

Sliced Optimal Partial Transport

Yikun Bai^{*1} & Bernhard Schmitzer^{*2} & Matthew Thorpe^{3,4} & Soheil Kolouri¹

¹Department of Computer Science, Vanderbilt University

²Institute of Computer Science, Göttingen University

³Department of Mathematics, University of Manchester

⁴The Alan Turing Institute

Abstract

Optimal transport (OT) has become exceedingly popular in machine learning, data science, and computer vision. The core assumption in the OT problem is the equal total amount of mass in source and target measures, which limits its application. Optimal Partial Transport (OPT) is a recently proposed solution to this limitation. Similar to the OT problem, the computation of OPT relies on solving a linear programming problem (often in high dimensions), which can become computationally prohibitive. In this paper, we propose an efficient algorithm for calculating the OPT problem between two non-negative measures in one dimension. Next, following the idea of sliced OT distances, we utilize slicing to define the Sliced OPT distance. Finally, we demonstrate the computational and accuracy benefits of the Sliced OPT-based method in various numerical experiments. In particular, we show an application of our proposed Sliced OPT problem in noisy point cloud registration and color adaptation. Our code is available at [Github Link](#).

1. Introduction

The Optimal Transport (OT) problem studies how to find the most cost-efficient way to transport one probability measure to another, and it gives rise to popular probability metrics like the Wasserstein distance. OT has attracted abundant attention in data science, statistics, machine learning, signal processing, and computer vision [1, 12, 13, 21, 24, 31, 37, 39, 47, 49]

A core assumption in the OT problem is the equal total amount of mass in the source and target measures (e.g., probability measures). Many practical problems, however, deal with comparing non-negative measures with varying total amounts of mass, e.g., shape analysis [9, 46], domain adaptation [17], color transfer [10]. In addition, OT

distances are often not robust to outliers and noise, as transporting outliers could be prohibitively expensive and might compromise the distance estimation. To address these issues, many variants of the OT problem have been recently proposed, for example, the optimal partial transport (OPT) problem [6, 18, 19], the Hellinger–Kantorovich distance [9, 36], unnormalized optimal transport [22], and Kantorovich–Rubinstein norm [25, 35]. These variants were subsequently unified under the name “unbalanced optimal transport” [11, 36].

The computational complexity of linear programming for balanced and partial OT problems is often a bottleneck for solving large-scale problems. Different approaches have been developed to address this issue. For instance, by entropic regularization, the problem becomes strictly convex and can be solved with the celebrated Sinkhorn–Knopp algorithm [14, 45] which has been extended to the unbalanced setting [10]. This approach can still be computationally expensive for small regularization levels. Other strategies exploit specific properties of ground costs. For example, if the ground cost is determined by the unique path on a tree, the problem can be efficiently solved in the balanced [33, 41] and the unbalanced setting [44]. In particular, balanced 1-dimensional transport problems with convex ground costs can be solved by the northwest corner rule, which essentially amounts to sorting the support points of the two input measures.

Based on this, another popular method is the sliced OT approach [5, 30, 43], which assumes the ground cost is consistent with the Euclidean distance (in 1-dimensional space). Furthermore, it has been shown [30, 32, 44] that the OT distance in Euclidean space can be approximated by the OT distance in 1-dimensional Euclidean space. Inspired by these works, in this paper, we propose the sliced version of OPT and an efficient computational algorithm for empirical distributions with uniform weights, i.e., measures of the form $\sum_{i=1}^n \delta_{x_i}$, where δ_x is the Dirac measure. Our contributions in this paper can be summarized as follows:

- We propose a primal-dual algorithm for 1-dimensional

*These authors contributed equally to this work.

OPT with a quadratic worst-case time complexity and linear or quadratic complexity in practice.

- In d -dimensional space, we propose the Sliced-OPT (SOPT) distance. Similar to the sliced OT distance, we prove that it satisfies the metric axioms and propose a computational method based on our 1-dimensional OPT problem solver.
- We demonstrate an application of SOPT in point cloud registration by proposing a SOPT variant of the *iterative closest point* (ICP) algorithm. Our approach is robust against noise. Also, we apply SOPT to a color adaptation problem (see the supplementary material).

2. Related Work

Linear programming. In the discrete case, the Kantorovich formulation [27] of OT problem is a (high-dimensional) linear program [28]. As shown in [6], OPT can be formulated as a balanced OT problem by introducing *reservoir* points, thus it could also be solved by linear programming. However, the time complexity is prohibitive for large datasets.

Entropy Regularization. Entropic regularization approaches add the transport plan’s entropy to the OT objective function and then apply the Sinkhorn-Knopp algorithm [14, 45]. The algorithm can be extended to the large-scale stochastic [23] and unbalanced setting [2, 10]. For moderate regularization these algorithms converge fast, however, there is a trade-off between accuracy versus stability and convergence speed for small regularization.

Sliced OT. Sliced OT techniques [5, 30, 32, 38, 43] rely on the closed-form solution for the balanced OT map in 1-dimensional Euclidean settings, i.e., the increasing rearrangement function given by the north-west corner rule. The main idea behind these methods is to calculate the expected OT distance between the 1-dimensional marginal distributions (i.e., slices) of two d -dimensional distributions. The expectation is numerically approximated via a Monte Carlo integration scheme. Other extensions of these distances include the generalized and the max-sliced Wasserstein distances [15, 29]. In the unbalanced setting, and for a particular case of OPT, [4] propose a fast (primal) algorithm, which has quadratic worst-case time complexity, and often linear complexity in practice. In particular, Bonneel et al. [4] assume that all the mass in the source measure must be transported to the target measure, i.e., no mass destruction happens in the source measure.

Other computational methods. When the transportation cost is a metric, network flow methods [25, 40] can be applied. For metrics on trees, an efficient algorithm based on dynamic programming with time complexity $\mathcal{O}(n \log^2 n)$ is proposed in [44]. However, in high dimensions existence

(and identification) of an appropriate metric tree remains challenging.

3. Background of Optimal (Partial) Transport

We first review the preliminary concepts of the OT and the OPT problems. In what follows, given $\Omega \subset \mathbb{R}^d, p \geq 1$, we denote by $\mathcal{P}(\Omega)$ the set of Borel probability measures and by $\mathcal{P}_p(\Omega)$ the set of probability measures with finite p ’th moment defined on a metric space (Ω, d) .

Optimal transport. Given $\mu, \nu \in \mathcal{P}(\Omega)$, and a lower semi-continuous function $c : \Omega^2 \rightarrow \mathbb{R}_+$, the OT problem between μ and ν in the *Kantorovich formulation* [27], is defined as:

$$\text{OT}(\mu, \nu) := \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\Omega^2} c(x, y) d\gamma(x, y), \quad (1)$$

where $\Gamma(\mu, \nu)$ is the set of all joint probability measures whose marginal are μ and ν . Mathematically, we denote as $\pi_{1\#}\gamma = \mu, \pi_{2\#}\gamma = \nu$, where π_1, π_2 are canonical projection maps, and for any (measurable) function $f : \Omega^2 \rightarrow \Omega$, $f_{\#}\gamma$ is the push-forward measure defined as $f_{\#}\gamma(A) = \gamma(f^{-1}(A))$ for any Borel set $A \subset \Omega$. When $c(x, y)$ is the p -th power of a metric, the p -th root of the induced optimal value is the **Wasserstein distance**, a metric in $\mathcal{P}_p(\Omega)$.

Optimal Partial Transport. The OPT problem, in addition to mass transportation, allows mass destruction on the source and mass creation on the target. Here the mass destruction and creation penalty will be linear. Let $\mathcal{M}_+(\Omega)$ denote the set of all positive Radon measures defined on Ω , suppose $\mu, \nu \in \mathcal{M}_+(\Omega)$, and $\lambda_1, \lambda_2 \geq 0$, the OPT problem is:

$$\begin{aligned} \text{OPT}_{\lambda_1, \lambda_2}(\mu, \nu) := & \inf_{\substack{\gamma \in \mathcal{M}_+(\Omega^2) \\ \pi_{1\#}\gamma \leq \mu, \pi_{2\#}\gamma \leq \nu}} \int c(x, y) d\gamma \quad (2) \\ & + \lambda_1(\mu(\Omega) - \pi_{1\#}\gamma(\Omega)) + \lambda_2(\nu(\Omega) - \pi_{2\#}\gamma(\Omega)) \end{aligned}$$

where the notation $\pi_{1\#}\gamma \leq \mu$ denotes that for any Borel set $A \subseteq \Omega$, $\pi_{1\#}\gamma(A) \leq \mu(A)$, and we say $\pi_{1\#}\gamma$ is *dominated* by μ , analogously for $\pi_{2\#}\gamma \leq \nu$; the notation $\mu(\Omega)$ denotes the total mass of measure μ . We denote the set of such γ by $\Gamma_{\leq}(\mu, \nu)$. When the transportation cost $c(x, y)$ is a metric, and $\lambda_1 = \lambda_2$, $\text{OPT}(\cdot, \cdot)$ defines a metric on $\mathcal{M}_+(\Omega)$ (see [11, Proposition 2.10], [42, Proposition 5], [34, Section 2.1] and [7, Theorem 4]). For finite λ_1 and λ_2 let $\lambda = \frac{\lambda_1 + \lambda_2}{2}$ and define:

$$\begin{aligned} \text{OPT}_{\lambda}(\mu, \nu) := & \text{OPT}_{\lambda, \lambda}(\mu, \nu) \quad (3) \\ = & \text{OPT}_{\lambda_1, \lambda_2}(\mu, \nu) - K_{\lambda_1, \lambda_2}(\mu, \nu) \end{aligned}$$

where,

$$K_{\lambda_1, \lambda_2}(\mu, \nu) = \frac{\lambda_1 - \lambda_2}{2} \mu(\Omega) + \frac{\lambda_2 - \lambda_1}{2} \nu(\Omega).$$

Since for fixed μ and ν , K_{λ_1, λ_2} is a constant, then without loss of generality, in the rest of the paper, we only consider $\text{OPT}_\lambda(\mu, \nu)$. The case where λ_i 's are not finite is discussed in Section 4.1.

Various equivalent formulations of OPT (3) have appeared in prior work, e.g., [18, 19, 42], which were later unified as a special case of unbalanced OT [11, 36]. We provide a short summary of these formulations and their relationship with unbalanced OT in the supplementary material. OPT has several desirable theoretical properties. For instance, by [42, Proposition 5], minimizing γ exists and they are concentrated on c -cyclical monotone sets. More concretely, we have the following proposition.

Proposition 3.1. *Let γ^* be a minimizer in (3), then the support of γ^* satisfies the c -cyclical monotonicity property: for any $n \in \mathbb{N}$, any $\{(x_i, y_i)\}_{i=1}^n \subset \text{supp}(\gamma^*)$ and any permutation $\sigma : [1 : n] \rightarrow [1 : n]$ we have*

$$\sum_{i=1}^n c(x_i, y_i) \leq \sum_{i=1}^n c(x_i, y_{\sigma(i)}).$$

In particular, in one dimension, for $c(x, y) = f(|x - y|)$ where $f : \mathbb{R} \rightarrow \mathbb{R}_+$ is an increasing and convex function, c -cyclical monotonicity is equivalent to

$$\begin{aligned} (x_1, y_1), (x_2, y_2) \in \text{supp}(\gamma^*), \\ \Rightarrow [x_1 \leq x_2 \text{ and } y_1 \leq y_2] \text{ or } [x_1 \geq x_2 \text{ and } y_1 \geq y_2]. \end{aligned}$$

See the supplementary material for proof. To further simplify (3), we show in the following lemma that the support of the optimal γ does not contain pairs of (x, y) whose cost exceeds λ .

Lemma 3.2. *There exists an optimal γ^* for (3) such that $\gamma^*(S) = 0$ where $S = \{(x, y) : c(x, y) \geq 2\lambda\}$.*

4. Empirical Optimal Partial Transport

In \mathbb{R}^d , suppose μ, ν are n and m -size empirical distributions, i.e., $\mu = \sum_{i=1}^n \delta_{x_i}$ and $\nu = \sum_{j=1}^m \delta_{y_j}$. The OPT problem (3), denoted as $\text{OPT}(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m)$ can be written as

$$\begin{aligned} \text{OPT}(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m) := \min_{\gamma \in \Gamma_{\leq}(\mu, \nu)} \sum_{i,j} c(x_i, y_j) \gamma_{ij} \\ + \lambda(n + m - 2 \sum_{i,j} \gamma_{ij}) \end{aligned} \quad (4)$$

where

$$\Gamma_{\leq}(\mu, \nu) := \{\gamma \in \mathbb{R}_+^{n \times m} : \gamma \mathbf{1}_m \leq \mathbf{1}_n, \gamma^T \mathbf{1}_n \leq \mathbf{1}_m\},^1$$

¹The rigorous notation should be $\Pi_{\leq}(1_n, 1_m)$. We would like to abuse the notation in order to simplify the presentation.

and $\mathbf{1}_n$ denotes the $n \times 1$ vector whose entries are 1 and analogously for $\mathbf{1}_m$. We show that the optimal plan γ for the empirical OPT problem is induced by a 1-1 mapping. A similar result is known for continuous measures μ and ν , see [18, Proposition 2.4 and Theorem 2.6].

Theorem 4.1. *There exists an optimal plan for $\text{OPT}(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m)$, which is induced by a 1-1 mapping, i.e., $\gamma_{ij} \in \{0, 1\}, \forall i, j$ and in each row and column, at most one entry of γ is 1.*

Combining this theorem and the cyclic monotonicity of 1D OPT, we can restrict the optimal mapping to strictly increasing maps.

Corollary 4.2. *For $\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m$ sorted point lists in \mathbb{R} , and a cost function $c(x, y) = h(x - y)$ where $h : \mathbb{R} \rightarrow \mathbb{R}$ is strictly convex, the empirical OPT problem $\text{OPT}(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m)$ can be further simplified to*

$$\text{OPT}(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m) := \min_L C(L) \quad (5)$$

where

$$C(L) := \sum_{i \in \text{dom}(L)} c(x_i, y_{L[i]}) + \lambda(n + m - 2|\text{dom}(L)|),$$

and $L : [1 : n] \rightarrow \{-1\} \cup [1 : m]$, $\text{dom}(L) := \{i : L[i] \neq -1\}$, $|\text{dom}(L)|$ is the cardinality of the set $\text{dom}(L)$, $L|_{\text{dom}(L)} : [1 : n] \hookrightarrow [1 : m]$ is a strictly increasing mapping.²

For convenience, we call any mapping $L : [1 : n] \rightarrow \{-1\} \cup [1 : m]$ a ‘‘transportation plan’’ (or ‘‘plan’’ for short). Furthermore, since L can be represented by a vector, we do not distinguish $L(i)$, $L[i]$ and L_i . Of course, there is a bijection between admissible L and γ and we use this equivalence implicitly in the sequel.

Importantly, the OPT problem is a convex optimisation problem and therefore has a dual form which is given by the following proposition.

Proposition 4.3. *The primal problem (4) admits the dual form*

$$\sup_{\substack{\Phi \in \mathbb{R}^n, \Psi \in \mathbb{R}^m \\ \Phi_i + \Psi_j \leq c(x_i, y_j) \forall i, j}} \sum_{i=1}^n \min\{\Phi_i, \lambda\} + \sum_{j=1}^m \min\{\Psi_j, \lambda\}.$$

Moreover, the following are necessary and sufficient conditions for $\gamma \in \Gamma_{\leq}(\mu, \nu)$, $\Phi \in \mathbb{R}^n$ and $\Psi \in \mathbb{R}^m$

²Here, mapping a point to $\{-1\}$ corresponds to destroying the point in the source, while unmatched points in the target are created. Hence, L uniquely represents the partial transport, and $|\text{dom}(L)|$ is the cardinality of $\text{dom}(L)$.

to be optimal for the primal and dual problems:

$$\begin{aligned} \Phi_i + \Psi_j &= c(x_i, y_j), \forall (x_i, y_j) \in \text{supp}(\gamma) \\ \Phi_i < \lambda &\Rightarrow [\pi_{1\#}\gamma]_i = 1 & \Psi_j < \lambda &\Rightarrow [\pi_{2\#}\gamma]_j = 1 \\ \Phi_i = \lambda &\Rightarrow [\pi_{1\#}\gamma]_i \in [0, 1] & \Psi_j = \lambda &\Rightarrow [\pi_{2\#}\gamma]_j \in [0, 1] \\ \Phi_i > \lambda &\Rightarrow [\pi_{1\#}\gamma]_i = 0 & \Psi_j > \lambda &\Rightarrow [\pi_{2\#}\gamma]_j = 0 \end{aligned}$$

4.1. A Polynomial Time Algorithm

Our algorithm finds the solutions γ , Φ , and Ψ to the primal-dual optimality conditions given in Proposition 4.3. We assume that, at iteration k , we have solved $\text{OPT}(\{x_i\}_{i=1}^{k-1}, \{y_j\}_{j=1}^m)$ in the previous iteration (stored in $L[1 : k-1]$) and we now proceed to solve $\text{OPT}(\{x_i\}_{i=1}^k, \{y_j\}_{j=1}^m)$ in the current iteration. Recall that we assume that $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$ are sorted and that $L[i]$ is the index determining the transport of mass from x_i , i.e., if $L[i] \neq -1$ then x_i is associated to $y_{L[i]}$. For simplicity we assume for now that all points in $\{x_i\}_{i=1}^n$ are distinct, and likewise for $\{y_j\}_{j=1}^m$. Duplicate points can be handled properly with some minor additional steps (see supplementary material). Let j^* be the index of the most attractive $\{y_j\}_{j=1}^m$ for the new point x_k under consideration of the dual variables, i.e., $j^* = \arg\min_{j \in [1:m]} c(x_k, y_j) - \Psi_j$ and set $\Phi_k = \min\{c(x_k, y_{j^*}) - \Psi_{j^*}, \lambda\}$ so that Φ_k is the largest possible value satisfying the dual constraints, but no greater than λ since at this point the dual objective becomes flat. We now distinguish three cases:

Case 1: If $\Phi_k = \lambda$, then destroying x_k is the most efficient action, i.e., we set $L[k] = -1$ and proceed to $k+1$.

Case 2: If $\Phi_k < \lambda$ and y_{j^*} is unassigned, then we set $L[k] = j^*$ and we can proceed to $k+1$.

Case 3: If $\Phi_k < \lambda$ and y_{j^*} is already assigned, we must resolve the conflict between x_k and the element currently assigned to y_{j^*} . This will be done by a sub-algorithm.

It is easy to see that in the first two cases, if L (or γ), Φ and Ψ satisfy the primal-dual conditions of Proposition 4.3 up until $k-1$, they will now satisfy them up until k . Let us now study the third case in more detail.

One finds that if y_{j^*} is already assigned, then it must be to x_{k-1} (proof in supplementary material). We now increase Φ_i for $i \in [k-1 : k]$ while decreasing Ψ_{j^*} (at the same rate) until one of the following cases occurs:

Case 3.1: Either Φ_{k-1} or Φ_k reaches λ . In this case, the corresponding x becomes unassigned and the other becomes (or remains) assigned to j^* . The conflict is resolved and we proceed to solve the problem for $k+1$.

Case 3.2: One reaches the point where $\Phi_k + \Psi_{j^*+1} = c(x_k, y_{j^*+1})$. In this case, x_k becomes assigned to y_{j^*+1} , the conflict is resolved and we move to $k+1$.

Case 3.3a: One reaches the point where $\Phi_{k-1} + \Psi_{j^*-1} = c(x_{k-1}, y_{j^*-1})$. If y_{j^*-1} is unassigned, we assign x_{k-1} to y_{j^*-1} , x_k to y_{j^*} . The conflict is resolved and we move on.

Case 3.3b: In the remaining case where y_{j^*-1} is already assigned, we will show that it must be to x_{k-2} . This means that the set of points involved in the conflict increases and we must perform a slight generalization of the above case 3 iteration until eventually one of the other cases occurs.

At each iteration we consider a contiguous set of $\{x_i\}_{i=i_{\min}}^{k-1}$ assigned monotonously to contiguous $\{y_j\}_{j=j_{\min}}^{j^*}$, where i_{\min} is initially set $i_{\min} = k-1$ and j_{\min} is the index of y_j that has been assigned to $x_{i_{\min}}$ (i.e., $j_{\min} = j^*$), and the additional point x_k . We increase Φ_i for $i \in [i_{\min} : k]$ and decrease Ψ_j for $j \in [j_{\min} : j^*]$, until either $\Phi_{k'}$ becomes equal to λ for some $k' \in [i_{\min} : k+1]$ (Case 3.1), $\Phi_k + \Psi_{j^*+1} = c(x_k, y_{j^*+1})$ (Case 3.2) or $\Phi_{i_{\min}} + \Psi_{j_{\min}-1} = c(x_{i_{\min}}, y_{j_{\min}-1})$ (Case 3.3). In Cases 3.1 and 3.2 the conflict can be resolved in an obvious way. The same holds for Case 3.3, when $y_{j_{\min}-1}$ is unassigned (Case 3.3a). Otherwise (Case 3.3b), one adds one more assigned pair to the chain and restarts the loop with the pair $(x_{i_{\min}-1}, y_{j_{\min}-1})$ added to the chain. Of course, trivial adaptations have to be made to account for boundary effects, e.g., Case 3.2 cannot occur if $j^* = m$ etcetera. A pseudocode description of the method is given by Algorithms 1 and 2, and a visual illustration of these algorithms is provided in Figure 1. Note that for the sake of legibility, we make some simplifications, e.g., boundary checks are ignored (see above), and we do not specify how to keep track of whether j^* is assigned or not. Also, updating the dual variables at each iteration of the sub-routine yields a cubic worst-case time complexity, but is easier to understand. A complete version of the algorithm with all checks, appropriate data structures, and quadratic complexity is given in the supplementary material. There we also prove the following claim:

Theorem 4.4. *Algorithm 1 is correct, i.e., it is well-defined and returns optimal primal and dual solutions L and (Φ, Ψ) to the 1-dimensional OPT problem for sorted $\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m$. A slight adaptation of the algorithm (given in the supplementary material) has a worst-case time complexity of $O(n \max\{m, n\})$.*

The algorithm can be adjusted to the case where $\lambda_1 = \infty$ by setting all instances of λ in Algorithms 1 and 2 to $+\infty$ except for the initialization of Ψ . This means that cases 1 and 3.1 never occur. The algorithm then reduces to a primal-dual version of that given in [4]. If $\lambda_2 = \infty$, then one simply flips the two marginals and proceeds as for $\lambda_1 = \infty$.

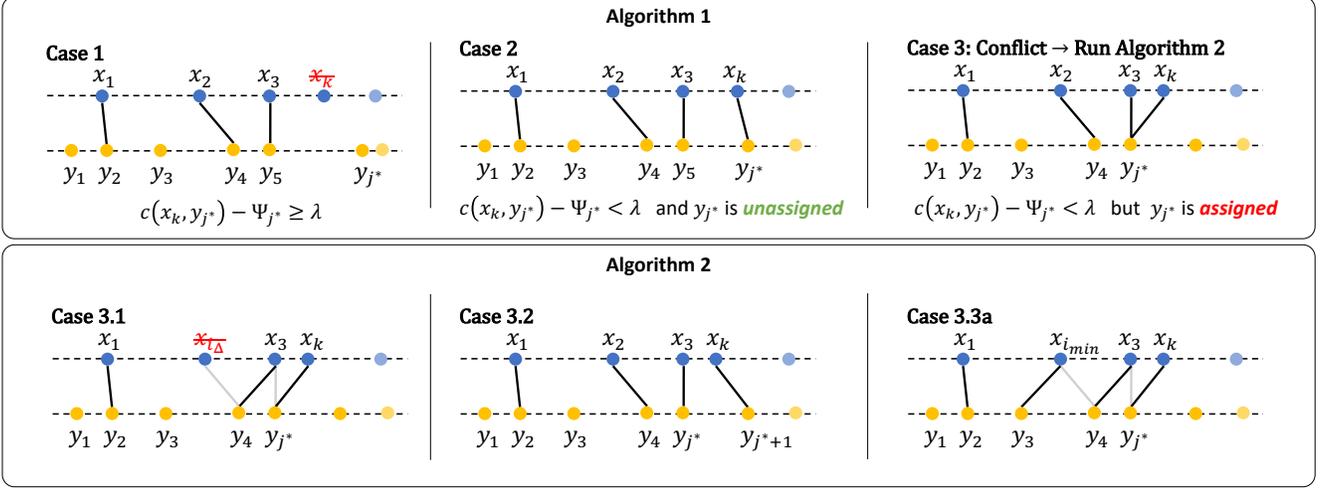


Figure 1. Depiction of Algorithms 1 and 2 for solving the optimal partial transport in one dimension.

Algorithm 1: opt-1d

Input: $\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m, \lambda$

Output: L, Ψ, Φ

- 1 Initialize $\Phi_i \leftarrow -\infty$ for $i \in [1 : n]$, $\Psi_j \leftarrow \lambda$ for $j \in [1 : m]$ and $L_i \leftarrow -1$ for $i \in [1 : n]$
 - 2 **for** $k = 1, 2, \dots, n$ **do**
 - 3 $j^* \leftarrow \operatorname{argmin}_{j \in [1:m]} c(x_k, y_j) - \Psi_j$
 - 4 $\Phi_k \leftarrow \min\{c(x_k, y_{j^*}) - \Psi_{j^*}, \lambda\}$
 - 5 **if** $\Phi_k = \lambda$ **then**
 - 6 [Case 1] No update on L
 - 7 **else if** j^* *unassigned* **then**
 - 8 [Case 2] $L_k \leftarrow j^*$
 - 9 **else**
 - 10 [Case 3] Run Algorithm 2.
-

4.2. Runtime Analysis

We test the wall clock time by sampling the point lists from uniform distributions and Gaussian mixtures. In particular, we set $\{x_i\}_{i=1}^n \stackrel{\text{iid}}{\sim} \text{Unif}[-20, 20]$, $\{y_j\}_{j=1}^m \stackrel{\text{iid}}{\sim} \text{Unif}[-40, 40]$, $\lambda \in \{20, 100\}$ and $\{x_i\}_{i=1}^n \stackrel{\text{iid}}{\sim} \frac{1}{5} \sum_{k=1}^5 \mathcal{N}(-4 + 2k, 1)$, $\{y_j\}_{j=1}^m \stackrel{\text{iid}}{\sim} \frac{1}{6} \sum_{k=1}^6 \mathcal{N}(-5 + 2k, 1)$, $\lambda \in \{2, 10\}$, $n \in \{2000, 2500, \dots, 10000\}$, and $m = n + 1000$. We compare our algorithm with POT (Algorithm 1 in [4]), an unbalanced Sinkhorn algorithm [10] (we set the entropic regularization parameter to $\lambda/40$), and linear programming in *python OT* [20]. Our algorithm, POT, and Sinkhorn algorithm are accelerated by *numba* and linear programming is written in C++. Note that POT [4] and the unbalanced Sinkhorn minimize a different model. In addition, for the latter, the performance depends strongly on

the strength of regularization and we found that it was not competitive in the regime of a low blur. For each (n, m) , we repeat the computation 10 times and compute the average. For our method and POT, the time of sorting is included, and for the linear programming and Sinkhorn algorithms, the time of computing the cost matrix is included. We also visualize our algorithm's solutions for $\{x_i\}_{i=1}^n \stackrel{\text{iid}}{\sim} \text{Unif}(-20, 20)$, $\{y_j\}_{j=1}^m \stackrel{\text{iid}}{\sim} \text{Unif}(-40, 40)$, $n = 8, m = 16$ and $\lambda \in \{1, 10, 100, 1000\}$ (see Figure 3). One can observe that as λ increases, larger transportations are permitted. The data type is 64-bit float number, and the experiments are conducted on a Google Colab Pro+ virtual machine.

5. Sliced Optimal Partial Transport

In practice, data has multiple dimensions and the 1-D computational methods cannot be applied directly. In the balanced OT setting the sliced OT approach [5, 29, 30, 32, 43] applies the 1-D OT solver on projections (i.e., slices) of two r -dimensional distributions. Inspired by these works, we extend the sliced OT technique into the OPT setting and introduce the so-called ‘‘sliced-unbalanced optimal transport’’ problem.

Definition 5.1. In \mathbb{R}^d space, given $\mu, \nu \in \mathcal{M}_+(\Omega)$ where $\Omega \subset \mathbb{R}^d$ and $\lambda : \mathbb{S}^{d-1} \rightarrow \mathbb{R}_{++}$ is an L^1 function, we define the sliced optimal partial transport (SOPT) problem as follows:

$$\text{SOPT}_\lambda(\mu, \nu) = \int_{\mathbb{S}^{d-1}} \text{OPT}_{\lambda(\theta)}(\langle \theta, \cdot \rangle_\# \mu, \langle \theta, \cdot \rangle_\# \nu) d\sigma(\theta) \quad (6)$$

where $\text{OPT}_\lambda(\cdot, \cdot)$ is defined in (3), $\sigma \in \mathcal{P}(\mathbb{S}^{d-1})$ is a probability measure such that $\text{supp}(\sigma) = \mathbb{S}^{d-1}$.

Algorithm 2: sub-opt

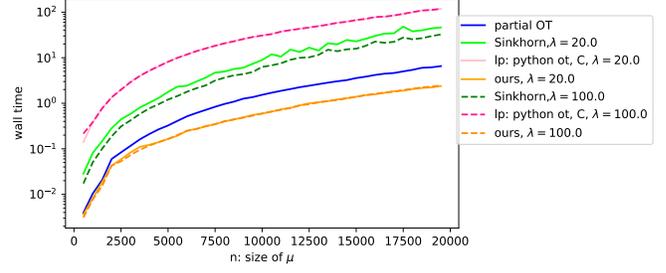
Input: $(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m, k, j^*, L, \Phi, \Psi)$
Output: (Updated L, Φ, Ψ , optimal for $\text{OPT}(\{x_i\}_{i=1}^k, \{y_j\}_{j=1}^m)$)

- 1 Initialize $i_{\min} \leftarrow k - 1, j_{\min} \leftarrow j^*$.
- 2 **while true do**
- 3 $i_{\Delta} \leftarrow \text{argmin}_{i \in [i_{\min}:k]} (\lambda - \Phi_i)$
- 4 $\lambda_{\Delta} \leftarrow \lambda - \Phi_{i_{\Delta}}$
- 5 $\alpha \leftarrow c(x_k, y_{j^*+1}) - \Phi_k - \Psi_{j^*+1}$
- 6 $\beta \leftarrow c(x_{i_{\min}}, y_{j_{\min}-1}) - \Phi_{i_{\min}} - \Psi_{j_{\min}-1}$
- 7 **if** $\lambda_{\Delta} \leq \min\{\alpha, \beta\}$ **then**
- 8 [Case 3.1]
- 9 $\Phi_i \leftarrow \Phi_i + \lambda_{\Delta}$ for $i \in [i_{\min} : k + 1]$
- 10 $\Psi_j \leftarrow \Psi_j - \lambda_{\Delta}$ for $j \in [j_{\min} : j^*]$
- 11 $L_{i_{\Delta}} \leftarrow -1, L_k \leftarrow j^*$
- 12 **for** $i \in [i_{\Delta} + 1 : k - 1]$ **do**
- 13 $L_i \leftarrow L_i - 1$
- 14 **return**
- 15 **else if** $\alpha \leq \min\{\lambda_{\text{diff}}, \beta\}$ **then**
- 16 [Case 3.2]
- 17 $\Phi_i \leftarrow \Phi_i + \alpha$ for $i \in [i_{\min} : k + 1]$
- 18 $\Psi_j \leftarrow \Psi_j - \alpha$ for $j \in [j_{\min} : j^*]$
- 19 $L_k \leftarrow j^* + 1$
- 20 **return**
- 21 **else**
- 22 $\Phi_i \leftarrow \Phi_i + \beta$ for $i \in [i_{\min} : k + 1]$
- 23 $\Psi_j \leftarrow \Psi_j - \beta$ for $j \in [j_{\min} : j^*]$
- 24 **if** $j_{\min} - 1$ *unassigned* **then**
- 25 [Case 3.3a]
- 26 $L_{i_{\min}} \leftarrow j_{\min} - 1, L_k \leftarrow j^*$
- 27 **for** $i \in [i_{\min} + 1 : k - 1]$ **do**
- 28 $L_i \leftarrow L_i - 1$
- 29 **return**
- 30 **else**
- 31 [Case 3.3b]
- 32 $i_{\min} \leftarrow i_{\min} - 1, j_{\min} \leftarrow j_{\min} - 1$

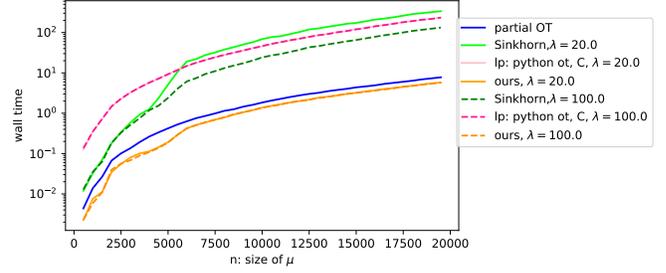
Generally, σ is set to the uniform distribution on the unit ball \mathbb{S}^{d-1} . Whenever $\text{supp}(\sigma) = \mathbb{S}^{d-1}$, $\text{SOPT}_{\lambda}(\mu, \nu)$ defines a metric.

Theorem 5.2. *Suppose $c : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$ is the p -th power of a metric on \mathbb{R} where $p \in [1, \infty)$, and $\lambda \in L^1(\mathbb{S}^{d-1}; \mathbb{R}_{++})$, then $(\text{SOPT}_{\lambda}(\mu, \nu))^{1/p}$ is a metric in $\mathcal{M}_+(\Omega)$.*

In practice, this integration is usually approximated using a Monte Carlo scheme that draws a finite number of i.i.d. samples $\{\theta_l\}_{l=1}^N$ from $\text{Unif}(\mathbb{S}^{d-1})$ and replaces the integral



(a) wall-clock time for uniform distributions



(b) Wall-clock time for Gaussian mixture distribution

Figure 2. We test the wall-clock time of our Algorithm 1, the partial OT solver (Algorithm 1 in [4]), the unbalanced Sinkhorn algorithm [10], and the linear programming solver in POT [20], which is written in C++. The maximum number of iterations for linear programming and the Sinkhorn algorithm is set to $200 \ln(n)$.

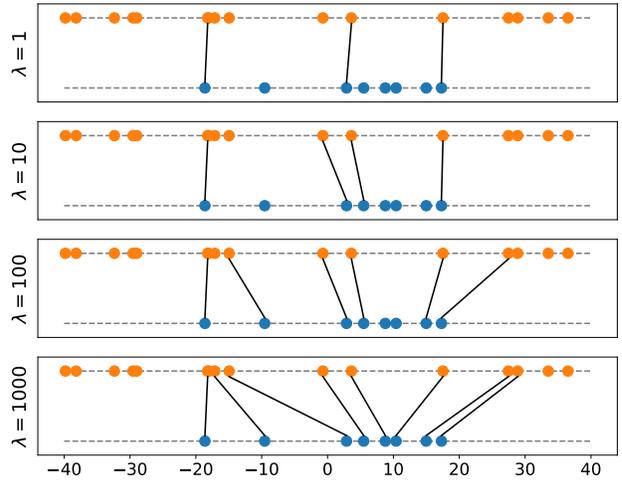


Figure 3. Output of the proposed algorithm for $\lambda \in [1, 10, 100, 1000]$ on a sample pair of measures.

with an empirical average:

$$\text{SOPT}_{\lambda}(\mu, \nu) \approx \frac{1}{N} \sum_{l=1}^N \text{OPT}_{\lambda_l}(\langle \theta_l, \cdot \rangle_{\#} \mu, \langle \theta_l, \cdot \rangle_{\#} \nu).$$

	10k,5%	10k,7%	9k,5%	9k,7%
ICP-D	1.10(1.59)	3.60(0.11)	1.65(1.13)	2.04(2.28)
ICP-U	3.72(0.53)	3.72(0.52)	3.69(0.63)	3.92(0.32)
SPOT	1.27(0.01)	1.40(0.15)	1.42(0.13)	1.53(1.50)
Ours	0.01(1e-3)	0.02(2e-3)	0.20(0.09)	0.33(0.03)

Table 1. We compute the mean (and variance, in parenthesis) of errors in the Frobenius norm between the ground truth and estimated transportation matrices for ICP(Du), ICP(Umeyama), SPOT, and our method. We vary the size of the source point cloud from 9k to 10k samples, the percentage of noise from 5% to 7% (on both source and target datasets).

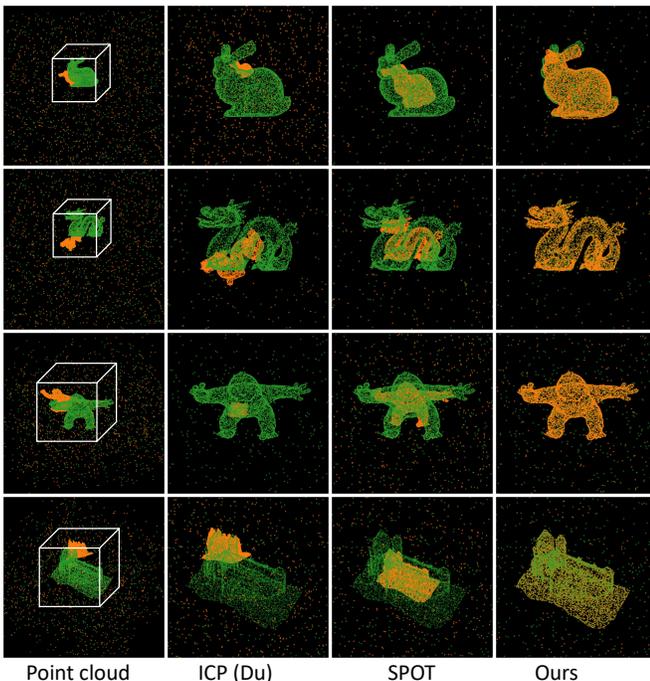


Figure 4. We visualize the results of ICP(Du), SPOT and our method. In each image, the target point cloud is in green and the source point cloud is in orange.

6. Application: Point Cloud Registration

Point cloud registration is a transformation estimation problem between two point clouds, which is a critical problem in numerous computer vision applications [26, 48]. In particular, given two 3-D point clouds, $x_i \sim \mu, i = 1, \dots, n$ and $y_j \sim \nu, j = 1, \dots, m$, one assumes there is an unknown mapping, T , that satisfies $\nu = T_{\#}\mu$. In many applications, the mapping T is restricted to have the following form, $Tx = sRx + \beta$, where R is a 3×3 dimensional rotation matrix, $s > 0$ is the scaling and β , called translation vector, is 3×1 vector. The goal is then to estimate the transform, T , based on the two point clouds.

The classic approach for solving this problem is Itera-

tive Closest Point Algorithms (ICP) introduced by [3, 8]. By [50]’s work, classical ICP can be extended into the uniformly scaled setting, with further developments by [16]. To address some issues of ICP methods (convergence to a local minimum, poor performance when the size of the two datasets are not equal), [4] proposed the Fast Iterative Sliced Transport algorithm (FIST) using sliced partial optimal transport (SPOT).

Problem setup and our method. We consider the uniform scaled point cloud registration problem and assume a subset of points in both the source and target point clouds are corrupted with additional uniformly distributed noise. We suppose prior knowledge of the proportion of noise is given (i.e., we have prior knowledge of a number of clean data).

In general, the registration problem can be iteratively solved, and each iteration contains two steps: estimating the correspondence and computing the optimal transform from corresponding points. The second step has a closed-form solution. For the first step, the ICP method estimates the correspondence by finding the closest y for each transformed x . Inspired by this work, we estimate the correspondence by using our SOPT solver. See the algorithm 3.

Algorithm 3: iterative-sopt

Input: $\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m, n_0$: the # of clean x , N : # of projections

Output: R, s, β

- 1 initialize R, s, β, λ , sample $\{\theta_i\}_{i=1}^N \subset \mathbb{S}^2$
- 2 **for** $l = 1, \dots, N$ **do**
- 3 $\hat{Y} \leftarrow sRX + \beta$
- 4 Compute transportation plan L of $\text{OPT}_{\lambda}(\theta_l^T \hat{Y}, \theta_l^T Y)$ by algorithm 1
- 5 $\forall i \in \text{dom}(L), \hat{y}_i \leftarrow \hat{y}_i + (\theta_l^T y_{L[i]} - \theta_l^T \hat{y}_i)\theta$
- 6 Compute R, s, β from $(X[\text{dom}(L)], \hat{Y}[\text{dom}(L)])$ by ICP (e.g., equations (39)-(42) in [50])
- 7 If $|\text{dom}(L)| > n_0$, decrease λ ; otherwise, increase λ .

Experiment. We illustrate our algorithm on different 3D point clouds, including **Stanford Bunny**, **Stanford dragon**, **Witch-castle** and **Mumble Sitting**. For each dataset, we generate transforms by uniformly sampling angles from $[-1/3\pi, 1/3\pi]$, translations from $[-2std, 2std]$, and scalings from $[0, 2]$, where std is the standard deviation of the dataset. Then we sample noise uniformly from the region $[-2M, 2M]^3$ where $M = \max_{i \in [1:n]}(\|x_i\|)$ and concatenate it to our point clouds. The number of points in the target (clean) data is fixed to be 10k and we vary the number of points in the source (clean) data from 9k to 10k, and the percentage of noise from 5% to 7%.

Performance. For accuracy, we compute the average and

Dataset(% Noise)	ICP				SPOT		Ours	
	Source	Target	Time/iter.	# Iter.	Time/iter.	# Iter.	Time/iter.	# Iter.
Bunny (5%)	9k	10k	0.66s	50	10.9s	100	0.31s	2000
Bunny (7%)	10k	10k	0.76s	130	13.2s	100	0.35s	2000
Dragon (5%)	9k	10k	0.66s	100	10.4s	100	0.31s	2000
Dragon (7%)	10k	10k	0.77s	100	13.1s	80	0.35s	1500
Mumble (5%)	9k	10k	0.65s	100	10.78s	100	0.32s	2000
Mumble (7%)	10k	10k	0.78s	100	13.18s	80	0.36s	1500
Castle (5%)	9k	10k	0.66s	150	10.7s	100	0.31s	2000
Castle (7%)	10k	10k	0.76s	350	13.7s	80	0.35s	1800

Table 2. This table reports the data for our method in the shape registration experiment: the number of source and target distributions, the percentage of noise, the computation times per iteration, and the number of iterations they took to converge for ICP(Du), SPOT, and our method. The source point cloud is in orange color and the target is in blue color.

standard deviation of error defined by the Frobenius norm between the estimated transformation and the ground truth (see Table 1). For accuracy, we observe ICP methods systematically fail, since the nearest neighbor matching technique induces a non-injective correspondence, which may result in a too-small scaling factor. SPOT can successfully recover the rotation, but it fails to recover the scaling. Our method is the only one that recovers the ground truth for the noise-corrupted data (since it utilizes prior knowledge).

For the running time, ICP has the fastest convergence time, which is generally 100-260 seconds, since finding the correspondence by the closest neighbor can be done immediately after the cost matrix is computed. SPOT requires 1000-1300 seconds and our method requires 500-700 seconds. The data type is 32-bit float number and the experiment is conducted on a Linux computer with AMD EPYC 7702P CPU with 64 cores and 256GB DDR4 RAM.

Lastly, we provide additional experimental results on a different application, namely image color adaptation, in the supplementary material to demonstrate the generality of the proposed metric.

7. Conclusion and future work

This paper proposes a fast computational method for solving the OPT problem for one-dimensional discrete measures. We provide computational and wall-clock analysis experiments to assess our proposed algorithm’s correctness and computational benefits. Then, utilizing one-dimensional slices of an r -dimensional measure, we propose the “sliced optimal partial transport (SOPT)” distance. Beyond our theoretical and algorithmic contributions, we provide empirical evidence that SOPT is practical for large-scale applications like point cloud registration and image color adaptation (see supplementary material). In point cloud registration, we show that compared with other classical methods, our SOPT-based approach adds robustness

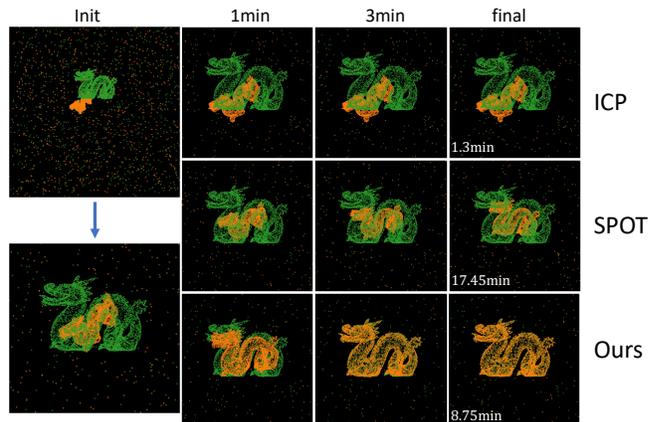


Figure 5. We visualize the processed point cloud for every methods with respect to time. The dataset is **Stanford dragon**. In each image, the point cloud in orange is source and the point cloud in green color is the target.

when the source and target point clouds are corrupted by noise. In the future, we will investigate the potential applications of SOPT in other machine learning tasks such as open set domain adaptation problems and measuring task similarities in continual and curriculum learning.

8. Acknowledgment

The authors thank Rana Muhammad Shahroz Khan for helping in code testing and thank Dr. Hengrong Du (hengrong.du@vanderbilt.edu) for helpful discussions. This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR00112190132. MT was supported by the European Research Council under the European Union’s Horizon 2020 research and innovation programme Grant Agreement No. 777826 (NoMADS).

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. [1](#)
- [2] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015. [2](#)
- [3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. [7](#)
- [4] Nicolas Bonneel and David Coeurjolly. SPOT: sliced partial optimal transport. *ACM Transactions on Graphics*, 38(4):1–13, 2019. [2](#), [4](#), [5](#), [7](#)
- [5] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. [1](#), [2](#), [6](#)
- [6] Luis A Caffarelli and Robert J McCann. Free boundaries in optimal transport and monge-ampere obstacle problems. *Annals of mathematics*, pages 673–730, 2010. [1](#), [2](#)
- [7] Yongxin Chen, Tryphon T Georgiou, Lipeng Ning, and Allen Tannenbaum. Matricial wasserstein-1 distance. *IEEE control systems letters*, 1(1):14–19, 2017. [2](#)
- [8] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. [7](#)
- [9] Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18(1):1–44, 2018. [1](#)
- [10] Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018. [1](#), [2](#), [5](#)
- [11] Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018. [1](#), [2](#), [3](#)
- [12] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30, 2017. [1](#)
- [13] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014. [1](#)
- [14] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. [1](#), [2](#)
- [15] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10648–10656, 2019. [2](#)
- [16] Shaoyi Du, Nanning Zheng, Shihui Ying, Qubo You, and Yang Wu. An extension of the icp algorithm considering scale factor. In *2007 IEEE International Conference on Image Processing*, volume 5, pages V–193. IEEE, 2007. [7](#)
- [17] Kilian Fatras, Thibault Séjourné, Rémi Flamary, and Nicolas Courty. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning*, pages 3186–3197. PMLR, 2021. [1](#)
- [18] Alessio Figalli. The optimal partial transport problem. *Archive for rational mechanics and analysis*, 195(2):533–560, 2010. [1](#), [3](#)
- [19] Alessio Figalli and Nicola Gigli. A new transportation distance between non-negative measures, with applications to gradients flows with dirichlet boundary conditions. *Journal de mathématiques pures et appliquées*, 94(2):107–130, 2010. [1](#), [3](#)
- [20] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. [5](#)
- [21] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a Wasserstein loss. *Advances in neural information processing systems*, 28, 2015. [1](#)
- [22] Wilfrid Gangbo, Wuchen Li, Stanley Osher, and Michael Puthawala. Unnormalized optimal transport. *Journal of Computational Physics*, 399:108940, 2019. [1](#)
- [23] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016. [2](#)
- [24] Aude Genevay, Gabriel Peyré, and Marco Cuturi. GAN and VAE from an optimal transport point of view. *arXiv preprint arXiv:1706.01807*, 2017. [1](#)
- [25] Kevin Guittet. *Extended Kantorovich norms: a tool for optimization*. PhD thesis, INRIA, 2002. [1](#), [2](#)
- [26] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690*, 2021. [6](#)
- [27] Leonid V Kantorovich. On a problem of monge. In *CR (Doklady) Acad. Sci. URSS (NS)*, volume 3, pages 225–226, 1948. [2](#)
- [28] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984. [2](#)
- [29] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced Wasserstein distances. *Advances in Neural Information Processing Systems*, 32, 2019. [2](#), [6](#)
- [30] Soheil Kolouri, Se Rim Park, and Gustavo K Rohde. The radon cumulative distribution transform and its application

- to image classification. *IEEE transactions on image processing*, 25(2):920–934, 2015. 1, 2, 6
- [31] 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. 1
- [32] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016. 1, 2, 6
- [33] Tam Le, Makoto Yamada, Kenji Fukumizu, and Marco Cuturi. Tree-sliced variants of wasserstein distances. *Advances in neural information processing systems*, 32, 2019. 1
- [34] Wonjun Lee, Rongjie Lai, Wuchen Li, and Stanley Osher. Generalized unnormalized optimal transport and its fast algorithms. *Journal of Computational Physics*, 436:110041, 2021. 2
- [35] Jan Lellmann, Dirk A Lorenz, Carola Schonlieb, and Tuomo Valkonen. Imaging with kantorovich–rubinstein discrepancy. *SIAM Journal on Imaging Sciences*, 7(4):2833–2859, 2014. 1
- [36] Matthias Liero, Alexander Mielke, and Giuseppe Savare. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018. 1, 3
- [37] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. Wasserstein GAN with quadratic transport cost. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4832–4841, 2019. 1
- [38] Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pages 4104–4113. PMLR, 2019. 2
- [39] Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted Boltzmann machines. *Advances in Neural Information Processing Systems*, 29, 2016. 1
- [40] James Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Proceedings of the Twentieth annual ACM symposium on Theory of Computing*, pages 377–387, 1988. 2
- [41] 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. 1
- [42] Benedetto Piccoli and Francesco Rossi. Generalized wasserstein distance and its application to transport equations with source. *Archive for Rational Mechanics and Analysis*, 211(1):335–358, 2014. 2, 3
- [43] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Berton. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011. 1, 2, 6
- [44] Ryoma Sato, Marco Cuturi, Makoto Yamada, and Hisashi Kashima. Fast and robust comparison of probability measures in heterogeneous spaces. *arXiv preprint arXiv:2002.01615*, 2020. 1, 2
- [45] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964. 1, 2
- [46] Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015. 1
- [47] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, pages 306–314. PMLR, 2014. 1
- [48] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 6
- [49] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. 1
- [50] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 7