Scalable Computation of Monge Maps with General Costs

Jiaojiao Fan *
Department of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
jiaojiaofan@gatech.edu

Shu Liu *
Department of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, USA
sliu459@gatech.edu

Shaojun Ma *
Department of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, USA
shaojunma@gatech.edu

Yongxin Chen
Department of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
yongchen@gatech.edu

Haomin Zhou
Department of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, USA
hmzhou@math.gatech.edu

Abstract

Monge map refers to the optimal transport map between two probability distributions and provides a principled approach to transform one distribution to another. In spite of the rapid developments of the numerical methods for optimal transport problems, computing the Monge maps remains challenging, especially for high dimensional problems. In this paper, we present a scalable algorithm for computing the Monge map between two probability distributions. Our algorithm is based on a weak form of the optimal transport problem, thus it only requires samples from the marginals instead of their analytic expressions, and can accommodate optimal transport between two distributions with different dimensions. Our algorithm is suitable for general cost functions, compared with other existing methods for estimating Monge maps using samples, which are usually for quadratic costs. The performance of our algorithms is demonstrated through a series of experiments with both synthetic and realistic data.

1 Introduction

In recent years we have witnessed great success of optimal transport (OT) (Villani 2008) based applications in machine learning community (Arjovsky et al., 2017; Krishnan & Martinez, 2018; Li et al., 2019; Makkava et al., 2020; Inoue et al., 2020; Ma et al., 2020; Fan et al., 2020; Haasler et al., 2020). As a crucial concept of OT, Wasserstein distance is used to evaluate the discrepancy between distributions due to its good properties such as symmetry and robustness. Given any two probability

*Equal contribution

Preprint. Under review.
distributions \( \rho_a \) and \( \rho_b \) both on \( \mathbb{R}^d \), the Wasserstein distance between \( \rho_a \) and \( \rho_b \) reads

\[
\inf_{\pi} \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y) d\pi(x, y) \bigg| \pi \in \Pi(\rho_a, \rho_b) \right\}
\]

which is known as the Kantorovich form of OT. We define \( c(x, y) \) as the cost function which quantifies the effort of moving one unit of mass from location \( x \) to location \( y \). We also define \( \Pi(\rho_a, \rho_b) \) as the set of joint distribution of \( \rho_a \) and \( \rho_b \). An optimal \( \pi^* \) to achieve the Wasserstein distance between two distributions is the solution of OT, meanwhile there exists an optimal map \( T \) such that \( T(x) \) and \( y \) have the same distribution when \( x \sim \rho_a \) and \( y \sim \rho_b \).

Solving (1) directly yields an intractable problem due to the curse of dimensionality for discretizing the space. By adding an entropic regularization. Sinkhorn distance (Cuturi [2013]) has been widely computed since it is friendly to high dimensional cases [Altschuler et al., 2017; Genevay et al., 2018; Li et al., 2019; Xie et al., 2020], but the algorithm does not scale well to a large number of samples and is not suitable to handle continuous probability measures (Genevay et al., 2016). However, a modern formulation in the work of Kantorovich, now known as the Monge–Kantorovich (MK) problem and its dual form is found to facilitate the computation of OT problem. Meanwhile, with the rising popularity of neural networks, many regularization-based OT problems have been formulated, such as entropic regularized OT (Seguy et al., 2017), Laplacian regularization (Flamary et al., 2014), Group-Lasso regularized OT (Court et al., 2016), Tsallis regularized OT (Muzellec et al., 2017) and OT with \( L^2 \) regularization (Dessein et al., 2018).

In particular, if we assume the cost function in the Kantorovich dual form as \( c(x, y) = \frac{1}{2} |x - y|^2 \), then we are evaluating well-known Wasserstein-2 distance denoted by \( W_2 \). Benamou & Brenier (2000) rewrites the OT problem from a fluid dynamics perspective, which is known as the Benamou-Brenier problem. The dynamic view of OT is beneficial for describing the density evolution under the influence of a vector field (Chen et al., 2016; 2020). But solving it leads to a partial differential equation (PDE) system, which is not a trivial task, especially in a high dimensional setting.

To overcome the drawbacks of fluid dynamics formulation, we propose a computationally efficient and scalable algorithm for estimating the Wasserstein distance and optimal map between probability distributions over continuous spaces. Particularly, we apply the Lagrangian multiplier to the MK problem, to this end we obtain a minimax problem. Our contribution can be summarized as follows: 1) We develop an scalable algorithm to compute the optimal transport map associated with general transport costs between any two distributions given their samples; 2) Our model is able to deal with the cases that \( \rho_a \) or \( \rho_b \) is not absolute continuous; 3) Our method is also capable of computing OT problems between distributions over spaces that do not share the same dimension. 4) We provide a rigorous error analysis of the algorithm; 5) We demonstrate its performance and its scaling properties in truly high dimensional settings.

**Related work:** Li et al. (2018); Gangbo et al. (2019) propose efficient approaches of computing OT problem, but the methods become infeasible in high dimensional cases. However, Kantorovich dual problem and application of DNN make OT computation more tractable. As a variety of models have been proposed to handle high dimensions via neural networks, we can classify OT models into static ones and dynamic ones. Static models include typical Kantorovich dual based work, such as Wasserstein-1 GAN (Arjovsky et al., 2017) and Wasserstein-\( 1 \) related variations (Gulrajani et al., 2017; Tolstikhin et al., 2017; Miyato et al., 2018; Tong Lin et al., 2018; Dukler et al., 2019; Xie et al., 2019), where cost function is chosen as \( L^1 \) norm, then the Wasserstein-\( 1 \) distance is computed via solving a maximization problem. However, when we solve for Wasserstein-\( 1 \) related problems, the non-trivial Lipschitz-\( 1 \) constraint is challenging to be theoretically and strictly realized. Utilizing input convex neural networks (ICNN) (Amos et al., 2017) to approximate the potential function is a good way to avoid Lipschitz-\( 1 \) constraint. For example, Fan et al. (2020) and Makkova et al. (2020) extend the semi-dual formulation of OT to new minimax problems.

The other one track of OT models are dynamic ones. By utilizing the structure of Wasserstein geodesic information (Villani, 2008; Benamou et al., 2010) or the distribution interpolation proposed
The subject of this work originates from the static MK problem. In particular, our algorithm is inspired by the recent work in estimation of optimal transport map and Wasserstein-p distance using dynamic view of OT (Liu et al., 2021), where samples’ density evolution is described via a pushforward map, Lagrange Multipliers are parametrized as neural networks. However our formulation extends to a general cost function setting without considering time dimension.

2 Background

Recall the Optimal Transport problem (1) on $\mathbb{R}^d$. In our study, we consider a more general OT problem from $\mathbb{R}^n$ to $\mathbb{R}^m$.

\[
\inf_{\pi} \left\{ \int_{\mathbb{R}^n \times \mathbb{R}^m} c(x,y) d\pi(x,y) \mid \pi \in \Pi(\rho_a, \rho_b) \right\} \tag{2}
\]

We will mainly focus on the cost function $c(\cdot, \cdot)$ satisfying the following conditions:

- There exists $a \in L^1(\rho_a)$, $b \in L^1(\rho_b)$, such that $c(x, y) \geq a(x) + b(y)$; \tag{3}
- $c$ is locally Lipschitz and superdifferentiable everywhere; \tag{4}
- $\partial_x c(x, \cdot)$ is injective for any $x \in \mathbb{R}^n$. \tag{5}

For the definitions of the bold terminologies listed above, please check Appendix

**Remark 1.** Since $\partial_x c(x, \cdot) : \mathbb{R}^m \to \mathbb{R}^n$, for most of the cost function $c$, (5) indicates that $m \leq n$.

The Kantorovich dual problem of the primal OT problem (1) is formulated as (Chap 5, Villani (2008))

\[
\sup_{(\psi, \phi) \in L^1(\rho_a) \times L^1(\rho_b)} \left\{ \int_{\mathbb{R}^m} \phi(y) \rho_a(y) dy - \int_{\mathbb{R}^n} \psi(x) \rho_a(x) dx \right\} \tag{6}
\]

It is not hard to tell that (6) is equivalent to

\[
\sup_{\psi \in L^1(\rho_a)} \left\{ \int_{\mathbb{R}^m} \psi c^+ (y) \rho_a(y) dy - \int_{\mathbb{R}^n} \psi(x) \rho_a(x) dx \right\} \tag{7}
\]

\[
\sup_{\phi \in L^1(\rho_b)} \left\{ \int_{\mathbb{R}^m} \phi(y) \rho_b(y) dy - \int_{\mathbb{R}^n} \phi c^- (x) \rho_b(x) dx \right\} \tag{8}
\]

Here we define $\psi c^+ (y) = \inf_y (\psi(y) + c(y, x))$ and $\phi c^- (x) = \sup_y (\phi(y) - c(x, y))$.

The following theorem states the equivalent relationship between the primal Optimal Transport problem and its Kantorovich dual (6), (7). The proof for a more general version can be found in Theorem 5.10 of Villani (2008).

**Theorem 1 (Kantorovich Duality).** Suppose $c$ is a cost function defined on $\mathbb{R}^n \times \mathbb{R}^m$ and satisfies (3). We denote $C(\rho_a, \rho_b)$ as the infimum value of (6) or (7). Then $C(\rho_a, \rho_b) = K(\rho_a, \rho_b)$.

In general, the optimal solution $\pi^*$ to (1) can be treated as a random transport plan, i.e. we are allowed to break the single particle into pieces and then transport each piece to certain positions according to the plan $\pi^*$. However, in this study, we mainly focus on the classical version of the OT problem, which is known as the Monge problem stated as follows

\[
\min_{T: \mathbb{R}^n \to \mathbb{R}^m, T \# \rho_a = \rho_b} \int_{\mathbb{R}^n} c(x, T(x)) \rho_a(x) dx \tag{9}
\]
Here $T$ is a measurable map on $\mathbb{R}^n$, we define the pushforward of distribution $\rho_a$ by $T$ as $T_*\rho_a(E) = \rho_b(T^{-1}(E))$ for any measurable set $E \subset \mathbb{R}^n$. The Monge problem seeks for the optimal deterministic transport plan from $\rho_a$ to $\rho_b$.

The following result states the existence and uniqueness of the optimal solution to the Monge problem. It is a simplified version of Theorem 10.28 combined with Remark 10.33 taken from [Villani (2008)].

**Theorem 2** (Existence, uniqueness and characterization of the optimal Monge map). Suppose the cost $c$ satisfies (3), (4), (5), we further assume that $\rho_a$ and $\rho_b$ are compactly supported and $\rho_a$ is absolute continuous with respect to the Lebesgue measure on $\mathbb{R}^n$. Then there exists a unique transport map $T_*$ solving the Monge problem. If we denote $\psi_*, \phi_*$ as the optimal solution of (7), (8) respectively. Then $\psi_*, \phi_*, T$ satisfy

$$\nabla \psi_*(x) + \partial_x c(x, T_*(x)) = 0, \quad \nabla \phi_*(x) - \partial_y c(x, T_*(x)) = 0, \quad \rho_a \text{ almost surely.}$$

### 3 Proposed method

In order to formulate a tractable algorithm for the general Monge problem (9), we first notice that (9) is a constrained optimization problem. Thus, it is natural to introduce the Lagrange multiplier $f$ for the constraint $T_*\rho_a = \rho_b$ and then reformulate (9) as a saddle point problem

$$\sup_f \inf_T \mathcal{L}(T, f)$$

with $\mathcal{L}$ defined as

$$\mathcal{L}(T, f) = \int_{\mathbb{R}^n} c(x, T(x))\rho_a(x)dx + \int_{\mathbb{R}^m} f(y)(\rho_b - T_\sharp\rho_a) dy$$

$$= \int_{\mathbb{R}^n} [c(x, T(x)) - f(T(x))]\rho_a(x) dx + \int_{\mathbb{R}^m} f(y)\rho_b(y) dy \quad (11)$$

We can verify that the max-min scheme (10) is equivalent to the Kantorovich dual problem (8). To this end, one only need to verify:

$$\inf_T \mathcal{L}(T, f) = -\int_{\mathbb{R}^n} \sup_\xi \{f(\xi) - c(x, \xi)\}\rho_a(x)dx + \int_{\mathbb{R}^m} f(y)\rho_b(y)dy$$

$$= \int_{\mathbb{R}^m} f(y)\rho_b(y)dy - \int_{\mathbb{R}^n} f^c(x)\rho_a(x)dx. \quad (12)$$

The following theorem guarantees that the max-min scheme (10) will find the optimal Monge map.

**Theorem 3** (Consistency). Suppose the optimal solution to (10) is $(T_*, f_*)$, then $T_*$ is the optimal Monge map to the problem (9) and $f_* = \phi_*$, where $\phi_*$ is the optimal solution to (8).

The proof to this theorem can be found in the supplementary material. In exact implementation, we will replace both the map $T$ and the dual variable $f$ by the neural networks $T_\theta, f_\eta$, with $\theta, \eta$ being the parameters of the networks. We aim at solving the following saddle point problems. The algorithm is summarized in Algorithm 1

$$\max_{\theta} \min_{\eta} \mathcal{L}(T_\theta, f_\eta) := \frac{1}{N} \sum_{k=1}^N c(X_k, T_\theta(X_k)) - f_\eta(T_\theta(X_k)) + f_\eta(Y_k) \quad (13)$$

where $N$ is the batch size and $\{X_k\}, \{Y_k\}$ are samples generated by $\rho_a$ and $\rho_b$ separately.

### 4 Error Analysis via Duality Gaps

In this section, we assume that $m = n = d$, i.e. we consider Monge problem between spaces with the same dimension $d$. Suppose we solve (10) to a certain stage and obtain the pair $(T, f)$, inspired by [Hütter & Rigollet (2020) and Makkuva et al. (2020)], we want to estimate a weighted $L^2$ error between our computed map $T$ and the optimal Monge map $T_*$.

Before we present our result, we introduce definition for $c$-concave functions. We mainly adopt the definition from Chapter 5 of [Villani (2008)].
We denote the duality gaps \( M_{\text{akkuva et al.}}(2020). \)

\[
\Psi(u) = \text{Computing Wasserstein distance and optimal map from } \text{Denote } T, \text{Posterior Error Analysis via Duality Gaps}
\]

Consider for any \( y \in \mathbb{R}^d \) and assume that there exists a function \( \varphi \in C^2(\mathbb{R}^d) \) such that \( f(y) = \inf_{x \in \mathbb{R}^d} \{ \varphi(x) + c(x, y) \} \).

Theorem 4 (Posterior Error Analysis via Duality Gaps). Assume \( f \in C^2(\mathbb{R}^d) \) is a \( c \)-concave function and assume that there exists \( \varphi \in C^2(\mathbb{R}^d) \) such that \( f(y) = \inf_{x \in \mathbb{R}^d} \{ \varphi(x) + c(x, y) \} \).

Consider for any \( y \in \mathbb{R}^d \) and the minimizer \( \hat{x}_y \in \arg\min \{ \varphi(x) + c(x, y) \} \neq \emptyset \), we further assume there exists function \( \lambda(\cdot) > 0 \) such that

\[
\lambda(y) I_n \geq \nabla^2_{xx}(\varphi(x) + c(x, y))|_{x=\hat{x}_y} \succ 0
\]

where \( I_n, O_n \) denotes \( n \times n \) identity matrix and zero matrix.

We denote the duality gaps

\[
\mathcal{E}_1(T, f) = \mathcal{L}(T, f) - \inf_{\tilde{T}} \mathcal{L}(\tilde{T}, f), \quad \mathcal{E}_2(f) = \sup_{f \in \mathcal{T}} \inf_{\tilde{f}} \mathcal{L}(\tilde{T}, \tilde{f}) - \inf_{\tilde{T}} \mathcal{L}(\tilde{T}, f)
\]

Suppose \( \mathcal{E}_1(T) \leq \epsilon_1, \mathcal{E}_2(T, f) \leq \epsilon_2 \).

Denote \( T_\star \) as the optimal Monge map of the OT problem \((9)\). Then there exists a strict positive weight function \( \beta(\cdot) > 0 \) (depending on \( c, T_\star, f \) and \( \varphi \), such that the weighted \( L^2 \) error between computed map \( T \) and optimal map \( T_\star \) is upper bounded by

\[
\| T - T_\star \|_{L^2(\beta\rho_a)} \leq \sqrt{2(\epsilon_1 + \epsilon_2)}.
\]

The proof of this theorem can be found in Appendix .

Remark 2. We can verify that \( c(x, y) = \frac{1}{2}\|x - y\|^2 \) or \( c(x, y) = -x \cdot y \) satisfy the conditions mentioned above. Then Theorem 4 recovers similar results proved in Hütter & Rigollet (2020) and Makkova et al. (2020).

Remark 3. Suppose \( c \) satisfies (14), (15), if \( c \) is also an analytical function, then \( c \) has the form \( \Psi(x) + \nabla u(x)^T y + \Phi(y) \), where \( \Psi, u, \Phi \) are analytical functions on \( \mathbb{R}^d \), and \( u \) is strictly convex.

## 5 Experiments

In Section 5.1 we test our algorithm on 2D datasets with most common \( L^2 \) cost. In Section 5.2, we show the effectiveness of our method on unequal dimension space examples. In Section 5.3, we conduct extensive experiments to study the obtained map with different cost functions such as \( L^p \) cost, KL divergence, etc. Finally, in Section 5.4 we show the quantitative results for a series of Gaussian marginal experiments. The details of the experiment including hyper parameter choices are deferred to the supplementary material.

**Algorithm 1** Computing Wasserstein distance and optimal map from \( \rho_a \) to \( \rho_b \)

1: **Input:** Marginal distributions \( \rho_a \) and \( \rho_b \), Batch size \( N \), Cost function \( c(x, T(x)) \).
2: Initialize \( T_\theta, f_\theta \).
3: for \( K \) steps do
4: \( \text{Sample } \{X_k\}_{k=1}^N \text{ from } \rho_a. \text{ Sample } \{Y_k\}_{k=1}^N \text{ from } \rho_b. \)
5: for \( K_1 \) steps do
6: Update (via gradient descent) \( \eta \) to decrease \([13]\)
7: end for
8: for \( K_2 \) steps do
9: Update (via gradient ascent) \( \theta \) to increase \([13]\)
10: end for
11: end for
5.1 Learning the 2D optimal map with $L^2$ cost

In this section, we compare our method with the baseline W2-OT (Makkva et al., 2020). W2-OT is developed to estimate Wasserstein-2 cost and as such learns the map $\nabla g(\cdot)$ with $L^2$ cost. As mentioned in related work, it utilizes a min-max optimization structure as well. Figure 2 depicts qualitative difference of our algorithm on the Square-Ring and Mixture of ten Gaussian datasets. Each example is represented in a row. It is observed that both methods could learn a reasonable map, however, the samples generated by our method could cover the support completely and uniformly. $\nabla g\sharp\rho_a$ leaves some blank space in the Square-Ring example, and excessively concentrates on the ten components in Gaussian mixture example.

![Figure 2](image)

Figure 2: Qualitative results of our algorithm for learning 2D map with $L^2$ cost. The first two columns represent the marginals $\rho_a$ and $\rho_b$. The marginal distributions are uniformly supported on a square and a ring in the first row. In the second row, they are standard Gaussian and Gaussian mixture with 10 components respectively. The maps generated by our method are demonstrated in (c)-(d) columns and W2-OT maps are in (e)-(f) columns.

5.2 Learning with unequal dimensions

Our algorithm framework enjoys a distinguishing quality that it can learn the map from a lower dimension space $\mathbb{R}^{d_x}$ to a manifold in a higher dimension space $\mathbb{R}^{d_y}(d_x \leq d_y)$. In this scenario, we make the input dimension of neural network $T$ to be $d_x$ and output dimension to be $d_y$. In case the cost function $c(x,y)$ requires dimensions are $x$ and $y$ are equal dimensional, we patch zeros behind each sample $X \sim \rho_a$ and complement to a counterpart sample $\tilde{X} = |X;0|$, where dimension of 0 is $d_y - d_x$. And the targeted min-max problem is replaced by

$$\max_\theta \min_\eta \frac{1}{N} \sum_{k=1}^{N} c(\tilde{X}_k, T_\theta(\tilde{X}_k)) - f_\eta(T_\theta(\tilde{X}_k)) + f_\eta(Y_k).$$

In Figure 3, we conduct two experiments for $d_x = 1$ and $d_y = 2$. Similarly, each row is shown as an example. The incomplete ellipse is a 1D manifold and our algorithm is able to learn a symmetric map from $\mathcal{N}(0,1)$ towards it. The second row is when the support of $\rho_b$ is in a higher dimension manifold than $\rho_a$. In this case, our method pushforwards $\rho_a$ to samples that are attempting to fill the space of the ball, which is very similar to the peano curve.

5.3 Effect of different costs

Next we test our algorithm with more general cost functions. We compare the results on the same set of problems but with different choice of costs, and illustrate the effects of cost $c$.

5.3.1 Examples in 2D space

$L^p$ cost In Figure 1 we consider the Monge problem with the cost $c(x,y) = |x-y|^p$ on $\mathbb{R}^2$, we assume $p > 1$. In order to reflect the difference between $L^p$-OT problems with different $p$ values, we consider $\rho_a = \frac{1}{2}(N(\mu_1, \sigma^2 I) + N(\mu_2, \sigma^2 I))$ and $\rho_b = \frac{1}{2}(N(\nu_1, \sigma^2 I) + N(\nu_2, \sigma^2 I))$, with $\mu_1 = (-2,2), \mu_2 = (-6,-4), \nu_1 = (2,2), \nu_2 = (6,-4)$ and $\sigma = 0.1$. Notice that $|\mu_1 - \nu_1|^2 +
We also compute the same problem for \(N(0, 1)\), and \(\rho_b\) are uniformly distributed on an incomplete ellipse and a ball respectively.\\

\[ |\mu_2 - \nu_2|^2 < |\mu_1 - \nu_2|^2 + |\mu_2 - \nu_1|^2 \text{ and } |\mu_1 - \nu_1|^2 + |\mu_2 - \nu_2|^2 < |\mu_1 - \nu_1|^2 + |\mu_2 - \nu_2|^2, \]

this indicates the difference between the Monge maps of \(L^2\)-OT and \(L^6\)-OT problems. Such difference is captured by our numerical results shown in Figure 1 in the introduction.

**Decreasing function as the cost** We consider the cost function \(c(x, y) = \phi(|x - y|)\) with \(\phi\) as a monotonically decreasing function. We test our algorithm for a specific example \(\phi(s) = \frac{1}{s^2}\). In this example, we compute for the optimal Monge map from \(\rho_a\) to \(\rho_b\) with \(\rho_a\) as a uniform distribution on \(\Omega_a\) and \(\rho_b\) as a uniform distribution on \(\Omega_b\), where we define

\[ \Omega_a = \{ (x, x) \mid 6^2 \geq x_1^2 + x_2^2 \geq 4^2 \}, \quad \Omega_b = \{ (x, x) \mid 2^2 \geq x_1^2 + x_2^2 \geq 1^2 \}. \]

We also compute the same problem for \(L^2\) cost. Figure 4 shows the transported samples as well as the differences between two cost functions.

**Monge problem on sphere** For a given sphere \(S\) with radius \(R\), for any two points \(x, y \in S\), we define the distance \(d(x, y)\) as the length of the geodesic joining \(x\) and \(y\). Now for given \(\rho_a, \rho_b\) defined on \(S\), we consider solving the following Monge problem on \(S\)

\[
\min_{T, T_1 \rho_a = \rho_b} \int_S d(x, T(x)) \rho_a(x) \, dx. \tag{16}
\]

Such sphere OT problem can be transferred to an OT problem defined on angular domain \(D = [0, 2\pi] \times [0, \pi]\), to be more specific, we consider \((\theta, \phi) (\theta \in [0, 2\pi], \phi \in [0, \pi])\) as the azimuthal and polar angle of the spherical coordinates. For two points \(x = (R \sin \phi_1 \cos \theta_1, R \sin \phi_1 \sin \theta_1, R \cos \phi_1), y = (R \sin \phi_2 \cos \theta_2, R \sin \phi_2 \sin \theta_2, R \cos \phi_2)\) on \(S\), the geodesic distance

\[ d(x, y) = c((\theta_1, \phi_1), (\theta_2, \phi_2)) = R \cdot \arccos(\sin \phi_1 \sin \phi_2 \cos(\theta_2 - \theta_1) + \cos \phi_1 \cos \phi_2). \]

Denote the corresponding distribution of \(\rho_a, \rho_b\) on \(D\) as \(\hat{\rho}_a, \rho_b\), now (16) can also be formulated as

\[
\min_{T, T_1 \hat{\rho}_a = \rho_b} \int_D c((\theta, \phi), \hat{T}(\theta, \phi)) \hat{\rho}_a(\theta, \phi) \, d\theta d\phi. \tag{17}
\]
We set \( \hat{\rho}_a = U([0, 2\pi]) \otimes U([0, \frac{\pi}{2}]) \) and \( \hat{\rho}_b = U([0, 2\pi]) \otimes U([\frac{3\pi}{4}, \pi]) \). We apply our algorithm to solve (17) and then translate our computed Monge map back to the sphere \( S \) to obtain the following results.

Figure 5: Monge map from \( \rho_a \) to \( \rho_b \) on the sphere: (a) blue samples from \( \rho_a \) (corresponds to \( \hat{\rho}_a \)) and orange samples from \( \rho_b \) (corresponds to \( \hat{\rho}_b \)); (b) blue samples from \( \rho_a \), orange samples are obtained from \( T^*_\rho \rho_b \), grey curves are geodesics connecting each transporting pairs; (c) our computed Monge map maps blue ring (\( \phi = \frac{\pi}{8} \)) to the orange curve (ground truth is \( \phi = \frac{7}{8} \pi \)); (d) our computed Monge map maps blue ring (\( \phi = \frac{\pi}{4} \)) to the orange curve (ground truth is the southpole).

5.3.2 Examples in 256D space

**KL divergence vs \( L^2 \) cost** In this experiment, we study the digits transfer map from MNIST (\( \rho_a \), scaled to \( 16 \times 16 \) dimensional) to US Postal (USPS) (\( \rho_b \), \( 16 \times 16 \) dimensional) handwritten digits data sets. The USPS data is derived from a project on recognizing handwritten digits on envelopes, mentioned in Hastie et al. (2003). The MNIST dataset, one of the most famous in digit recognition, is created by LeCun et al. (2010). As we see from Figure 6, the style of MNIST digit number is thinner in (a) while the style of USPS digit number is larger and rounder in (b). We choose cost functions as \( \| \cdot \|_2^2 \) and KL divergence respectively. To apply KL divergence, we force \( L^1 \) norm of each sample is equal to one by a softmax normalization.

5.3.3 Examples in 784D space

**\( L^2 \) vs \( L^1 \) cost** In this experiment we choose MNIST as our data set (\( 28 \times 28 \) dimensional) and assume each handwritten digit from 0 to 9 follows a specific distribution respectively. Our task is to learn the Monge mapping between 0 and 1, 2 and 5, 3 and 7, 4 and 8, 6 and 9. In each round we treat 0, 2, 3, 4 and 6 as \( \rho_a \) and 1, 5, 7, 8, 9 as \( \rho_b \), respectively. We present generated digits that follow \( \rho_a \) and \( \rho_b \) in Figure 7, the learned map \( T \) pushforward the digit distributions to target ones. Here we set \( \frac{1}{2} \| \cdot \|_2^2 \) as our cost function, thus we are computing Wasserstein-2 distance. With the same training settings, we also show the result of choosing \( \| \cdot \|_1 \) as the cost function in Figure 8.

5.4 Learning the map between Gaussians with \( L^2 \) cost

In this section, we test our method on Gaussian marginal setting to investigate the quantitative performance. We follow the experiment setup exactly in W2GN (Korotin et al., 2021). The error is quantified as \( L^2 \)-UVP = 100 \cdot \frac{\| T - T^* \|_2^2 / \text{Var}(\rho_b)}{\| \rho_a \|} \%). The marginals \( \rho_a, \rho_b \) are two randomly generated centered Gaussian distributions. We refer to Korotin et al. (2021, Section 5.1, Section C.4)
for the performance of W2-OT and W2GN methods. In Figure 8, we rerun the experiment in each dimension for 5 times and report the error bar. When $d < 64$, the $L^2$-UVP is lower than $1\%$, which is on par with the performance of W2-OT and W2GN. And $L^2$-UVP in $d \geq 64$ is still bounded by $3\%$.

6 Conclusion

In this paper we present a novel method to compute Monge map between two given distributions with freely chosen cost functions. In particular, we consider applying Lagrange multipliers on MK problem, which leads to a saddle point problem. By further introducing neural networks into our optimization, we obtain a scalable algorithm that can handle most general costs and even the case where the dimensions of marginals are unequal. Our method not only computes sample based Wasserstein distance, but also provides optimal map. Our scheme is shown to be effective through a series of experiments with both low dimensional and high dimensional settings. However, the limitation of our work is that it’s more sensitive when dimension is less than 10 and it couldn’t catch the state-of-art methods in some of $L^2$ applications. It will become an useful tool for machine learning applications such as domain adaption that requires transforming data distributions. It will also be potentially used in areas outside machine learning, such as robotics. On the negative side, since the algorithm is a general tool to transform data distributions, it could be potentially used in many other activities involving maneuvering data distributions.
References

Altschuler, J., Niles-Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, pp. 1964–1974, 2017.

Amos, B., Xu, L., and Kolter, J. Sinkhorn distances: Lightspeed computation of optimal transport. In *International Conference on Machine Learning*, pp. 146–155, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. In *arXiv preprint arXiv:1701.07875*, 2017.

Benamou, J. and Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Benamou, J., Froese, B., and Oberman, A. Two numerical methods for the elliptic monge-ampere equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(4):737–758, 2010.

Chen, Y., Georgiou, T., and Pavon, M. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. In *Journal of Optimization Theory and Applications*, 2016.

Chen, Y., Georgiou, T., and Pavon, M. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2020.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Neural Information Processing Systems*, 2013.

Desser, A., Papadakis, N., and Rouas, J. Regularized optimal transport and the rot mover’s distance. *The Journal of Machine Learning Research*, 19(1):590–642, 2018.

Dukler, Y., Li, W., Tong Lin, A., and Montúfar, G. Wasserstein of wasserstein loss for learning generative models. 2019.

Fan, J., Taghvaei, A., and Chen, Y. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020.

Flamary, R., Courty, N., Rakotomamonjy, A., and Tuia, D. Optimal transport with laplacian regularization. In *Advances in Neural Information Processing Systems, Workshop on Optimal Transport and Machine Learning*, 2014.

Gangbo, W., Li, W., Osher, S., and Puthawala, M. Unnormalized optimal transport. *Journal of Computational Physics*, 399:108940, 2019.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pp. 3440–3448, 2016.

Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with sinkhorn divergences. *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, 2018.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Neural Information Processing Systems*, 2017.

Haasler, I., Singh, R., Zhang, Q., Karlsson, J., and Chen, Y. Multi-marginal optimal transport and probabilistic graphical models. In *arXiv preprint arXiv:2006.14113*, 2020.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer, Berlin/Heidelberg, 2003. ISBN 0387952845.

Hütter, J.-C. and Rigollet, P. Minimax estimation of smooth optimal transport maps, 2020.
Inoue, D., Ito, Y., and Yoshida, H. Optimal transport-based coverage control for swarm robot systems: Generalization of the voronoi tessellation-based method. *IEEE Control Systems Letters*, 5(4):1483–1488, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.

Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=bEoxzW_EXsa

Krishnan, V. and Martínez, S. Distributed optimal transport for the deployment of swarms. *IEEE Conference on Decision and Control (CDC)*, pp. 4583–4588, 2018.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. In http://yann.lecun.com/exdb/mnist/, 2010.

Li, R., Ye, X., Zhou, H., and Zha, H. Learning to match via inverse optimal transport. In *J. Mach. Learn. Res.*, pp. 20, pp.80–1, 2019.

Li, W., Ryu, E., Osher, S., Yin, W., and Gangbo, W. A parallel method for earth mover’s distance. *Journal of Scientific Computing*, 75(1):182–197, 2018.

Liu, S., Ma, S., Chen, Y., Zha, H., and Zhou, H. Learning high dimensional wasserstein geodesics. In *arXiv preprint arXiv:2102.02992*, 2021.

Ma, S., Liu, S., Zha, H., and Zhou, H. Learning stochastic behaviour of aggregate data. In *arXiv preprint arXiv:2002.03513*, 2020.

Makkuva, A. and Taghvaei, A., Oh, S., and Lee, J. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681, 2020.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Mroueh, Y., Li, C., Sercu, T., Raj, A., and Cheng, Y. Sobolev gan. In *arXiv preprint arXiv:1711.04894*, 2020.

Muzellec, B., Nock, R., Patrini, G., and Nielsen, F. Tsallis regularized optimal transport and ecological inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.

Seguy, V., Damodaran, B., Flamary, R., Courty, N., R., A., and Blondel, M. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *arXiv preprint arXiv:1711.01558*, pp. 3440–3448, 2017.

Tong Lin, A., Li, W., Osher, S., and Montúfar, G. Wasserstein proximal of gans. 2018.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Xie, Y., Chen, M., Jiang, H., Zhao, T., and Zha, H. On scalable and efficient computation of large scale optimal transport. In *International Conference on Machine Learning*, pp. 6882–6892, 2019.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pp. 433–453, 2020.

Yang, L. and Karniadakis, G. Potential flow generator with $l_2$ optimal transport regularity for generative models. In *arXiv preprint arXiv:1908.11462*, 2019.
A Definitions

**Definition 2** (Superdifferentiability). For function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we say $f$ is superdifferentiable at $x$, if there exists $p \in \mathbb{R}^n$, such that

$$f(z) \geq f(x) + \langle p, z - x \rangle + o(|z - x|).$$

**Definition 3** (Locally Lipschitz). Let $U \subseteq \mathbb{R}^n$ be open and let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be given. Then

1. $f$ is Lipschitz if there exists $L < \infty$ such that
   $$\forall x, z \in \mathbb{R}^n, \quad |f(z) - f(x)| \leq L|x - z|.$$

2. $f$ is said to be locally Lipschitz if for any $x_0 \in \mathbb{R}^n$, there is a neighbourhood $O$ of $x_0$ in which $f$ is Lipschitz.

B Proof of Theorem 3

**Theorem 3** (Consistency). Suppose the optimal solution to (10) is $(T_*, f_*)$, then $T_*$ is the optimal Monge map to the problem (9) and $f_* = \phi_*$, where $\phi_*$ is the optimal solution to (8).

*Proof of Theorem 3* According to (12), we are able to tell that the optimal solution $f_*$ equals $\phi_*$. Furthermore, at the optimal point $(T_*, f_*)$, we have

$$T_*\rho_a = \rho_b, \quad T_*(x) \in \arg\max_{\xi \in \mathbb{R}^n} \{f_*(\xi) - c(x, \xi)\}.$$

The second equation leads to

$$f_*^{-}(x) = f_*(T_*(x)) - c(x, T_*(x)).$$

Then we have

$$\int_{\mathbb{R}^n} c(x, T_*(x))\rho_a(x) \, dx = \int_{\mathbb{R}^m} f_*(T_*(x))\rho_a(x) \, dx - \int_{\mathbb{R}^n} f_*^{-}(x)\rho_a(x) \, dx$$

$$= \int_{\mathbb{R}^m} f_*(y)\rho_b(y) \, dy - \int_{\mathbb{R}^n} f_*^{-}(x)\rho_a(x) \, dx$$

$$\leq \int_{\mathbb{R}^n \times \mathbb{R}^m} [f_*(y) - f_*^{-}(x)]d\pi(x, y) \leq \int_{\mathbb{R}^n \times \mathbb{R}^m} c(x, y)d\pi(x, y)$$

for any $\pi \in \Pi(\rho_a, \rho_b)$. Here the second equality is due to $T_*\rho_a = \rho_b$, the last inequality is due to the definition of $f_*^{-}(x) = \sup_y \{f_*(y) - c(x, y)\}$.

As a result, $T_*$ solves (9) and thus is the unique optimal Monge map.

C Proof of Theorem 4

**Theorem 4** (Posterior Error Analysis via Duality Gaps). We consider the Monge problem from $\mathbb{R}^n$ to $\mathbb{R}^d$ with the cost function $c \in C^2(\mathbb{R}^n \times \mathbb{R}^n)$ satisfying the conditions mentioned in Theorem 3. Despite that we further assume $c$ satisfies:

$$\partial_{xy}c(x, y), \text{ as an } n \times n \text{ matrix, is invertible and self-adjoint.}$$

(18)

$$\partial_{yy}c(x, y) \text{ is independent of } x; \text{ i.e. } \partial_{yy}c(x, y) = M(y) \text{ with } M(y) \text{ a matrix function of } y. \quad (19)$$

We further denote

$$\sigma(x, y) = \sigma_{\min}(\partial_{xy}c(x, y))$$

(20)

as the minimum singular value of matrix $\partial_{xy}c(x, y)$, since the matrix is invertible, $\sigma(x, y) > 0$ for any $x, y \in \mathbb{R}^n$.

Assume that $f \in C^2(\mathbb{R}^d)$ is $c$-convex function and there exists $\varphi \in C^2(\mathbb{R}^d)$ such that $f(y) = \inf_x \{\varphi(x) + c(x, y)\}$. For any

$$\hat{x}_y \in \arg\min \{\varphi(x) + c(x, y)\} \neq \emptyset,$$
we further assume there exists function $\lambda(\cdot) > 0$ such that
\[
\lambda(y)I_n \succeq \nabla^2_x (\varphi(x) + c(x, y)) \bigg|_{x=x^*} > O_n
\]

If we denote the duality gaps
\[
E_1(T, f) = \mathcal{L}(T, f) - \inf_{\tilde{T}} \mathcal{L}(\tilde{T}, f)
\]
\[
E_2(f) = \sup_{\tilde{f}} \inf_{\tilde{T}} \mathcal{L}(\tilde{f}, \tilde{T}) - \inf_{\tilde{T}} \mathcal{L}(\tilde{T}, f)
\]
Suppose $E_1(T) \leq \epsilon_1$, $E_2(T, f) \leq \epsilon_2$.

If we denote $T_*$ as the optimal Monge map of the OT problem (9), then there exists a strict positive weight function $\beta(\cdot) > 0$ (defined in (32)) depending on $c$, $T_*$, $f$ and $\varphi$, such that the weighted $L^2$ error between computed map $T$ and optimal map $T_*$ is upper bounded by
\[
\|T - T_*\|_{L^2(\beta\rho_n)} \leq \sqrt{2(\epsilon_1 + \epsilon_2)}.
\]

**Lemma 1.** Suppose $n \times n$ matrix $A$ is self-adjoint, i.e. $A = A^T$, with minimum singular value $\sigma_{\min}(A) > 0$. Also assume $n \times n$ matrix $H$ is self-adjoint and satisfies $\lambda I_n \succeq H \succ O_n$. Then $AH^{-1}A \succeq \frac{\sigma_{\min}(A)^2}{\lambda} I_n$.

**Proof of Lemma 1.** One can first verify that $H^{-1} \succeq \frac{1}{\lambda} I_n$ by diagonalizing $H^{-1}$. To prove this lemma, we only need to verify that for arbitrary $v \in \mathbb{R}^n$,
\[
v^T AH^{-1} A v = (Av)^T H^{-1} Av \geq \frac{|Av|^2}{\lambda} \geq \frac{\sigma_{\min}(A)^2}{\lambda} |v|^2
\]
Thus $AH^{-1}A - \frac{\sigma_{\min}(A)^2}{\lambda} I_n$ is non-negative definite. 

The following lemma is crucial for proving our results, it analyzes the concavity of the target function $f(\cdot) - c(\cdot, \cdot)$ with $f$ $c$-concave.

**Lemma 2** (Concavity of $f(\cdot) - c(x, \cdot)$ as $f$ $c$-concave). Suppose the cost function $c(x, y)$ and $f$ satisfy the conditions mentioned in Theorem 4. Denote the function $\Psi_x(y) = f(y) - c(x, y)$, keep all notations defined in Theorem 4 then we have
\[
\nabla^2 \Psi_x(y) \preceq -\frac{\sigma(x, y)^2}{\lambda(y)} I_n.
\]

**Proof of Lemma 2** We first compute
\[
\nabla^2 \Psi_x(y) = \nabla^2 f(y) - \partial^2_{xy} c(x, y)
\]
We can compute the Hessian $\nabla^2 f(y)$ as:
\[
\nabla^2 f(y) = D_y (D_y \inf_x \{\varphi(x) + c(x, y)\}) = D_y (\partial_y c(\hat{x}(y), y)) = \partial^2_{x} c(\hat{x}(y), y) \nabla \hat{x}(y) + \partial^2_{yy} c(\hat{x}(y), y).
\]
The second equality is due to the envelope theorem.

Notice that $f$ is $c$-convex, thus, there exists $\varphi$ such that $f(y) = \inf_x \{\varphi(x) + c(x, y)\}$. Let us also denote $\Phi(x, y) = \varphi(x) + c(x, y)$.

Now for a fixed $y \in \mathbb{R}^n$, We pick one
\[
\hat{x}_y = \arg\min_x \{\varphi(x) + c(x, y)\}
\]
Since we assumed that $\varphi \in C^2(\mathbb{R}^n)$ and $c \in C^2(\mathbb{R}^n \times \mathbb{R}^n)$, we have
\[
\partial_x \Phi(\hat{x}_y, y) = \nabla \varphi(\hat{x}_y) + \partial_x c(\hat{x}_y, y) = 0
\]
At the same time, since \( \hat{x}_y \) is the minimum point of the \( C^2 \) function \( \Phi(\cdot, y) \), in condition (18), we have made positive definite assumption on the Hessian of \( \Phi(\cdot, y) \) at \( \hat{x}_y \).

\[
\partial_{xx}^2 \Phi(\hat{x}_y, y) = \nabla_{xx}^2 (\varphi(x) + c(x, y)) \bigg|_{x=\hat{x}_y} = \nabla_{xx}^2 \varphi(\hat{x}_y) + \partial_{xx}^2 c(\hat{x}_y, y) \succ 0.
\]

Since \( \partial_{xx}^2 \Phi(\hat{x}_y, y) \) is positive definite, it is also invertible. We can now apply the implicit function theorem to show that the equation \( \partial_x \Phi(x, y) = 0 \) determines an implicit function \( \hat{x}(\cdot) \), which satisfies \( \hat{x}(y) = \hat{x}_y \) in a small neighbourhood \( U \subset \mathbb{R}^n \) containing \( y \). Furthermore, one can show that \( \hat{x}(\cdot) \) is continuously differentiable at \( y \). We will denote \( \hat{x}_y \) as \( \hat{x}(y) \) in our following discussion.

Now differentiating (24) with respect to \( y \) yields

\[
\partial_{xx}^2 \Phi(\hat{x}(y), y) \nabla \hat{x}(y) + \partial_{xx}^2 c(\hat{x}(y), y) = 0 \tag{25}
\]

On one hand, (25) tells us

\[
\nabla \hat{x}(y) = -\partial_{xx} \Phi(\hat{x}(y), y)^{-1} \partial_{xy} c(\hat{x}(y), y). \tag{26}
\]

On the other hand, notice that \( c \in C^2(\mathbb{R}^n \times \mathbb{R}^n) \), thus \( \partial_{xy} c = \partial_{yx} c \). By (25), we have

\[
\partial_{yy} \Phi(\hat{x}(y), y) \nabla \hat{x}(y) = -\partial_{xx}^2 \Phi(\hat{x}(y), y) \nabla \hat{x}(y) \nabla \hat{x}(y)
= -\left( \nabla^2 \varphi(\hat{x}(y)) + \partial_{xx}^2 c(\hat{x}(y), y) \right) \nabla \hat{x}(y) \nabla \hat{x}(y) \tag{27}
\]

Plugging (27) into (23), also recall (22), this yields

\[
\nabla^2 \Psi_x(y) = -(\nabla^2 \varphi(\hat{x}(y)) + \partial_{xx}^2 c(\hat{x}(y), y)) \nabla \hat{x}(y) \nabla \hat{x}(y) + \partial_{yy}^2 c(\hat{x}(y), y) - \partial_{yy}^2 c(x, y)
\]

Now by (18), since \( \partial_{yy} c(x, y) \) is independent of \( x \), \( \partial_{yy}^2 c(\hat{x}(y), y) - \partial_{yy}^2 c(x, y) = 0 \). Thus \( \nabla^2 \Psi_x(y) \) equals

\[
\nabla^2 \Psi_x(y) = -(\nabla^2 \varphi(\hat{x}(y)) + \partial_{xx}^2 c(\hat{x}(y), y)) \nabla \hat{x}(y) \nabla \hat{x}(y)
= -\partial_{xx}^2 \Phi(\hat{x}(y), y) \nabla \hat{x}(y) \nabla \hat{x}(y) \tag{28}
\]

To further simplify (28), recall (26), we have

\[
\nabla^2 \Psi_x(y) = -\partial_{xy} c(\hat{x}(y), y) \partial_{xx} \Phi(\hat{x}(y), y)^{-1} \partial_{xy} c(\hat{x}(y), y).
\]

By (21), we have

\[
\lambda(y) I_n \succeq \partial_{xx} \Phi(\hat{x}(y), y) \succ O_n
\]

By condition (18), \( \partial_{xy} c \) is self-adjoint, and (20) tells \( \sigma_{\min}(\partial_{xy} c(x, y)) = \sigma(x, y) \). Now applying lemma [1] we have

\[
\nabla^2 \Psi_x(y) \preceq -\frac{\sigma(x, y)^2}{\lambda(y)} I_n.
\]

\[\square\]

Now we can prove main result in Theorem [4].

**Proof of Theorem [4].** In this proof, we denote \( f \) as \( \int_{\mathbb{R}^d} \) for simplicity.

We first recall

\[
\mathcal{L}(T, f) = \int f(y) \rho_0(y) \, dy - \int (f(T(x)) - c(x, T(x))) \rho_0(x) \, dx,
\]

also recall definition (22), \( f^{\min}(x) = \sup_y \{ f(y) - c(x, y) \} \), we can write

\[
\mathcal{E}_1(T, f) = -\int [f(T(x)) - c(x, T(x))] \rho_0 \, dx + \inf_T \left\{ \int [f(\bar{T}(x)) - c(x, \bar{T}(x))] \rho_0 \, dx \right\}
= \int [f^{\min}(x) - (f(T(x)) - c(x, T(x)))] \rho_0 \, dx
\]

14
We denote 
\[ T_f(x) = \text{argmax}_y \{ f(y) - c(x, y) \} = \text{argmax}_y \{ \Psi_x(y) \}, \]
then we have 
\[ \nabla \Psi_x(T_f(x)) = 0. \] (29)

On the other hand, one can write:
\[ \mathcal{E}_1(T, f) = \int [(f(T_f(x)) - c(x, T_f(x))) - (f(T(x)) - c(x, T(x)))] \]
\[ = \int [\Psi_x(T_f(x)) - \Psi_x(T(x))] \rho_a(x) \, dx \]
\[ \text{For a fixed } x, \text{ since } \Psi_x(\cdot) \in C^2(\mathbb{R}^n), \text{ then} \]
\[ \Psi_x(T(x)) - \Psi_x(T_f(x)) = \nabla \Psi_x(T_f(x))(T(x) - T_f(x)) + \frac{1}{2}(T(x) - T_f(x))^T \nabla^2 \Psi_x(\eta(x))(T(x) - T_f(x)) \]
with \( \eta(x) = (1 - \theta_x)T(x) + \theta_x T_f(x) \) for certain \( \theta_x \in (0, 1) \). By (29) and Lemma 2, we have 
\[ \Psi_x(T(x)) - \Psi_x(T_f(x)) \leq -\frac{\sigma(x, \eta(x))^2}{2\lambda(\eta(x))} |T(x) - T_f(x)|^2. \]

Thus we have:
\[ \mathcal{E}_1(T, f) = \int [\Psi_x(T_f(x)) - \Psi_x(T(x))] \rho_a(x) \, dx \geq \int \frac{\sigma(x, \eta(x))^2}{2\lambda(\eta(x))} |T(x) - T_f(x)|^2 \rho_a(x) \, dx \] (30)

On the other hand, let us denote the optimal Monge map from \( \rho_a \) to \( \rho_b \) as \( T_* \), by Kontorovich duality, we have 
\[ \sup_f \inf_T \mathcal{L}(T, f) = \inf_{T, T_*} \int c(x, T(x)) \rho_a \, dx = \int c(x, T_*(x)) \rho_a \, dx \]
Thus we have 
\[ \mathcal{E}_2(f) = \int c(x, T_*(x)) \rho_a \, dx - \left( \int f(y) \rho_b \, dy - \int f^{c^{-}}(x) \rho_a \, dx \right) \]
\[ = \int c(x, T_*(x)) \rho_a \, dx - \left( \int f(T_*(x)) \rho_a \, dx - \int f^{c^{-}}(x) \rho_a \, dx \right) \]
\[ = \int [f^{c^{-}}(x) - (f(T_*(x)) - c(x, T_*(x)))] \rho_a \, dx \]

Similar to the previous treatment, we have 
\[ \mathcal{E}_2(f) = \int [\Psi_x(T_f(x)) - \Psi_x(T_*(x))] \rho_a(x) \, dx \]
Apply similar analysis as before, we will also have 
\[ \mathcal{E}_2(f) \geq \int \frac{\sigma(x, \xi(x))^2}{2\lambda(\xi(x))} |T_*(x) - T_f(x)|^2 \rho_a(x) \, dx \] (31)
with \( \xi(x) = (1 - \tau_x)T_*(x) + \tau_x T_f(x) \) for certain \( \tau_x \in (0, 1) \). 

Now we set 
\[ \beta(x) = \min \left\{ \frac{\sigma(x, \eta(x))}{2\lambda(\eta(x))}, \frac{\sigma(x, \xi(x))}{2\lambda(\xi(x))} \right\}, \] (32)
combining (30) and (31), we obtain 
\[ \epsilon_1 + \epsilon_2 \geq \mathcal{E}_1(T, f) + \mathcal{E}_2(f) \]
\[ \geq \int \beta(x)(|T(x) - T_f(x)|^2 + |T_*(x) - T_f(x)|^2) \rho_a \, dx \]
\[ \geq \int \frac{\beta(x)}{2} |T(x) - T_*(x)|^2 \rho_a \, dx \]
This leads to \( \| T - T_* \|_{L^2(\rho_a)} \leq \sqrt{2(\epsilon_1 + \epsilon_2)}. \)
D Experiment details

In terms of training, the cost functions we choose, structures of neural networks and details of training process will be introduced here. We ran experiments using a NVIDIA RTX 2080 GPU for the experiments in Section 5.1, Section 5.2, Section 5.3.2 and NVIDIA RTX 3090 GPU for Section 5.4. And we ran experiments on CPU for all the other experiments. If we train on GPU cards with inner iteration \(K_3 = 4\), the training time is about 10 minutes for 2D examples.

For general settings, for all experiments we use the Adam optimizer ([Kingma & Ba 2014](#)) and vanilla feedforward networks unless specified. The activation functions are all PReLU unless specified.

D.1 Normal 2D cases

For our method, the networks \(T_\theta\) and \(f_\eta\) each has 5 layers and 32 hidden neurons. The batch size \(N = 100\). \(K_1 = 4\), \(K_2 = 1\). The learning rate is \(10^{-3}\). The number of iterations \(K = 12000\).

For the W2-OT, the setup is all the same except \(K_1 = 10\). The number of iterations \(K = 12000\).

D.2 Unequal dimensions

For the incomplete ellipse example, the networks \(T_\theta\) and \(f_\eta\) each has 5 layers with 10 hidden neurons. The batch size \(N = 100\). \(K_1 = 6\), \(K_2 = 1\). The learning rate is \(10^{-5}\). The number of iterations \(K = 12000\).

For the ball example, the networks \(T_\theta\) has 12 layers and \(f_\eta\) has 5 layers. Both of them have 32 hidden neurons. The batch size \(N = 100\). \(K_1 = 4\), \(K_2 = 1\). The learning rate is \(10^{-3}\). The number of iterations \(K = 15000\).

D.3 Effects of different cost

D.3.1 Examples in 2D space

- **\(L^p\) cost**  In this example, we set \(T_\theta(x) = x + F_\theta(x)\) and optimize over \(\theta\). For either \(L^2\) or \(L^6\) case we set both \(F_\theta\) and the Lagrange multiplier \(f_\eta\) as six layers fully connected neural networks, with PReLU activation functions, each layer has 36 nodes. The training batch size \(N = 1200\). We set \(K = 2000\), \(K_1 = 8\), \(K_2 = 4\).

- **Decreasing cost function**  In this example, we set \(T_\theta(x) = x + F_\theta(x)\) and optimize over \(\theta\). For either \(\frac{1}{|x|^p}\) or \(|x - y|^2\) case we set both \(F_\theta\) and the Lagrange multiplier \(f_\eta\) as six layers fully connected neural networks, with PReLU and Tanh activation functions respectively, each layer has 36 nodes. The training batch size \(N = 2000\). We set \(K = 4000\), \(K_1 = 8\), \(K_2 = 4\). We choose rather small learning rate in this example to avoid gradient blow up, we set \(0.5 \times 10^{-5}\) as the learning rate for \(\theta\) and \(10^{-5}\) as the learning rate for \(\eta\).

- **On sphere**  In this example, we set \(T_\theta(x) = x + F_\theta(x)\) and optimize over \(\theta\). We set both \(F_\theta\) and the Lagrange multiplier \(f_\eta\) as six layers fully connected neural networks, with PReLU activation functions, each layer has 8 nodes The training batch size \(N = 200\). We set \(K = 4000\), \(K_1 = 8\), \(K_2 = 4\). We choose rather small learning rate in this example to avoid gradient blow up, we set \(0.5 \times 10^{-5}\) as the learning rate for \(\theta\) and \(10^{-5}\) as the learning rate for \(\eta\).

D.3.2 Examples in 256D space

For both \(L^2\) and KL divergence experiments, the network \(T_\theta\) has 4 layers and \(f_\eta\) has 5 layers. Both of them have 512 hidden neurons. The batch size \(N = 100\). \(K_1 = 10\), \(K_2 = 1\). The learning rate is \(10^{-4}\). The number of iterations \(K = 10000\). The running time is about 30 minutes.

D.3.3 Examples in 784D space

Here for either \(L^1\) or \(L^2\) case we set both mapping function and the Lagrange multiplier as six layers fully connected neural networks, with ReLU and Tanh activation functions respectively, each layer has 512 nodes, the learning rate is \(10^{-4}\). The batch size is 500 and we train the model with the same general techniques mentioned before, specially we set \(K_1 = 8\) and \(K_2 = 1\). The algorithm is stable
and it converges very fast. For $L^1$ the average outer iteration to get a convergent result is 8000, total computation time lasts for 3 hours, while for $L^2$ the average iteration for convergence is 4000 and total computation time is around 1 to 2 hours. From our experiments it seems that $L^1$ based model is more sensitive to the initial weights of the networks.

D.4 Map between Gaussians

The networks $T_\theta$ and $f_\eta$ each has 5 layers. Both of them have $[10, 16, 32, 100, 160]$ hidden neurons for experiments in dimension $[2, 8, 16, 32, 64, 128]$ respectively. The batch size $N = 100$. $K_1 = 6$, $K_2 = 1$. The learning rate is $10^{-4}$. The number of iterations $K = 24000$. 