Learning Cost Functions for Optimal Transport

Haodong Sun ∗ Haomin Zhou † Hongyuan Zha ‡ Xiaojing Ye §

Abstract

Learning the cost function for optimal transport from observed transport plan or its samples has been cast as a bi-level optimization problem. In this paper, we derive an unconstrained convex optimization formulation for the problem which can be further augmented by any customizable regularization. This novel framework avoids repeatedly solving a forward optimal transport problem in each iteration which has been a thorny computational bottleneck for the bi-level optimization approach. To validate the effectiveness of this framework, we develop two numerical algorithms, one is a fast matrix scaling method based on the Sinkhorn-Knopp algorithm for the discrete case, and the other is a supervised learning algorithm that realizes the cost function as a deep neural network in the continuous case. Numerical results demonstrate promising efficiency and accuracy advantages of the proposed algorithms over existing state of the art methods.

1 Introduction

Optimal transport (OT) is an important mathematical subject that links a wide range of concepts, including differential geometry, partial differential equations, optimization, probability theory and more recently machine learning applications [4 26 20]. Let X and Y be two measure spaces (e.g., Euclidean spaces), and \( c(x, y) : Z := X \times Y \to \mathbb{R}^+ \) the cost function that quantifies the effort of moving one unit of mass from location \( x \) to location \( y \). Then, given any two probability distributions \( \mu \) and \( \nu \) on \( X \) and \( Y \), respectively, the optimal transport problem aims at finding the joint distribution \( \pi^* \) on \( Z \) that minimizes the total cost, i.e., \( \pi^* \) solves the following constrained convex minimization problem:

\[
\min_{\pi} \left\{ \int_Z c(x, y)\pi(x, y) \, dx \, dy \ \middle| \ \pi \in \Pi(\mu, \nu) \right\},
\]

(1)

where the constraint set \( \Pi(\mu, \nu) \) is given by

\[
\Pi(\mu, \nu) := \left\{ \pi : Z \to \mathbb{R}^+ \ \middle| \ \int_Y \pi \, dy = \mu, \int_X \pi \, dx = \nu \right\}
\]

(2)

and \( \pi^* \) is called the optimal transport plan between \( \mu \) and \( \nu \) under \( c \). The optimal function value of (1) is also called the earth mover’s distance (EMD) between the two distributions \( \mu \) and \( \nu \), suggesting the minimal total effort to move a pile of earth shaped as \( \mu \) to form the pile shaped as \( \nu \). The discrete counterpart of (1) can also be defined, where \( c \) reduces to a cost matrix in \( \mathbb{R}^{m \times n} \), and \( \mu \in \Delta^{m-1} := \{ \mu \in \mathbb{R}^m \mid \mu \geq 0, \sum_{i=1}^m \mu_i = 1 \} \) and \( \nu \in \Delta^{n-1} \) become two probability vectors, respectively. Then the discrete OT problem can be written as a linear program:

\[
\min_{\pi \in \mathbb{R}^{m \times n}} \left\{ \langle c, \pi \rangle \mid \pi \geq 0, \ \pi 1_n = \nu, \ \pi^\top 1_m = \nu \right\},
\]

(3)

where \( \langle c, \pi \rangle := \text{tr}(c^\top \pi) = \sum_{i,j} c_{ij} \pi_{ij} \) is the discretized total cost under \( \pi \), and \( 1_n := [1, \ldots, 1]^\top \in \mathbb{R}^n \). Note that a direct discretization of (1) can yield a computationally intractable problem (3): \( m \) represents the...
number of grid points (or bins) for discretizing the space $X$, and hence they grow exponentially fast in the dimensionality $d$ of $X$, e.g., $X = \mathbb{R}^d$ or $X = [0, 1]^d$. Similar for $n$ and $Y$.

The past two decades have witnessed great advancements in the theory of OT. Moreover, OT has received significant interests in the machine learning community in the past few years, and has been employed in a variety of applications including domain adaption \cite{domain_adaptation}, regularization \cite{regularization}, parameter estimation \cite{parameter_estimation}, dictionary learning \cite{dictionary_learning}, Kalman filtering \cite{kalman_filtering}, image processing \cite{image_processing}, graph partition \cite{graph_partition}, information geometry \cite{information_geometry}, among many others.

**Motivation and Approach.** In this paper, we consider an inverse problem of OT — learning the cost function $c$ when given certain information about $\pi^*$. As we can see, $c$ in OT \cite{OT} plays the paramount role in shaping the optimal transport plan. It is the main parameter defining the EMD between probability distributions. In many existing applications of OT, the cost is simply handcrafted, often chosen as the Euclidean distance such as $c(x, y) = \|x - y\|$. However, these manually chosen cost functions may not be able to capture the underlying structure and feature of the data, and hence the induced EMD and optimal transport plan can be severely biased, which calls for the development of a general framework for solving the inverse OT problem.

We propose a novel inverse OT framework to learn the cost functions from the observed transport plan, and develop efficient numerical algorithms for inverse OT in both discrete and continuous settings. For the discrete setting, the transport plan $\hat{\pi}$ is essentially a nonnegative matrix, which can be directly given or easily estimated from samples by binning. In the continuous case, however, $\hat{\pi}$ is only given by its i.i.d. samples of form $(x, y) \sim \hat{\pi}$. Specifically, we focus on entropy regularized OT formulation, and propose a variational problem of learning the cost function such that the induced optimal transport plan is close to the observed transport plan or its samples. This variational problem yields a bi-level optimization problem, which can be challenging to solve in general. However, surprisingly, by leveraging the dual form of entropy regularized OT, we manage to show that this bi-level optimization can be reformulated as an unconstrained and convex problem in the cost function before adding any customizable regularization in Section 3. We extend the discussion and generalization of this framework to other problems in Supplementary Material A.

Our proposed novel framework makes it possible to develop a fast matrix scaling algorithm in the discrete setting. A significant advantage of our framework over existing ones is that we can avoid solving a forward optimal transport problem in each iteration which is required by the bi-level optimization formulation. As a consequence, the computational complexity for the inverse OT problem is comparable to that of the Sinkhorn algorithm for solving a single forward OT problem. For the continuous case, we propose a deep learning approach that parametrizes the cost function as deep neural network, which is completely mesh-free and has great potential for handling high dimensional cases, e.g., when $X$ and $Y$, as Euclidean spaces, are high dimensional.

**Highlights of our framework.** Our proposed framework targets at the cost learning problem in OT. Compared to the existing approaches, which all aim at recovering the cost function and exploiting its properties for optimal transport but vary in specific problem formulations and applications domains, our framework features the following properties: (i) Existing methods formulate the cost learning as bi-level optimization or its variations, which require solving forward OT problem repeatedly for hundreds and even thousands of times to recover the cost. In contrast, we show that our inverse OT formulation can be converted to a standard convex optimization with customizable regularization, the complexity of which is comparable to that of one forward OT; (ii) In contrast to existing work, we provide detailed characterization of the solution and its uniqueness in our inverse OT formulation. We show cases where our framework does not suffer the common ambiguity issue of underdetermined inverse problems and can recover the desired ground truth cost robustly; (iii) Existing work are all carried out in the discrete setting, which essentially renders a cost matrix to be learned. Our framework can be applied to both discrete and continuous settings. To the best of our knowledge, our work is the first in the literature that tackles cost function learning in continuous case, which enables the application of OT with adaptive cost in problems involving high-dimensional data.

**Presentation outline.** The remainder of this paper is organized as follows. We first provide an overview of existing cost learning approaches in OT and the relations to other metric learning problems in Section 2. In Section 3, we propose an inverse optimal transport approach for cost function learning, and derive a novel framework based on the dual of the inverse OT formulation. In Section 4, we develop two prototype
algorithms to recover the cost matrix (function) in the discrete (continuous) setting, and discuss their properties and variations. We validate the proposed framework by several numerical experiments in Section 5. Section 6 concludes this paper.

2 Related Work

Cost learning for OT. The problem of cost learning for optimal transport has received considerable attention in the past few years. In [12, 13], the cost matrix is parametrized as a bilinear function of the feature vectors of the two sides in optimal transport. The parameter of the bilinear function, i.e., the interaction matrix, is recovered from the observed matchings, which are hypothesized to be samples drawn from the optimal transport plan that maximizes the total social surpluses. The interaction matrix quantifies coupling surplus that are important in the study of quantitative economics. In [17], a primal-dual matrix learning algorithm is proposed to allow more flexible parametrization of the cost matrix and also takes into account inaccurate marginal information for robust learning. In [8], a set of distributions are given where each pair is also associated with a weight coefficient, and the cost matrix is learned by minimizing the weighted sum of the earth mover’s distances between these pairs induced by this cost. Given class labels of documents which are represented as histograms of words, the cost matrix is parametrized as Mahalanobis distance between feature vectors of the words and learned such that the induced earth mover’s distance between similar documents are small [15]. In [27], the cost matrix is learned such that the induced earth mover’s distances between histograms labeled as similar are separated from those between dissimilar histograms, which mimics the widely used metric learning setup. In [30], the cost matrix is induced by a kernel mapping, which is jointly learned with a feature-to-label mapping in a label distribution learning framework. This work is extended to a multi-modal, multi-instance, and multi-label learning problem in a follow-up work [29]. In [28], the cost matrix is parameterized as the exponential of negative squared distance between features, where the feature map is learned such that the induced EMD is small for those with same labels and large otherwise. A cost matrix learning method based on the Metropolis-Hasting sampling algorithm is proposed in [25]. In [18], the Sinkhorn iteration is unrolled into a deep neural network with cost matrix as unknown parameter, which is then trained using given side information. The present work targets at the cost learning problem for OT as in the aforementioned ones, but contrasts favorably to them as highlighted in Section 1.

Relation to general metric learning. The aforementioned methods and the work presented in this paper aim at learning the cost matrix/function, which is related but different from the standard metric learning [5] in machine learning. In standard metric learning, the goal is to directly learn the distance that quantifies the similarity between features or data points given in the samples. In contrast, the learning problem in inverse OT aims at recovering the cost function (also known as the ground metric) that induces the earth mover’s distance (or more generally the Wasserstein distance) and optimal transport plan, optimal couplings, or optimal matchings exhibited by the data. In inverse OT, we only observe the couplings/matchings which are not labeled as similar or not. Hence, we cannot directly assess the distance or cost between features. Instead, we need to learn the cost based on the relative frequency of the matchings in the observed data, which is a compounded effect of the cost function and the intra-population competitions. Moreover, the cost function is critical to reveal the underlying mechanism of optimal transport and matchings, and can be used to predict or recommend optimal matchings given new but different marginal distributions [12, 13, 17].

Relation to Riemann distance learning on the probability manifolds. The cost function learning problem for optimal transport is also related to Riemannian metric learning on probability simplex. In [9, 16], the Riemannian metric, i.e., distances between probability distributions or histograms on the manifold of probability simplex, is directly learned. In contrast, the goal in this work is to learn the cost function that reveals the interaction between features, which can also automatically induce a metric on the probability simplex if the cost function satisfies proper conditions. Moreover, the learned cost function from inverse OT can be used to provide insights of observed matchings and generate interpretable predictions on new data associations, which are extremely important and useful in many real-world applications.
3 Proposed Framework

3.1 Some Preliminaries on Regularized OT

We focus on the entropy regularized OT problem, where the objective function in \(1\) is supplemented by the (negative) entropy of the unknown distribution \(\pi\). The entropy regularization takes into account of the uncertainty and incompleteness of observed data, which can be an advantage over the OT without regularization \([20]\). The entropy regularized OT problem (29) as follows:

\[
\min_{\pi \in \Pi(\mu, \nu)} \left\{ \int_Z c(x, y)\pi(x, y) \, dx \, dy - \varepsilon H(\pi) \right\}, \tag{4}
\]

where \(H(\pi) := -\int_Z \pi \log \pi \, dx \, dy + \int_Z \pi \, dx \, dy\) denotes the entropy of \(\pi\).

By introducing the Lagrangian multipliers \(\alpha : X \to \mathbb{R} \) and \(\beta : Y \to \mathbb{R}\) to handle the linear constraints in \(\Pi(\mu, \nu)\) in \([2]\), we can obtain the dual problem of \(1\) as follows,

\[
\max_{\alpha, \beta} \int_X \alpha \mu \, dx + \int_Y \beta \nu \, dy - \varepsilon \int_Z e^{(\alpha + \beta - c) / \varepsilon} \, dx \, dy. \tag{5}
\]

Denote \((\alpha^c, \beta^c)\) the optimal solution of the dual problem \(5\) for the fixed cost function \(c\), then the optimal solution \(\pi^c\) of the primal problem \(4\), also under \(c\), is given by

\[
\pi^c(x, y) = e^{(\alpha^c(x) + \beta^c(y) - c(x, y)) / \varepsilon}. \tag{6}
\]

Due to the closed-form expression of \(\pi^c\) in terms of \((\alpha^c, \beta^c)\) in \(6\), we can just focus on solving the dual problem \(5\) rather than the primal one \(4\). In the discrete setting, we can apply the Sinkhorn algorithm that performs row and column scalings to obtain the optimal solution of \(5\), which is much more scalable than linear programming based solutions for \(1\) \([7, 20]\).

3.2 Inverse OT and Its Dual Formulation

As stated in Section 1, our goal in this paper is to learn the cost function \(c\) from the observed optimal transport plan. Suppose that the marginal distributions are given as \(\mu\) and \(\nu\), and we observed sample transport plan \(\hat{\pi} \in \Pi(\mu, \nu)\) (we will elaborate the formation of \(\hat{\pi}\) in the next section). Then we propose the following inverse OT problem to learn the underlying cost function such that the induced optimal transport plan matches \(\hat{\pi}\) by solving:

\[
\min_c D_{KL}(\hat{\pi} \| \pi^c) + R(c), \tag{7}
\]

where \(D_{KL}(\hat{\pi} \| \pi) := \int_Z \hat{\pi} \log (\hat{\pi} / \pi) \, dx \, dy\) is the Kullback-Leibler (KL) divergence from \(\pi\) to \(\hat{\pi}\), \(R(c)\) (including its weight) represents the regularization (or constraint) on the cost function \(c\), which is to be specified later, and \(\pi^c\) is the optimal transport plan induced by \(c\), i.e., the solution of \(4\). The model \(7\) is straightforward to interpret: we want to find the cost function \(c\) such that the induced \(\pi^c\) is close to the observed transport plan \(\hat{\pi}\) in the KL sense, and meanwhile it retains certain regularity or satisfies constraint described by \(R(c)\).

The problem \(7\) is a typical bi-level optimization due to the constraint that \(\pi^c\) solves another minimization \(4\). In general, bi-level optimization problems such as \(7\) are considered very challenging to solve: the standard approach requires repeated evaluations of the gradients of the main objective function with respect to the current estimate \(c\), and each of such evaluations amounts to solving one instance of the forward problem \(4\). However, we show that \(7\) is equivalent to an unconstrained and convex problem in \(c\) before adding \(R(c)\) by leveraging the dual form of the entropy regularized OT \(5\). To this end, we recall that the optimal transport plan \(\pi^c\) induced by \(c\) has a direct dependency on the optimal dual variables \((\alpha^c, \beta^c)\) of \(5\), as given in \(6\). Substituting \(\pi^c\) in \(6\) back into \(7\), and eliminating terms independent of \(c\), we obtain an equivalent minimization:

\[
\min_c \left\{ R(c) - \int_X \alpha^c \mu \, dx - \int_Y \beta^c \nu \, dy + \int_Z c \hat{\pi} \, dx \, dy \right\}, \tag{8}
\]
where we have used the fact \( \hat{\pi} \in \Pi(\mu, \nu) \) to obtain

\[
\int_Z \alpha^c \hat{\pi} \, dx \, dy = \int_X \alpha^c \left( \int_Y \hat{\pi} \, dy \right) \, dx = \int_X \alpha^c \mu \, dx,
\]

and a similar equality for \( \int_Y \beta^c \nu \, dy \). Despite the simplicity of formulation (8), \( \alpha^c \) and \( \beta^c \) depend on \( c \) indirectly as they are the optimal solution of the dual problem (5) for the given \( c \).

To eliminate \( \alpha^c \) and \( \beta^c \) in (8), we leverage an important optimality property of \((\alpha^c, \beta^c)\) for the dual problem (5):

\[
\int_Z c \hat{\pi} \, dx \, dy - \int_X \alpha \mu \, dx - \int_Y \beta \nu \, dy + \varepsilon = E(\alpha, \beta, c) = \min_{\alpha, \beta} E(\alpha, \beta, c)
\]

where the functional \( E \) is defined by

\[
E(\alpha, \beta, c) := \int_Z c \hat{\pi} \, dx \, dy - \int_X \alpha \mu \, dx - \int_Y \beta \nu \, dy + \varepsilon \int_Z e^{(\alpha + \beta - c)/\varepsilon} \, dx \, dy,
\]

and we have used the fact that \( \pi^c(x, y) = e^{(\alpha + \beta - c)/\varepsilon} \) is the optimal transport plan solving (4) according to (6), and hence satisfies \( \int_Z \pi^c \, dx \, dy = 1 \) to eliminate the integral of exponential in (10). Substituting (9) into (8), merging the minimizations, and eliminating the singled-out constant \( \varepsilon \), we obtain the final minimization for inverse OT:

\[
\min_{\alpha, \beta, c} E(\alpha, \beta, c) + R(c)
\]

We summarize the result of the derivations above in the following theorem.

**Theorem 1.** The inverse OT formulations (7) and (11) are equivalent in the sense that \( c^\ast \) solves (7) \iff \((\alpha^c, \beta^c, c^\ast)\) solves (11).

The variational problem (11) establishes the main framework for our algorithmic development of cost function learning in the next section. Compared to (7), the new formulation (11) is a regular optimization with a convex functional \( E \) (as shown later) and a customizable regularization (or constraint) \( R \), and hence has the potential to be solved much more efficiently than standard bi-level optimization problem. Interestingly, if \( c \) is given and fixed, then the inverse OT (11) reduces to the dual problem of the forward, entropy regularized OT (5).

The main feature of (11) is that the functional \( E(\alpha, \beta, c) \) is convex in \((\alpha, \beta, c) \in \mathcal{W} := C(X) \times C(Y) \times C(X \times Y) \). However, unlike (5), \( E(\alpha, \beta, c) \) is not strictly convex in its variable \((\alpha, \beta, \mathcal{C})\), which is normal as inverse problems are generally underdetermined given data (\( \hat{\pi} \) is our case). To address the uniqueness of solution to (11), we can exploit the behavior of \( E \) over the quotient space induced by the following equivalence relation.

**Definition 1.** Denote \((\alpha, \beta, c) \sim (\bar{\alpha}, \bar{\beta}, c) \) iff \( \alpha(x) + \beta(y) - c(x, y) = \bar{\alpha}(x) + \bar{\beta}(y) - \bar{c}(x, y) \) for all \((x, y) \in Z\), then \( \sim \) defines an equivalence relation over \( \mathcal{W} \), and hence a quotient space \( \mathcal{W} := \mathcal{W} / \sim \). Denote \([[(\alpha, \beta, c)] \subset \mathcal{W} \) the equivalence set of \((\alpha, \beta, c)\), and \( P : \mathcal{W} \rightarrow \mathcal{W} \) with \( P(\alpha, \beta, c) = [([\alpha, \beta, c]) \) the associated canonical projection.

We have the following result regarding the functional \( E(\alpha, \beta, c) \) in (11), and the proof is provided in Supplementary Material.

**Theorem 2.** The functional \( E(\alpha, \beta, c) \) in (11) is convex in \((\alpha, \beta, c)\). Moreover, \( E \) is constant over \([[(\alpha, \beta, c)]) \) for any \((\alpha, \beta, c) \in \mathcal{W} \). Let \( \bar{E} : \mathcal{W} \rightarrow \mathbb{R} \) such that \( \bar{E}([[(\alpha, \beta, c)]) = E(\alpha, \beta, c) \), then \( \bar{E} \) is well defined. If \((\alpha^\ast, \beta^\ast, c^\ast) \) is a solution of (11), the corresponding optimal transport plan \( \pi^c \) in (7) is \( e^{(\alpha^\ast(x) + \beta^\ast(y) - c^\ast(x, y))/\varepsilon} \), and \([[(\alpha^\ast, \beta^\ast, c^\ast)]) \) is the unique minimizer of \( \bar{E} \) over \( \mathcal{W} \).

According to Theorem 2 our inverse OT formulation (11) is convex as long as the customizable regularization \( R(c) \) is convex in \( c \). In this case, we can employ convex optimization schemes to solve (11) for the cost function, which are computationally much cheaper than solving general bi-level optimizations. Moreover, Theorem 2 implies that (11) admits a unique equivalence set that minimizes the functional \( E \). Therefore, even without \( R(c) \), we can characterize the entire optimal solution set using one minimizer of \( E(\alpha, \beta, c) \). As we will show later, certain mild regularization \( R(c) \) on \( c \) can further narrow down the search of the desired cost \( c \) to a single point within the equivalence set minimizing \( E \).
4 Algorithmic Development

In this section, we develop prototype numerical algorithms for cost function learning based on the inverse OT formulation \((11)\). Due to the substantial differences between the discrete and continuous settings, where the cost function is merely a matrix in the former case but a function defined on the continuous space \(Z\) in the latter, we consider the algorithmic developments separately.

4.1 Discrete Case

In the discrete case, the marginal distributions \(\mu\) and \(\nu\) are two probability vectors from \(\Delta^{m-1}\) and \(\Delta^{n-1}\), respectively. Therefore, the cost \(c\) and transport plan \(\pi\) are both \(m \times n\) matrices. Suppose that we can summarize the observed matching pairs into the matching matrix \(\hat{\pi}\), e.g., \(\pi_{ij} = N_{ij}/N\), where \(N_{ij}\) is the number of couples of an individual from the \(i\)th class corresponding to \(\mu_i\) and another individual from the \(j\)th class corresponding to \(\nu_j\), and \(N = \sum_{i=1}^{m} \sum_{j=1}^{n} N_{ij}\), then the inverse OT formulation \((11)\) reduces to the following optimization problem:

\[
\min_{\alpha, \beta, c} \{ R(c) - \langle \alpha, \mu \rangle - \langle \beta, \nu \rangle + \langle c, \hat{\pi} \rangle + s(\alpha, \beta, c) \} \tag{12}
\]

where \(\alpha \in \mathbb{R}^m, \beta \in \mathbb{R}^n, c \in \mathbb{R}^{m \times n}\), and \(s(\alpha, \beta, c) := \varepsilon \sum_{i=1}^{m} \sum_{j=1}^{n} e^{(\alpha_{ij} + \beta_{ij} - c_{ij})/\varepsilon} \). To solve the minimization problem, we can apply a matrix scaling algorithm by modifying the Sinkhorn-Knopp algorithm that alternates \(\alpha, \beta, c\) in \((12)\). More specifically, the updates of \(\alpha, \beta, c\) are identical to that in the Sinkhorn-Knopp algorithm for the forward entropy regularized OT \([7]\). The update of the cost matrix \(c\) is performed by applying a similar matrix scaling and then the proximal operator of \(R(c)\) defined by

\[
\text{prox}_{\gamma R}(\hat{c}) = \arg \min_c \left\{ R(c) + \frac{1}{2\gamma} \| c - \hat{c} \|^2 \right\}.
\]

The algorithm is summarized in Algorithm 1, where all exponential, logarithm, division operations are performed component-wisely. As inverse problems are mostly underdetermined, additional information can be essential to narrow down the search to the desired solution. Due to the convexity of \(E(\alpha, \beta, c)\) in \([11]\), we only need to impose proper convex regularization or constraint to convex set. For convenience, we denote \(J = [I_m^\top \otimes I_n; I_m \otimes 1_n^\top; 1_m \otimes I_n]\), where \(I_n\) is the \(n \times n\) identity matrix, and \([; ;]\) stacks the arguments vertically by following the standard MATLAB syntax. Denote \(\phi = [\alpha; \beta; c] \in \mathbb{R}^{m+n+mn}\) (where \(c \in \mathbb{R}^{mn}\) stacks the columns of \(c \in \mathbb{R}^{m \times n}\) in order vertically, we use the matrix and vector forms of \(c\) interchangeably hereafter), then \(J\phi = 0\) iff \(\alpha_i + \beta_j = c_{ij}\) for all \(i, j\). The following result characterizes a sufficient condition for unique minimizers of \((12)\) if \(R(c)\) imposes a convex constraint of \(c\) into the set \(C\), and the proof is provided in Supplementary Material \([13]\). Note that in this case \((12)\) reduces to \(\min_{(\alpha, \beta, c) \in W} E(\alpha, \beta, c)\) over the manifold \(W = \mathbb{R}^m \times \mathbb{R}^n \times C\) describing the constraint on \(\phi\), and the proximal operator \(\text{prox}_{\gamma R}\) in the c-step in Algorithm \([11]\) is simply the projection onto \(C\).

**Theorem 3.** Suppose \(R(c)\) imposes the projection onto a closed convex set \(C\) in \(\mathbb{R}^{m \times n}\) and \((12)\) attains minimum at \(\phi^* := (\alpha^*, \beta^*, c^*)\). If \(T_{\phi^*} W \cap \ker(J) = \{0\}\), where \(T_{\phi^*} W\) is the tangent space of \(W\) at \(\phi^*\), then \(\phi^*\) is the unique minimizer of \((12)\).
Symmetric matrices with null diagonal. If $C$ is the set of symmetric matrices with null diagonal, then the projection of a (square) matrix $c$ onto $C$ can be computed simply by $c \leftarrow (1/2) \cdot (c + c^\top)$ and then $c_{ii} \leftarrow 0$ for all $i$. We can show that such projection yields a unique solution of (12) in the next corollary, where the proof is provided in Supplementary Material [13].

**Corollary 1.** Suppose $C = \{ c \in \mathbb{R}^{n \times n} \mid c = c^\top, c_{ii} = 0, \forall i \}$, then (12) admits a unique solution $c^*$.  

Despite of its simple projection, the constraint $C$ includes a large number of cost matrices used in practice. In particular, any (multiple of) norm-like cost matrix, i.e., $kc$ with $c_{ij} = |i - j|^p$ (up to any permutation of indices) for any nonzero $k, p \in \mathbb{R}$ (if $p < 0$ then also require $c_{ii} = 0$ separately) is included in $C$.  

Linear affinity matrix. In this case, $C = \{ G^\top A D \mid A \in \mathbb{R}^{p \times q} \}$ for given $G = [g_1, \ldots, g_m] \in \mathbb{R}^{p \times m}$ and $D = [d_1, \ldots, d_n] \in \mathbb{R}^{q \times n}$. Here $G$ and $D$ represent the feature vector matrices for the two populations in matching, respectively, and $A \in \mathbb{R}^{p \times q}$ is the interaction (affinity) matrix to be learned. This is an application considered in [12] [13], where $g_k \in \mathbb{R}^p$ and $d_j \in \mathbb{R}^q$ are the feature vectors of the $i$th class of men and $j$th class of women in the marriage market for $i \in [m]$ and $j \in [n]$, and $A_{kj} \in \mathbb{R}$ is the complementary coefficient of the $k$th feature of men and $l$th feature of women for $k \in [p]$ and $l \in [q]$. Then the projection of $-\varepsilon \log K$ onto the set $C$ is given by $\text{prox}_\varepsilon (-\varepsilon (G^\top)^\top \log K) D^+$, where $G^+ + D^+$ are the Moore-Penrose pseudoinverse of $G$ and $D$, respectively, and can be pre-computed before applying Algorithm [1].

If the regularization $R(c)$ is imposed as an additive penalty term in the objective function in (11), then we need to compute the proximal operator $\text{prox}_{\gamma R}(c)$ in the $c$-step of Algorithm [1] with $\gamma \to 0$ gradually along with iterations. We can also employ the alternating direction method of multipliers (ADMM) with an additional Lagrange multiplier $\lambda \in \mathbb{R}^{m \times n}$, and execute the $c$-step as $c \leftarrow \text{prox}_{\gamma R}(-\varepsilon \log K - \lambda)$ followed by the multiplier update step $\lambda \leftarrow \lambda + (c + \varepsilon \log K)$ in Algorithm [1]. The main computational cost is still in the proximal operator. We provide several closed-form solutions for such operator as follows.

Low rank matrices. In this case, we can set $R(c) = \|c\|_*$, the nuclear norm of the matrix $c \in \mathbb{R}^{m \times n}$, i.e., the sum of singular values of the matrix $c$. Nuclear norm is a convex relaxation of the matrix rank and much more computationally tractable than the latter. The closed-form solution of $\text{prox}_{\gamma R}(\hat{c})$ is $U \max(\Sigma - \gamma I, 0) V^\top$, where $U \Sigma V^\top$ is the singular value decomposition (SVD) of the matrix $\hat{c}$.

Sparse matrices. If $c$ is known to be a sparse matrix, we can set $R(c) = \sum_{i=1}^m \sum_{j=1}^n |c_{ij}|$ which is a convex relaxation of the nonzero component counter of $c$. The proximal operator $c \leftarrow \text{prox}_{\gamma R}(\hat{c})$ has a closed form known as the soft shrinkage: $c_{ij} = \text{sign}(\hat{c}_{ij}) \max(|\hat{c}_{ij}| - \gamma, 0)$ for all $i, j$.

### 4.2 Continuous Case

In the continuous case, $\mu, \nu,$ and $\pi$ represent probability density functions on $X, Y$, and $Z = X \times Y$, respectively. In this case, we are only given i.i.d. samples of these distributions, namely, $x^{(i)} \sim \mu$, $y^{(i)} \sim \nu$, and $(x^{(i)}, y^{(i)}) \sim \pi \in \Pi(\mu, \nu)$. The cost function $c(x, y)$ is a function defined on the continuous space $Z$ where $X \subset \mathbb{R}^{d_X}$ and $Y \subset \mathbb{R}^{d_Y}$ for potentially high dimensionality $d_X$ and $d_Y$. As mentioned in Section [1], discretization of the spaces $X$ and $Y$ results in exponentially increasing $m$ and $n$, for which the computation in Algorithm [1] becomes prohibitive just as in any discrete forward OT problems.

Our approach [11] is a regular optimization and enables a natural solution to overcome the issue of discretization using deep neural networks, which is a significant advantage over bi-level optimization formulations. More precisely, we can parametrize the cost function $c$, as well as the functions $\alpha$ and $\beta$, in [11] as deep networks. In this case, $\alpha, \beta,$ and $c$ are neural networks with output layer dimension 1 and input layer dimensions $d_X, d_Y$, and $d_X + d_Y$, respectively. In particular, the design of the network architecture of $c$ may take the regularization $R(c)$ into consideration. For example, if $R(c)$ suggests that $c \geq 0$, then we can set the activation function in the output layer as the rectified linear unit (ReLU) $\sigma(x) := \max(x, 0)$. We can also introduce an encoder $h_{\eta}$ to be learned, such that the cost $c_{\eta}$ is the standard Euclidean distance between encoded features, e.g., $c_{\eta}(x, y) = |h_{\eta}(x) - h_{\eta}(y)|$. Nevertheless, in general, the architectures of the $\alpha, \beta,$ and $c$ are rather flexible, and can be customized adaptively according to specific applications. We will present several numerical results with architecture specifications used in our experiments.
In (14), the empirical expectations are defined by the sample averages:
\[
\hat{\alpha}, \hat{\beta}, \hat{\omega} \text{ we can use scaling } L \text{ descent algorithm to }
\]

In our experiments, we simply used the uniform sampling (15).
\[
\text{variance with properly chosen } \Sigma. \text{ Other methods for approximating the integrals can also be applied.}
\]

To formalize our deep neural net approach for solving continuous inverse OT, we let \( \theta \) and \( \eta \) denote the parameters of \( \alpha \) and \( \beta \) (in actual implementations, \( \alpha \) and \( \beta \) are two separate neural networks with different parameters \( \theta_a \) and \( \theta_b \) respectively, but we use \( \theta \) for both to avoid overloaded notations). In addition, we use \( \eta \) to denote the parameters of \( c \). Then we can solve for the optimal \( (\theta^*, \eta^*) \) by minimizing the loss function \( L \) of the network parameters \( (\theta, \eta) \) based on (14) as follows:
\[
\min_{\theta, \eta} L(\theta, \eta) := R(c_\eta) - \mathbb{E}_\mu[\alpha_\theta] - \mathbb{E}_\nu[\beta_\eta] + \mathbb{E}_{\hat{\pi}}[c_\eta] + \varepsilon \int_Z d\pi(x) \rho(\alpha_\theta(x) + \beta_\eta(y) - c_\eta(x,y))/\varepsilon \ dx dy.
\] (14)

In (14), the empirical expectations are defined by the sample averages:
\[
\mathbb{E}_\mu[\alpha_\theta] := N_\mu^{-1} \sum_{i=1}^{N_\mu} \alpha_\theta(x^{(i)}), \quad \mathbb{E}_\nu[\beta_\eta] := N_\nu^{-1} \sum_{i=1}^{N_\nu} \beta_\eta(y^{(i)}), \quad \mathbb{E}_{\hat{\pi}}[c_\eta] := N_\pi^{-1} \sum_{i=1}^{N_\pi} c_\eta(x^{(i)}, y^{(i)}),
\]
where \( D_\mu := \{x^{(i)} | i \in [N_\mu]\}, D_\nu := \{y^{(i)} | i \in [N_\nu]\}, \) and \( D_\pi := \{(x^{(i)}, y^{(i)}) | i \in [N_\pi]\} \) are observed pairings of \( \hat{\pi} \). If only \( \hat{\pi} \) is available, we can also substitute the samples for \( \mu \) and \( \nu \) by the first and second coordinates of the samples in \( \{(x^{(i)}, y^{(i)}) | i \in [N_\pi]\} \).

The last integral in (14) can be approximated by numerical integration methods, such as Gauss quadrature and sample-based integrations. For example, if \( Z \) is bounded, we can sample \( N_\pi \) collocation points \( \{(x^{(i)}, y^{(i)}) | i \in [N_\pi]\} \) from \( Z \) uniformly, and approximate the integral by
\[
\int_Z G_{\theta, \eta}(x,y) \ dx \ dy \approx \frac{1}{N_\pi} \sum_{i=1}^{N_\pi} G_{\theta, \eta}(x^{(i)}, y^{(i)}),
\] (15)

where \( G_{\theta, \eta}(x, y) := e^{(\alpha_\theta(x) + \beta_\eta(y) - c_\eta(x,y))/\varepsilon} \). A more appealing method for sample-based integration is to use an importance sampling strategy: we first estimate the mode(s) of the function \( G_{\theta, \eta}(x, y) \), and draw i.i.d. samples drawn from \( \mu \) and \( \nu \) respectively, and \( D_{\pi} := \{(x^{(i)}, y^{(i)}) | i \in [N_\pi]\} \) are observed pairings of \( \pi \). If only \( \pi \) is available, we can also substitute the samples for \( \mu \) and \( \nu \) by the first and second coordinates of the samples in \( \{(x^{(i)}, y^{(i)}) | i \in [N_\pi]\} \).

The advantages of this importance sampling strategy include the capability of integral over unbounded domain \( Z \) and smaller sample approximation variance with properly chosen \( \omega \) and \( \Sigma \). Other methods for approximating the integrals can also be applied. In our experiments, we simply used the uniform sampling (15).

Now we have all the ingredients in the loss function \( L(\theta, \eta) \) in (14). We can apply (stochastic) gradient descent algorithm to \( L \) and find an optimal solution \( (\theta^*, \eta^*) \). In each iteration, we can use all the samples available for the empirical expectations, or only sample a mini-batch, for the computation of the gradient of \( L \) with respect to \( (\theta, \eta) \). Otherwise, the optimization is standard in deep neural network training. Moreover, we can use scaling \( \alpha_\theta \leftarrow \alpha_\theta / \varepsilon \), \( \beta_\eta \leftarrow \beta_\eta / \varepsilon \), and \( c_\eta \leftarrow c_\eta / \varepsilon \) and hence \( L(\theta, \eta) \) can be minimized with \( \varepsilon = 1 \) in (14). Then we can scale \( c_\eta \) back by multiplying \( \varepsilon \) after \( (\theta^*, \eta^*) \) is obtained. This algorithm is summarized in Algorithm 2.

5 Numerical Experiments

Experiment setup. We evaluate the proposed cost learning algorithms (Algorithms 1 and 2) to several synthetic and real datasets. All algorithms and numerical experiments are implemented in Python (TensorFlow framework is used in the continuous setting) and performed on a machine equipped with 2.20GHz
We follow the setting of [13] and apply Algorithm 1 to Discrete inverse OT on real marriage data. We perform the same test of Algorithm 1 with varying entropy regularization weight $\varepsilon$. For fixed $\varepsilon$, we test Algorithm 1 for each problem size $n = 10, 20, 50, 100, 200$ (e.g., $\mu, \nu \in \mathbb{R}^n$), run the algorithm until the relative error of $c$ reaches $10^{-2}$, and record the average and standard deviation of the CPU time for 20 instances. We plot the CPU time (in seconds) versus the problem size $n$ (in log-log) in Figure 1d.

**Discrete inverse OT on synthetic data.** We first test Algorithm 1 to recover cost matrix $c$ using observed transport plan matrix $\hat{\pi}$ in the discrete case. We set $m = n$ and set ground truth $c^*$ with $c^*_{ij} = |\frac{i-j}{n}|^p$ for $i, j \in [n]$ and $p = 0.5, 1, 2, 3$. Then we generate $\hat{\pi}$ for each $c^*$ with varying $\varepsilon = 10^1, 10^2, 10^{-1}, 10^{-2}$ using Sinkhorn algorithm, and apply Algorithm 1 to $\hat{\pi}$ and see if we can recover the original $c^*$. To this end, we set the constraint set $C = \{c \in \mathbb{R}^{n \times n} | c = c^T, c_{ii} = 0, \forall i \in [n]\}$, for which the projection $\text{prox}_{\varepsilon, R}$ can be computed simply by $c \leftarrow (c + c^T)/2$ followed by $\text{diag}(c) \leftarrow 0$. We also threshold $c$ to nonnegative values by applying $\max(\cdot, 0)$, which seems to further improve efficiency for this problem. Figure 2 shows the results of Algorithm 1. For fixed $\varepsilon = 10^{-1}$, we generate 20 random pairs of $(\mu, \nu) \in \mathbb{R}^n \times \mathbb{R}^n$ and corresponding $\hat{\pi}$ for problem size $n = 100$, and apply Algorithm 1 to recover cost matrix $c$. Figures 1a and 1b show the average (center line) and standard deviation (shade boundary) over the 20 instances of the objective function $\text{Obj}$ versus iteration number. In all cases, the true $c^*$ is accurately recovered with relative error approximately $10^{-4}$ or lower after 500 iterations. For fixed $p = 2$, we also perform the same test of Algorithm 1 with varying entropy regularization weight $\varepsilon = 10^1, 10^{-0}, 10^{-1}, 10^{-2}$. We again run 20 instances and plot decay of relative error (in logarithm) versus iteration number. The result is shown in Figure 1c. With the same settings for $p = 2$ and varying $\varepsilon$, we test Algorithm 1 for each problem size $n = 10, 20, 50, 100, 200$ (e.g., $\mu, \nu \in \mathbb{R}^n$), run the algorithm until the relative error of $c$ reaches $10^{-2}$, and record the average and standard deviation of the CPU time for 20 instances. We plot the CPU time (in seconds) versus the problem size $n$ (in log-log) in Figure 1d.

**Discrete inverse OT on real marriage data.** We follow the setting of [13] and apply Algorithm 1 to...
In this paper, we considered the problem of learning the cost function for OT. We propose an inverse OT approach to learn the cost functions such that the induced OT plan is close to the observed plan or its samples. Unlike the bi-level optimization in the literature, we propose a novel inverse OT formulation to learn the cost function by minimizing an unconstrained convex functional, which can be further augmented by customizable regularization on the cost. We developed two prototype numerical algorithms to recover the cost in the discrete and continuous cases separately. Numerical results show very promising efficiency and accuracy of these algorithms.

### Table 1: Part of the affinity matrix learned by Algorithm 1

Entries show the complementarity coefficients of husband (H) and wife (W) for five features: education (edu), irresponsibility (irresp), disciplined (discip), ordered (order), and detail.

| H \ W | edu | irresp | discip | order | detail |
|-------|-----|-------|-------|-------|-------|
| edu   | 0.728 | 0.029 | 0.042 | -0.128 | -0.055 |
| irresp | 0.081 | 0.166 | 0.032 | -0.007 | 0.001 |
| discip | -0.129 | 0.130 | 0.193 | -0.010 | -0.012 |
| order | -0.171 | 0.012 | -0.054 | 0.066 | 0.039 |
| detail | -0.137 | 0.017 | -0.054 | 0.058 | 0.035 |

Continuous inverse OT on synthetic data. Next, we test Algorithm 2 to recover the cost function \( c \) using observed samples of \( \hat{\pi} \) in the continuous case. In this test, we set the true cost function \( c^*(x, y) = |x - y|^p \) where \( X = Y = [0, 1] \subset \mathbb{R} \) and \( p = 0.5, 1, 2, 3 \). Then we use the Sinkhorn algorithm to generate the corresponding transport plan \( \hat{\pi} \), and draw samples \( D_{\hat{\pi}} = \{(x^{(i)}, y^{(i)}) | i \in [N_{\hat{\pi}}]\} \) from \( \hat{\pi} \). With the samples \( D_{\hat{\pi}} \) only, we apply Algorithm 2 to recover the cost \( c^* \). In (14), we require \( c : Z \to \mathbb{R} \) to be nonnegative and symmetric in its arguments \( x \) and \( y \). Therefore \( R(c) \) is the projection onto the set of functions \( \{c : Z \subset \mathbb{R} | c(x, y) = c(y, x) \geq 0\} \). To this end, we set the input layer of \( c_v \) to take \( \xi \in \mathbb{R}^d \) where \( \xi_i = |x_i - y_i| \) for all \( i \). Other parameters are provided in Supplementary Material D. The result is shown in Figure 2, which demonstrate that Algorithm 2 faithfully recovers the cost function.

### 6 Conclusion

In this paper, we considered the problem of learning the cost function for OT. We propose an inverse OT approach to learn the cost functions such that the induced OT plan is close to the observed plan or its samples. Unlike the bi-level optimization in the literature, we propose a novel inverse OT formulation to learn the cost function by minimizing an unconstrained convex functional, which can be further augmented by customizable regularization on the cost. We developed two prototype numerical algorithms to recover the cost in the discrete and continuous cases separately. Numerical results show very promising efficiency and accuracy of these algorithms.
Figure 2: The relative error (left) and objective function value (right) versus SGD iteration of Algorithm 2 on continuous inverse OT with synthetic data and varying $p$.

References

[1] S. S. Abadeh, V. A. Nguyen, D. Kuhn, and P. M. M. Esfahani. Wasserstein distributionally robust kalman filtering. In Advances in Neural Information Processing Systems, pages 8474–8483, 2018.

[2] T. Abrishami, N. Guillen, P. Rule, Z. Schutzman, J. Solomon, T. Weighill, and S. Wu. Geometry of graph partitions via optimal transport. arXiv preprint arXiv:1910.09618, 2019.

[3] S.-i. Amari, R. Karakida, and M. Oizumi. Information geometry connecting wasserstein distance and kullback-leibler divergence via the entropy-relaxed transportation problem. Information Geometry, 1(1):13–37, 2018.

[4] L. Ambrosio, L. A. Caffarelli, Y. Brenier, G. Buttazzo, and C. Villani. Optimal transportation and applications, volume 1813 of Lecture Notes in Mathematics. Springer-Verlag, Berlin, 2003. Lectures from the C.I.M.E. Summer School held in Martina Franca, September 2–8, 2001, Edited by Caffarelli and S. Salsa.

[5] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709, 2013.

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

[7] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in Neural Information Processing Systems, pages 2292–2300, 2013.

[8] M. Cuturi and D. Avis. Ground metric learning. The Journal of Machine Learning Research, 15(1):533–564, 2014.

[9] T. M. Dagnew and U. Castellani. Supervised learning of diffusion distance to improve histogram matching. In International Workshop on Similarity-Based Pattern Recognition, pages 28–37. Springer, 2015.

[10] A. Dessein, N. Papadakis, and C.-A. Deledalle. Parameter estimation in finite mixture models by regularized optimal transport: A unified framework for hard and soft clustering. arXiv preprint arXiv:1711.04366, 2017.

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

[12] A. Dupuy and A. Galichon. Personality traits and the marriage market. Journal of Political Economy, 122(6):1271–1319, 2014.

[13] A. Galichon and B. Salanié. Cupid’s invisible hand: Social surplus and identification in matching models. Available at SSRN 1804623, 2015.

[14] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. Sample complexity of sinkhorn divergences. arXiv preprint arXiv:1810.02733, 2018.
[15] G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger. Supervised word mover’s distance. In *Advances in Neural Information Processing Systems*, pages 4862–4870, 2016.

[16] T. Le and M. Cuturi. Unsupervised riemannian metric learning for histograms using aitchison transformations. In *International Conference on Machine Learning*, pages 2002–2011, 2015.

[17] R. Li, X. Ye, H. Zhou, and H. Zha. Learning to match via inverse optimal transport. *Journal of Machine Learning Research*, 20(80):1–37, 2019.

[18] R. Liu, A. Balsubramani, and J. Zou. Learning transport cost from subset correspondence. *arXiv preprint arXiv:1909.13203*, 2019.

[19] N. Papadakis. *Optimal transport for image processing*. PhD thesis, 2015.

[20] G. Peyré, M. Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[21] A. Rolet, M. Cuturi, and G. Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638, 2016.

[22] B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

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

[24] S. Shafieezadeh-Abadeh, D. Kuhn, and P. M. Esfahani. Regularization via mass transportation. *arXiv preprint arXiv:1710.10016*, 2017.

[25] A. M. Stuart and M.-T. Wolfram. Inverse optimal transport. *arXiv:1905.03950*, 2019.

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

[27] F. Wang and L. J. Guibas. Supervised earth move’s distance learning and its computer vision applications. In *European Conference on Computer Vision*, pages 442–455. Springer, 2012.

[28] L. Xu, H. Sun, and Y. Liu. Learning with batch-wise optimal transport loss for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3333–3342, 2019.

[29] Y. Yang, Y.-F. Wu, D.-C. Zhan, Z.-B. Liu, and Y. Jiang. Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2594–2603. ACM, 2018.

[30] P. Zhao and Z.-H. Zhou. Label distribution learning by optimal transport. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
Supplementary Materials

A Characterization of Inverse Problems Bypassing Bi-level Optimization

In general, inverse problems are formulated as optimization problems with complicated constraints. In a typical setting, a set of (unknown) parameters of the problem determines an intermediate variable, and such relation forms the constraint; the intermediate variable is then compared with the measurement data, and the objective (loss) function quantifies their discrepancy. The goal is then to find a (the) set of parameters such that the objective function is minimized.

In the inverse OT problem considered in this work, the cost function $c$ of OT is the set of unknown parameter. The induced optimal transport plan $\pi^*$ is the intermediate variable, which is compared to the observed data $\hat{\pi}$ using KL divergence as a metric of discrepancy. In neural ODE, the initial value and/or parameters in the defining function (e.g., network parameters) is the set of unknown parameters, the solution of the ODE is the induced intermediate variable, and the objective function is the squared error between the ODE solution and observations. Although the solution of ODE can be considered as an optimal trajectory that fits the ODE determined by the unknown parameters, one often employs numerical ODE solver to approximate the solution, which makes the sense of optimality part a little faded.

When the constraint itself involves an optimization, the constrained problem is called bi-level optimization, which is considered very challenging computationally. Common approaches to bi-level optimization require solving the optimization problem given in the constrained repeatedly, which results in high computational cost of bi-level optimization.

We showed in this work that the bi-level optimization problem of inverse OT can be reduced to an unconstrained optimization, which is much easier to solve than the original one. Here we provide certain characterizations of problems which may potentially enjoy similar solution approach. In brief, the discrepancy measure in the objective function and the optimization problem described by the constraint need to be paired up to make the reduction possible.

To follow the notations used in this work, we use $L(\pi, \gamma; c)$ to denote the Lagrange function of the optimization problem in the constraint. Here $\pi$ and $\gamma = (\alpha, \beta)$ correspond to the primal and dual variables, and $c$ is the unknown parameter that we want to obtain ultimately. In inverse OT, $L(\pi, \gamma; c) = \langle c, \pi \rangle - \epsilon H(\pi) - \langle \gamma, A\pi - b \rangle$, where $A\pi = b$ describes the marginal constraint $\pi \in \Pi(\mu, \nu)$ with $A = [1_m \otimes I_n; I_m \otimes 1_n]$ and $b = [\mu; \nu]$. We then define the primal and dual objective functions:

\[
F(\pi; c) := L(\pi, \gamma; \pi; c) = \max_{c} L(\pi, \gamma; c), \tag{16}
\]
\[
G(\gamma; c) := L(\pi(\gamma; c), \gamma; c) = \min_{\pi} L(\pi, \gamma; c), \tag{17}
\]

where we used the following notations to denote the optimal dual (primal) variable when the primal (dual) variable is fixed:

\[
\gamma(\pi; c) := \operatorname{arg \ max}_{\gamma} L(\pi, \gamma; c), \tag{18}
\]
\[
\pi(\gamma; c) := \operatorname{arg \ min}_{\pi} L(\pi, \gamma; c). \tag{19}
\]

If the strong duality of $L$ holds (as in OT), we also use $(\pi^*(c), \gamma^*(c))$ to denote the optimal solution of the minimax problem:

\[
\max_{\pi} \min_{\gamma} L(\pi, \gamma; c) = \min_{\gamma} \max_{\pi} L(\pi, \gamma; c) \tag{20}
\]

Note that $\pi^*(c) = \pi(\gamma^*(c); c) = \operatorname{arg \ min}_{\pi} F(\pi; c)$ and $\gamma^*(c) = \gamma(\pi^*(c); c) = \operatorname{arg \ max}_{\gamma} G(\gamma; c)$.

Consider the bi-level optimization problem of form

\[
\min_{\pi} D(\hat{\pi}, \pi^*(c)), \quad \text{subject to} \quad \pi^*(c) = \operatorname{arg \ min}_{\pi} F(\pi; c). \tag{21}
\]
Suppose that the objective function $D$ is defined by (up to constant shifting and scaling)

$$D(\hat{x}, \pi) = F(\hat{x}; c) - F(\pi; c)$$  \hspace{1cm} (22)

for the given data $\hat{x}$, then one can readily see that $D(\hat{x}, \pi^*(c)) \geq 0$ since $\pi^*(c) = \arg\min_\pi F(\pi; c)$. Moreover $D(\hat{x}, \pi^*(c)) = 0$ means that $\hat{x}$ is equally good as $\pi^*(c)$ in terms of minimizing $F(\pi; c)$ for the given $c$. Therefore $D(\hat{x}, \pi^*(c))$ is a promising metric to evaluate the quality of a parameter $c$: the “loss” $D$ is always nonnegative, and vanishes only if $\hat{x}$ is a minimizer of $F(\pi; c)$.

In summary, if the objective function in the bi-level optimization problem (21) is defined as in (22), then the problem can be readily reduced to a simpler unconstrained problem:

$$\min_c \left\{ D(\hat{x}, \pi^*(c)) \mid \pi^*(c) = \arg\min_\pi F(\pi; c) \right\}$$

$$= \min_c F(\hat{x}; c) - F(\pi^*(c); c)$$

$$= \min_c F(\hat{x}; c) - G(\pi^*(c); c)$$

$$= \min_c F(\hat{x}; c) - \max_\gamma G(\pi^*(c); c)$$

$$= \min_c F(\hat{x}; c) - G(\pi^*; c)$$

\hspace{1cm} (23)

where the second equality is due to the definition of primal and dual objective functions $\{16\}$ and $\{17\}$ and the strong duality $\{20\}$, the third equality is due to the optimality of $\pi^*(c)$.

The inverse OT formulation we had in this work is exactly one special case of (23): for fixed $\varepsilon > 0$, we had $F(\pi; c) = \langle \pi, c \rangle - \varepsilon H(\pi) + \varepsilon H(\pi)$ and $G(\pi; c) = \langle \gamma, b \rangle - \varepsilon (e^{(A^T \gamma - c - \varepsilon) / \varepsilon} - 1)$, where $\varepsilon_{A \pi = b}(\pi) = 0$ if $A \pi = b$ and $+\infty$ otherwise, and we set the loss function to $D(\hat{x}, \pi) = D_{KL}(\hat{x} \mid\mid \pi)$. Therefore,

$$D(\hat{x}, \pi^*(c)) = D_{KL}(\hat{x} \mid\mid \pi^*(c)) = \langle \hat{x}, \log(\hat{x} \mid\mid \pi^*(c)) \rangle$$

$$= -H(\hat{x}) + \langle \hat{x}, 1 \rangle - \langle \hat{x}, \log(\pi^*(c)) \rangle$$

$$= -H(\hat{x}) + \langle \hat{x}, 1 \rangle - \langle \hat{x}, \log e^{(A^T \gamma - c - \varepsilon) / \varepsilon} \rangle$$

$$= -H(\hat{x}) + \langle \hat{x}, 1 \rangle + \varepsilon^{-1} \langle \hat{x}, c \rangle - \varepsilon^{-1} \langle A \hat{x}, \gamma^*(c) \rangle$$

$$= \varepsilon^{-1} F(\hat{x}; c) + \langle \pi^*(c), 1 \rangle - \varepsilon^{-1} \langle A \hat{x}, \gamma^*(c) \rangle$$

$$= \varepsilon^{-1} F(\hat{x}; c) + \langle \pi^*(c), 1 \rangle - \varepsilon^{-1} \langle b, \gamma^*(c) \rangle$$

$$= \varepsilon^{-1} F(\hat{x}; c) + \langle e^{(A^T \gamma - c - \varepsilon) / \varepsilon}, 1 \rangle - \varepsilon^{-1} \langle b, \gamma^*(c) \rangle$$

$$= \varepsilon^{-1} F(\hat{x}; c) - \varepsilon^{-1} G(\gamma^*(c); c)$$

$$= \varepsilon^{-1} (F(\hat{x}; c) - \varepsilon^{-1} G(\gamma^*(c); c))$$

where the third equality is due to $\pi^*(c) = \pi(\gamma^*(c); c)$ as shown under $\{20\}$, the fourth equality is due to the expression of the optimal primal variable using dual variable, the sixth equality is due to the definition of $F$ and that $\hat{x}$ and $\pi^*(c)$ are probabilities, the seventh is from $A \pi = b$, the ninth due to the definition of $G$, and the last equality due to strong duality.

\section{Proofs}

\textbf{Proof of Theorem $\{3\}$} Let $\phi : X \times Y \to \mathbb{R}^3$ be defined by $\phi(x, y) = (\alpha(x), \beta(y), c(x, y))^T$. Denote $\zeta = (1, 1, -1)^T \in \mathbb{R}^3$. Then the functional $E$ in $\{11\}$ is given by

$$E(\phi) = \int_Z (-\mu(x), -\nu(y), \pi(x, y)) \cdot \phi(x, y) \, dx \, dy + \varepsilon \int_Z e^{(\zeta \cdot \phi) / \varepsilon} \, dx \, dy.$$  \hspace{1cm} (24)

For any fixed $\psi : Z \to \mathbb{R}^3$, we define the function

$$f(\epsilon) = E(\phi + \epsilon \psi).$$  \hspace{1cm} (24)
Then we can verify that
\[
    f'(\epsilon) = \int_Z (-\mu, -\nu, \pi) \cdot \psi \, dx \, dy + \int_Z e^{(\zeta(\phi+\epsilon\psi))/\epsilon}(\zeta \cdot \psi) \, dx \, dy
\]
and that
\[
    f''(\epsilon) = \frac{1}{\epsilon} \int_Z e^{(\zeta(\phi+\epsilon\psi))/\epsilon}(\zeta \cdot \psi)^2 \, dx \, dy \geq 0
\]
for any \( \epsilon \in \mathbb{R} \). Since \( \psi \) is arbitrary, we know \( E \) is a convex functional of \( \phi \).

To show that \( E \) is constant over the equivalence set \( \{ \phi \} = [(\alpha, \beta, c)] \), we recall that \( (\alpha, \beta, c) \sim (\tilde{\alpha}, \tilde{\beta}, \tilde{c}) \) if \( \alpha(x) + \beta(y) - c(x,y) = \tilde{\alpha}(x) + \tilde{\beta}(y) - \tilde{c}(x,y) \) for all \((x,y)\). Define \( \alpha_\delta(x) = \alpha(x) - \delta \alpha(x) \) for every \( x \in X \) and \( \beta_\delta(y) = \beta(y) - \delta \beta(y) \) for every \( y \in Y \), then it is obvious that
\[
    \tilde{c}(x,y) = \tilde{\alpha}(x) + \tilde{\beta}(y) - \alpha(x) - \beta(y) + c(x,y) = -\alpha_\delta(x) - \beta_\delta(y) + c(x,y). \tag{27}
\]

Hence we have
\[
    E(\tilde{\alpha}, \tilde{\beta}, \tilde{c}) = \int_Z \tilde{\pi} \, dx \, dy - \int_X \mu \tilde{\alpha} \, dx - \int_Y \nu \tilde{\beta} \, dy + \epsilon \int_Z e^{(\tilde{\alpha} + \tilde{\beta} - \delta)/\epsilon} \, dx \, dy
\]
\[
    = \int_Z (-\alpha_\delta - \beta_\delta + c)\tilde{\pi} \, dx \, dy - \int_X \mu(\alpha - \alpha_\delta) \, dx - \int_Y \nu(\beta - \beta_\delta) \, dy + \epsilon \int_Z e^{(\alpha_\delta + \beta_\delta)/\epsilon} \, dx \, dy
\]
\[
    = E(\alpha, \beta, \gamma) - \int_Z (\alpha + \beta(\delta)\tilde{\pi} \, dx \, dy + \int_X \mu \alpha_\delta \, dx + \int_Y \mu \beta_\delta \, dy
\]
\[
    = E(\alpha, \beta, \gamma),
\]

since \( \tilde{\pi} \in \Pi(\mu, \nu) \) implies \( \int_Z \alpha_\delta \tilde{\pi} \, dx \, dy = \int_X \alpha_\delta(\int_Y \tilde{\pi} \, dy) \, dx = \int_X \mu \alpha_\delta \, dx \) etc. Therefore \( E \) is constant over \([(\alpha, \beta, c)]\), and hence \( \tilde{E} : W(= W/\sim) \to \mathbb{R} \) given by \( \tilde{E}(\phi) := E(\phi) \) for all \( \phi \in W \) is well defined.

For the minimizer \((\alpha^*, \beta^*, c^*)\) of \( \Pi \), we know that \((\alpha^*, \beta^*)\) minimizes \( E(\alpha, \beta, c) \) for \( c = c^* \), i.e.,
\[
    (\alpha^*, \beta^*) = \arg \min_{\alpha,\beta} E(\alpha, \beta, c^*) = \arg \min_{\alpha,\beta} \left\{ \epsilon \int_Z e^{(\alpha + \beta - c^*)/\epsilon} \, dx \, dy - \int_X \mu \alpha \, dx - \int_Y \nu \beta \, dy \right\}.
\]

Hence \((\alpha^*, \beta^*)\) is the optimal dual variable of the forward OT with cost \( c^* \), and therefore the optimal transport plan (optimal primal variable) is \( \pi^*(x,y) = e^{(\alpha^*(x) + \beta^*(y) - c^*(x,y))/\epsilon} \) for all \((x,y)\in Z\).

Finally, we need to show that \([\phi^*]\) is the unique minimizer of \( \tilde{E} \) over \( \tilde{W} \) if \( \phi^* \) minimizes \( E \). For any nonzero \([\psi]\in \tilde{W} \), we define
\[
    g(\epsilon) = \tilde{E}([\phi] + [\epsilon[\psi]]) = \tilde{E}([\phi + \epsilon[\psi]]) = E(\phi + \epsilon[\psi]). \tag{28}
\]

Following the same derivation above, we can show that
\[
    g''(\epsilon) = \frac{1}{\epsilon} \int_Z e^{(\zeta(\phi+\epsilon\psi))/\epsilon}(\zeta \cdot \psi)^2 \, dx \, dy
\]
Since \([\psi]\neq 0\), we know \( \zeta \cdot \psi(x,y) \neq 0 \) at some \((x,y)\in Z\). Since \( \psi \) is continuous, we know that there exists \( \delta > 0 \) and an open neighborhood \( U \subset Z \) (with measure \(|U| > 0\)) of \((x,y)\) such that \( e^{(\zeta(\phi+\epsilon\psi))/\epsilon}(\zeta \cdot \psi)^2 \geq \delta > 0 \) for all \((x,y)\in U\). This implies that
\[
    g''(\epsilon) \geq \frac{1}{\epsilon} \int_U \delta \, dx \, dy = \epsilon^{-1} \delta |U| > 0.
\]

Hence \( g \) is strictly convex at every \([\phi]\in \tilde{W} \). Therefore \([\phi^*]\) is the unique minimizer of \( \tilde{E} \) over \( \tilde{W} \). \qed

Proof of Theorem 3 In the discrete case, we use the notation \( \phi = [\alpha; \beta; c] \in \mathbb{R}^{m+n+m} \) and \( E(\phi) = \langle [-\mu; -\nu; c], \phi \rangle + \epsilon(\zeta^\phi/\epsilon, 1_m, n) \), where the exponential is component-wise. Hence the Hessian of \( E(\phi) \) is
\[
    \nabla^2 E(\phi) = \epsilon^{-1} J^T \text{diag}(e^{J\phi/\epsilon}) J \succeq 0.
\]
If \( \phi^* \) is a minimizer and \( T_\phi W \cap \ker(J) = \{0\} \), then for any nonzero \( \psi \in T_\phi W \), we have \( J\psi \neq 0 \). Hence, \( \psi^T (\nabla^2 E) \psi = \epsilon^{-1} (J\psi)^T \text{diag}(e^{J\phi^*/\epsilon})(J\psi) > 0 \), since \( \text{diag}(e^{J\psi^*/\epsilon}) > 0 \). Therefore \( \phi^* = (\alpha^*, \beta^*, c^*) \) is the unique global minimizer. \qed
Proof of Corollary 4. Since prox$_L(\cdot)$ is the projection onto $C$, we know $W = \{\phi | L\phi = 0\}$ where $L$ denotes the matrix such that $Lc \in \mathbb{R}^{2n^2+n}$ stacks $c^T$ and diag$(c)$ vertically. Therefore $T_\phi W = W$ for all $\phi \in W$. Suppose $\psi = (\alpha, \beta, c) \in T_\phi W \cap \ker(J)$, then $Lc = 0$ and $c_{ij} = \alpha_i + \beta_j$ for all $i,j$. Therefore $c_{ii} = \alpha_i + \beta_i = 0$, and hence $\alpha_i = -\beta_i$, for all $i$. Moreover, there is

$$c_{ij} = \alpha_i + \beta_j = \alpha_i - \alpha_j.$$ 

Similarly, $c_{ji} = \alpha_j - \alpha_i$. Since $c$ is symmetric, we know that $\alpha_i - \alpha_j = \alpha_j - \alpha_i$, which implies $\alpha_i = \alpha_j$. Therefore $c_{ij} = \alpha_i + \beta_j = \alpha_i - \alpha_j = 0$ for all $i,j$, and hence $c = 0$. Moreover, $\alpha = -\beta = \xi 1^T$ for some constant $\xi \in \mathbb{R}$. So $T_\phi W \cap \ker(J) = \{(\xi 1_n, -\xi 1_n, 0) \in \mathbb{R}^{2n^2+n} | \xi \in \mathbb{R}\}$. By Theorem 2, if $\phi^* = (\alpha^*, \beta^*, c^*)$ solves (12), then $E$ attains minimum only at $\{\phi^* + \psi | \psi \in T_\phi W \cap \ker(J)\} \subset [\phi^*]$, which all yield the same cost matrix $c^*$.

\[\square\]

C Additional Related Work

Computational OT. The computation of OT has been a long standing challenge and is still under active research. Most existing work focus on the discrete setting (3), which is a special type of linear program (LP). However, the cubic computation complexity for general LP solvers prohibits fast numerical solution for large $m$ and $n$. In [7], a modification of (3) with an additional entropy regularization term in the objective function is proposed:

$$\min_{\pi \in \mathbb{R}^{m \times n}} \left\{ \langle \epsilon, \pi \rangle - \epsilon H(\pi) | \pi 1_n = \nu, \pi^T 1_m = \nu \right\},$$

(29)

where $H(\pi) := -\langle \pi, \log \pi - 1 \rangle = -\sum_{i,j} \pi_{ij} (\log \pi_{ij} - 1)$ is the (normalized) Shannon entropy of $\pi$, and $\epsilon > 0$ is a prescribed weight of the entropy regularization. Due to the entropy term, the troublesome inequality constraint in the original OT (3) is eliminated, and the objective function in (29) becomes strictly convex which admits unique solution. Moreover, the dual problem of (29) is unconstrained, which can be solved by a fast matrix scaling algorithm called the Sinkhorn (or Sinkhorn-Knopp) algorithm [7]. Sinkhorn algorithm has been the common approach to solve (regularized) OT (29) numerically in the discrete setting since then. Its property, convergence, and relation to the original OT (3) are also extensively studied, for instance, in [11] [22]. A more comprehensive treatment of computational OT in discrete setting, especially in the regularized form, can be found in [20]. Recently, the continuous OT problem is considered where the dual variables are parameterized as deep neural networks [23]. The sample complexity of OT is also studied in [14].

D Additional Experimental Results

D.1 More Experimental Results on Continuous Synthetic Data

With the true cost function $c^*(x, y) = \|x - y\|^p$ where $p = 0.5, 1, 2, 3$, we apply Algorithm 2 to recover the cost $c^*$. We plot the true cost functions and recovered cost functions in Figure 4. We also test Algorithm 2 to recover an asymmetric cost function $c^*(x, y) = \|x - 2y\|^2$. The comparison between the true cost and recovered cost is shown in Figure 3.

For parameters, we set the batch size equal to 5000 and tune the learning rate from $1e^{-3}$ to $1e^{-5}$ for all of these experiments. The maximum number of iterations is set to be 5000. We set $\epsilon = 1e^{-1}$. For network architectures, we set the input layer of $c_{ij}$ to take $\xi \in \mathbb{R}^2$ where $\xi_i = |x_i - y_i|$ for all $i$ when cost function is symmetric. For more stable training, we also try other variations like $|x_i - y_i|^2, x_i - y_i$. For asymmetric cost $c^*(x, y) = \|x - 2y\|^2$, we simply set $\xi_i = |x_i - 2y_i|$ for all $i$. For all neural networks in this work, the input and output layer has only one unit, and each of three hidden layers consists of 20 units with tanh function as activation function. Only the output layer of the cost network uses ReLU activation function.

D.2 Experimental Results on Marriage Data
Figure 3: Asymmetric Cost function recovered by Algorithm 2

Figure 4: Symmetric Cost function recovered by Algorithm 2
Table 2: The estimation of the affinity matrix

| Husbands/Wives Education | Education | Height | Weight | Health | Irresponsible | Disciplined | Ordered | Clumsy | Detail |
|-------------------------|-----------|--------|--------|--------|---------------|-------------|---------|--------|--------|
| Education               | 0.728     | 0.029  | 0.042  | 0.038  | -0.128        | -0.055      | -0.128  | -0.169 | -0.055 |
| Height                  | -0.180    | 0.085  | 0.055  | 0.057  | -0.097        | -0.024      | 0.157   | 0.071  | 0.001  |
| Weight                  | -0.120    | 0.052  | 0.034  | 0.065  | -0.087        | -0.038      | -0.118  | 0.071  | 0.001  |
| Health                  | -0.099    | -0.065 | -0.023 | -0.054 | -0.160        | 0.010       | 0.011   | 0.039  | 0.002  |
| Irresponsible           | 0.081     | 0.028  | 0.034  | 0.065  | 0.163         | -0.041      | 0.014   | -0.012 | 0.001  |
| Disciplined             | -0.129    | 0.038  | 0.028  | 0.054  | 0.193         | 0.130       | 0.012   | 0.049  | -0.036 |
| Ordered                 | -0.060    | 0.030  | 0.022  | 0.034  | -0.054        | 0.086       | 0.058   | 0.013  | -0.028 |
| Clumsy                  | -0.137    | 0.030  | 0.022  | 0.028  | 0.054         | 0.022       | 0.017   | 0.035  | 0.035  |
| Detail                  |           |        |        |        |               |             |         |        |        |