machine Learning* **111**(11), 4159–4182 (2022)
- [34] Pitie, F., Kokaram, A.C., Dahyot, R.: N-dimensional probability density function transfer and its application to color transfer. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2, pp. 1434–1439 (2005). IEEE
- [35] Ferradans, S., Papadakis, N., Peyré, G., Aujol, J.-F.: Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences* **7**(3), 1853–1882 (2014)
- [36] Li, Z., Lei, N., Shi, J., Xue, H.: Real-world super-resolution under the guidance of optimal transport. *Machine Vision and Applications* **33**(3), 48 (2022)
- [37] Xiao, J., Liu, T., Zhao, R., Lam, K.-M.: Balanced distortion and perception in single-image super-resolution based on optimal transport in wavelet domain. *Neurocomputing* **464**, 408–420 (2021)
- [38] Petzka, H., Fischer, A., Lukovnicov, D.: On the regularization of wasserstein gans. arXiv preprint arXiv:1709.08894 (2017)
- [39] Sanjabi, M., Ba, J., Razaviyayn, M., Lee, J.D.: On the convergence and robustness of training gans with regularized optimal transport. *Advances in Neural Information Processing Systems* **31** (2018)
- [40] Chen, S., Figalli, A.: Partial w_2, p regularity for optimal transport maps. *Journal of Functional Analysis* **272**(11), 4588–4605 (2017)
- [41] Lei, N., Su, K., Cui, L., Yau, S.-T., Gu, X.D.: A geometric view of optimal transportation and generative model. *Computer Aided Geometric Design* **68**, 1–21

(2019)

- [42] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
- [43] Metropolis, N., Ulam, S.: The monte carlo method. *Journal of the American statistical association* **44**(247), 335–341 (1949)
- [44] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
- [45] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
- [46] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
- [47] Zhang, Z., Luo, P., Loy, C.C., Tang, X.: From facial expression recognition to interpersonal relation prediction. *International Journal of Computer Vision* **126**, 550–569 (2018)
- [48] Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems* **31** (2018)
- [49] Srivastava, A., Valkov, L., Russell, C., Gutmann, M.U., Sutton, C.: Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems* **30** (2017)
- [50] Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
- [51] Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016)
- [52] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. *Advances in neural information processing systems* **29** (2016)
- [53] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017)
- [54] Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A.M., Mohamed, S., Goodfellow, I.: Many paths to equilibrium: Gans do not need to decrease a divergence at every step. arXiv preprint arXiv:1710.08446 (2017)

- [55] Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
- [56] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN (2017)
- [57] Berthelot, D., Schumm, T., Metz, L.: Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717 (2017)
- [58] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
- [59] Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776 (2017)
- [60] Hoshen, Y., Li, K., Malik, J.: Non-adversarial image synthesis with generative latent nearest neighbors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5811–5819 (2019)
- [61] Lucic, M., Kurach, K., Michalski, M., et al.: Are gans created equal? a large-scale study. Advances in neural information processing systems **31** (2018)
- [62] Zhai, S., Cheng, Y., Feris, R., Zhang, Z.: Generative adversarial networks as variational training of energy based models. arXiv preprint arXiv:1611.01799 (2016)
- [63] Abbasnejad, M.E., Shi, Q., Abbasnejad, I., Hengel, A.v.d., Dick, A.: Bayesian conditional generative adversarial networks. arXiv preprint arXiv:1706.05477 (2017)
- [64] Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks. arXiv preprint arXiv:1706.04987 (2017)
- [65] Bonneel, N., Van De Panne, M., Paris, S., Heidrich, W.: Displacement interpolation using lagrangian mass transport. In: Proceedings of the 2011 SIGGRAPH Asia Conference, pp. 1–12 (2011)
- [66] Flamary, R., Courty, N., Gramfort, A., Alaya, M.Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al.: Pot: Python optimal transport. The Journal of Machine Learning Research **22**(1), 3571–3578 (2021)