# Fast and Accurate Cost-Scaling Algorithm for the Semi-Discrete Optimal Transport

**Pankaj K. Agarwal[1], Sharath Raghvendra[2], Pouyan Shirzadian[2], and Keegan Yao[1]**

[1]Duke University, [2]Virginia Tech

## Abstract

Given a continuous probability distribution $\mu$ and a discrete distribution $\nu$ in the $d$-dimensional space, the semi-discrete Optimal Transport (OT) problem asks for computing a minimum-cost plan to transport mass from $\mu$ to $\nu$. In this paper, given any parameter $\varepsilon > 0$, we present an algorithm that computes a semi-discrete transport plan $\tilde{\tau}$ with cost $\text{¢}(\tilde{\tau}) \leq \text{¢}(\tau^*) + \varepsilon$ in $n^{O(d)} \log \frac{\mathrm{D}}{\varepsilon}$ time; here, $\tau^*$ is the optimal transport plan, D is the diameter of the supports of $\mu$ and $\nu$, and we assume we have access to an oracle that outputs the mass of $\mu$ inside a constant-complexity region in $O(1)$ time. Our algorithm works for several ground distances including the $L_p$-norm and the squared-Euclidean distance.

## 1 Introduction

Optimal transport (OT) is a powerful tool for comparing probability distributions and computing maps between them. Put simply, the optimal transport problem deforms one distribution to the other with smallest possible cost. When both input distributions are discrete, we refer to the problem as the *discrete optimal transport* problem. The problem is called *semi-discrete* when one input distribution is discrete and the other is a continuous one. Classically, the OT problem has been extensively studied within the mathematics, statistics, and operations research [37, 38, 47]. In recent years, optimal transport has seen rapid rise in various machine learning and computer vision applications as a meaningful metric between distributions. The discrete optimal transport has been extensively used in generative models [7, 18, 24, 43], supervised learning [27, 35], computer vision applications [9, 26], and parameter estimation [11, 34]. The semi-discrete OT has also been used in various statistical and machine learning applications, such as variational inference [6] and blue noise generation [17, 41]. There are several provable and efficient exact and approximation algorithms for the discrete OT problem. However, very little is known for the semi-discrete OT problem. See the book [40] for a review of the various applications of and algorithms for the OT problem. In this paper, we present a new provable polynomial time combinatorial additive approximation algorithm for the $d$-dimensional semi-discrete OT problem.

**Problem Definition.** Let $\mu$ be a continuous probability distribution (i.e., density) defined over a compact bounded support $A \subset \mathbb{R}^d$, and let $\nu$ be a discrete distribution, where the support of $\nu$, denoted by $B$, is a set of $n$ points in $\mathbb{R}^d$. Let $\mathrm{d}(\cdot, \cdot)$ be the ground distance between a pair of points in $\mathbb{R}^d$. A coupling $\tau \colon A \times B \to \mathbb{R}_{\geq 0}$ is called a *transport plan* for $\mu$ and $\nu$ if for all $a \subseteq A$, $\sum_{b \in B} \tau(a, b) = \mu(a)$ (where $\mu(a)$ is the mass of $\mu$ inside $a$) and for all $b \in B$, $\int_A \tau(a, b)\, da = \nu(b)$. The cost of the transport plan $\tau$ is given by $\text{¢}(\tau) := \int_A \sum_{b \in B} \mathrm{d}(a, b) \tau(a, b)\, da$. The goal is to find

[*] This work appears as part of a paper in the proceedings of the Symposium on Discrete Algorithms (SODA) 2024 [3], and the full version of the paper can be accessed at `https://arxiv.org/abs/2311.02172`

Accepted to the Optimal Transport and Machine Learning Workshop at NeurIPS 2023.

a minimum-cost (semi-discrete) transport plan satisfying $\mu$ and $\nu$[1]. For any parameter $\varepsilon > 0$, a transport plan $\tau$ between $\mu$ and $\nu$ is called *$\varepsilon$-close* if the cost of $\tau$ is within an additive error of $\varepsilon$ from the cost of the optimal transport plan $\tau^*$, i.e., $\cent(\tau) \le \cent(\tau^*) + \varepsilon$.

The problem of computing semi-discrete OT between $\mu$ and $\nu$ reduces to the problem of finding a set of weights $y : B \to \mathbb{R}_{\ge 0}$ so that, for any point $b \in B$, the Voronoi cell of $b$ in the additively weighted Voronoi diagram has a mass equal to $\nu(b)$, i.e., $\mathrm{Vor}(b) = \{x \in \mathbb{R}^d \mid \mathrm{d}(x, b) - y(b) \le \mathrm{d}(x, b') - y(b'), \forall b' \in B\}$, $\mu(\mathrm{Vor}(b)) = \nu(b)$, and the mass of $\mu$ in $\mathrm{Vor}(b)$ is transported to $b$; see [8]. One can thus define an optimal semi-discrete transport plan by describing the weights of points in $B$. For arbitrary distributions, weights can have large bit (or algebraic) complexity, so our goal will be to compute the weights accurately up to $s = O(\log \varepsilon^{-1})$ bits, which in turn will return an $\varepsilon$-close semi-discrete OT plan.

**Related work.** Many known algorithms for semi-discrete OT compute an $\varepsilon$-close transport plan using the first- and second-order numerical solvers [8, 10, 14, 17, 29, 30, 33, 38]. These algorithms start with an initial set of weights for points in $B$ and iteratively improve the weights until the mass inside the Voronoi cell of any point $b \in B$ is an additive factor $\varepsilon$ away from $\nu(b)$. One can use these solvers to compute an $\varepsilon$-close transport plan by executing $\mathrm{poly}(n, 1/\varepsilon)$ iterations. Each iteration requires computation of several weighted Voronoi diagrams, each of which takes $n^{O(d)}$ time. Another widely used approach is to draw samples from the continuous distribution and convert the semi-discrete OT problem to a discrete instance [23]; however, due to sampling errors, this approach provides an additive approximation. Van Kreveld *et. al.* [46] presented a $(1 + \varepsilon)$-approximation OT algorithm for the restricted case when the continuous distribution is uniform over a collection of simple geometric objects (e.g. segments, simplices, etc.), by sampling roughly $n^2$ points. Their running time is roughly $n^2 \varepsilon^{-O(d)} \mathrm{poly}\log(n)$.

The discrete OT problem under any metric can be modeled and solved as an uncapacitated minimum-cost flow problem [39, 44]. There has also been extensive work on the design of near-linear time $(1 + \varepsilon)$-approximation algorithms for the optimal transport problem [9, 21, 22, 28, 45]; the running time of all these algorithms are exponentially dependent on the dimension, which make them less usable for the high-dimensional instances that arise in machine learning applications. Due to the lack of fast (relative) approximation algorithms in high dimensions, researchers have designed algorithms with additive guarantees on approximation [2, 4, 5, 12, 16, 19, 25, 32, 42].

**Our contributions.** We present a cost-scaling algorithm that computes an $\varepsilon$-close transport plan for a semi-discrete instance in $n^{O(d)} \log(\mathrm{D}/\varepsilon)$ time, assuming that we have access to an oracle that, given a constant complexity region $\varphi$, returns $\mu(\varphi)$:

**Theorem 1.1.** *Let $\mu$ be a continuous distribution defined on a compact bounded set $A \subset \mathbb{R}^d$ for some fixed $d \ge 1$, $\nu$ a discrete distribution with a support $B \subset \mathbb{R}^d$ of size $n$, and $\varepsilon > 0$ a parameter. Suppose there exists an* ORACLE *which, given a constant complexity region $\varphi$, returns $\mu(\varphi)$ in $Q$ time. Then an $\varepsilon$-close transport plan can be computed in $Qn^{O(d)} \log(\frac{\mathrm{D}}{\varepsilon})$ time, where $\mathrm{D}$ is the diameter of $A \cup B$.*

To the best of our knowledge, our algorithm is the first one to compute an $\varepsilon$-close transport plan in time that is polynomial in both $n$ and $\log(\varepsilon^{-1})$. Earlier algorithms had an $\varepsilon^{-O(1)}$ factor in the run time[2]. Our algorithm not only computes an $\varepsilon$-close transport plan, it also finds the optimal dual weights within an additive error of $\varepsilon$, i.e., it computes optimal dual-weights up to $O(\log \varepsilon^{-1})$ bits of accuracy. Our algorithm works for any ground distance where the bisector of two points under the distance function $\mathrm{d}(\cdot, \cdot)$ is an algebraic variety of constant degree. Consequently, it works for several important distances, including the $L_p$-norm and the squared-Euclidean distance. The previous best-known algorithm by Kitagawa [29] for the semi-discrete OT has an execution time $n^{\Omega(d)}\mathrm{D}/\varepsilon$; furthermore, their algorithm only approximates the cost and does not necessarily provide any guarantees for the transport plan or the dual weights of $B$.

[1] Apparently the semi-discrete OT was introduced by Cullen and Purser [15] without reference to optimal transport.

[2] Mérigot and Thibert had conjectured that an algorithm for computing an $\varepsilon$-close OT for semi-discrete setting with runtime $(n \log \varepsilon^{-1})^{O(1)}$ might follow using a scaling framework [36, Remark 24]. Our result proves their conjecture in the affirmative.

For each scale $\delta$, our algorithm starts with a set of dual weights assigned to $B$ and constructs an instance of discrete OT by using the arrangement of $4n+1$ shifts of the Voronoi cell of each point in $B$. This discrete instance, which is of size $n^{O(d)}$, is then solved using a primal-dual solver. The optimal dual weights for this discrete instance are then used to refine the dual weights of $B$. These refined dual weights act as the starting dual weights for the next scale $\delta/2$. Starting with $\delta = \mathrm{D}$, our algorithm executes $O(\log(\mathrm{D}/\varepsilon))$ scales and stops when $\delta \leq \varepsilon$ . In order to show that the semi-discrete transport plan computed in scale $\delta$ is $\delta$-close, we introduce a set of exponentially many $\delta$-feasibility constraints and show that any transport plan that satisfies these is a $\delta$-close transport plan. We then show that, in scale $\delta$, the semi-discrete OT plan and the duals computed by our polynomial time algorithm satisfies all of these exponentially many constraints and therefore, is $\delta$-close.

**Organization.** In Section 2, we first describe the overall framework, then provide details of the algorithm, and finally analyze the running time of our algorithm. Then, in Section 3, we provide a discussion on the correctness of our algorithm. Finally, we show in Section 4 that our algorithm computes a set of accurate dual weights as well. We provide the proofs of all claims and lemmas in the full version [3].

## 2 Computing a Highly Accurate Semi-Discrete Optimal Transport

Given a continuous distribution $\mu$ over a compact bounded set $A \subset \mathbb{R}^d$, a discrete distribution $\nu$ over a set $B \subset \mathbb{R}^d$ of $n$ points, and a parameter $\varepsilon > 0$, we present a cost-scaling algorithm for computing an $\varepsilon$-close transport plan from $\mu$ to $\nu$. We first describe the overall framework, then provide details of the algorithm and analyze its efficiency, and finally prove its correctness.

In our algorithm, we use a black-box primal-dual discrete OT solver PD-OT$(\mu', \nu')$ that given two discrete distributions $\mu'$ and $\nu'$ defined over two point sets $A'$ and $B'$, returns a transport plan $\sigma$ from $\mu'$ to $\nu'$ and a dual weight $y(v)$ for each point $v \in A' \cup B'$ such that for any pair $(a,b) \in A' \times B'$,

$$\begin{aligned} y(b) - y(a) &\leq \mathrm{d}(a,b), && (1)\\ y(b) - y(a) &= \mathrm{d}(a,b) \quad \text{if } \sigma(a,b) > 0. && (2) \end{aligned}$$

Standard primal-dual methods [31] construct a transport plan while maintaining (1) and (2). For concreteness, we use Orlin's algorithm [39] that runs in $O(|A' \cup B'|^3)$ time.

### 2.1 The Scaling Framework.

The algorithm works in $O(\log(\mathrm{D}\varepsilon^{-1}))$ rounds, where D is the diameter of $A \cup B$. In each round, we have a parameter $\delta > 0$ that we refer to as the *current scale*, and we also maintain a dual weight $y(b)$ for every point $b \in B$. Initially, in the beginning of the first round, $\delta = \mathrm{D}$ and $y(b) = 0$ for all $b \in B$. Execute the following steps $s = \lceil \log_2(\mathrm{D}\varepsilon^{-1}) \rceil$ times[3].

(i) *Construct a discrete OT instance:* Using the current values of dual weights of $B$, as described below, construct a discrete distribution $\hat{\mu}_\delta$ with a support set $X_\delta$, where $|X_\delta| = n^{O(d)}$, and define a (discrete) distance function $\mathrm{d}_\delta : X_\delta \times B \to \{0, \dots, 4n+1\}$.

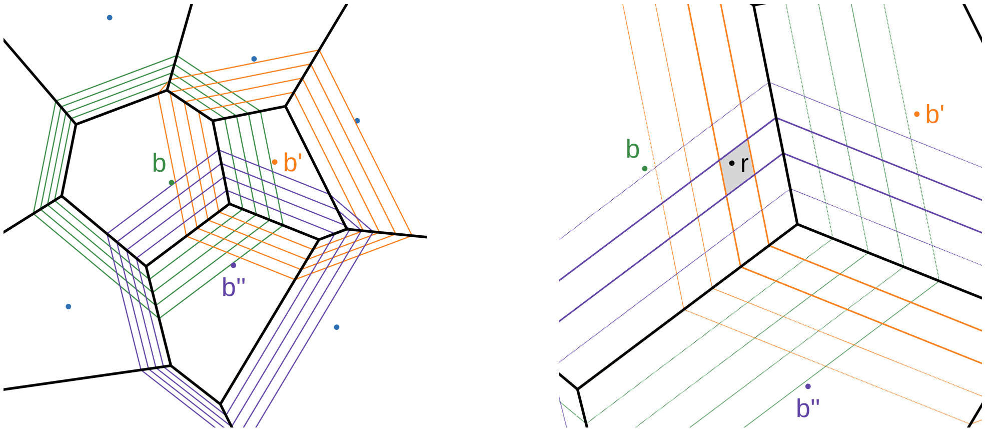
(ii) *Solve OT instance:* Compute an optimal transport plan between discrete distributions $\hat{\mu}_\delta$ and $\nu$ using the procedure PD-OT$(\hat{\mu}_\delta, \nu)$. Let $\sigma_\delta$ be the coupling and $\hat{y} : B \to \mathbb{R}$ be the dual weights returned by the procedure.

(iii) *Update dual weights:* $y(b) \leftarrow y(b) + \delta\hat{y}(b)$ for each point $b \in B$.

(iv) *Update scale:* $\delta \leftarrow \delta/2$.

Our algorithm terminates when $\delta \leq \varepsilon$. We now describe the details of step (i) of our algorithm, which is the only non-trivial step. Let $y(\cdot)$ be the dual weights of $B$ at the start of scale $\delta$.

[3] Computing an $\varepsilon$-close transport plan requires $O(\log(\mathrm{D}/\varepsilon))$ iterations. When the goal, on the other hand, is to obtain accurate dual weights up to $O(\log \varepsilon^{-1})$ bits, we need to execute our algorithm for $O(\log(n\mathrm{D}/\varepsilon))$ iterations. See Section 4.



(a) (b)

Figure 1: (a) The (expanded) Voronoi cells $V_\cdot^i$ of three points $b, b', b'' \in B$, (b) A region $\varphi \in \mathcal{A}(\mathcal{V})$ (highlighted in gray) with a representative point $r \in X_\delta$, where $\mathrm{d}_\delta(b, r) = 0$ since $r \in V_b^1$, $\mathrm{d}_\delta(r, b') = 1$ since $r \in V_{b'}^2 \setminus V_{b'}^1$, and $\mathrm{d}_\delta(r, b'') = 2$ since $r \in V_{b''}^3 \setminus V_{b''}^2$. The ground distance $\mathrm{d}(\cdot, \cdot)$ in this figure is squared Euclidean.

**Constructing a discrete OT instance.** We construct the discrete instance by constructing a family of Voronoi diagrams and overlaying some of their cells. For a weighted point set $P \subset \mathbb{R}^d$ with weights $w : P \to \mathbb{R}$ and a distance function $\mathrm{d} : \mathbb{R}^d \times P \to \mathbb{R}_{\geq 0}$, we define the *weighted distance* from a point $p \in P$ to any point $x \in \mathbb{R}^d$ as $\mathrm{d}_w(x, p) = \mathrm{d}(x, p) - w(p)$. For a point $p \in P$, its *Voronoi cell* (also known as *Laguerre cell*) is $\mathrm{Vor}_w(p) = \{x \in \mathbb{R}^d \mid \mathrm{d}_w(x, p) \leq \mathrm{d}_w(x, p'), \forall p' \in P\}$, and the *Voronoi diagram* (also known as *Laguerre diagram*) $\mathrm{VD}_w(P)$ is the decomposition of $\mathbb{R}^d$ induced by Voronoi cells; see [20].

For $i \in [1, 4n + 1]$ and a point $b \in B$, we define a Voronoi cell $V_b^i$ using a weight function $w_i : B \to \mathbb{R}_{\geq 0}$, as follows. We set $w_i(b) = y(b) + i\delta$ and $w_i(b') = y(b')$ for all $b' \neq b$. We set $V_b^i = \mathrm{Vor}_{w_i}(b)$ in $\mathrm{VD}_{w_i}(B)$. By construction, $V_b^1 \subseteq V_b^2 \subseteq \cdots \subseteq V_b^{4n+1}$. Set $\mathcal{V}_b = \{V_b^i \mid i \in [1, 4n + 1]\}$ and $\mathcal{V} = \bigcup_{b \in B} \mathcal{V}_b$ (See Figure 1(a)). Let $\mathcal{A}(\mathcal{V})$ be the *arrangement* of $\mathcal{V}$, the decomposition of $\mathbb{R}^d$ into (connected) cells induced by $\mathcal{V}$; each cell of $\mathcal{A}(\mathcal{V})$ is the maximum connected region lying in the same subset of regions of $\mathcal{V}$ [1].

For each cell $\varphi$ in $\mathcal{A}(\mathcal{V})$, we choose a representative point $r_\varphi$ arbitrarily and set its mass to $\hat{\mu}_\delta(r_\varphi) = \mu(\varphi)$, where for any region $\rho$ in $\mathbb{R}^d$, $\mu(\rho) = \int_\rho \mu(a)\, da$ is the mass of $\mu$ inside $\rho$ (Here we assume the mass to be 0 outside the support $A$ of $\mu$). Set $X_\delta = \{r_\varphi \mid \varphi \in \mathcal{A}(\mathcal{V})\}$. The resulting mass distribution on $X_\delta$ is $\hat{\mu}_\delta$.

The (discrete) distance $\mathrm{d}_\delta(r, b)$ between any point $b \in B$ and a point $r \in X_\delta$ is defined as

$$\mathrm{d}_\delta(r, b) = \begin{cases} 0, & \text{if } r \in V_b^1, \\ i, & \text{if } r \in V_b^{i+1} \setminus V_b^i, \;\; i \in [1, 4n], \\ 4n + 1, & \text{if } r \notin V_b^{4n+1}. \end{cases}$$

See Figure 1(b). Since each $V_b^i$ is defined by $n$ algebraic surfaces of constant degree, assuming the bisector of two points under the distance function $\mathrm{d}(\cdot, \cdot)$ is an algebraic variety of constant degree, $\mathcal{A}(\mathcal{V})$ has $n^{O(d)}$ cells and a point in every cell of $\mathcal{A}(\mathcal{V})$ can be computed in $n^{O(d)}$ time [13]. Hence, $|X_\delta| = n^{O(d)}$. This completes the construction of $X_\delta$, $\hat{\mu}_\delta$, and $\mathrm{d}_\delta$.

**Computing a semi-discrete transport plan.** At the end of any scale $\delta$, we compute a $\delta$-close transport plan $\tau_\delta$ from the discrete transport plan $\sigma_\delta$ as follows: For any edge $(r_\varphi, b) \in X_\delta \times B$, we arbitrarily transport $\sigma_\delta(r_\varphi, b)$ mass from the points inside the region $\varphi$ to the point $b$. A simple construction of such transport plan is to set, for any region $\varphi$, any point $a \in \varphi$, and any point $b \in B$, $\tau_\delta(a, b) = \frac{\mu(a)}{\hat{\mu}_\delta(r_\varphi)} \sigma_\delta(r_\varphi, b)$. Our algorithm will only compute the transport plan at the end of the last scale, i.e., $\delta \leq \varepsilon$.

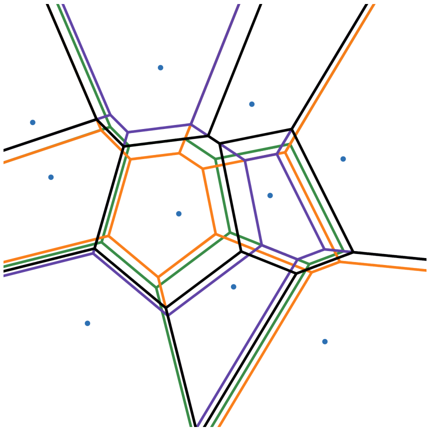


Figure 2: The weighted Voronoi diagrams for four different weight vectors in $\mathbb{W}_\delta$. The ground distance in this figure is squared Euclidean.

**Efficiency analysis.** Our algorithm runs $O(\log(\mathrm{D}\varepsilon^{-1}))$ scales, where in each scale, it constructs a discrete OT instance in $n^{O(d)}$ time and solves the OT instance using a polynomial-time primal-dual OT solver. Since the size of the discrete OT instance is $n^{O(d)}$, solving it also takes $n^{O(d)}$ time, resulting in a total execution time of $n^{O(d)}\log(\mathrm{D}\varepsilon^{-1})$ for our algorithm.

## 3 Proof of Correctness

In the discrete setting, cost scaling algorithms obtain an $\varepsilon$-close transport plan that satisfies (2) and an additive $\varepsilon$ relaxation of (1). For our proof, we extend these relaxed feasibility conditions to a semi-discrete setting and show that the transport plan computed by our algorithm for a scale $\delta$ satisfies these conditions. We use the relaxed feasibility conditions to show that our transport plan is $\delta$-close. Thus, our algorithm returns an $\varepsilon$-close transport plan from $\mu$ to $\nu$ at the end of the last scale ($\delta \leq \varepsilon$).

**$\delta$-feasible transport plan.** For points $B = \{b_1, b_2, \dots, b_n\}$, let $w = \langle w_1, \dots, w_n \rangle$ be an $n$-dimensional vector representing a weight assignment to the points in $B$. We say that the vector $w$ is *valid* if each $w_i$ is a non-negative integer multiple of $\delta$ and bounded by $(8n+2)\mathrm{D}$. Consider the set $\mathbb{W}_\delta$ of all valid vectors, i.e., $\mathbb{W}_\delta = (\delta\mathbb{Z} \cap [0, (8n+2)\mathrm{D}])^n$. For any $\delta$, consider a decomposition of the support $A$ of the continuous distribution $\mu$ into a set of regions $\mathscr{A}_\delta$, where each region $\varrho$ in $\mathscr{A}_\delta$ satisfies the following condition:

(P1) Any two points $x$ and $y$ in $\varrho$ have the same weighted nearest neighbor in $B$ with respect to any valid weight vector $w \in \mathbb{W}_\delta$,

where a point $b$ is a *weighted nearest neighbor* of a point $a \in A$ with respect to weights $w(\cdot)$ if $\mathrm{d}_w(a,b) = \min_{b' \in B} \mathrm{d}_w(a,b')$. For a valid vector $w \in \mathbb{W}_\delta$, let $\mathrm{VD}_w(B)$ denote the weighted Voronoi diagram constructed for the points in $B$ with weights $w$. The partitioning $\mathscr{A}_\delta$ is simply the overlay of all weighted Voronoi diagrams $\mathrm{VD}_w(B)$ across all valid weight vectors $w \in \mathbb{W}_\delta$ (See Figure 2). For each region $\varrho \in \mathscr{A}_\delta$, let $r_\varrho$ denote an arbitrary representative point inside $\varrho$.

For any point $b \in B$, let $y(b)$ be the dual weight of $b$, where $y(b)$ is a non-negative integer multiple of $\delta$. For each region $\varrho \in \mathscr{A}_\delta$, we derive a dual weight $y_\delta(r_\varrho)$ for its representative point as follows. Let $b_\varrho \in B$ be the weighted nearest neighbor of $r_\varrho$ with respect to weights $y(\cdot)$. We set the dual weight of $r_\varrho$ as

$$y_\delta(r_\varrho) \leftarrow y(b_\varrho) - \mathrm{d}(r_\varrho, b_\varrho) - \delta. \tag{3}$$

We say that a transport plan $\tau$ from $\mu$ to $\nu$ along with the set of dual weights $y(\cdot)$ for points in $B$ is *$\delta$-feasible* if, for each point $b \in B$ and each region $\varrho \in \mathscr{A}_\delta$,

$$\begin{aligned} y(b) - y_\delta(r_\varrho) &\leq \mathrm{d}(r_\varrho, b) + \delta, && (4)\\ y(b) - y_\delta(r_\varrho) &\geq \mathrm{d}(r_\varrho, b) \qquad \text{if } \tau(\varrho, b) > 0. && (5) \end{aligned}$$

In the following lemma, we show that any $\delta$-feasible transport plan $\tau, y(\cdot)$ from $\mu$ to $\nu$ is $\delta$-close.

**Lemma 3.1.** *Suppose $\tau, y(\cdot)$ is any $\delta$-feasible transport plan from $\mu$ to $\nu$ and let $\tau^*$ denote any optimal transport plan from $\mu$ to $\nu$. Then, $\cent(\tau) \leq \cent(\tau^*) + \delta$.*

Let $y(\cdot)$ denote the set of dual weights maintained by our algorithm at the beginning of scale $\delta$. For any point $b \in B$ and any region $\varrho \in \mathscr{A}_\delta$, we define a *slack* on condition (4) for the pair $(\varrho, b)$, denoted by $s_\delta(\varrho, b)$, as

$$s_\delta(\varrho, b) := \left\lfloor \frac{\mathrm{d}(r_\varrho, b) + \delta - y(b) + y_\delta(r_\varrho)}{\delta} \right\rfloor \delta.$$

Next, we show that for each scale $\delta$, the semi-discrete transport plan $\tau_\delta$ and dual weights $(y + \delta\hat{y})(\cdot)$ for the points in $B$ computed by our algorithm at the end of the scale is a $\delta$-feasible transport plan.

**$\delta$-feasibility of the computed transport plan.** We begin by relating the decomposition $\mathscr{A}_\delta$ to the partitioning $\mathcal{A}(\mathcal{V})$ that is constructed in step (i) of our algorithm. We also relate the distance $\mathrm{d}_\delta$ computed by our algorithm to the slacks $s_\delta$.

In any scale $\delta$, it can be shown that for each point $b \in B$ and each $i \in [1, 4n + 1]$, the $i$-expansion $V_b^i$ can be seen as the Voronoi cell of $b$ in the weighted Voronoi diagram constructed with respect to some valid weight vector in $\mathbb{W}_\delta$. Hence, by the construction of $\mathscr{A}_\delta$, each region $\varrho \in \mathscr{A}_\delta$ completely lies inside some region $\varphi \in \mathcal{A}(\mathcal{V})$, i.e., each region in $\mathcal{A}(\mathcal{V})$ consists of a collection of regions in $\mathscr{A}_\delta$.

We observe that for each point $b \in B$, all regions with a zero slack to $b$ lie inside the 1-expansion $V_b^i$, all regions with a slack $i\delta$ to $b$, for $i \in [1, 4n]$, lie inside the region sandwiched between $i$ and $i + 1$ expansions of the weighted Voronoi cell of $b$, and all regions $\varrho$ with a slack $> 4n\delta$ to $b$ are outside $V_b^{4n+1}$. Using this observation, in the next lemma, we establish a connection between the slacks and the distances $\mathrm{d}_\delta$.

**Lemma 3.2.** *For any region $\varphi \in \mathcal{A}(\mathcal{V})$, any region $\varrho \in \mathscr{A}_\delta$ inside $\varphi$, and any point $b \in B$, if $\mathrm{d}_\delta(r_\varphi, b) \leq 4n$, then $s_\delta(\varrho, b) = \mathrm{d}_\delta(r_\varphi, b)\delta$. Furthermore, if $\mathrm{d}_\delta(r_\varphi, b) = 4n + 1$, then $s_\delta(\varrho, b) \geq (4n + 1)\delta$.*

Recall that $X_\delta$ denotes the set of representative points of the regions in $\mathcal{A}(\mathcal{V})$ and $\hat{\mu}_\delta$ is the discrete distribution over $X_\delta$ computed by our algorithm at step (i). In the following lemma, we show that any optimal transport plan $\sigma^*$ from $\hat{\mu}_\delta$ to $\nu$ under distance function $\mathrm{d}_\delta$ does not transport mass on edges $(r_\varphi, b) \in X_\delta \times B$ with cost $\mathrm{d}_\delta(r_\varphi, b) > 4n$.

**Lemma 3.3.** *For any scale $\delta$, let $\sigma^*$ be any optimal transport plan from $\hat{\mu}_\delta$ to $\nu$. For any point $b \in B$ and any region $\varphi \in \mathcal{A}(\mathcal{V})$, if $\sigma^*$ transports mass from $r_\varphi$ to $b$, then $\mathrm{d}_\delta(r_\varphi, b) \leq 4n$.*

*Proof.* Let $\tau_{2\delta}, y(\cdot)$ be the $2\delta$-feasible transport plan computed by our algorithm at scale $2\delta$. Let $\sigma_{2\delta}$ denote a transformation of $\tau_{2\delta}$ into a discrete transport plan from $\hat{\mu}_\delta$ to $\nu$ by simply setting, for each region $\varphi \in \mathcal{A}(\mathcal{V})$, $\sigma_{2\delta}(r_\varphi, b) := \tau_{2\delta}(\varphi, b)$. Let $\sigma^*$ be any optimal transport plan from $\hat{\mu}_\delta$ to $\nu$, where the cost of each edge $(r_\varphi, b)$ is set to $\mathrm{d}_\delta(r_\varphi, b)$. Define a directed graph $\mathcal{G}$ on the vertex set $X_\delta \cup B$ as follows. For any pair $(r, b) \in X_\delta \times B$, if $\sigma^*(r, b) > \sigma_{2\delta}(r, b)$, then we add an edge, called a *forward edge*, directed from $r$ to $b$ with a capacity $\sigma^*(r, b) - \sigma_{2\delta}(r, b)$; otherwise, if $\sigma^*(r, b) < \sigma_{2\delta}(r, b)$, then we add an edge, called a *backward edge*, directed from $b$ to $r$ with a capacity $\sigma_{2\delta}(r, b) - \sigma^*(r, b)$. This completes the construction of the directed graph.

For the sake of contradiction, suppose there is a pair $(r^*, b^*) \in X_\delta \times B$ with $\mathrm{d}_\delta(r^*, b^*) > 4n$ such that $\sigma^*(r^*, b^*) > 0$. As shown in the full version of our paper, the edge $(r^*, b^*)$ is a forward edge and is contained in a simple directed cycle $C = \langle b_1, r_1, \ldots, b_k, r_k, b_{k+1} = b_1 \rangle$ in $\mathcal{G}$. Note that by the construction of $\mathcal{G}$, the edges of $C$ alternate between forward and backward edges. Define the cost of the cycle $C$ as

$$w(C) := \sum_{i=1}^{k} \mathrm{d}_\delta(r_i, b_{i+1}) - \sum_{i=1}^{k} \mathrm{d}_\delta(r_i, b_i);$$

i.e., the cost of $C$ is simply the total distance of its forward edges minus the total distance of its backward edges. Since $\sigma^*$ is an optimal transport plan from $\hat{\mu}_\delta$ to $\nu$, any directed cycle $C$ on $\mathcal{G}$ has a non-positive cost. Since $C$ is a simple cycle, the length of $C$ is at most $2n$. Furthermore, as shown in the full version of our paper, any backward edge has a distance at most 4, i.e., for each

$i \in [1, k]$, $\mathrm{d}_\delta(r_i, b_i) \leq 4$. Finally, by construction, all edges have a non-negative distance. Therefore,

$$w(C) = \sum_{i=1}^{k} \mathrm{d}_\delta(r_i, b_{i+1}) - \sum_{i=1}^{k} \mathrm{d}_\delta(r_i, b_i) \geq \mathrm{d}_\delta(r^*, b^*) - \sum_{i=1}^{k} 4 \geq \mathrm{d}_\delta(r^*, b^*) - 4n > 0,$$

which is a contradiction of the fact that all simple cycles have a non-positive cost. Hence, $\sigma^*$ cannot transport mass on edges $(r^*, b^*)$ with distance $\mathrm{d}_\delta(r^*, b^*) > 4n$. □

Let $\sigma_\delta, \hat{y}(\cdot)$ be the optimal transport plan from $\hat{\mu}_\delta$ to $\nu$ computed at step (ii) of our algorithm, and recall that $\tau_\delta$ is the transport plan from $\mu$ to $\nu$ computed at the end of scale $\delta$. In the following lemma, we show that $\tau_\delta, (y + \delta\hat{y})(\cdot)$ is a $\delta$-feasible transport plan.

**Lemma 3.4.** *For each scale $\delta$, let $(y + \delta\hat{y})(\cdot)$ denote the set of dual weights for points in $B$ computed at step (iii) of our algorithm. Then, the transport plan $\tau_\delta, (y + \delta\hat{y})(\cdot)$ is a $\delta$-feasible transport plan.*

*Proof.* Let $y_\delta(\cdot)$ denote the set of dual weights derived for the representative points of regions in $\mathscr{A}_\delta$ using Equation (3) at the beginning of scale $\delta$. Consider a set of dual weights $y'_\delta$ that assigns, for each region $\varrho \in \mathscr{A}_\delta$ inside a region $\varphi \in \mathcal{A}(\mathcal{V})$, a dual weight $y'_\delta(r_\varrho) := y_\delta(r_\varrho) + \delta\hat{y}(r_\varphi)$. In what follows, we show that the transport plan $\tau_\delta$ along with dual weights $(y + \delta\hat{y})(\cdot)$ and $y'_\delta(\cdot)$ satisfy $\delta$-feasibility conditions (4) and (5). Using this, in the full version of our paper, we show that by reassigning the dual weights of the representative points of regions in $\mathscr{A}_\delta$ as in Equation (3), the $\delta$-feasibility conditions (4) and (5) remain satisfied; hence, we conclude that $\tau, (y + \delta\hat{y})(\cdot)$ is $\delta$-feasible, as claimed.

For any region $\varphi \in \mathcal{A}(\mathcal{V})$, any region $\varrho \in \mathscr{A}_\delta$ inside $\varphi$, and any point $b \in B$,

- by Lemma 3.2, $\mathrm{d}_\delta(r_\varphi, b)\delta \leq s_\delta(\varrho, b)$. Combining with feasibility condition (1),

  $$\begin{aligned}(y + \delta\hat{y})(b) - y'_\delta(r_\varrho) &= (y(b) + \delta\hat{y}(b)) - (y_\delta(r_\varrho) + \delta\hat{y}(r_\varphi)) \\ &= (y(b) - y_\delta(r_\varrho)) + \delta(\hat{y}(b) - \hat{y}(r_\varphi)) \\ &\leq (y(b) - y_\delta(r_\varrho)) + \mathrm{d}_\delta(r_\varphi, b)\delta \leq (y(b) - y_\delta(r_\varrho)) + s_\delta(\varrho, b) \\ &\leq (y(b) - y_\delta(r_\varrho)) + (\mathrm{d}(r_\varrho, b) + \delta - y(b) + y_\delta(r_\varrho)) \\ &= \mathrm{d}(r_\varrho, b) + \delta,\end{aligned}$$

  leading to $\delta$-feasibility condition 4.

- if $\tau_\delta(\varrho, b) > 0$, then $\sigma_\delta$ transports mass from $r_\varphi$ to $b$, i.e., $\sigma_\delta(r_\varphi, b) > 0$. In this case, by Lemma 3.3, $\mathrm{d}_\delta(r_\varphi, b) \leq 4n$ and by Lemma 3.2, $s_\delta(\varrho, b) = \mathrm{d}_\delta(r_\varphi, b)\delta$. Combining with feasibility condition (2),

  $$\begin{aligned}(y + \delta\hat{y})(b) - y'_\delta(r_\varrho) &= (y(b) + \delta\hat{y}(b)) - (y_\delta(r_\varrho) + \delta\hat{y}(r_\varphi)) \\ &= (y(b) - y_\delta(r_\varrho)) + \delta(\hat{y}(b) - \hat{y}(r_\varphi)) \\ &= (y(b) - y_\delta(r_\varrho)) + \mathrm{d}_\delta(r_\varphi, b)\delta = (y(b) - y_\delta(r_\varrho)) + s_\delta(\varrho, b) \\ &\geq (y(b) - y_\delta(r_\varrho)) + (\mathrm{d}(r_\varrho, b) - y(b) + y_\delta(r_\varrho)) \\ &= \mathrm{d}(r_\varrho, b),\end{aligned}$$

  leading to $\delta$-feasibility condition 5.

□

## 4 Optimal Dual Weights

In this section, we show that in addition to computing an $\varepsilon$-close transport cost in the semi-discrete setting, our algorithm can also compute the set of dual weights for the points in $B$ accurately, up to $O(\log \varepsilon^{-1})$ bits. To obtain such accurate set of dual weights, we execute our algorithm for $O(\log(n\mathrm{D}/\varepsilon))$ iterations so that the final value of $\delta$ when the algorithm terminates is at most $\varepsilon/5n$. In the following, we show that the dual weight computed for each point in $B$ at the last scale is $\varepsilon$-close to the optimal dual weight value.

Note that any edge in the graph constructed in Step (i) of our algorithm has a cost at most $4n+1$. Consequently, in Step (ii), the largest dual weight returned by the primal-dual solver is at most $4n+1$[4] and in Step (iii), the dual weight of any point $b \in B$ changes by at most $(4n+1)\delta$. Since the dual weight of $b$ becomes the optimal dual weight in the limit, to bound the difference between the current dual weight and the optimal, it suffices if we bound the total change in the dual weights for all scales after scale $\delta \leq \varepsilon/5n$. The difference between the optimal dual weight and the current dual weight is at most

$$(4n+1)\sum_{i=1}^{\infty} \delta/2^i = (4n+1)\delta \leq (4n+1)(\varepsilon/5n) \leq \varepsilon.$$

Therefore, after $O(\log(n\mathrm{D}/\varepsilon))$ iterations of the algorithm, the difference in the optimal dual weight $y(b)$ and the current dual weight of $b$ is at most $\varepsilon$.

## References

[1] Pankaj K Agarwal and Micha Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surveys*, 30(4):412–458, 1998.

[2] Pankaj K Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Rachita Sowle. A higher precision algorithm for computing the 1-Wasserstein distance. In *The Eleventh Internat. Conf. on Learning Representations*, 2022.

[3] Pankaj K. Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Keegan Yao. Fast and accurate approximations of the optimal transport in semi-discrete and discrete settings. *arXiv preprint arXiv:2311.02172*, 2023.

[4] 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.

[5] Jason Altschuler, Francis Bach, Alessandro Rudi, and Jonathan Niles-Weed. Massively scalable Sinkhorn distances via the nyström method. *Advances in Neural Information Processing Systems*, 32, 2019.

[6] Luca Ambrogioni, Umut Guclu, and Marcel van Gerven. Wasserstein variational gradient descent: From semi-discrete optimal transport to ensemble variational inference. *arXiv preprint arXiv:1811.02827*, 2018.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Internat. Conf. on machine learning*, pages 214–223. PMLR, 2017.

[8] Franz Aurenhammer, Friedrich Hoffmann, and Boris Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20(1):61–76, 1998.

[9] Arturs Backurs, Yihe Dong, Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. Scalable nearest neighbor search for optimal transport. *In Internat. Conf. on Machine Learning*, pages 497–506, 2020.

[10] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[11] Espen Bernton, Pierre E. Jacob, Mathieu Gerber, and Christian P. Robert. On parameter estimation with the Wasserstein distance. *Information and Inference: A J. of IMA*, 8(4):657–676, 2019.

[12] Jose Blanchet, Arun Jambulapati, Carson Kent, and Aaron Sidford. Towards optimal running times for optimal transport. *arXiv preprint arXiv:1810.07717*, 2018.

[4] Any set of dual weights returned by the algorithm can be translated by a fixed value so that the smallest dual weight becomes 0. Assuming this, it is easy to see that the largest dual weight is $4n+1$.

[13] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real Algebraic Geometry*, volume 36. Springer, 2013.

[14] Rick Chartrand, Brendt Wohlberg, Kevin Vixie, and Erik Bollt. A gradient descent solution to the Monge-Kantorovich problem. *Applied Math. Sci.*, 3(22):1071–1080, 2009.

[15] Michael JP Cullen and R James Purser. An extended lagrangian theory of semi-geostrophic frontogenesis. *J. of Atmospheric Sci.*, 41(9):1477–1497, 1984.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. Blue noise through optimal transport. *ACM Trans. on Graphics*, 31(6):1–11, 2012.

[18] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced Wasserstein distance. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, pages 3483–3491, 2018.

[19] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *Internat. Conf. on Machine Learning*, pages 1367–1376. PMLR, 2018.

[20] Steven Fortune. Voronoi diagrams and delaunay triangulations. *Comput. in Euclidean Geom.*, pages 225–265, 1995.

[21] Kyle Fox and Jiashuai Lu. A near-linear time approximation scheme for geometric transportation with arbitrary supplies and spread. In *Proc. 36th Annual Sympos. Comput. Geom.*, pages 45:1–45:18, 2020.

[22] Kyle Fox and Jiashuai Lu. A deterministic near-linear time approximation scheme for geometric transportation. *arXiv preprint arXiv:2211.03891*, 2022.

[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.

[24] Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with Sinkhorn divergences. *In Internat. Conf. on Artificial Intelligence and Stat.*, page 1608–1617, 2018.

[25] Wenshuo Guo, Nhat Ho, and Michael Jordan. Fast algorithms for computational optimal transport and Wasserstein barycenter. *In Internat. Conf. on Artificial Intelligence and Stat.*, pages 2088–2097, 2020.

[26] Rishi Gupta, Piotr Indyk, and Eric Price. Sparse recovery for earth mover distance. In *Proc. 48th Annual Allerton Conf. on Communication, Control, and Comput.*, pages 1742–1744. IEEE, 2010.

[27] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Wasserstein regularization for sparse multi-task regression. In *The 22nd Internat. Conf. on Artificial Intelligence and Stat.*, pages 1407–1416. PMLR, 2019.

[28] Andrey Boris Khesin, Aleksandar Nikolov, and Dmitry Paramonov. Preconditioning for the geometric transportation problem. *arXiv preprint arXiv:1902.08384*, 2019.

[29] Jun Kitagawa. An iterative scheme for solving the optimal transportation problem. *Calculus of Variations and Partial Differential Equations*, 51(1):243–263, 2014.

[30] Jun Kitagawa, Quentin Mérigot, and Boris Thibert. Convergence of a Newton algorithm for semi-discrete optimal transport. *J. of European Math. Society*, 21(9):2603–2651, 2019.

[31] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quart.*, 2(1-2):83–97, 1955.

[32] Nathaniel Lahn and Sharath Raghvendra. A faster algorithm for minimum-cost bipartite matching in minor-free graphs. In *Proc. Thirtieth Annual ACM-SIAM Sympos. Discrete Algorithms*, pages 569–588. SIAM, 2019.

[33] Bruno Lévy and Erica L Schwindt. Notions of optimal transport theory and how to implement them on a computer. *Comput. & Graphics*, 72:135–148, 2018.

[34] Huidong Liu, G. U. Xianfeng, and Dimitris Samaras. A two-step computation of the exact GAN Wasserstein distance. *In Internat. Conf. on Machine Learning*, pages 3159–3168, 2018.

[35] Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of Sinkhorn approximation for learning with Wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

[36] Quentin Merigot and Boris Thibert. Optimal transport: discretization and algorithms. In *Handbook of Numerical Analysis*, volume 22, pages 133–212. Elsevier, 2021.

[37] Jean-Marie Mirebeau. Discretization of the 3d monge- ampere operator, between wide stencils and power diagrams. *ESAIM: Math. Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 49(5):1511–1523, 2015.

[38] Vladimir I Oliker and Laird D Prussner. On the numerical solution of the equation $\frac{\partial^2 z}{\partial x^2}\frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial x \partial y}\right) = f$ and its discretizations, i. *Numerische Mathematik*, 54(3):271–293, 1989.

[39] James Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Proc. Twentieth Annual ACM Sympos. Theory of Comput.*, pages 377–387, 1988.

[40] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Found. and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[41] Hongxing Qin, Yi Chen, Jinlong He, and Baoquan Chen. Wasserstein blue noise sampling. *ACM Trans. on Graphics (TOG)*, 36(5):1–13, 2017.

[42] Kent Quanrud. Approximating optimal transport with linear programs. *arXiv preprint arXiv:1810.05957*, 2018.

[43] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. *In Internat. Conf. on Learning Representations*, 2018.

[44] Ravi Seshadri and Karthik K Srinivasan. Algorithm for determining path of maximum reliability on a network subject to random arc connectivity failures. *Transport. Research Rec.*, 2467(1): 80–90, 2014.

[45] Jonah Sherman. Generalized preconditioning and undirected minimum-cost flow. In *Proc. Twenty-Eighth Annual ACM-SIAM Sympos. Discrete Algo.*, pages 772–780, 2017.

[46] Marc van Kreveld, Frank Staals, Amir Vaxman, and Jordi Vermeulen. Approximating the earth mover's distance between sets of geometric objects. *arXiv preprint arXiv:2104.08136*, 2021.

[47] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.