Partial Optimal Transport for Support Subset Selection

Anonymous authors
Paper under double-blind review

Abstract

In probabilistic terms, optimal transport aims to find a joint distribution that couples two distributions, which minimizes the cost of transforming one distribution to another. Any feasible coupling necessarily maintains the support of both distributions. However, maintaining the entire support is not ideal when only a subset of one of the distributions, namely the source, is assumed to align with the other target distribution. For these cases, which are common in machine learning applications, we propose to relax the constraints on the joint distribution allowing it to underrepresent a subset of the source by overrepresenting other subsets of the source by a constant factor. This is a special case of partial optimal transport. In the discrete distribution case, such as in the case of two samples from continuous random variables, optimal transport with the relaxed constraints is a linear program. When sufficiently relaxed, the solution has a source marginal with only a subset of its original support. We demonstrate the usefulness of this support subset selection in applications such as color transfer, partial point cloud alignment, and semisupervised machine learning, where a part of data is curated to have reliable labels and another part is unlabeled or has unreliable labels. Our experiments show that optimal transport under the relaxed constraint can improve the performance of these applications by allowing for more flexible alignment between distributions.

1 Introduction

Measuring, and subsequently minimizing, the dissimilarity between two distributions or data samples are ubiquitous tasks in machine learning. Recently, the theory of optimal transport and the family of Wasserstein distances have seen applications across the spectrum of machine learning problems including computer vision (Solomon et al., 2014; Kolouri et al., 2017; Rabin et al., 2011; Garg et al., 2020), generative modeling (Arjovsky et al., 2017; Gulrajani et al., 2017; Salimans et al., 2018; Genevay et al., 2018; Tolstikhin et al., 2018; Kolouri et al., 2018; Deshpande et al., 2018), natural language processing (Xu et al., 2018), and domain adaptation (Kirchmeyer et al., 2022). Optimal transport is widely applicable since it combines the statistical and geometric aspects of data and provides a correspondence to couple two samples or distributions.

However, a limitation of standard optimal transport is the strict constraint on the complete transfer of mass between the two distributions being compared. This can be problematic in cases where such a transfer is not necessary or desirable, such as when dealing with distributions with different support, over- or under-representation, or the presence of outliers in a portion of the data. In order to deal with this kind of unbalanced scenarios partial optimal transport problems (Figalli, 2010; Caffarelli & McCann, 2010; Bonneel & Coeurjolly, 2019; Chapel et al., 2020) have been proposed. Partial optimal transport relaxes the marginal constraints on the transport plan to inequalities, allowing transportation plans that cover only a fraction of the total mass. The marginal constraints can also be relaxed to divergence-based regularizations, such as Kullback-Leibler divergence and total-variation distance (Chizat et al., 2018; Peyré et al., 2019). The solution of partial optimal transport often selects a subset (also known as an active region) of the support. This property is exploited in the partial Wasserstein covering problem, which has applications in active learning (Kawano et al., 2022).

In practice, a key limiting factor is the scalability of solving the linear program corresponding to partial or standard optimal transport for large data sets. While efficient gradient based methods cannot be applied to it directly, a number of regularization based remedies, including entropic (Cuturi, 2013; Cuturi & Peyré, 2018), quadratic, and group-lasso regularization (Flamary et al., 2016; Blondel et al., 2018) have been shown to give approximate solutions. In particular, the Sinkhorn algorithm provides a solution to the entropically regularized standard optimal transport problem. The Sinkhorn algorithm has been widely applied due to its simple implementation consisting of alternating projections to the feasible sets of marginal constraints. In Chizat et al. (2018), the authors propose to use Dykstra's algorithm[1]to solve the entropically regularized partial optimal transport problem.

In this paper, we study a discrete case of partial optimal transport (Figalli, 2010; Caffarelli & McCann, 2010) where the solution is a joint distribution with one fixed marginal and one marginal that is constrained to be pointwise less than or equal to a constant factor $c \geq 1$ of the original, typically uniform, marginal. We propose an entropically regularized version to efficiently find an approximate solution that we solve with an algorithm that combines Sinkhorn-like projections with the proximal gradient method. To yield solutions closer to the original, unregularized, linear program, we follow the inexact Bergman proximal gradient-based method adopted by Xie et al. (2020) in the context of the classical optimal transport problem. The contributions of this paper are the following:

- We motivate and formulate a partial optimal transport problem for selecting a subset of a source distribution, which may exhibit overrepresentation or outliers, for a fixed target distribution.
- We develop an accelerated proximal gradient method-based algorithm to solve the entropically regularized version and incorporate the inexact Bregman proximal method-based approach from Xie et al. (2020) to mitigate the effects of entropic regularization.
- We demonstrate the resultant algorithm for color adaptation, partial distribution alignment, partial point cloud registration problems, and positive-unlabeled learning (Bekker & Davis, 2020).
- We incorporate the subset selection-based approach into a semi-supervised loss function for training a neural network-based classifier, which computes the optimal transport based on the learning representation.

2 Methodology

In Section 2.1, relevant preliminaries related to discrete optimal transport along with formulation of subset selection problem are discussed. In Section 2.2, the entropically regularized support subset selection problem and an algorithm to solve it are discussed. In Section 2.3, an inexact Bregman proximal-based approach to better approximate the solution of the unregularized support selection problem is detailed.

Notation: The set of the first n natural numbers $\{1, 2, \ldots, n\}$, is denoted by $[n]$. The set of integers is denoted by $\mathbb{Z}$. The ceiling function defined on real numbers $x \in \mathbb{R}$ is $\lceil x \rceil = \min\{n \in \mathbb{Z} : n \geq x\}$. The floor function defined on real numbers $x \in \mathbb{R}$ is $\lfloor x \rfloor = \max\{n \in \mathbb{Z} : n \leq x\}$. The n-dimensional real vector space is denoted by $\mathbb{R}^n$. Vectors are typeset in lowercase bold ($\boldsymbol{x}$); matrices are in uppercase bold ($\boldsymbol{X}$); and bold is dropped when an element are referenced by subscripts (x_i, X_{ij}). When needed for clarity, elements will be referenced by subscripts on square brackets ($[\boldsymbol{x}_1]_i, [\boldsymbol{X}_2]_{ij}$). The set of non-negative vectors in $\mathbb{R}^n$, known as the non-negative orthant, is denoted by $\mathbb{R}^n_+$. The n-dimensional vector with all elements equal to unity is denoted by $\mathbf{1}_n$ and the m-by-n matrix with all unity elements is denoted by $\mathbf{1}_{m\times n}$. For vectors and matrices, the symbol $\preccurlyeq$ denotes element-wise less than or equal to, and $\succcurlyeq$ denotes element-wise greater than or equal to. The set denoted by $\boldsymbol{\Delta}_n = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the probability simplex. The element-wise product for vectors and matrices is denoted by the $\odot$ symbol. The element-wise division for vectors and matrices is denoted by the $\oslash$ symbol. The diagonal operator is a matrix valued map $\boldsymbol{D} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, such that $[\boldsymbol{D}(\boldsymbol{x})]_{ii} = x_i \ \forall \ i \in [n]$ and $[\boldsymbol{D}(\boldsymbol{x})]_{ij} = 0 \ \forall \ i \neq j \in [n]$. For $\boldsymbol{x} \in \mathbb{R}^n$, the ℓ_1, ℓ_2 and ℓ_∞ norms are given by $\|\boldsymbol{x}\|_1 = \sum_i |x_i|$, $\|\boldsymbol{x}\|_2 = (\sum_i |x_i|^2)^{\frac{1}{2}}$ and $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$, respectively. Both, the Euclidean inner-product for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ given by $\sum_i x_i y_i$, and the Frobenius inner-product for $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m\times n}$ given by $\sum_{i=1}^m \sum_{j=1}^n X_{ij} Y_{ij}$, are denoted by $\langle \cdot, \cdot \rangle$. The element-wise exponent of a vector or

---

[1] Dykstra's algorithm can find solutions at the intersection of convex, not necessarily affine, sets.

a matrix is denoted by $\mathbf{exp}(\cdot)$ and the element-wise logarithm of a vector or a matrix is denoted by $\mathbf{log}(\cdot)$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, its non-negative part is represented by $\boldsymbol{X}_+$ or $[\boldsymbol{X}]_+$, with matrix components given by $\big[\boldsymbol{X}_+\big]_{ij} = \max\{X_{ij}, 0\}\ \forall\ i \in [m],\ j \in [n]$. Similarly, the non-positive part of matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is denoted by $\boldsymbol{X}_-$ or $[\boldsymbol{X}]_-$ and contains matrix elements $\big[\boldsymbol{X}_-\big]_{ij} = \min\{X_{ij}, 0\}\ \forall\ i \in [m]$ and $j \in [n]$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, both $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ denote the non-negative and non-positive parts respectively. We denote the indicator function of the singleton set $\{\boldsymbol{z}\}$ as $\delta_{\boldsymbol{z}}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} = \boldsymbol{z} \\ 0, & \boldsymbol{x} \neq \boldsymbol{z} \end{cases}$. The set of vectors $\{\boldsymbol{e}_i\}_{i=1}^n$ form the standard basis for $\mathbb{R}^n$, where $[\boldsymbol{e}_i]_i = 1$ and $[\boldsymbol{e}_i]_j = 0$ for $i \neq j$.

2.1 Problem Formulation

We consider the discrete optimal transport between two weighted samples of size m and n corresponding to random variables $X \sim \mu$ defined on $\{\boldsymbol{x}^{(i)}\}_{i=1}^m \subset \mathbb{R}^d$ and $Y \sim \nu$ defined on $\{\boldsymbol{y}^{(j)}\}_{j=1}^n \subset \mathbb{R}^d$, with probability measures $\mu = \sum_{i=1}^m \mu_i \delta_{\boldsymbol{x}^{(i)}}$ and $\nu = \sum_{j=1}^n \nu_j \delta_{\boldsymbol{y}^{(j)}}$ for probability masses $\boldsymbol{\mu} \in \boldsymbol{\Delta}_m$ (with $\{\mu_i = \mu(\boldsymbol{x}^{(i)})\}_{i=1}^m$) and $\boldsymbol{\nu} \in \boldsymbol{\Delta}_n$ (with $\{\nu_j = \nu(\boldsymbol{y}^{(j)})\}_{j=1}^n$), respectively. Let $\mathrm{d} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ denote a distance function. In practice, this is often the Euclidean distance matrix between the points $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$. Given $1 \leq p \leq \infty$, the p-Wasserstein distance (to the p-power) between μ and ν is expressed in terms of the cost matrix $\boldsymbol{M}$, where $M_{ij} = \mathrm{d}^p(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(j)})\ \forall i \in [m],\ j \in [n]$ is the cost associated with transporting $\boldsymbol{x}^{(i)}$ to $\boldsymbol{y}^{(j)}$, as

$$\begin{aligned} \mathcal{W}_p^p(\mu, \nu) := \min_{\boldsymbol{P} \succcurlyeq 0} \quad & \langle \boldsymbol{P}, \boldsymbol{M} \rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu},\ \boldsymbol{P}^\top \mathbf{1}_m = \boldsymbol{\nu}, \end{aligned} \tag{1}$$

where $\boldsymbol{P}$ is the transport map and $\mathbf{1}_m^\top \boldsymbol{P}\mathbf{1}_n = \mathbf{1}_m^\top \boldsymbol{\mu} = \boldsymbol{\nu}^\top \mathbf{1}_n = 1$. The constraints ensure that any solution $\boldsymbol{P}^*$ is a joint distribution that couples the target marginal $\boldsymbol{\mu}$ and the source marginal $\boldsymbol{\nu}$. (In the computational optimal transport literature, μ is referred to as the target and ν as the source.) Therefore, any Wasserstein distance requires the complete mass transfer between a fixed source and target.

In partial optimal transport (Figalli, 2010), the marginal equality constraints are replaced by the inequalities $\boldsymbol{P}\mathbf{1}_n \preccurlyeq \boldsymbol{\mu}$, $\boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\nu}$, and the equality $\mathbf{1}_m^\top \boldsymbol{P}\mathbf{1}_n = \rho$; $\boldsymbol{\mu} \in \mathbb{R}_+^m$ and $\boldsymbol{\nu} \in \mathbb{R}_+^n$ may not have equal mass; and the transport map need only transport a fraction of the total mass $\rho \in [0, \min\{\|\boldsymbol{\mu}\|_1, \|\boldsymbol{\nu}\|_1\}]$. Motivated by machine learning scenarios with a trusted target sample of data and an additional source of data which cannot be assumed to be of uniform quality, we focus on a special case, where the target is fixed with $\|\boldsymbol{\mu}\|_1 = 1$, ensuring the total mass constraint, but allow the source mass to redistribute among the source points to a new marginal $\boldsymbol{\nu}^* \leq c\boldsymbol{\nu}$, where $c \geq 1$ is a scaling factor. The relaxed constraint is $\boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq c\boldsymbol{\nu}$. The resulting partial optimal transport problem[2] is

$$\begin{aligned} \min_{\boldsymbol{P} \succcurlyeq 0} \quad & \langle \boldsymbol{P}, \boldsymbol{M} \rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu},\ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq c\boldsymbol{\nu}. \end{aligned} \tag{2}$$

Let $\boldsymbol{P}_c^*$ denote an optimal solution, then the source's new mass is $\boldsymbol{\nu}_c^* = \boldsymbol{P}_c^{*\top} \mathbf{1}_m$. Since $\mathbf{1}_m^\top \boldsymbol{\mu} = \mathbf{1}_m^\top \boldsymbol{P}_c^* \mathbf{1}_n = 1$, $\|\boldsymbol{\nu}_c^*\|_1 = 1$. Intuitively, this problem allows the new mass of some source points that have relatively lower cost to increase by a factor of c of the original mass, which enables higher cost source points to have less or even zero mass. In other words, due to total unit mass constraint, the mass increment at one source point results in its decrement at other source points. The subset of the source points selected is $\mathrm{supp}(\boldsymbol{\nu}_c^*)$, where $\mathrm{supp}(\cdot)$ indicates the support of a vector, i.e., the indices of the points with non-zero mass.[3]

---

[2] An equivalent set of constraints are $\boldsymbol{P}\mathbf{1}_n \preccurlyeq \boldsymbol{\mu},\ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq c\boldsymbol{\nu}, \mathbf{1}_m^\top \boldsymbol{P}\mathbf{1}_n = 1$.

[3] Support subset selection can be extended to both marginals, by scaling the total mass of the transport plan to unity, which results in mass assignments constraints for both source and target being scaled by the same constant factor $\frac{1}{\rho}$:

$$\begin{aligned} \min_{\boldsymbol{P} \succcurlyeq 0} \quad & \langle \boldsymbol{P}, \boldsymbol{M} \rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n \preccurlyeq \boldsymbol{\mu},\ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\nu}, \mathbf{1}_m^\top \boldsymbol{P}\mathbf{1}_n = \rho \end{aligned} \quad \overset{\boldsymbol{T} = \frac{1}{\rho}\boldsymbol{P}}{\iff} \quad \begin{aligned} \min_{\boldsymbol{T} \succcurlyeq 0} \quad & \langle \boldsymbol{T}, \boldsymbol{M} \rangle \\ \text{s.t.} \quad & \boldsymbol{T}\mathbf{1}_n \preccurlyeq \frac{1}{\rho}\boldsymbol{\mu},\ \boldsymbol{T}^\top \mathbf{1}_m \preccurlyeq \frac{1}{\rho}\boldsymbol{\nu},\ \mathbf{1}_m^\top \boldsymbol{T}\mathbf{1}_n = 1. \end{aligned}$$

To explore the relaxed constraint set, we consider the case of a uniformly distributed mass $\boldsymbol{\nu} = \frac{1}{n}\mathbf{1}_n$ and express the constraint as $\boldsymbol{P}^\top \mathbf{1}_m \leq \frac{1}{L}\mathbf{1}_n$, where $0 < L \leq n$ and $c = \frac{n}{L}$. For a fixed value of L, the set of feasible source marginal distributions form a polyhedral set $\Xi_n^{(L)} \subseteq \boldsymbol{\Delta}_n$ bounded by linear inequalities parameterized by L. The set $\Xi_3^{(L)}$ for different values of L are given in Figure 1.

Remark. *For the partial optimal transport problem 2 with $n > 2$ and $\boldsymbol{\nu} = \frac{1}{n}\mathbf{1}_n$, by defining $L = \frac{n}{c}$, the inequality constraint can be written as $\boldsymbol{P}^\top \mathbf{1}_m \leq \frac{1}{L}\mathbf{1}_n$. Extreme points of $\Xi_n^{(L)}$ can be characterized as follows:*

- *For $0 < L \leq 1$, the entire probability simplex $\boldsymbol{\Delta}_n$ is feasible due to the fact that in this case the vertices of the probability simplex correspond to extreme points of feasible set.*
- *For $1 < L < 2$ and $n > 2$, the feasible set $\Xi_n^{(L)}$ has $n(n-1)$ number of vertices, which can can be written as as convex combination $\frac{1}{L}\boldsymbol{e}_i + (1 - \frac{1}{L})\boldsymbol{e}_j$, where $i, j \in [n]$ and $i \neq j$.*
- *For $L = 2$, the feasible set $\Xi_n^{(2)}$ has $\frac{n(n-1)}{2}$ vertices given as $\frac{1}{2}(\boldsymbol{e}_i + \boldsymbol{e}_j)$ for $i \neq j$.*
- *More generally, the number of extreme points of the feasible set $\Xi_n^{(L)}$ is*

  $$\frac{n!}{\lfloor L\rfloor! \lceil 1 - \frac{\lfloor L\rfloor}{L}\rceil! (n - \lfloor L\rfloor - \lceil 1 - \frac{\lfloor L\rfloor}{L}\rceil)!},$$

  with the vertices given as the set of possible multi-set permutations of the vector:

  $$\begin{bmatrix} \overbrace{\frac{1}{L} \quad \frac{1}{L} \quad \cdots \quad \frac{1}{L}}^{\lfloor L\rfloor \ terms} & 1 - \frac{\lfloor L\rfloor}{L} & \overbrace{0 \quad 0 \quad \cdots \quad 0}^{n-1-\lfloor L\rfloor \quad terms} \end{bmatrix}^\top.$$

Since the amount of mass to be transported is constant, when the scaling parameter c is increased (or equivalently, when the value of L is decreased) starting from $c = 1$, the behavior of the coupling $\boldsymbol{P}_c^*$ in the transportation problem is affected. This behavior depends on the structure of the cost matrix $\boldsymbol{M}$ and leads to a redistribution of mass, assigning more mass to certain points and less to others. At $c = 1$, all source constraints are active, meaning that all constraints in the problem are considered. However, as the value of c is increased, constraints become inactive, which constraints depends on the structure of the cost matrix $\boldsymbol{M}$ and the distributions.Once c reaches c^*, all inequality constraints are inactive and can be discarded. After this breakpoint c^*, any further increments in c do not affect the resulting transport plans. In other words, for $c \geq c^*$, the transport plan $\boldsymbol{P}_c^*$ remains the same as $\boldsymbol{P}_{c^*}^*$. The exact value of c^* can be determined analytically. The analytical expression for c^* depends on the properties of the cost matrix $\boldsymbol{M}$ and the constraints involved,

$$c^* = \max_{j\in[n]} \frac{\sum_j Q_{ij}}{\nu_j}, \tag{3}$$

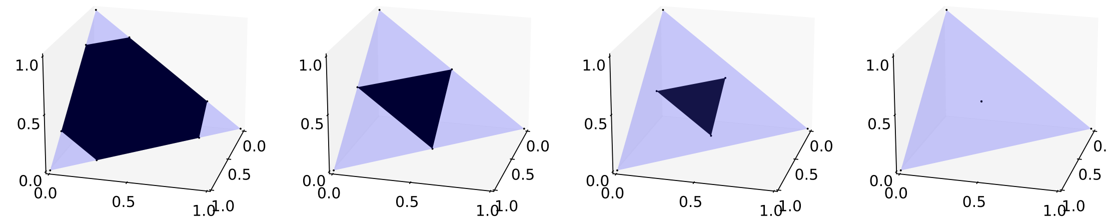
where the matrix $\boldsymbol{Q}$ is found by nearest neighbor search,

$$Q_{ij} = \begin{cases} \mu_i, & j \in \arg\min_{k\in[n]} M_{ik} \\ 0, & \text{otherwise} \end{cases}, \quad i \in [m], j \in [n]. \tag{4}$$

In cases without an initial source mass $\boldsymbol{\nu}$, a designer can provide an upper-bound on the mass assignments to source points by $\boldsymbol{\zeta} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\zeta}\|_1 \geq 1$ (equality corresponds to standard optimal transport), which adds the flexibility in designing partial optimal transport problems that allow variable ranges of masses for the source points. For target distribution μ and source upper-bounding measure $\zeta = \sum_{j=1}^m \zeta_j \delta_{\boldsymbol{y}^{(j)}}$, which is necessarily not a probability measure, the support subset selection problem can be stated as

$$\begin{aligned} \mathcal{S}_p(\mu, \zeta) := \min_{\boldsymbol{P}\succcurlyeq 0} \quad & \langle \boldsymbol{P}, \boldsymbol{M}\rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu},\ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\zeta}, \end{aligned} \tag{5}$$

and related to the p-Wasserstein distance by $\mathcal{S}_p(\mu, c\nu) \leq \mathcal{W}_p^p(\mu, \nu)$ for $c \geq 1$. In the next section, we will discuss the entropic regularization of the support subset selection problem and algorithms to solve it.



(a) (b) (c) (d)

Figure 1: The feasible set $\Xi_3^{(L)}$ for different values of L. For $0 < L \leq 1$ the whole probability simplex is feasible. (a) $L = 1.3$, six vertices will exist for $1 < L < 2$. (b) $L = 2$ yields 3 vertices. (c) $L = 2.2$, $2 < L < 3$ also give 3 vertices. (d) $L = 3$ yields a singleton set corresponding to a uniform distribution.

2.2 Support Subset Selection with Entropic Regularization

The support subset selection problem 5 is a linear program, which can be exactly solved by the simplex method or interior point methods, both of which do not scale well with the dimension of transport map (Cuturi, 2013). In order to apply efficient gradient based optimization to linear programs, entropic regularization has been added to linear objective functions (Li & Fang, 1997). Cuturi applied entropic regularization to the optimal transport problem to efficiently approximate the Wasserstein distance using Sinhkorn's matrix scaling algorithm (Cuturi, 2013; Sinkhorn, 1964). For fixed target distribution μ and upper-bounding source measure ζ with mass $\boldsymbol{\zeta} \in \mathbb{R}_+^n$, the proposed entropically regularized support subset selection problem is

$$\begin{aligned} \mathcal{S}_p^{(\gamma)}(\mu, \zeta) := \min_{\boldsymbol{P} \succcurlyeq 0} \quad & \langle \boldsymbol{P}, \boldsymbol{M} \rangle + \gamma \langle \boldsymbol{P}, \mathbf{log}(\boldsymbol{P}) - \mathbf{1}_{m \times n} \rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu}, \ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\zeta}, \end{aligned} \tag{6}$$

where γ is the regularization parameter. It is important to mention that the regularization term $\langle \boldsymbol{P}, \mathbf{log}(\boldsymbol{P}) - \mathbf{1}_{m \times n} \rangle$ is negative entropy, which is 1-strongly convex with respect to the ℓ_1 and ℓ_2 norms in the feasible set: $\{\boldsymbol{P} : \boldsymbol{P} \succcurlyeq 0, \ \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu}, \ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\zeta}\}$ (Beck, 2017). The Lagrangian of equation 6 is

$$\mathcal{L}(\boldsymbol{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \left\langle \boldsymbol{P}, \ \boldsymbol{M} + \gamma(\mathbf{log}\boldsymbol{P} - \mathbf{1}_m \mathbf{1}_n^\top) + \boldsymbol{\alpha} \mathbf{1}_n^\top + \mathbf{1}_m \boldsymbol{\beta}^\top \right\rangle - \langle \boldsymbol{\alpha}, \boldsymbol{\mu} \rangle - \langle \boldsymbol{\beta}, \boldsymbol{\zeta} \rangle, \tag{7}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are the Lagrange multipliers. Note that we have adopted the approach of Cuturi (2013) and do not explicitly enforce the simplex constraint on $\boldsymbol{P}$, which would lead to the log-sum-exp formulation as in Cuturi & Peyré (2018); Lin et al. (2022); Guminov et al. (2021). Taking the element-wise derivative of $\mathcal{L}$ with respect to $\boldsymbol{P}$ and setting it to zero yields

$$\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{D}\Big(\mathbf{exp}(-\boldsymbol{\alpha}/\gamma)\Big) \mathbf{exp}(-\boldsymbol{M}/\gamma) \boldsymbol{D}\Big(\mathbf{exp}(-\boldsymbol{\beta}/\gamma)\Big). \tag{8}$$

Substituting $\tilde{\boldsymbol{P}}$ back into Lagrangian results in the dual problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad & \left\{ f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \gamma \mathbf{1}_m^\top \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \mathbf{1}_n + \langle \boldsymbol{\alpha}, \boldsymbol{\mu} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\zeta} \rangle \right\} \\ \text{s.t.} \quad & \boldsymbol{\beta} \succcurlyeq 0. \end{aligned} \tag{9}$$

The constraint set $\boldsymbol{\beta} \succcurlyeq 0$ is closed. The indicator function of the constraint set $\boldsymbol{\beta} \succcurlyeq 0$ is defined as

$$I_+(\boldsymbol{\beta}) = \begin{cases} 0, & \text{for } \boldsymbol{\beta} \succcurlyeq 0 \\ \infty, & \text{otherwise.} \end{cases}$$

Therefore, we can convert problem 9 into an unconstrained composite optimization problem,

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad f(\boldsymbol{\alpha}, \boldsymbol{\beta}) + I_+(\boldsymbol{\beta}). \tag{10}$$

Since $f(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is convex and $I_+(\boldsymbol{\beta})$ is proper, closed, and convex, we can apply the accelerated proximal gradient algorithm to solve the composite optimization problem. Defining the Gibbs kernel $\boldsymbol{K} = \mathbf{exp}(-\frac{\boldsymbol{M}}{\gamma})$, the partial gradient $\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is

$$\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\zeta} - \mathbf{exp}(-\boldsymbol{\beta}/\gamma) \odot \left(\boldsymbol{K}^\top \mathbf{exp}(-\boldsymbol{\alpha}/\gamma)\right). \tag{11}$$

The proximal projection for the non-negative orthant's indicator function I_+ is computed by setting any negative entries to zero.

Algorithm 1 outlines our accelerated proximal gradient algorithm to solve the dual form of subset selection problem with entropic regularization. Similar to the standard entropically regularized optimal transport problem, the dual variable $\boldsymbol{\alpha}$ is updated with a Sinkhorn-like update at iteration k as

$$\boldsymbol{\alpha}^{(k+1)} = \gamma \mathbf{log}\Big(\left(\boldsymbol{K} \mathbf{exp}(-\boldsymbol{\beta}^{(k)}/\gamma)\right) \oslash \boldsymbol{\mu} \Big). \tag{12}$$

Whereas, $\boldsymbol{\beta}$ is updated at iteration k using accelerated proximal gradient based update rule (Beck, 2017; Beck & Teboulle, 2009) with step size $\frac{1}{\eta_s^{(k)}}$

$$\boldsymbol{\beta}^{(k+1)} = \left[\boldsymbol{\xi}^{(k)} - \frac{1}{\eta_s^{(k)}} \nabla_{\boldsymbol{\xi}} f(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\xi}^{(k)})\right]_+ = \left[\boldsymbol{\xi}^{(k)} - \frac{1}{\eta_s^{(k)}} \left(\boldsymbol{\zeta} - \mathbf{exp}(-\boldsymbol{\xi}^{(k)}/\gamma) \odot \boldsymbol{K}^\top \mathbf{exp}(-\boldsymbol{\alpha}^{(k+1)}/\gamma)\right)\right]_+, \tag{13}$$

which uses equation 11 to compute the gradient with respect to the variable $\boldsymbol{\xi}^{(k)}$ before applying the proximal operator $[\cdot]_+$. In Algorithm 1, we use a constant step size $\frac{1}{\eta_s^{(k)}} = \gamma$ (since the primal problem 9 is γ-strongly convex and its semi-dual is $\frac{1}{\gamma}$-Lipschitz smooth Cuturi & Peyré (2016), see Appendix C for details), but another option is a backtracking line search (Beck, 2017).

By incorporating the update of $\boldsymbol{\alpha}^{(k+1)}$ as in equation 12 directly into the gradient $\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})$, the algorithm can be written entirely in terms of $\boldsymbol{\beta}^{(k)}$. This shows that Algorithm 1 consists of standard accelerated proximal gradient updates and has a $\mathcal{O}(1/k^2)$ convergence rate. As shown by Beck (2017), the required number of iterations k_ε to achieve an ε sub-optimal solution of equation 10 using Algorithm 1 is upper-bounded as

$$k_\varepsilon + 1 \leq \sqrt{\frac{2}{\gamma\varepsilon}} \cdot \|\boldsymbol{\beta}^{(\mathrm{i})} - \boldsymbol{\beta}^*\|, \tag{14}$$

where $\boldsymbol{\beta}^{(\mathrm{i})}$ is the initialization and $\boldsymbol{\beta}^*$ is the optimal solution.

If Algorithm 1 is allowed to run until its convergence, it returns the optimal coupling $\boldsymbol{P}^*$, but in practice, if Algorithm 1 does not reach convergence, $\hat{\boldsymbol{P}}^* \in \mathbb{R}_+^{m \times n}$ may violate the primal constraints on its marginals as these are not ensured by an approximate dual solution. For some applications a projection of $\hat{\boldsymbol{P}}^*$ to satisfy one or both of the marginal constraints may be required. While not explored in this paper due to the additional computational cost, projection to the feasible set can be done by the fast dual proximal gradient (FDPG) algorithm from Beck & Teboulle (2014); Beck (2017) in conjunction with Altschuler et al. (2017)'s Algorithm-2.

2.3 Support Subset Selection with the Bregman Proximal-point Method

Although the entropic regularization of the coupling distribution enables an efficient approximation of the support subset selection problem 5, the entropic regularization yields denser coupling distributions as compared to the unregularized problem. The denser coupling distributions result in a new marginal mass $\boldsymbol{\nu}^*$ that is also not sparse, yielding complete support rather than a subset of the source points. Different approaches have been proposed to maintain the computational benefits of entropic regularization while yielding solutions closer to the unregularized problem (Schmitzer, 2019; Xie et al., 2020).

In this paper, we follow Xie et al. (2020) and adapt an inexact Bregman proximal gradient for the negative entropy function (Teboulle, 1992) to the partial optimal transport case. The Bregman proximal gradient approach uses a Bregman proximal operator by replacing the usual Euclidean distance in the proximal

Algorithm 1: Fast proximal gradient algorithm to solve the dual problem 10 of the entropically regularized support subset selection problem 6

```
Inputs  : Target distribution $\boldsymbol{\mu}$, mass assignment bounding vector $\boldsymbol{\zeta}$, cost matrix $\boldsymbol{M}$,
          entropic regularization parameter $\gamma$, initial dual variable, $\boldsymbol{\beta}^{(\mathrm{i})} \in \mathbb{R}_+^n$, and iteration
          limit max-iter.
Outputs : $\hat{\boldsymbol{P}}^*$, which approaches the optimal coupling $\boldsymbol{P}^*$
Function EntropicSS($\boldsymbol{\mu}, \boldsymbol{\zeta}, \boldsymbol{\beta}^{(\mathrm{i})}, \gamma, \boldsymbol{M}$, max-iter):
    Initialization: $t_0 \leftarrow 1$, $\boldsymbol{\beta}^{(0)} \leftarrow \boldsymbol{\beta}^{(\mathrm{i})}$, $\boldsymbol{\xi}^{(0)} \leftarrow \boldsymbol{\beta}^{(\mathrm{i})}$, $\boldsymbol{K} \leftarrow \mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{M})$
    for $k \leftarrow 0$ to max-iter $- 1$ do
        $\boldsymbol{\alpha}^{(k+1)} \leftarrow \gamma \mathbf{log}\Big( \big(\boldsymbol{K}\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{\beta}^{(k)})\big) \oslash \boldsymbol{\mu} \Big)$
        $\boldsymbol{\beta}^{(k+1)} \leftarrow \Big[\boldsymbol{\xi}^{(k)} - \gamma \nabla_{\boldsymbol{\xi}} f(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\xi}^{(k)})\Big]_+$
        $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
        $\boldsymbol{\xi}^{(k+1)} \leftarrow \boldsymbol{\beta}^{(k+1)} + (\frac{t_k - 1}{t_{k+1}})(\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)})$
    end
    $\boldsymbol{\alpha}^* \leftarrow \boldsymbol{\alpha}^{(k+1)}$
    $\boldsymbol{\beta}^* \leftarrow \boldsymbol{\beta}^{(k+1)}$
    $\hat{\boldsymbol{P}}^* \leftarrow \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \boldsymbol{D}\big(\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{\alpha}^*)\big)\boldsymbol{K}\boldsymbol{D}\big(\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{\beta}^*)\big)$
return $\hat{\boldsymbol{P}}^*$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$
```

method (Parikh & Boyd, 2014) with the Bregman divergence associated with a continuously differentiable and strictly convex function (Beck, 2017). In the case of negative entropy, the Bregman divergence is the Kullback–Leibler divergence. Let $\phi(\boldsymbol{P}) = \langle \boldsymbol{P}, \mathbf{log}(\boldsymbol{P}) - \mathbf{1}_{m\times n}\rangle$ denote the negative entropy of a non-negative matrix $\boldsymbol{P}$, then given a non-negative matrix $\boldsymbol{P}^{'} \in \mathbb{R}_+^{m\times n}$, the Bregman divergence is

$$\mathcal{B}_\phi(\boldsymbol{P}||\boldsymbol{P}^{'}) := \langle \boldsymbol{P}, \mathbf{log}(\boldsymbol{P} \oslash \boldsymbol{P}^{'})\rangle - \langle \boldsymbol{P}, \mathbf{1}_{m\times n}\rangle + \langle \boldsymbol{P}^{'}, \mathbf{1}_{m\times n}\rangle. \tag{15}$$

For the subset selection problem 5, the Bregman proximal point evaluated at $\boldsymbol{P}^{(t)}$, is

$$\begin{aligned} \mathbf{Breg\text{-}prox}_\phi(\boldsymbol{P}^{(t)}) = \underset{\boldsymbol{P} \succcurlyeq 0}{\arg\min} \quad & \langle \boldsymbol{P}, \boldsymbol{M}\rangle + \lambda \mathcal{B}_\phi(\boldsymbol{P}||\boldsymbol{P}^{(t)}) \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu}, \ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\zeta}, \end{aligned} \tag{16}$$

where λ is positive scaling factor. By substituting $\mathcal{B}_\phi(\boldsymbol{P}||\boldsymbol{P}^{(t)})$ from equation 15 into equation 16 and ignoring the constant term $\langle \boldsymbol{P}^{(t)}, \mathbf{1}_{m\times n}\rangle$ we obtain

$$\begin{aligned} \mathbf{Breg\text{-}prox}_\phi(\boldsymbol{P}^{(t)}) = \underset{\boldsymbol{P} \succcurlyeq 0}{\arg\min} \quad & \langle \boldsymbol{P}, \boldsymbol{M} - \mathbf{log}(\boldsymbol{P}^{(t)})\rangle + \lambda \langle \boldsymbol{P}, \mathbf{log}(\boldsymbol{P}) - \mathbf{1}_{m\times n}\rangle \\ \text{s.t.} \quad & \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu}, \ \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq \boldsymbol{\zeta}, \end{aligned} \tag{17}$$

which corresponds to the entropically regularized subset selection problem 6 with parameters γ and $\boldsymbol{M}$ in 6 replaced by λ and $\boldsymbol{M} - \mathbf{log}(\boldsymbol{P}^{(t)})$, respectively. Thus, solving the entropy-regularized support subset selection problem is required to solve an iteration of the proximal-step evaluation problem in equation 17. It has been shown in Xie et al. (2020) that as $t \to \infty$, the iterations $\boldsymbol{P}^{(t+1)} = \mathbf{Breg\text{-}prox}_\phi(\boldsymbol{P}^{(t)})$ converge to an optimal solution of the original unregularized problem. Therefore, to solve 5 we can iteratively invoke Algorithm 1 to obtain $\boldsymbol{P}^{(t+1)} = \mathbf{Breg\text{-}prox}_\phi(\boldsymbol{P}^{(t)})$, while replacing γ and $\boldsymbol{M}$ in problem 6 by λ and $\boldsymbol{M} - \lambda\boldsymbol{P}^{(t)}$ in problem 17, respectively.

Algorithm 2 outlines the steps to solve the subset support selection using the Bregman proximal-point method, where the inner loop is solved by Algorithm 1. The nested loops of the exact proximal point algorithm result in high computational costs, but this can circumvented by choosing a lower number of iterations for the inner loop—stopping before its convergence. This is justified by the observation that the majority of the progress towards optimal solutions by gradient based methods is achieved during the first few iterations. Recently, an inertial variant of Bregman proximal point method for the optimal transport has been proposed (Yang & Toh, 2022), which may further accelerate the Bregman proximal point method, but to the best of our knowledge there are no guarantees for accelerated convergence.

Algorithm 2: Bregman Proximal Point Algorithm to approximately solve 5 via 17

```
Inputs        : Target distribution $\boldsymbol{\mu}$, mass assignment upper bounding vector $\boldsymbol{\zeta}$, cost matrix $\boldsymbol{M}$,
                Bregman scaling parameter $\lambda$, and initial dual variable, $\boldsymbol{\beta}^{(\mathrm{i})} \in \mathbb{R}^n_+$, inner-iteration
                limit max-inner-iter and outer-iteration limit max-outer-iter
Outputs       : $\hat{\boldsymbol{P}}^*$
Initialization: $\boldsymbol{\beta}^{(0)} \leftarrow \boldsymbol{\beta}^{(\mathrm{i})}, \boldsymbol{P}^{(0)} \leftarrow \frac{1}{mn}\mathbf{1}_{m\times n}$
for $t \leftarrow 0$ to max-outer-iter $- 1$ do
    // repeatedly invoke the function EntropicSS from the Algorithm 1.
    $\boldsymbol{P}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)} \leftarrow$ EntropicSS $\left(\boldsymbol{\mu}, \boldsymbol{\zeta}, \boldsymbol{\beta}^{(t)}, \lambda, \boldsymbol{M} - \mathbf{log}(\boldsymbol{P}^{(t)}), \textit{max-inner-iter}\right)$
end
$\hat{\boldsymbol{P}}^* = \boldsymbol{P}^{(t+1)}$
```

Due to early stopping, Algorithm 2 can yield infeasible solutions that do not satisfy the marginal constraints. In practice, the number of iterations depends on problem in the hand. For applications related to point cloud registration, color transfer and PU learning, where number of data points in a data-batch is small, the algorithm is allowed to run with a large number of iterations yielding a highly accurate and feasible solution. Whereas, for the applications related to neural network training where training efficiency is more important than the solution accuracy, the algorithm is allowed to run for a smaller number of iterations.

2.4 Point Cloud Registration with Subset Selection

Point cloud registration is a well studied problem that tries to find a correspondence of points in one sample (cloud) to another sample (Zhang et al., 2021; Zang et al., 2019). Practical settings include points sampled from the boundaries of 3D images such as captured by LIDAR or the points on the edges of objects in 2D images (Xu et al., 2023). More generally, points correspond to data points in two or more samples. In both of these cases it is useful to consider the case that the two samples exist in different coordinate frames such that there is an affine transformation needed to align the samples before finding the correspondence.

Partial optimal transport for point cloud registration is motivated by cases of occlusion in 2D or 3D imagery. In the case of data, it could be that one sample has dropped modes either by the nature of the data gathering or generating process. Our proposed subset selection algorithms are applicable to cases where the source is assumed to have a complete or overcomplete representation of the target, i.e., only a subset of the target is available and all target points should be maintained.

We propose to use support subset selection as a loss function for optimizing affine transformations in partial point cloud registration. This can be posed as a bi-level optimization problem

$$\min_{\Theta} \min_{\boldsymbol{P} \succcurlyeq 0} \quad \langle \boldsymbol{P}, \hat{\boldsymbol{M}}(\Theta) \rangle \quad \text{s.t.} \quad \boldsymbol{P}\mathbf{1}_n = \boldsymbol{\mu}, \; \boldsymbol{P}^\top \mathbf{1}_m \preccurlyeq c\boldsymbol{\nu}, \tag{18}$$

where $\Theta = [\boldsymbol{A}, \boldsymbol{b}]$ are the parameters of the affine transform, the entries of the cost matrix $\hat{\boldsymbol{M}}(\Theta)$ are $[\hat{\boldsymbol{M}}(\Theta)]_{ij} = \|\boldsymbol{x}_i - \hat{\boldsymbol{y}}_j^{\Theta}\|_2^2 \quad i \in [m], j \in [n]$ for fixed target $\{\boldsymbol{x}_i\}_{i=1}^m$ and transformed source $\{\hat{\boldsymbol{y}}_j^{\Theta} = \boldsymbol{A}\boldsymbol{y}_j + \boldsymbol{b}\}_{j=1}^n$. To find a solution to equation 18, we use an iterative alternating algorithm with two steps. In the first step, given the affine transformation we obtain an approximate solution $\hat{\boldsymbol{P}}^*$ to the subset selection problem equation 5 via Algorithm 2. In the second step, we used automatic differentiation of the cost $\langle \hat{\boldsymbol{P}}^*, \hat{\boldsymbol{M}}(\Theta) \rangle$ and perform gradient based update for the parameters $\Theta = [\boldsymbol{A}, \boldsymbol{b}]$. It is important to mention that we follow the approach adopted by Xie et al. (2020) for gradient evaluation. Therefore, during an iteration, once the subset set selection map $\hat{\boldsymbol{P}}^*$ is obtained, it is deemed constant for the iteration in consideration, therefore the gradient is: $\nabla_{\Theta} \langle \hat{\boldsymbol{P}}^*, \hat{\boldsymbol{M}}(\Theta) \rangle = \sum_{i,j} [\hat{\boldsymbol{P}}^*]_{ij} \nabla_{\Theta} [\hat{\boldsymbol{M}}(\Theta)]_{ij}$. More specifically, we used PyTorch based automatic differentiation for gradient evaluation (Paszke et al., 2017) and the Adam optimizer (Kingma & Ba, 2014) with learning rate of 0.5 for gradient based updates of parameters Θ. Instead of using the entire set of points, we can also sample mini-batches from each point cloud. To initialize the affine mapping parameters we simply set $\boldsymbol{A}$ and $\boldsymbol{b}$ to the identity matrix and zero vector, respectively. However, since the bi-level optimization problem is not convex, even though the subset selection problem at each iteration is convex, in practice the algorithm could be allowed to run with multiple initialization to obtain the best fit.

The bi-level optimization approach we employ is similar to the standard approaches in point cloud registration (Myronenko & Song, 2010; Arun et al., 1987). The standard approach discussed in Myronenko & Song (2010) also involves two step iterations, but the standard is to solve the subproblem exactly via ordinary least squares. If the coupling matrix during an iteration is given by $\boldsymbol{P}^*$, the next subproblem is to find the affine transformation parameters $\Theta = [\boldsymbol{A}, \boldsymbol{b}]$ that minimize the weighted squared errors $\sum_{i,j}[\boldsymbol{P}^*]_{ij}\|\boldsymbol{x}_i-(\boldsymbol{A}\boldsymbol{y}_j+\boldsymbol{b})\|_2^2$. The solution can be found analytically in terms of the source mass vector $\boldsymbol{\nu}^* = \boldsymbol{P}^{*\top}\mathbf{1}_m$, weighted means of the target point cloud $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]^\top$ and the source point cloud $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top$ as $\bar{\boldsymbol{x}} = \boldsymbol{X}^\top\boldsymbol{\mu}$ and $\bar{\boldsymbol{y}} = \boldsymbol{Y}^\top\boldsymbol{\nu}^*$, and centered point clouds $\tilde{\boldsymbol{X}} = \boldsymbol{X} - \mathbf{1}_m\bar{\boldsymbol{x}}^\top$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{Y} - \mathbf{1}_n\bar{\boldsymbol{y}}^\top$, as

$$\begin{aligned}\boldsymbol{A} &= \left(\tilde{\boldsymbol{X}}^\top\boldsymbol{P}^*\tilde{\boldsymbol{Y}}\right)\left(\tilde{\boldsymbol{Y}}^\top\boldsymbol{D}(\boldsymbol{\nu}^*)\tilde{\boldsymbol{Y}}\right)^\dagger,\\ \boldsymbol{b} &= \bar{\boldsymbol{x}} - \boldsymbol{A}\bar{\boldsymbol{y}},\end{aligned} \tag{19}$$

where $(\cdot)^\dagger$ indicates the Moore-Penrose pseudo-inverse. Also in contrast to Myronenko & Song (2010), instead of using complete source and target point clouds to obtain affine transformations, during every iteration we draw batches from both source and target point clouds to obtain the coupling matrix $\boldsymbol{P}^*$ and update the affine transformation parameters Θ. The advantage of this mini-batch based approach is an implicit regularization and faster updates for affine transformation parameters.

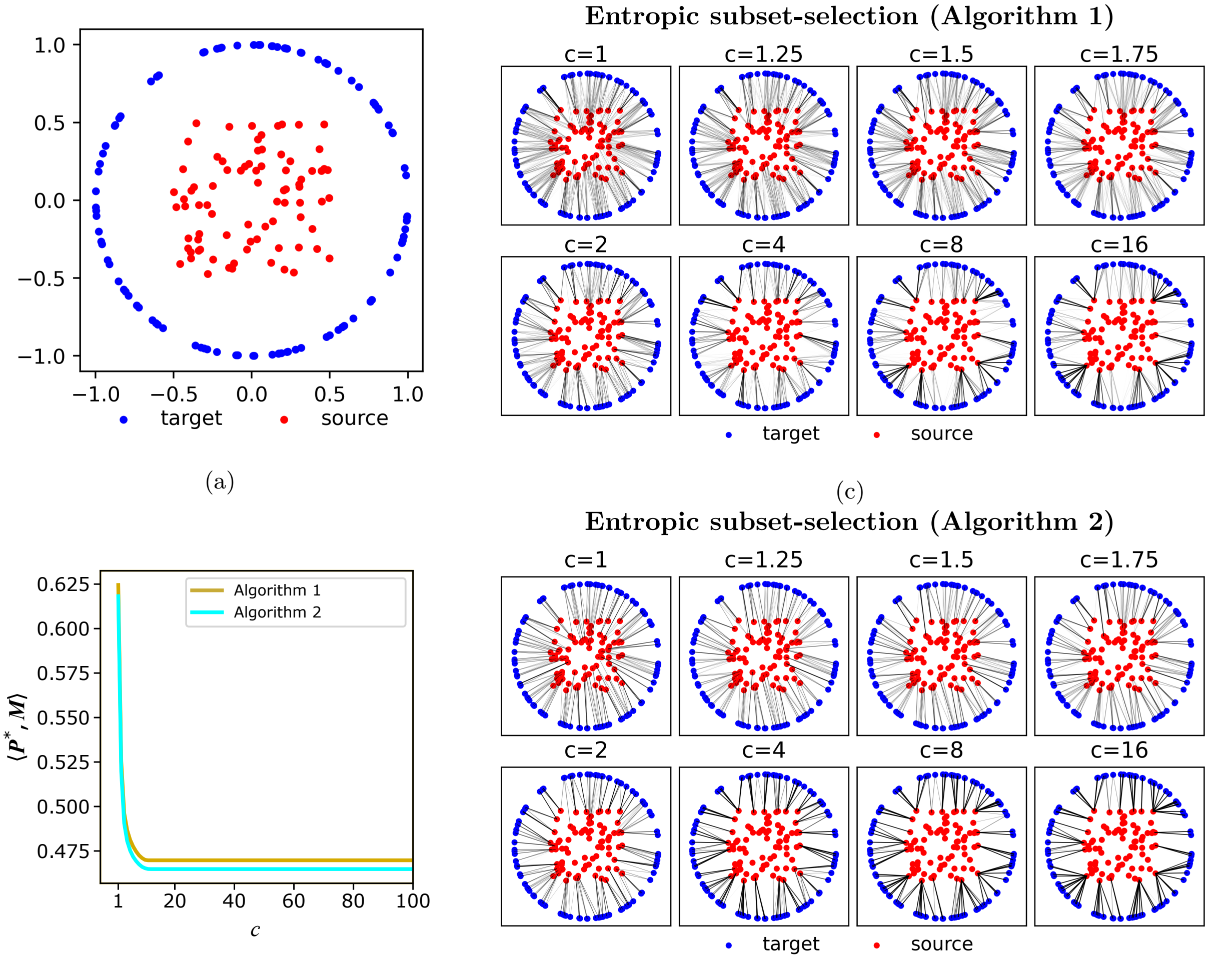
3 Experimental Results and Discussion

In this section we discuss the application of subset selection to different test case. Subsections 3.1 and 3.2 discuss the application of subset selection in toy data sets: point-clouds in 2D and 3D with and without affine transformations and color transfer, respectively. Subsection 3.3 discusses subset selection for positive-unlabeled learning tasks. Subsection 3.4 discusses the application of subset selection for semi-supervised training of neural networks. All the experiments done in this paper use $p = 2$ and the Euclidean distance to define the cost matrix. Unless stated otherwise, experiments use $\boldsymbol{\mu} = \frac{1}{m}\mathbf{1}_m$ and $\boldsymbol{\zeta} = \frac{c}{n}\mathbf{1}_n$, where $c \geq 1$ is the scaling factor.

3.1 Subset Selection on Point Clouds

Circle and Square: In order to demonstrate the proposed algorithms and highlight the difference between regular optimal transport and subset selection, we consider a target sample of points from a circle centered at the origin and a source sample of points from a 2D uniform distribution also centered at the origin. We allow the scaling parameter c to vary between 1 and 100, obtain the optimal transport plans $\boldsymbol{P}^*$ using both Algorithm 1 and Algorithm 2, and evaluate the cost values $\langle\boldsymbol{P}^*, \boldsymbol{M}\rangle$. Results for this toy case are shown in Figure 2. It can be observed that as c is increased the transport cost decreases until it saturates to the cost of the greedy solution $\sum_{i\in[m]}\frac{1}{m}\min_{j\in[n]}[\boldsymbol{M}]_{ij}$, which corresponds to $c = c^*$ where the transport map could be found by greedily choosing nearest source point for each target point as in equation 3. Figure 2 also illustrates the transport couplings for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$. A key observation is that transport maps obtained with Algorithm 2 are sparser as compared to the denser maps obtained using Algorithm 1. Additionally, they achieve smaller values of transport cost. Therefore in the subsequent, we focus on results from Algorithm 2 in the main body; results for Algorithm 1 are in Appendix A.

Fragmented Hypercube with Mode Dropping: We demonstrate the utility of the support subset selection algorithm for partial point cloud registration on a toy case with one dropped mode and an affine transformation between the source and the target. Specifically, we consider data sampled from a uniform distribution over a hypercube (specifically, a square in 2D or a cube in 3D), which is then fragmented, where



(a)

(b)

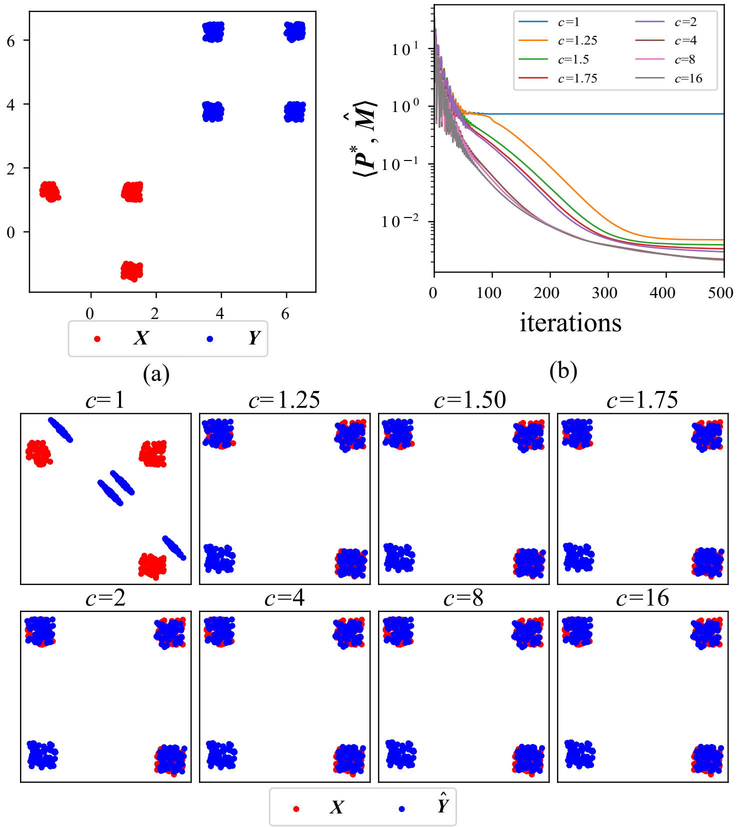
(c)

(d)

Figure 2: (a) The toy data generated by uniformly sampling $m = 100$ points from a circle centered at the origin with unit diameter as the target. The source contains $n = 80$ points generated by sampling uniformly from $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. (b) The optimal costs $\langle \boldsymbol{P}^*, \boldsymbol{M} \rangle$ obtained using Algorithms 1 and 2 versus c for $c \in [1, 100]$. Algorithm 1 is ran for 10,000 iterations with $\gamma = 0.1$. Algorithm 2 is ran for *max-outer-iter* $=$ 100, *max-inner-iter* $= 100$ and $\lambda = 0.1$. (c) and (d) Support subset selection results obtained for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$ using the Algorithms 1 and 2, respectively.

the target has one less fragment than the source. To generate the source we sample n points $\{\boldsymbol{v}_i\}_{i=1}^n$ from the uniform distribution over a unit hypercube centered at the origin $[-\frac{1}{2}, \frac{1}{2}]^d, \quad d \in \{2, 3\}$. These points are then fragmented into 2^d fragments according to their quadrant $\tilde{\boldsymbol{y}}_i = \boldsymbol{v}_i + (d-1)\,\mathrm{sign}(\boldsymbol{v}_i)$ and then offset to obtain the source points as $\boldsymbol{y}_i = \tilde{\boldsymbol{y}}_i + 5(d-1)$ for $i \in [n]$. The target data is generated similarly: a sample of $\hat{m} > m$ points $\{\boldsymbol{z}_i\}_{i=1}^{\hat{m}}$ is obtained from $[-\frac{1}{2}, \frac{1}{2}]^d$, then points with all negative coordinates are discarded, leaving m points, which are fragmented into $2^d - 1$ fragments to obtain the target set $\{\boldsymbol{x}_i\}_{i=1}^m$ via $\boldsymbol{x}_i = \boldsymbol{z}_i + (d-1)\,\mathrm{sign}(\boldsymbol{z}_i)$ for $i \in [m]$. Examples of the data for 2D and 3D are shown in Figure 3(a) and Figure 4(a), respectively.

Due to the translation by $5(d-1)$ of the source point coordinates, direct application of the transport map will not yield a meaningful registration. Instead we use the bi-level optimization algorithm described in Section 2.4. The target and the transformed source after applying the affine transformation obtained using Algorithm 2 for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$ are displayed in Figure 3(c) and Figure 4(c), respectively. Clearly, the $c = 1$ case corresponding to the complete optimal transport fails to identify a meaningful affine



(a)

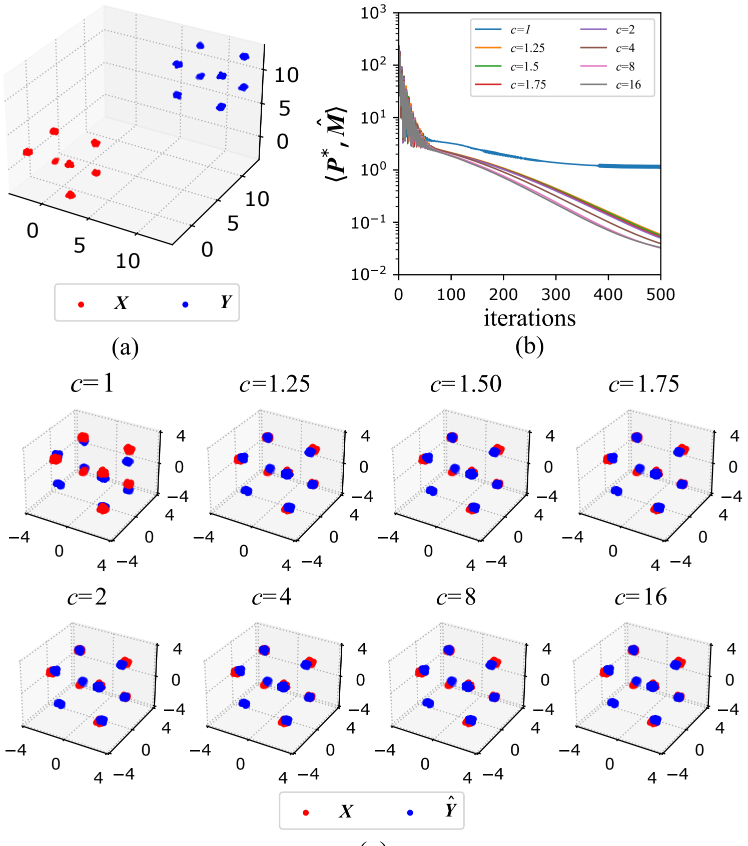
(b)

(c)

Figure 3: Results for affine transformation optimization with subset selection for partial optimal transport. Target points $\boldsymbol{X}$ are sampled from a 2D fragmented hypercube centered at the origin with negative coordinates removed, whereas source points $\boldsymbol{Y}$ are sampled from a translated fragmented hypercube. (a) Source and target sample points. (b) Loss function curves for scaling parameter $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$. (c) Target and transformed source points after application of optimized affine transformation. Subset selection problems are solved using the Algorithm 2 with $\lambda = 0.1$, *max-outer-iter* $= 200$ and *max-inner-iter* $= 200$.

transformation, instead skewing and rotating the source fragments to minimize the Wasserstein distance to the target. The figures also display the cost $\langle \boldsymbol{P}^*, \hat{\boldsymbol{M}}\rangle$ across iterations. It can be observed that, like the previous toy examples as the value of scaling factor c is increased, initially the value of the optimal loss $\langle \boldsymbol{P}^*, \hat{\boldsymbol{M}}\rangle$ decreases but after certain values of c, it saturates and stops decreasing and stays constant afterwards.

Partial Point Cloud for 3D Shapes: We further applied this form of subset selection based point cloud registration to point clouds for 3D objects when the target points are only taken from a portion of the entire 3D point cloud. Results for the Stanford bunny and armadillo (Turk & Levoy, 1994; Krishnamurthy & Levoy, 1996) are shown in Figure 5. It can be observed that for the case $c = 1$, which corresponds to complete optimal transport, the entire set of source points are coupled to the target point cloud which results in a distorted affine transform. For $c \in \{2, 5, 10, 20\}$, subset selection allows an appropriate subset of the source points to be well-fit by an affine transform to the target point cloud.



(a) (b)

(c)

Figure 4: Results for affine transformation optimization with subset selection for partial optimal transport. Target points $\boldsymbol{X}$ are sampled from a 3D fragmented hypercube centered at the origin with negative coordinates removed, whereas source points $\boldsymbol{Y}$ are sampled from a translated fragmented hypercube. (a) Source and target sample points. (b) Loss function curves for scaling parameter $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$. (c) Target and transformed source points after application of the optimized affine transformation. Subset selection problems are solved using the Algorithm 2 with $\lambda = 0.1$, *max-outer-iter* $= 200$ and *max-inner-iter* $= 200$.

3.2 Color Transfer

Color transfer is the problem of finding a correspondence in the colors of pixels (represented as points in a 3D color space) between two images and then using this map to assign the colors of the source image to the target image (Reinhard et al., 2001). Color transfer is essentially an optimal transport problem in the color space, but with the added context that the pixels have their image coordinates, which are not used by the algorithm. For practical application to high resolution images, the pixel colors are first quantized using k-means clustering, as using partial optimal transport on the full set of pixel colors is computationally demanding. While in standard optimal transport the relative mass of each color cluster has to be preserved, here we exploit our formulation of partial optimal transport as support subset selection to allow a subset of colors to be used at a higher proportion than in the original source and allow a subset of colors to be completely discarded. For example, if a color cluster represents 1% of the original source's pixels, then it could represent up to c% of the target's pixels.

We apply k-means clustering to the set of vectors in RGB color space representing the source's M pixels and the target's N pixels separately to obtain $m \ll M$ color centroids $\{\boldsymbol{x}_i\}_{i=1}^m \subset \mathbb{R}_+^3$ for the target image and $n \ll N$ color centroids $\{\boldsymbol{y}_j\}_{j=1}^n \subset \mathbb{R}_+^3$ for the source image, with $\boldsymbol{\mu} \in \boldsymbol{\Delta}_m$ and $\boldsymbol{\mu} \in \boldsymbol{\Delta}_n$ being the vectors of proportion of colors in the target and source image color clusters, respectively. After that, we defined the

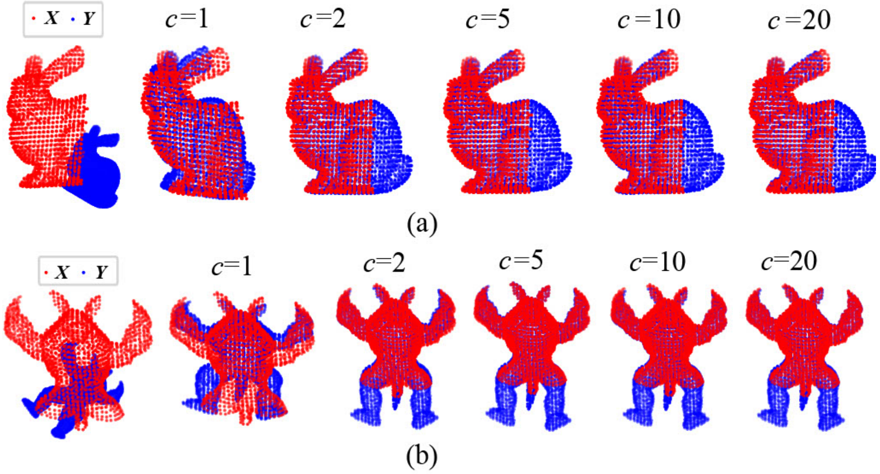


Figure 5: Affine transformation optimization for partial alignment of point clouds where the source point-cloud $\boldsymbol{Y}$ can be perfectly aligned (after rotation and scaling) with a subset of the target point cloud $\boldsymbol{X}$. We use the optimization algorithm described in Section 2.4, where Algorithm 2 is employed to obtain the coupling $\boldsymbol{P}^*$ given the affine transformation parameters $\boldsymbol{A}$ and $\boldsymbol{b}$, which are updated using equation 19. (a) Stanford bunny point cloud. (b) Stanford armadillo point cloud.

cost matrix between the color centroids as $\boldsymbol{M}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2, \forall i \in [m], j \in [n]$ and obtained the support subset selection map $\boldsymbol{P}^* \in \mathbb{R}_+^{m\times n}$ using Algorithm 2, such that $\boldsymbol{P}^*\mathbf{1}_n = \boldsymbol{\mu}$ and $\boldsymbol{P}^{*\top}\mathbf{1}_m \preccurlyeq c\boldsymbol{\nu}$. The support subset selection is then used to obtain the barycenter projections by solving (Blondel et al., 2018)

$$\hat{\boldsymbol{x}}_i = \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^3} \sum_{j=1}^{n} P^*_{ij}\|\boldsymbol{x} - \boldsymbol{y}_j\|_2^2, \quad \forall i \in [m]. \tag{20}$$

The analytic solution of the barycenter projections can be compactly written as

$$\hat{\boldsymbol{X}} = (\boldsymbol{P}^* \oslash (\boldsymbol{\mu}\mathbf{1}_n^\top))\boldsymbol{Y} \in \mathbb{R}^{m\times 3}, \tag{21}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^\top \in \mathbb{R}^{m\times 3}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m]^\top \in \mathbb{R}^{n\times 3}$ are matrices of the color centroids. Each pixel in the target image is assigned the corresponding barycenter projection $\hat{\boldsymbol{x}}_{\pi(i)}$, where $\pi(i) \in [m]$ is the cluster assignment for the i-th pixel of target image, $i \in [M]$.

We apply this color transfer scheme to images freely available though a Creative Commons licence, the "Louisiana Nature Scene Barataria Preserve" by Neil O as target and "Autumn in Toronto" by Bahman A-Mahmoodi as source. The color transfer results with $m = n = 128$ and Algorithm 2 with $\lambda = 0.1$ are shown in Figure 6. It can be observed that the results for larger values of c are smoother and sharper as compared to the optimal transport case $c = 1$. This due to fact that larger values of c allow certain colors to be reused more than their prevalence in the source image and allow some colors to be discarded, which enables smoother transitions in colors for areas of the target images with smooth color gradients. Similar observations can be seen in Figure 7 which uses the same settings and MATLAB test images: "peppers" as target and "corn" as source.

3.3 Subset Selection for Positive-Unlabeled Learning

In this section, we discuss the application of subset selection to the one-class semi-supervised classification scheme known as positive-unlabeled (PU) learning (Bekker & Davis, 2020). In PU learning, the training sample consists of purely positively labeled instances, and the unlabeled test sample consists of both positive and negative instances. Previous work often assumes that a prior on the probability of positive instances in unlabeled data is known (Kato et al., 2019; Hsieh et al., 2019; Chapel et al., 2020). Partial optimal transport is then used to find a subset with cardinality proportional to the prior of the test sample (the source) that

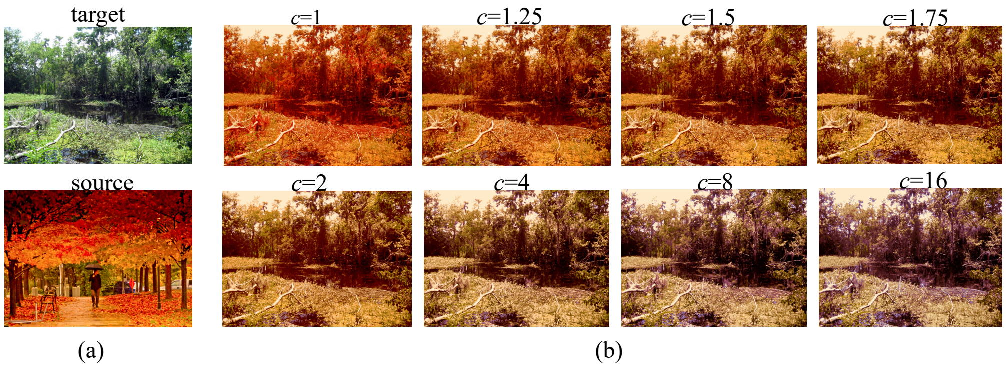


Figure 6: Color transfer results for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$ for "Louisiana Nature Scene Barataria Preserve" by Neil O as target and "Autumn in Toronto" as source by Bahman A-Mahmoodi as target. The value of c for each image is indicated at the top of image.

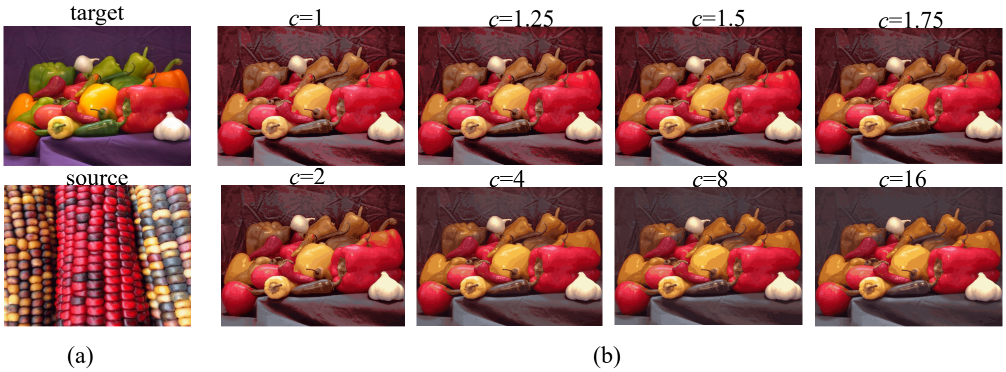


Figure 7: Color transfer results for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$ for MATLAB image "peppers" as target and "corn" as source. The value of c for each image is indicated at the top of image.

corresponds to all or a subset of the training sample (the target). We argue that all of the target mass should be preserved in cases of a relatively small and curated positive training sample. This motivates the application of our proposed subset selection approach to find the subset of the source that covers the positive target.

We applied subset selection to PU learning using the experimental settings adapted from Chapel et al. (2020), who previously explored using partial optimal transport and the partial Gromov-Wasserstein distance on various UCI, MNIST, colored-MNIST, and Caltech-office data sets. For the UCI, MNIST, and colored-MNIST data sets, we randomly draw $m = 400$ positive points and $n = 800$ unlabeled points. For Caltech-office data sets, we randomly sampled $m = 100$ positive points from the first domain and $n = 100$ unlabeled data points. Following the experiments from Chapel et al. (2020) and Kato et al. (2019), for multi-class data sets, we chose the data points from the first class as positive and a random mixture of all classes as unlabeled, and the prior probability of positive class in the unlabeled set π is set to be exactly the proportion of positives in unlabeled sample $\pi = \frac{n_+}{n}$, where n_+ is the number of true positives. This informs the partial Wasserstein and partial Gromov-Wasserstein optimal transport problems on the amount of mass to be transported as $\rho = \pi$, whereas for subset selection we set the scaling parameter to be $c = \frac{n}{\lceil \pi n \rceil}$. Classification accuracy is evaluated by assigning positive predictions to the n_+ largest source mass assignments and negative predictions to the remaining source points. We also compute the ROC curve by using the source mass assignment $\boldsymbol{\nu}^*$ to rank the unlabeled source points. We ran the experiment 10 times and report the average of classification accuracy and the area under the ROC curve (ROC-AUC) in Table 1. It can be observed that subset-selection performs better than both the partial Wasserstein and partial Gromov-Wasserstein optimal transport in terms of accuracy in 6 out of the 8 UCI and MNIST data sets. For the Caltech-office data sets with domain transfer, the partial Gromov-Wasserstein optimal transport does better than both partial Wasserstein optimal transport and the subset selection. In terms of ROC-AUC, subset selection does the best in 13 out of the 16 data sets, which is not surprising since the relative ranking is more meaningful than when the mass assignments are restricted to be binary valued $\{0, p\}$ as in Chapel et al. (2020).

Dataset	π	**Accuracy**			**ROC-AUC**		
		PW	PGW	SS	PW	PGW	SS
mushrooms	0.518	95.15	94.85	**96.63**	0.9657	0.3336	**0.9883**
shuttle	0.786	95.13	93.63	**96.20**	0.9321	0.6215	**0.9718**
pageblocks	0.898	91.90	90.38	**92.40**	0.8036	0.7197	**0.8513**
usps	0.167	98.28	95.55	**98.48**	0.9815	0.5096	**0.9927**
connect-4	0.658	**60.95**	58.03	60.73	0.5692	0.5126	**0.5871**
spambase	0.394	78.80	68.40	**79.28**	0.7952	0.5834	**0.8369**
mnist	0.1	99.08	98.23	**99.18**	0.9874	0.7638	**0.9971**
mnist-colored	0.1	91.58	**96.78**	91.88	0.8189	0.6619	**0.9521**
surf C $\rightarrow$ surf C	0.1	90.00	87.20	**90.40**	**0.8576**	0.4622	0.7333
surf C $\rightarrow$ surf A	0.1	81.60	**86.80**	81.60	0.4546	0.4764	**0.4889**
surf C $\rightarrow$ surf W	0.1	82.20	**86.40**	82.20	0.4707	0.4807	**0.5056**
surf C $\rightarrow$ surf D	0.1	80.00	**87.00**	80.00	0.3756	0.4328	**0.4444**
decaf C $\rightarrow$ decaf C	0.1	**94.00**	86.20	93.00	0.9498	0.5713	**0.9566**
decaf C $\rightarrow$ decaf A	0.1	80.20	**88.20**	80.00	0.3986	**0.5031**	0.4242
decaf C $\rightarrow$ decaf W	0.1	80.20	**88.60**	80.20	0.4299	0.5827	**0.6282**
decaf C $\rightarrow$ decaf D	0.1	80.80	**92.20**	81.00	0.4546	**0.5042**	0.4617

Table 1: PU learning on data sets in (Chapel et al., 2020). For subset selection (SS) accuracy is evaluated by assigning label 1 to the largest mass assignments to and label 0 to the remaining mass assignments. In Chapel et al. (2020), the mass assignment are constrained to be binary valued in the set $\{0, p\}$, the data points with mass assignments 0 are labeled 0 and the data points with mass p are labeled 1.

3.3.1 PU Learning on MNIST/EMNIST

To further illustrate how Algorithm 2 operates on PU learning, we apply it to the case where the positive training sample (target) consists of MNIST digit images and the unlabeled test sample contains 50% points (MNIST digits) and 50% negative points (alphabetic letters from EMNIST). When $c = 1$, which is equivalent to standard optimal transport, initially all the images in the unlabeled source sample are assigned uniform masses. As c is increased, we hypothesize that the true positive MNIST digits will been assigned larger mass and remain in the selected support, whereas the EMNIST letters will receive relatively lower or zero mass. Our hypothesis is confirmed by the results displayed in Figure 8(a), which displays the ROC curve across different choices of c, and in Figure 8(c), which displays the area under the ROC curve (AUC). As c is increased, source points with largest mass assignments are mostly MNIST digits. Likewise, Figure 8(e) shows the images with highest mass for different values of c which are mainly MNIST numbers or EMNIST letters with close resemblance to a number. Figure 8(b) visualizes the distribution of source point masses by graphing the sorted masses for different values of c. From these curves the cardinality of the subset is easily seen for different values of c. Notably, for values of $c \leq 4$ there are exists a subset of the selected source points with uniform mass, but for larger values of c, the mass is non-uniform across all instances. These changes correspond to the change in slope of the entropy of the distribution for different values of c is displayed in Figure 8(d).

3.3.2 PU Learning for CIFAR-10 Neural-Network Representations

We now consider the proposed subset selection algorithm for PU learning on the CIFAR-10 data set, where a single class from the training set is treated as the positive target and a mixture of all classes from the test set is the unlabeled source. Fundamentally, the performance of optimal transport methods on PU learning depends on the distance metric defining the cost matrix. Thus, the method performs poorly if a Euclidean distance

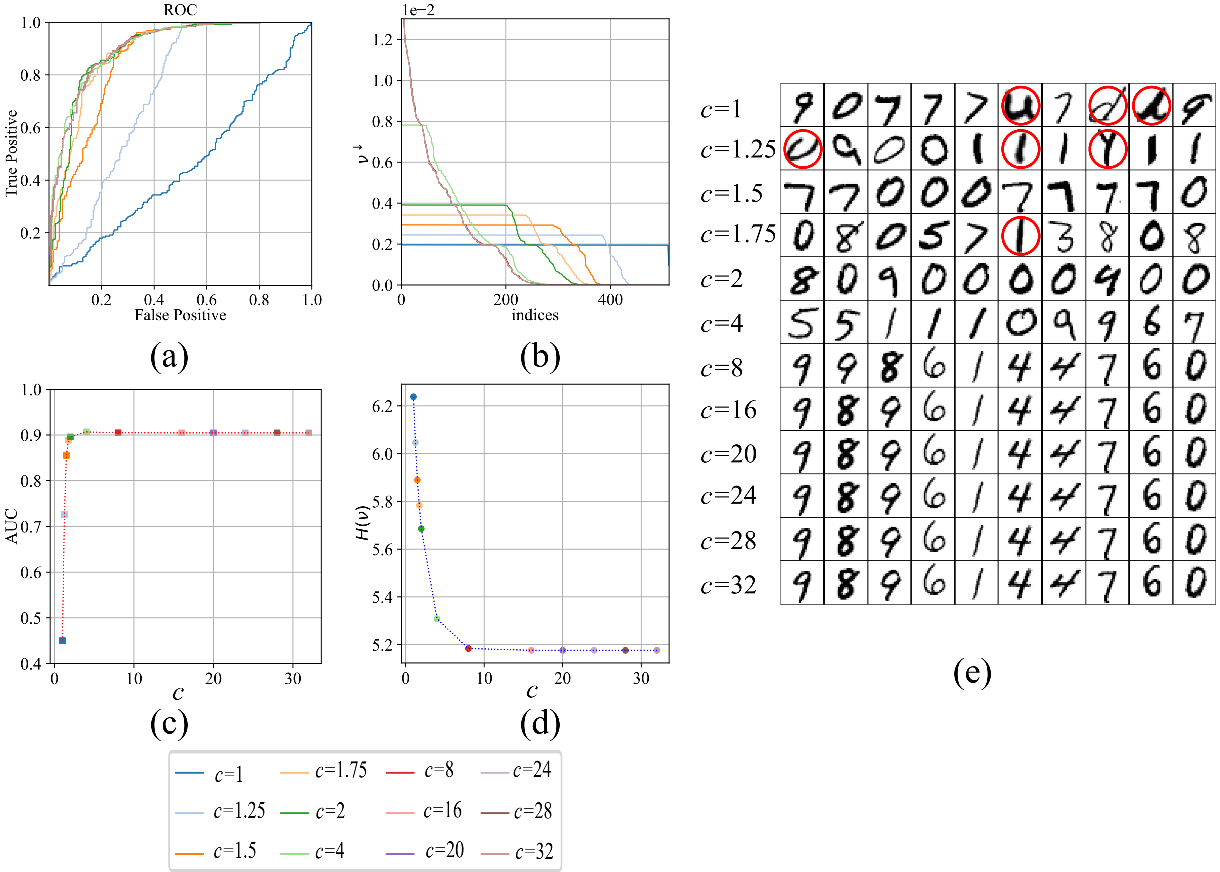


Figure 8: Subset selection results obtained using Algorithm 2 with parameters $\lambda = 1$, *max-outer-iter* $= 250$ and *max-inner-iter* $= 20$. Source sample $n = 512$ consists of 50% digit images from MNIST and 50% letter images from EMNIST. The target sample contains $m = 512$ digit images drawn from MNIST. (a) ROC curves for different values for $c \in \{1, 1.25, 1.50, 1.75, 2, 4, 8, 16, 20, 24, 28, 32\}$. (b) Mass assignments to source images in descending order $\boldsymbol{\nu}^{\downarrow}$. (c) AUC of ROC versus c. (d) Entropy $H(\boldsymbol{\nu})$ of the mass assignments $\boldsymbol{\nu}$ versus c.

metric is applied to complex data such as natural images. Instead, a learning representation extracted from a pretrained neural network can be used. Here each image is represented as the vector of activations of the penultimate layer of the pre-trained ResNet-20 classifier (trained on CIFAR-10), and the Euclidean distance between the activation vectors defines the cost matrix for the transport problem.

The results are given in Figure 9. The results are similar to the previous MNIST/EMNIST data set. Mass is uniformly distributed across a subset of images for values of $1 < c < 8$. When the subset is greater than the proportion of positive instances in the unlabeled source, then the relative ranking of mass is not reliable: the top instances for $c \in \{1.5, 2, 4\}$ are images from the target class, but $c \in \{1.25, 1.75\}$ have images resembling it from other classes. As c is increased above $c = 8$ the mass assignment is non-uniform, but constant for further increment in c. Values of c greater than 2 have an AUC >90%.

3.4 Subset Selection for Semi-supervised Learning

We consider the semi-supervised training of a classifier where the training set is divided into a reliably labeled (curated) target set and an unlabeled or noisily labeled source set. We apply our proposed subset selection algorithm to perform partial optimal transport of the unlabeled source to the labeled target. The transport plan is computed without knowledge of any labels but defines how the source points will be labeled,

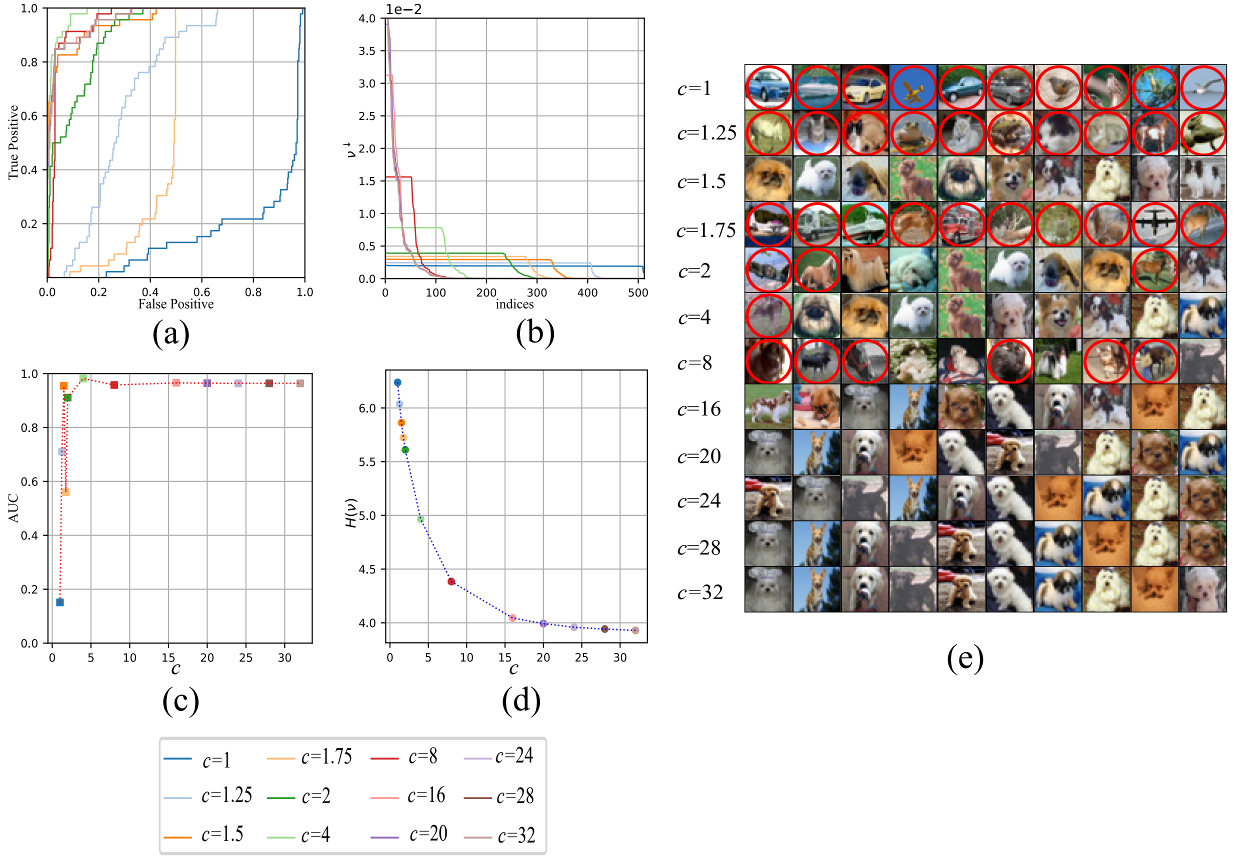


Figure 9: Subset selection results obtained using algorithm 2 with parameters $\lambda = 1$, *max-outer-iter* $= 250$ and *max-inner-iter* $= 20$. Target and source consist of ResNet-20 embeddings of $m = 512$ CIFAR-10 dog images and $n = 512$ randomly sampled CIFAR-10 images, respectively. (a) ROC curves for $c \in \{1, 1.25, 1.50, 1.75, 2, 4, 8, 16, 20, 24, 28, 32\}$. (b) Mass assignments to source images in descending order $\boldsymbol{\nu}^{\downarrow}$. (c) AUC of the ROC versus c. (d) Entropy $H(\boldsymbol{\nu})$ of mass assignments $\boldsymbol{\nu}$ with versus c. (e) Source images with 10 largest mass assignments.

and subset selection removes points that cannot easily be aligned to labeled training points. Additionally, the new mass assignment source points may be relatively higher for unlabeled points relatively close to labeled training points and lower for unlabeled points from existing points. Used in this way, the optimal transport with subset selection automatically tunes how far to propagate labels in a manner that takes into consideration the geometry and distribution of the curated target data set rather than only the local distances.

However, using the distances defined directly in the input space may not be suitable, and a pre-trained representation may not exist for various tasks. Instead, we propose to use the internal learning representation from the neural network classifier while it is being optimized with the semi-supervised loss function.

Let $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{M}$ denote the labeled portion of the training set with the input $\boldsymbol{x}_i \in \mathcal{X}$ and label encoded as a one-hot vector $\boldsymbol{L}_i \in \{0,1\}^k \subset \boldsymbol{\Delta}_k$, $\|\boldsymbol{L}_i\|_1 = 1$ for $i \in [M]$, and $\mathcal{T} = \{\boldsymbol{y}_j\}_{j=1}^{N}$ denote the unlabeled portion, $\boldsymbol{y}_j \in \mathcal{X}$ for $j \in [N]$. We consider a neural-network classifier with soft-max activation $\boldsymbol{f}(\cdot\,;\boldsymbol{\theta}) : \mathcal{X} \to \boldsymbol{\Delta}_k$ with parameters $\boldsymbol{\theta}$ trained on data with k classes. The neural network's internal representation is a function $\boldsymbol{g}(\cdot\,;\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}^d$. The Euclidean distance between the internal representation of data points provides the distance function, $\mathrm{d}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|\boldsymbol{g}(\boldsymbol{x}_i;\boldsymbol{\theta}) - \boldsymbol{g}(\boldsymbol{y}_j;\boldsymbol{\theta})\|_2$, which is parameterized by the network's parameters.

We train the neural network using mini-batches and a semi-supervised cross-entropy loss. Equal-sized batches are drawn uniformly from the pooled training data set of size $M+N$. Let $\boldsymbol{\tau}$ and $\boldsymbol{\sigma}$ denote the length-m and length-n vectors of indices of the labeled and unlabeled points in a given batch, respectively, where $m+n$ is the constant batch size. The m-by-n ground cost matrix $\boldsymbol{M}(\boldsymbol{\theta})$ is defined using the squared distances among the batch's latent representations, $M_{ij}(\boldsymbol{\theta}) = \mathrm{d}^2_{\boldsymbol{\theta}}(\boldsymbol{x}_{\tau_i}, \boldsymbol{y}_{\sigma_j}) = \|\boldsymbol{g}(\boldsymbol{x}_{\tau_i};\boldsymbol{\theta}) - \boldsymbol{g}(\boldsymbol{y}_{\sigma_j};\boldsymbol{\theta})\|_2^2$.

Given the cost matrix and hyper-parameters (including $c \leq n$), the subset selection transport plan $\boldsymbol{P}^* \in [0,1]^{m\times n}$ is obtained using Algorithm 2. Given the matrix of one-hot encoded labels $\boldsymbol{L} = [\boldsymbol{L}_{\tau_1}, \ldots, \boldsymbol{L}_{\tau_m}]^\top \in \{0,1\}^{m\times k}$, the matrix of pseudo-labels assigned by the algorithm of the unlabeled mini-batch points is computed $\tilde{\boldsymbol{L}} = n\boldsymbol{P}^{*\top}\boldsymbol{L} \in [0,c]^{n\times k}$, where $[\boldsymbol{P}^{*\top}\boldsymbol{L}]_{jl} = \frac{1}{n}\tilde{L}_{jl} \in [0,1]$ is the estimate of the joint probability that mini-batch unlabeled instance $j \in [n]$ belongs to class $l \in [k]$.[4] Given the pseudo-labels, the semi-supervised cross-entropy loss function for a batch is

$$\mathrm{loss}(\boldsymbol{\theta}) = -\Big[\sum_{i=1}^{m}\sum_{l=1}^{k}\frac{1}{m}\boldsymbol{L}_{il}\log(f_l(\boldsymbol{x}_{\tau_i};\boldsymbol{\theta})) + \sum_{j=1}^{n}\sum_{l=1}^{k}\frac{1}{n}\tilde{\boldsymbol{L}}_{jl}\log(f_l(\boldsymbol{y}_{\sigma_j};\boldsymbol{\theta}))\Big]. \tag{22}$$

Our approach is similar to other recent work (Damodaran et al., 2020) that also employs optimal transport using a learning representation. While we address the semi-supervised case, Damodaran et al. (2020) address supervised learning in the presence of label noise and perform self optimal transport within batches to correct for label noise.

As baseline comparisons, we compare our semi-supervised approach to supervised training with either only the labeled portion or with noisy labels on the unlabeled portion. Due to the curated labeled set, the latter is not the typical label noise scenario; however, the division of a training set into a curated portion and a portion with label noise is relevant to practical scenarios. While our semi-supervised approach does not use noisy labels, future extensions could consider how to leverage the noisy labels too.

In order to evaluate our approach we used MNIST, Fashion-MNIST (FMNIST), and CIFAR-10. We split training data sets into 80/20 proportions for training and validation. We further split the training part into a reliably labeled and unreliably labeled parts. Labels for the unreliably labeled part are generated by uniformly corrupting the true labels to other classes depending on the noise level. For each of our experiments, the subset selection transport underlying the loss done is found via Algorithm 2 with $\lambda = 0.01$, *max-outer-iter* = *max-inner-iter* = 20 with a batch-size of 512. We used PyTorch framework for our experiments. A ResNet-18 model architecture is used on the CIFAR-10 data set. We trained the ResNet-18 for 180 epochs using Adam optimizer with an initial learning rate of 0.001, which is scheduled to be halved after every 60 epochs. The model architectures containing two convolutional layers for MNIST and FMNIST are given in Appendix B. The neural network classification models for MNIST are trained using stochastic gradient descent with a learning rate of 0.001, whereas models for FMNIST are trained using Adam with a learning rate of 0.001 and weight decay 1e-4. Model training for MNIST and Fashion-MNIST are done on a desktop system with Intel Core-i7 9700 with 32 GB memory and NVIDIA GeForce RTX 2070 GPU. ResNet-18 based models for CIFAR-10 are trained using Lambda-labs cloud resources with 30 vCPUs, 200 GB memory, and NVIDIA A10 GPUs.

In the first step of experiments, we split the training set for each data set into 50/50 proportions for unreliably and unreliably labeled parts. Unreliably labeled data is generated by uniformly corrupting the labels with a 80% chance (noise level 0.8). Validation accuracies for each data set are displayed in the Table 2. Notably, the performance for $c = 1$ is higher than training with noisy labels, which shows that the semi-supervised training performs better than training with data with a high noise level. (Because the algorithm is not run to convergence, the mass assignments for unlabeled points may not be be exactly uniform in the $c = 1$ case.) The performance of subset selection is consistently higher for values of $c > 1$ compared to the $c = 1$, and the validation accuracies do not exhibit much change between $c = 2$ and $c = 20$. Therefore, we further evaluated our approach by varying both noise levels and clean and noisy proportions only for $c = 2$ and $c = 20$.

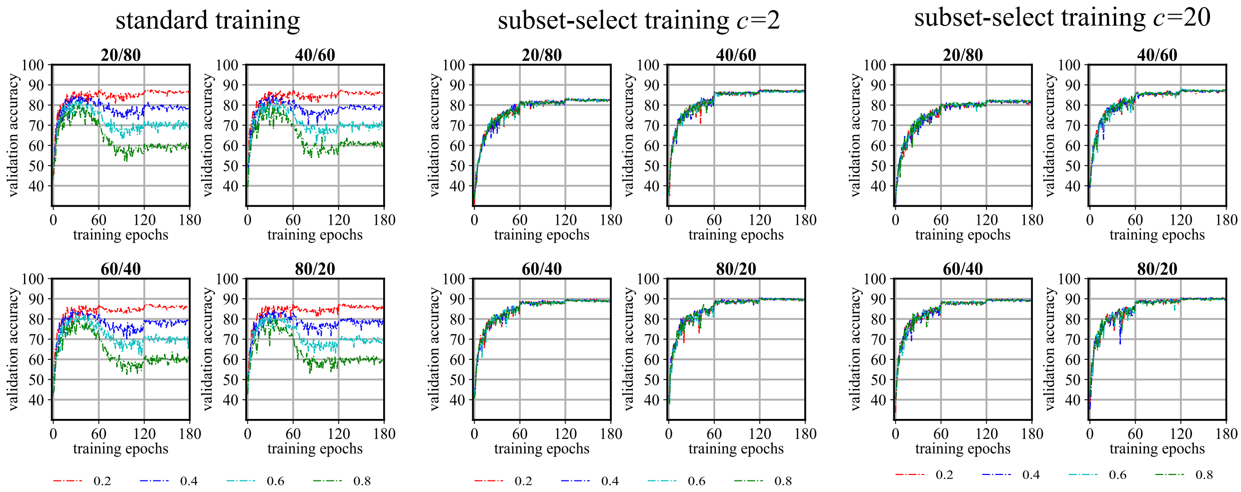
---

[4] It can be seen that the total sum of this joint is 1, $\underbrace{\mathbf{1}_n^\top \boldsymbol{P}^{*\top}}_{\boldsymbol{\mu}^\top}\underbrace{\boldsymbol{L}\mathbf{1}_k}_{\mathbf{1}_m} = 1$.

Dataset	Architecture	stand.	Subset Selection								
			$c=1$	$c=2$	$c=3$	$c=4$	$c=5$	$c=6$	$c=7$	$c=8$	$c=20$
MNIST	2-layer conv-net	96.57	97.15	97.39	97.33	97.40	97.43	97.32	97.38	**97.44**	97.35
FMNIST	2-layer conv-net	88.68	90.14	90.30	90.27	**90.51**	90.23	90.38	90.21	90.37	90.16
CIFAR-10	ResNet-18	79.10	87.18	89.45	89.27	89.21	89.53	89.54	**89.72**	89.47	89.57

Table 2: Validation accuracies for different values for neural network classification models trained with 50% reliably labeled points and 50% points with noisy labels (noise level 0.8). Standard training (stand.) treats them equally, but subset selection treats them as unlabeled and assign pseudo-labels. Subset selection is done using the Algorithm 2 with $\lambda = 0.1$ and *max-outer-iter* = *max-inner-iter* = 20.

Progress of validation accuracies on CIFAR-10 are displayed in Figure 10 for clean/noisy proportions in $\{20/80, 40/60, 60/40, 80/20\}$ with noise levels $\{0.2, 0.4, 0.6, 0.8\}$. It can be observed that standard neural network training process with label noise can divided into three phases, first in which the validation accuracy increases until a peak. In the second phase, validation accuracy decreases, where the magnitude of the decrement depends on the noise level: it decreases less for low noise levels and more for higher noise levels. In the third phase, validation accuracy increases again and then oscillates around a constant value. This kind of phenomenon is more pronounced for larger noise levels (Zheng et al., 2020). In contrast, for the proposed subset selection based semi-supervised learning the validation accuracy does not go down after hitting its peak during the training process. This indicates that the transport map tend to assign correct pseudo-labels to the data points nearest the labeled data points and does not introduce label noise.



(a) (b) (c)

Figure 10: Progress of validation accuracies while training ResNet-18 for CIFAR-10 classification. (a) Uniform noise levels are varied between 0.2, 0.4, 0.6, 0.8 for standard training. (b) and (c) Training with the subset selection based semi-supervised loss at different values of c, does not use the unreliable labels, and outperforms standard training with noisy labels when either the proportion of reliably labeled data is 40% or the noise level is 0.4 or greater. Subset selection is done using the Algorithm 2 with $c = 20$, $\lambda = 0.1$, *max-outer-iter* = *max-inner-iter* = 20.

The test set accuracies are displayed in Table 3 for supervised training on only the labeled data versus the semi-supervised training. The semi-supervised training with subset selection outperforms training performs better on 2 of 3 data sets under a 20/80 split of labeled and unlabeled, but does not outperform supervsed learning for the 40/60 split. Thus, the semi-supervised loss function equation 22 is most beneficial when there is a higher ratio of unlabeled to labeled points.

Dataset	Architecture	Labeled/Unlabeled %	Labeled only	Semi-supervised with subset selection
MNIST	2-layer conv-net	20/80	**98.19**	96.09
		40/60	**98.30**	97.14
F-MNIST	2-layer conv-net	20/80	88.35	**88.42**
		40/60	**89.93**	89.48
CIFAR-10	ResNet-18	20/80	81.62	**82.37**
		40/60	**87.77**	86.74

Table 3: Test accuracies on MNIST, Fashion-MNIST, and CIFAR-10. Subset selection is done using the Algorithm 2 with $c = 20$, $\lambda = 0.1$, *max-outer-iter* = *max-inner-iter* = 20.

4 Discussion and Further Work

In this paper, we have focused on selecting a subset of one distribution's support as a special case of partial optimal transport Figalli (2010); Chapel et al. (2020). This is useful to find meaningful alignment when the support of the target distribution is assumed to be a subset of the source distribution. Results on the partial point cloud alignment, color transfer, PU learning, and semi-supervised learning all demonstrate the utility of this approach.

In particular, the results from the PU learning show that the proposed subset selection is useful when there is known target distribution (an existing training or validation set) and an additional source distribution, which has additional diversity, but also outliers, compared to the target. One application of PU learning is to filter a source of new data for relevant examples for further modeling. In this case, a user would want to balance the diversity (entropy) of the filtered source with its purity. Future work, could explore this approach for source distributions created from synthetic generation mechanisms. While not explored in a machine learning context, it is possible that the partial optimal transport with affine (or nonlinear) transformation can be applied to account for global covariate shifts in these cases.

In our experiments related to semi-supervised learning, we employed optimal transport between a labeled target and unlabeled source, to assign pseudo-labels to source points that cover the labeled data distribution, while ignoring ambiguous cases, during training. In future extension, we can consider how to use class information in the optimal transport planning, perhaps by using class conditional optimal transport, as currently the transport plan is not informed of the known target labels nor the classifier's boundaries. Another line of exploration is how to use the support subset selection to correct noisily labeled source.

While not explored here, support subset selection on both distributions can be useful (perhaps in generative modeling due to the established theory of partial optimal transport for continuous distributions). However, the proposed algorithms are not directly applicable, but with slight modifications similar optimization algorithms could be applied.

Another key contribution of this work is the proposed support subset selection algorithm using the inexact Bregman proximal point algorithm (Algorithm 2), which as shown in Appendix A yields a solution with a sparse source marginal similar to solutions to the original linear program 5—unlike the entropically regularized solution from Algorithm 1. We also demonstrate that the mass assignments of the linear program solution are piece-wise linear as a function of c. While not fully investigated here, this behavior could be exploited to find the sequence of breakpoints where points leave the support and where points leave the active set of constraints (indicated by being on the upper diagonal).

Recently, Gromov-Wasserstein optimal transport has seen applications in graph-matching and generative modeling (Brogat-Motte et al., 2022; Li et al., 2023; Titouan et al., 2019; Nekrashevich et al., 2023; Bunne et al., 2019; Mémoli, 2009; Bunne et al., 2019). Due to inherent ability to match structural correspondences across spaces, partial Gromov-Wasserstein optimal transport can be used to solved robust graph-alignment problems. Recently, efficient locally convergent solutions for a relaxed Gromov-Wasserstein distance have been proposed (Peyré et al., 2016; Li et al., 2023). Future work can explore the subset selection case of the partial Gromov-Wasserstein optimal transport, where one domain is expected to have a complete or overcomplete source distribution compared to the target. This may be useful in robust domain adaptation, semi-supervised domain adaptation, and metric alignment.

Broader Impact Statement

This paper discusses the application of optimal transport with support subset selection, which can be applied in wide variety of machine learning contexts. To the best of our knowledge, this work does not violate any of TMLR ethics guidelines.

References

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A Rewriting System for Convex Optimization Problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-Linear Time Approximation Algorithms for Optimal Transport via Sinkhorn Iteration. *Advances in Neural Information Processing Systems*, 30, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017.

K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611974997`.

Amir Beck and Marc Teboulle. A fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Amir Beck and Marc Teboulle. A Fast Dual Proximal Gradient Algorithm for Convex Minimization and Applications. *Operations Research Letters*, 42(1):1–6, 2014.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109:719–760, 2020.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and Sparse Optimal Transport. In *International Conference on Artificial Intelligence and Statistics*, pp. 880–889. PMLR, 2018.

Nicolas Bonneel and David Coeurjolly. SPOT: sliced partial optimal transport. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence d'Alché Buc. Learning to predict graphs with fused gromov-wasserstein barycenters. In *International Conference on Machine Learning*, pp. 2321–2335. PMLR, 2022.

Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning Generative Models across Incomparable Spaces. In *International Conference on Machine Learning*, pp. 851–861. PMLR, 2019.

Luis A Caffarelli and Robert J McCann. Free Boundaries in Optimal Transport and Monge-Ampere Obstacle Problems. *Annals of Mathematics*, pp. 673–730, 2010.

Laetitia Chapel, Mokhtar Z Alaya, and Gilles Gasso. Partial Optimal Tranport with Applications on Positive-Unlabeled Learning. *Advances in Neural Information Processing Systems*, 33:2903–2913, 2020.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling Algorithms for Unbalanced Optimal Transport Problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Advances in Neural Information Processing Systems*, 26, 2013.

Marco Cuturi and Gabriel Peyré. A Smoothed Dual Approach for Variational Wasserstein Problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016. doi: 10.1137/15M1032600. URL `https://doi.org/10.1137/15M1032600`.

Marco Cuturi and Gabriel Peyré. Semidual Regularized Optimal Transport. *SIAM Review*, 60(4):941–965, 2018.

Bharath Bhushan Damodaran, Rémi Flamary, Vivien Seguy, and Nicolas Courty. An entropic optimal Transport Loss for Learning Deep Neural Networks under Label Noise in Remote Sensing Images. *Computer Vision and Image Understanding*, 191:102863, 2020.

Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative Modeling Using the Sliced Wasserstein Distance. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3483–3491, 2018.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Alessio Figalli. The Optimal Partial Transport Problem. *Archive for Rational Mechanics and Analysis*, 195 (2):533–560, 2010.

Rémi Flamary, Nicholas Courty, Davis Tuia, and Alain Rakotomamonjy. Optimal Transport for domain Adaptation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 1, 2016.

Divyansh Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Wasserstein Distances for Stereo Disparity Estimation. *Advances in Neural Information Processing Systems*, 33:22517–22529, 2020.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning Generative Models with Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 30, 2017.

Sergey Guminov, Pavel Dvurechensky, Nazarii Tupitsa, and Alexander Gasnikov. On a Combination of Alternating Minimization and Nesterov's Momentum. In *International Conference on Machine Learning*, pp. 3886–3898. PMLR, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from Positive, Unlabeled and Biased Negative Data. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2820–2829. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/hsieh19c.html`.

Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *International Conference on Learning Representations*, 2019.

Keisuke Kawano, Satoshi Koide, and Keisuke Otaki. Partial Wasserstein Covering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7115–7123, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Matthieu Kirchmeyer, Alain Rakotomamonjy, Emmanuel de Bezenac, and patrick gallinari. Mapping Conditional Distributions for Domain Adaptation under Generalized Target Shift. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=sPfB2PI87BZ`.

Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal Mass Transport: Signal Processing and Machine-Learning Applications. *IEEE Signal Processing Magazine*, 34 (4):43–59, 2017.

Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffmann. Sliced Wasserstein Distance for Learning Gaussian Mixture Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.

Venkat Krishnamurthy and Marc Levoy. Fitting Smooth Surfaces to Dense Polygon Meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 313–324, 1996.

Jiajin Li, Jianheng Tang, Lemin Kong, Huikang Liu, Jia Li, Anthony Man-Cho So, and Jose Blanchet. A Convergent Single-Loop Algorithm for Relaxation of Gromov-Wasserstein in Graph Data. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=0jxPyVWmiiF`.

Xing-Si Li and Shu-Cherng Fang. On The Entropic Regularization Method for Solving Min-Max Problems with Applications. *Mathematical methods of operations research*, 46(1):119–130, 1997.

Tianyi Lin, Nhat Ho, and Michael Jordan. On Efficient Optimal Transport: An Analysis of Greedy and Accelerated Mirror Descent Algorithms. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3982–3991. PMLR, 09–15 Jun 2019.

Tianyi Lin, Nhat Ho, and Michael I Jordan. On the Efficiency of Entropic Regularized Algorithms for Optimal Transport. *Journal of Machine Learning Research*, 23(137):1–42, 2022.

Andriy Myronenko and Xubo Song. Point Set Registration: Coherent Point Drift. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

Facundo Mémoli. Spectral Gromov-Wasserstein distances for shape matching. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 256–263, 2009. doi: 10.1109/ICCVW.2009.5457690.

Maksim Nekrashevich, Alexander Korotin, and Evgeny Burnaev. Neural Gromov-Wasserstein Optimal Transport. *arXiv preprint arXiv:2303.05978*, 2023.

Neal Parikh and Stephen Boyd. Proximal Algorithms. *Foundations and Trends® in Optimization*, 1(3): 127–239, 2014.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in Pytorch. In *NIPS 2017 Autodiff Workshop*, 2017.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pp. 2664–2672. PMLR, 2016.

Gabriel Peyré, Marco Cuturi, et al. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein Barycenter and its Application to Texture Mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color Transfer Between Images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs Using Optimal Transport. In *International Conference on Learning Representations*, 2018.

Bernhard Schmitzer. Stabilized Sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

Richard Sinkhorn. A Relationship between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein Propagation for Semi-Supervised Learning. In *International Conference on Machine Learning*, pp. 306–314. PMLR, 2014.

Marc Teboulle. Entropic Proximal Mappings with Applications to Nonlinear Programming. *Mathematics of Operations Research*, 17(3):670–690, 1992.

Vayer Titouan, Rémi Flamary, Nicolas Courty, Romain Tavenard, and Laetitia Chapel. Sliced Gromov-Wasserstein. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 14726–14736. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/9615-sliced-gromov-wasserstein.pdf`.

Ilya Tolstikhin, Olivier Bousquet, Sylvian Gelly, and Bernhard Schölkopf. Wasserstein Auto-Encoders. In *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

Greg Turk and Marc Levoy. Zippered Polygon Meshes from Range Images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pp. 311–318, 1994.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *Uncertainty in Artificial Itelligence*, pp. 433–453. PMLR, 2020.

Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. Distilled Wasserstein Learning for Word Embedding and Topic Modeling. *Advances in Neural Information Processing Systems*, 31, 2018.

Ningli Xu, Rongjun Qin, and Shuang Song. Point Cloud Registration for Lidar and Photogrammetric Data: A Critical Synthesis and Performance Analysis on Classic and Deep Learning Algorithms. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, pp. 100032, 2023.

Lei Yang and Kim-Chuan Toh. Bregman Proximal Point Algorithm Revisited: A New Inexact Version and Its Inertial Variant. *SIAM Journal on Optimization*, 32(3):1523–1554, 2022.

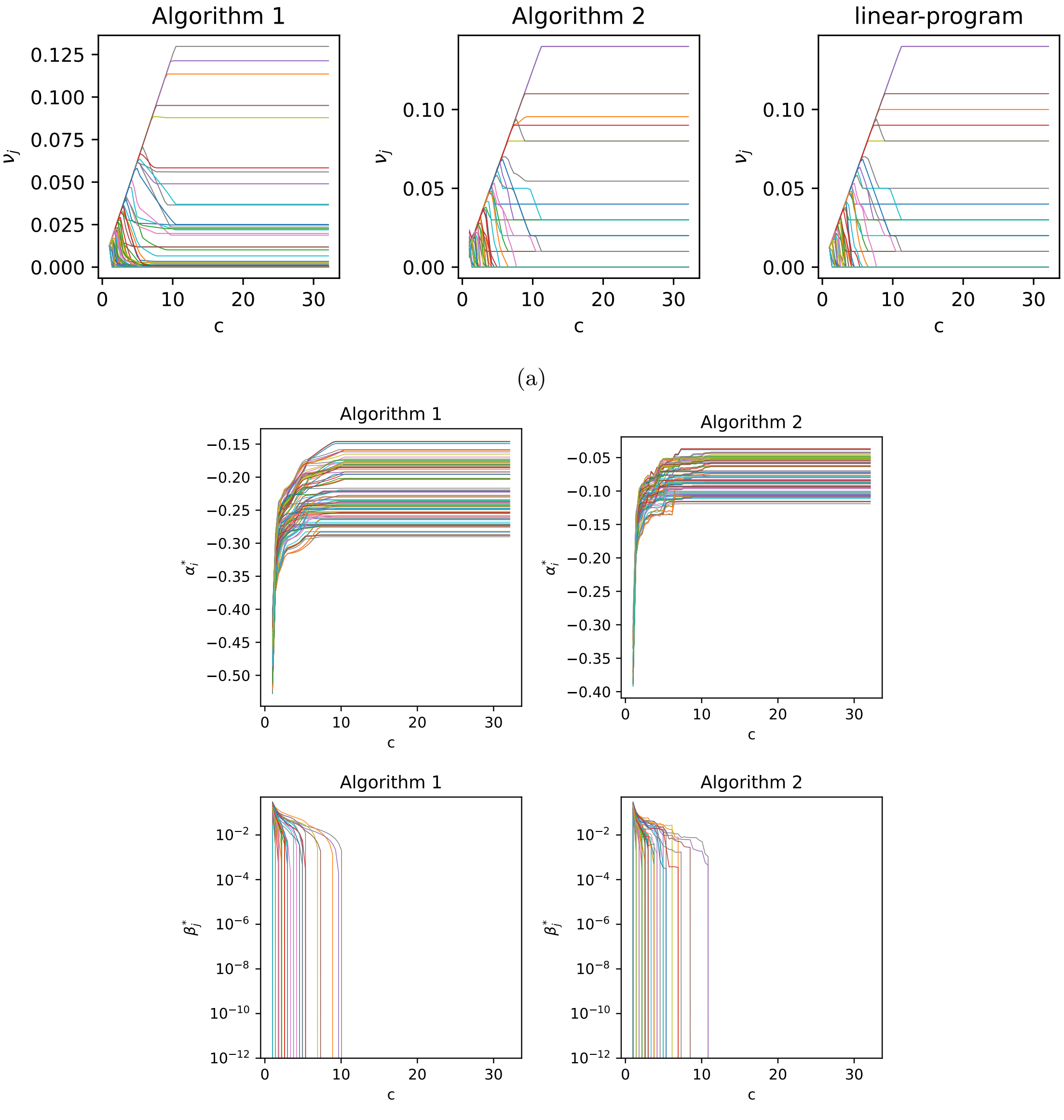
Yufu Zang, Roderik Lindenbergh, Bisheng Yang, and Haiyan Guan. Density-adaptive and Geometry-aware Registration of TLS Point Clouds based on Coherent Point Drift. *IEEE Geoscience and Remote Sensing Letters*, 17(9):1628–1632, 2019.

Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2021.

Pengxiang Zheng, Songzhu andWu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-Bounded Correction of Noisy Labels. In *International Conference on Machine Learning*, 2020.

A Appendix: Entropic Regularization Results

First we show the variation of mass assignments ν_j^*, for $j \in [n]$ as a function of c for the toy problem in the Figure 2 with transport plans obtained using Algorithm 1, Algorithm 2, and the solution to the unregularized linear program obtained from CVXPY (Diamond & Boyd, 2016; Agrawal et al., 2018). It can be be observed sparsity patterns of $\boldsymbol{\nu}^*$ obtained using Algorithm 2 matches closer to the linear program solution, whereas mass assignments obtained using Algorithm 1 are more dense with less mass assignments equal to 0.

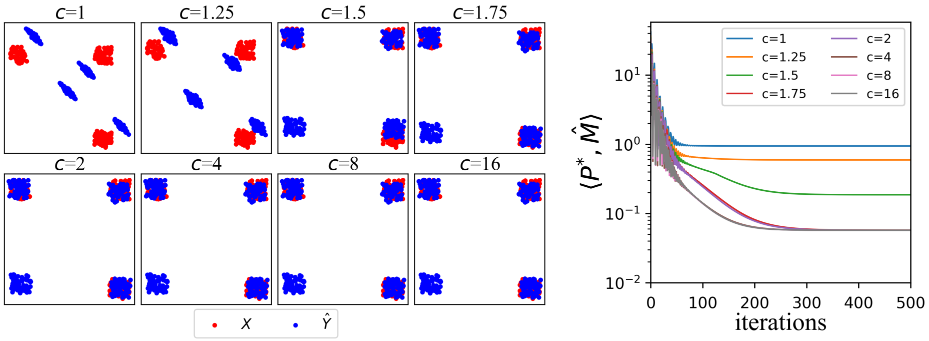


(a)

(b)

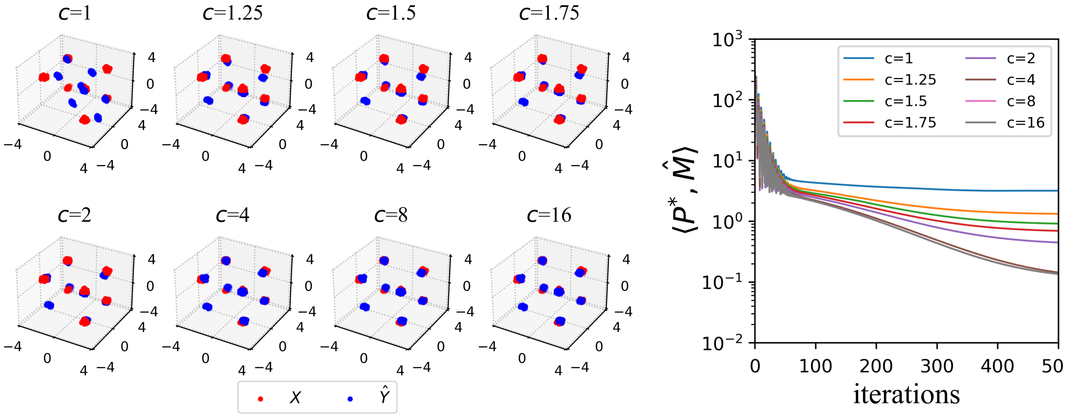
Figure 11: (a) The variation of mass assignment vector $\boldsymbol{\nu}^*$ with c for toy problem in Figure 2, using Algorithm 1, Algorithm 2, and linear programming solution obtained using CVXPY Agrawal et al. (2018); Diamond & Boyd (2016). (b) Optimal dual variables $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ obtained using using Algorithms 1 and 2 as c is varied. It can be observed that above certain threshold c^* ($c^* = 11.19$ as calculated using the method discussed in section 2.1) all dual potentials corresponding to inequality $\boldsymbol{\beta}$ go to zero, this is due to fact that corresponding constraints become inactive and therefore superfluous for all values of c larger than c^*.

Fragmented Hypercubes: In this appendix, results for point cloud registration and color transfer for Algorithm 1 are displayed, which can be compared to the results for Algorithm 2 in the main body. Figure 12 shows the results for affine transformation optimization in 2D, which can be compared with the results in Figure 3(a). Visually it is clear that the alignment is much worse for values of $c \in \{1.25, 1.5\}$, but quantitatively it is worse for all values of $c > 1$ as Algorithm 2 achieves cost below 10^{-2} at 500 iterations. Figure 13 shows the results in 3D using Algorithm 1, which can be compared with the results in Figure 4(a). In this case, results are quantitatively worse for all values of $c > 1$ as Algorithm 2 achieves cost below 10^{-1} at 500 iterations.



(a) (b)

Figure 12: Results for affine transformation optimization with subset selection for partial optimal transport. Target points $\boldsymbol{X}$ are sampled from a 2D fragmented hypercube centered at the origin with negative coordinates removed, whereas source points $\boldsymbol{Y}$ are sampled from a translated fragmented hypercube. (a) Target and transformed source points after application of optimized affine transformation. Subset selection problems are solved using the Algorithm 1 with $\gamma = 0.01$ with *max-iter* $= 4000$.(b) Loss function curves for scaling parameter $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$.



(a) (b)

Figure 13: Results for affine transformation optimization with subset selection for partial optimal transport. Target points $\boldsymbol{X}$ are sampled from a 3D fragmented hypercube centered at the origin with negative coordinates removed, whereas source points $\boldsymbol{Y}$ are sampled from a translated fragmented hypercube. (a) Target and transformed source points after application of optimized affine transformation. Subset selection problems are solved using the Algorithm 1 with $\gamma = 0.01$ with *max-iter* $= 4000$.(b) Loss function curves for scaling parameter $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$.

Partial point cloud registration: The results for partial point cloud registration with entropically regularized subset selection (Algorithm 1) for the Stanford bunny and armadillo point clouds. It is clear that the entropically regularized form alone fails to find a meaningful correspondence, transforming the source such that is completely covered by the partial point cloud.

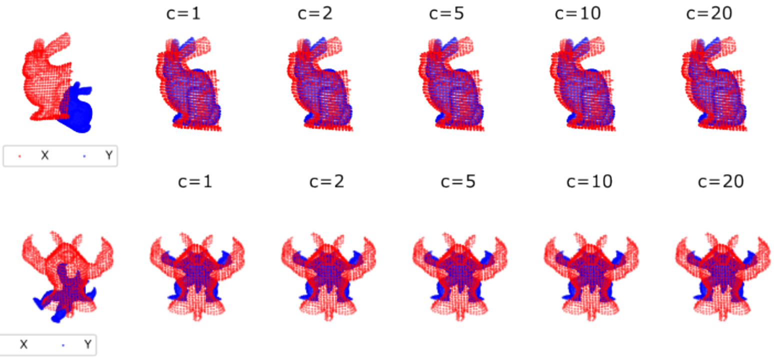


Figure 14: Bunny and Armadillo partial point cloud registration using entropically regularized subset selection Algorithm 1 *gamma* $= 0.05$, fails to find a accurate alignment of the source with the partially occluded target.

Color Transfer: The results for color transfer with the entropically regularized subset selection Algorithm 1 are shown in Figure 15, which can be compared to results from Algorithm 2 shown in Figure 6 and Figure 7. Namely, for the first image “Louisiana Nature Scene Barataria Preserve” the entropically regularized results appear more monochromatic with less distinct colors. In the second set of images, there is no visual difference between the outputs of Algorithm 1 and Algorithm 2.

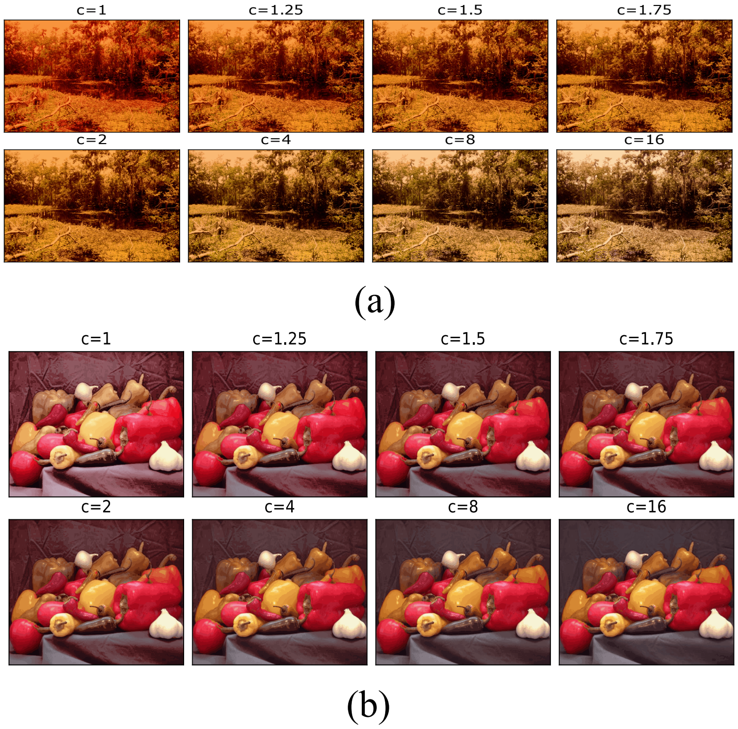


Figure 15: Color transfer results for $c \in \{1, 1.25, 1.5, 1.75, 2, 4, 8, 16\}$ using Algorithm 1.

B Appendix: Neural Network Model Architectures

For semi-supervised learning, we used neural networks to both perform the classification and provide a learning representation space in which to perform the optimal transport to assign pseudo-labels to both unlabeled points. For the CIFAR-10 data set, we used the ResNet-18 (He et al., 2016) architecture. For MNIST and Fashion-MNIST, we used custom, but simple, model architectures. Both architectures contain two convolutional layers, followed by three fully-connected layers. The model used to train Fashion-MNIST (FMNIST) classifier contains additional batch-normalization layer between the convolutional layers. Optimization algorithms along with related hyper-parameters are in Section 3.4. The code below details the exact architectures along with types and shapes of all transformations.

```python
import torch
import torch.nn as nn
import torch.nn.functional as F

class MNIST_classifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(in_channels=1, out_channels=5,kernel_size=(5,5))
        self.conv2 = nn.Conv2d(in_channels=5, out_channels=1,kernel_size=(5,5))
        self.fc1 = nn.Linear(400, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 10)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = torch.flatten(x, 1)
        x = F.relu(self.fc1(x))
        x_rep = F.relu(self.fc2(x))
        x = self.fc3(x_rep)
        return x, x_rep

class FMNIST_classifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(in_channels=1, out_channels=32, kernel_size=(5, 5))
        self.batchN1 = nn.BatchNorm2d(num_features=32)
        self.conv2 = nn.Conv2d(in_channels=32, out_channels=64, kernel_size=(5, 5))
        self.fc1 = nn.Linear(in_features=64*4*4, out_features=128)
        self.fc2 = nn.Linear(in_features=128, out_features=64)
        self.fc3 = nn.Linear(in_features=64, out_features=10)

    def forward(self, x):
        x = self.conv1(x)
        x = F.relu(F.max_pool2d(input=x, kernel_size=2, stride=2))
        x = self.batchN1(x)
        x = self.conv2(x)
        x = F.relu(F.max_pool2d(input=x, kernel_size=2, stride=2))
        x = torch.flatten(x, 1)
        x = F.relu(self.fc1(x))
        x_rep = self.fc2(x)
        x = self.fc3(x_rep)
        return x, x_rep
```

Listing 1: Models used for training classifiers for MNIST and Fashion-MNIST

C Appendix: Lipschitz Smoothness of Dual

The dual form 9 considered in this paper does not explicitly enforce the primal problem's marginal simplex constraints on the transport plan. Consequently, the dual form is not necessarily Lipschitz smooth (Lin et al., 2019; Cuturi & Peyré, 2018; Lin et al., 2022). But in the proposed algorithm, we first update the dual variable $\boldsymbol{\alpha}$ using the Sinkhorn-like update, which implicitly enforces the simplex constraint, making the semi-dual problem $\frac{1}{\gamma}$-Lipschitz smooth with respect to ℓ_1, ℓ_2 and ℓ_∞ norms. This justifies the use of $\eta_s^{(k)} = \frac{1}{\gamma}$ in the accelerated proximal-gradient based approach to solve 9.

Recall that the Lagrangian of 9 given in equation 7 is

$$\mathcal{L}(\boldsymbol{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \left\langle \boldsymbol{P},\ \boldsymbol{M} + \gamma(\mathbf{log}(\boldsymbol{P}) - \mathbf{1}_m\mathbf{1}_n^\top) + \boldsymbol{\alpha}\mathbf{1}_n^\top + \mathbf{1}_m\boldsymbol{\beta}^\top \right\rangle - \langle \boldsymbol{\alpha}, \boldsymbol{\mu}\rangle - \langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle. \tag{23}$$

By Slater's conditions, the problem 9 is strongly dual therefore

$$\min_{P \succcurlyeq 0} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \min_{P \succcurlyeq 0} \mathcal{L}(\boldsymbol{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{24}$$

In order to find the minimum of the Lagrangian with respect to $\boldsymbol{P}$, one takes its element-wise derivative with respect to $\boldsymbol{P}$ and obtains $\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{D}\big(\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{\alpha})\big)\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{M})\boldsymbol{D}\big(\mathbf{exp}(-\frac{1}{\gamma}\boldsymbol{\beta})\big)$, which can then be substituted back into the Lagrangian 23 to obtain the problem

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \left\{ g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\gamma \mathbf{1}_m^\top \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{1}_n - \langle \boldsymbol{\alpha}, \boldsymbol{\mu}\rangle - \langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle \right\}, \text{ s.t. } \boldsymbol{\beta} \succcurlyeq 0, \tag{25}$$

which can be converted to the convex minimization problem 9 by defining $f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -g(\boldsymbol{\alpha}, \boldsymbol{\beta})$, as in

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \left\{ f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \gamma \mathbf{1}_m^\top \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{1}_n + \langle \boldsymbol{\alpha}, \boldsymbol{\mu}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle \right\}, \text{ s.t. } \boldsymbol{\beta} \succcurlyeq 0, \tag{26}$$

The partial gradients of $f(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are

$$\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\mu} - \mathbf{exp}\big(-\frac{\boldsymbol{\alpha}}{\gamma}\big) \odot \boldsymbol{K}\mathbf{exp}(-\frac{\boldsymbol{\beta}}{\gamma}), \tag{27a}$$

$$\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\zeta} - \mathbf{exp}\big(-\frac{\boldsymbol{\beta}}{\gamma}\big) \odot \boldsymbol{K}^\top\mathbf{exp}(-\frac{\boldsymbol{\alpha}}{\gamma}). \tag{27b}$$

For twice continuously differentiable functions, the Lipschitz smoothness parameter is determined by the Hessian. The Hessian for $f(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is

$$\boldsymbol{H}_f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \begin{bmatrix} \nabla_{\boldsymbol{\alpha}}^\top \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) & \nabla_{\boldsymbol{\beta}}^\top \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \nabla_{\boldsymbol{\alpha}}^\top \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) & \nabla_{\boldsymbol{\beta}}^\top \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) \end{bmatrix}, \tag{28}$$

where

$$\nabla_{\boldsymbol{\alpha}}^\top \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\gamma}\boldsymbol{D}\left(\mathbf{exp}\big(-\frac{\boldsymbol{\alpha}}{\gamma}\big) \odot \boldsymbol{K}\mathbf{exp}(-\frac{\boldsymbol{\beta}}{\gamma})\right) = \frac{1}{\gamma}\boldsymbol{D}\big(\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{1}_n\big), \tag{29a}$$

$$\nabla_{\boldsymbol{\beta}}^\top \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\gamma}\boldsymbol{D}\left(\mathbf{exp}\big(-\frac{\boldsymbol{\beta}}{\gamma}\big) \odot \boldsymbol{K}^\top\mathbf{exp}(-\frac{\boldsymbol{\alpha}}{\gamma})\right) = \frac{1}{\gamma}\boldsymbol{D}\big(\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta})^\top\mathbf{1}_m\big), \tag{29b}$$

$$\nabla_{\boldsymbol{\beta}}^\top \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\gamma}\left(\boldsymbol{K} \odot \mathbf{exp}\big(-\frac{\boldsymbol{\alpha}}{\gamma}\big)\mathbf{exp}(-\frac{\boldsymbol{\beta}^\top}{\gamma})\right) = \frac{1}{\gamma}\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{29c}$$

$$\nabla_{\boldsymbol{\alpha}}^\top \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\gamma}\left(\boldsymbol{K}^\top \odot \mathbf{exp}\big(-\frac{\boldsymbol{\beta}}{\gamma}\big)\mathbf{exp}(-\frac{\boldsymbol{\alpha}^\top}{\gamma})\right) = \frac{1}{\gamma}\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}, \boldsymbol{\beta})^\top. \tag{29d}$$

The Sinkhorn update for $\boldsymbol{\alpha}$ in equation 12 ensures that after each update of $\boldsymbol{\alpha}$, the transport plan lies on the probability simplex and matches the target marginal $\boldsymbol{\mu} = \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})\mathbf{1}_n$. Defining $\tilde{\boldsymbol{\nu}} = \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})^\top\mathbf{1}_m$, the Hessian equation 28 at $(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})$ is compactly written as

$$\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)}) = \frac{1}{\gamma}\begin{bmatrix} \boldsymbol{D}(\boldsymbol{\mu}) & \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)}) \\ \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})^\top & \boldsymbol{D}(\tilde{\boldsymbol{\nu}}) \end{bmatrix}.$$

We use the induced-norms of the Hessian $\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})$ to characterize the smoothness at $(\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\beta}^{(k)})$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the induced norm $\|\cdot\|_{p,q}$ is defined as

$$\|\boldsymbol{A}\|_{p,q} := \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_p \le 1} \|\boldsymbol{A}\boldsymbol{x}\|_q. \tag{30}$$

The twice continuously differentiable function $f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is L-Lipschitz with respect to ℓ_p norm, if $\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{p,q} \leq L$, where ℓ_q is the dual norm of the ℓ_p norm. Since the Hessian $\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is a non-negative matrix, one can observe that all its matrix entries are less than $\frac{1}{\gamma}\max\{\mu_{\max},\tilde{\nu}_{\max}\}$, where $\mu_{\max}$ and $\tilde{\nu}_{\max}$ are maximum entries of $\boldsymbol{\mu}$ and $\tilde{\boldsymbol{\nu}}$ respectively. Therefore for $p=1$, if one can find the column index k corresponding to a matrix entry with value $\frac{1}{\gamma}\max\{\mu_{\max},\tilde{\nu}_{\max}\}$, then $\boldsymbol{x}=\boldsymbol{e}_k$ is the vertex of the ℓ_1 norm-ball where $\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\boldsymbol{x}\|_\infty = \frac{1}{\gamma}\max\{\mu_{\max},\tilde{\nu}_{\max}\}$. Thus,

$$\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{1,\infty} = \frac{1}{\gamma}\max\{\mu_{\max},\tilde{\nu}_{\max}\} \leq \frac{1}{\gamma}, \tag{31}$$

which proves the function $f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is $\frac{1}{\gamma}$-Lipschitz with respect to the ℓ_1 norm. Since all the matrix entries of the Hessian $\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ are less than $\frac{1}{\gamma}$, its spectral radius is less than $\frac{1}{\gamma}$ (Horn & Johnson, 2012)(Theorem 8.1.18) and

$$\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{2,2} = \lambda_{\max}(\boldsymbol{H}_f) \leq \frac{1}{\gamma}. \tag{32}$$

Therefore, the function $f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is $\frac{1}{\gamma}$-Lipschitz with respect to the ℓ_2 norm. For $p=\infty$, one can maximize the norm $\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\boldsymbol{x}\|_1$, at the vertex of the ℓ_∞ ball where all entries are unit magnitude, in particular $\boldsymbol{x}=\mathbf{1}_{m+n}$, which results into

$$\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\mathbf{1}_{m+n} = \frac{1}{\gamma}\begin{bmatrix}\boldsymbol{\mu}+\tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\mathbf{1}_n \\ \tilde{\boldsymbol{P}}(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})^\top\mathbf{1}_m+\tilde{\boldsymbol{\nu}}\end{bmatrix} = \frac{2}{\gamma}\begin{bmatrix}\boldsymbol{\mu}\\ \tilde{\boldsymbol{\nu}}\end{bmatrix}. \tag{33}$$

Therefore,

$$\|\boldsymbol{H}_f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{\infty,1} = \frac{2}{\gamma}\left\|\begin{matrix}\boldsymbol{\mu}\\ \tilde{\boldsymbol{\nu}}\end{matrix}\right\|_1 = \frac{4}{\gamma}, \tag{34}$$

and the function $f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is $\frac{4}{\gamma}$-Lipschitz with respect to the ℓ_∞ norm. In summary, the partial gradients are all $\frac{1}{\gamma}$-Lipschitz smooth with respect to ℓ_1, ℓ_2 and ℓ_∞ norms. Additionally, considering the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, seperately, one can see that

$$\|\nabla_{\boldsymbol{\alpha}}^\top\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{1,\infty} = \frac{\mu_{\max}}{\gamma} \leq \frac{1}{\gamma},\ \|\nabla_{\boldsymbol{\alpha}}^\top\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{2,2} \leq \frac{1}{\gamma},\quad \|\nabla_{\boldsymbol{\alpha}}^\top\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{\infty,1} = \frac{1}{\gamma},$$

and

$$\|\nabla_{\boldsymbol{\beta}}^\top\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{1,\infty} = \frac{\tilde{\nu}_{\max}}{\gamma} \leq \frac{1}{\gamma},\ \|\nabla_{\boldsymbol{\beta}}^\top\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{2,2} \leq \frac{1}{\gamma},\quad \|\nabla_{\boldsymbol{\beta}}^\top\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})\|_{\infty,1} = \frac{1}{\gamma}.$$

Therefore, $f(\boldsymbol{\alpha}^{(k+1)},\boldsymbol{\beta}^{(k)})$ is separately $\frac{1}{\gamma}$-Lipschitz for both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with respect to ℓ_1, ℓ_2 and ℓ_∞ norms.