APPENDIX OF "A HIGHER PRECISION ALGORITHM FOR COMPUTING THE 1-WASSERSTEIN DISTANCE"

Pankaj K. Agarwal[1*], **Sharath Raghvendra**[2], **Pouyan Shirzadian**[2], **and Rachita Sowle**[2]

[1]Duke University, [2]Virginia Tech

A MISSING DETAILS OF SECTION 2.1

In this section, we analyze the additive error of the 1-Wasserstein distance computed in Section 2.1.

Recall that the algorithm of Section 2.1 computes a transport plan in two steps. In the first step, it computes a transport plan σ_1 by arbitrarily transporting supplies to demands within each non-empty cell of the grid. Suppose U_1 supplies are transported in this step. In the second step, for each surplus (resp. deficit) cell $\square$, the algorithm moves the excess supplies (resp. demands) of the points inside $\square$ to the center point $c_\square$ and creates the input instance $\mathcal{A} \cup \mathcal{B}$. Then, it computes an $\varepsilon/2$-close transport plan σ_2 on $\mathcal{A} \cup \mathcal{B}$, which transports $U - U_1$ supplies. The algorithm reports $w(\sigma_1) + w(\sigma_2)$ as an approximate 1-Wasserstein distance.

We define some notations that are used in the analysis. For any point set $A \cup B$ and any transport plan σ on $A \cup B$, a demand point $a \in A$ (resp. supply point $b \in B$) is called a *free point* with respect to σ if the total supplies transported into a (resp. from b) by σ is less than $-\eta(a)$ (resp. $\eta(b)$). The transport plan σ is a *partial transport plan* if there exists free points in $A \cup B$ with respect to σ; otherwise, σ is a *complete transport plan*. For any partial transport plan σ, let A_F^σ (resp. B_F^σ) denote the set of free points of A (resp. B) with respect to σ. For any free demand point $a \in A_F^\sigma$ (resp. free supply point $b \in B_F^\sigma$), let $\eta_\sigma(a)$ (resp. $\eta_\sigma(b)$) denote the *excess weight* of a (resp. b); i.e, the amount of demands of a (resp. supplies of b) that is not transported by σ. More precisely, define $\eta_\sigma(a) := -|\eta(a)| + \sum_{b' \in B} \sigma(a, b')$ (resp. $\eta_\sigma(b) := \eta(b) - \sum_{a' \in A} \sigma(a', b)$).

Suppose σ^* denote an optimal transport plan on $A \cup B$. In this section, we show that the reported cost is ε-close to the optimal transport cost; i.e, we show that $w(\sigma_1) + w(\sigma_2) \leq w(\sigma^*) + \varepsilon U$.

Bounding the reported cost: Since the diameter of each cell of the grid is $\varepsilon/2$, each edge carrying a positive amount of supplies in σ_1 has a length of at most $\varepsilon/2$; therefore, $w(\sigma_1) \leq \varepsilon U_1/2$. Thus, it remains to bound the cost of σ_2 by $w(\sigma^*) + \varepsilon U - \varepsilon U_1/2$. We bound $w(\sigma_2)$ as follows.

- First, in Lemma A.1, we show that there exists a transport plan σ_F on the set of free points $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ such that σ_F transports all the excess weights $\eta_{\sigma_1}(\cdot)$ on the points in $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ and has a cost of at most $w(\sigma^*) + w(\sigma_1)$.
- Second, we transform σ_F into a transport plan σ_2^* on the instance $\mathcal{A} \cup \mathcal{B}$ and show, in Lemma A.2, that the cost of σ_2^* is at most $w(\sigma_F) + \frac{\varepsilon}{2}(U - U_1)$.
- Finally, using the fact that σ_2 is an $\frac{\varepsilon}{2}$-close transport plan on $\mathcal{A} \cup \mathcal{B}$, we bound the cost of σ_2 by $w(\sigma_2^*) + \frac{\varepsilon}{2}(U - U_1)$.

Combining all the above mentioned bounds, we get

$$w(\sigma_2) \leq w(\sigma_2^*) + \frac{\varepsilon}{2}(U - U_1) \leq w(\sigma_F) + \varepsilon(U - U_1) \leq w(\sigma^*) + w(\sigma_1) + \varepsilon(U - U_1). \quad (1)$$

As mentioned earlier, $w(\sigma_1) \leq \frac{\varepsilon}{2} U_1$. Therefore,

$$w(\sigma_1) + w(\sigma_2) \leq w(\sigma^*) + 2w(\sigma_1) + \varepsilon(U - U_1) \leq w(\sigma^*) + \varepsilon U,$$

as desired. It remains to show the details of the steps of our analysis. In the following, first, we define some notations that are used in describing the steps of bounding $w(\sigma_2)$. Then, we describe the details of each step of our analysis.

[*]Following convention from Theoretical Computer Science, all authors are ordered alphabetically.

Notations and definitions: Define $\sigma' := \sigma^* - \sigma_1$ to be a function that assigns $\sigma'(a,b) = \sigma^*(a,b) - \sigma_1(a,b)$ to any pair of points $(a,b) \in A \times B$. The function σ' has the following properties:

(1) $\sum_{b' \in B} \sigma'(a,b') = |\eta_{\sigma_1}(a)|$ for any $a \in A_F^{\sigma_1}$ and $\sum_{b' \in B} \sigma'(a,b') = 0$ for any $a \in A \setminus A_F^{\sigma_1}$,

(2) $\sum_{a' \in A} \sigma'(a',b) = \eta_{\sigma_1}(b)$ for any $b \in B_F^{\sigma_1}$ and $\sum_{a' \in A} \sigma'(a',b) = 0$ for any $b \in B \setminus B_F^{\sigma_1}$.

Consider a directed graph $\mathcal{G}$ on the point set $A \cup B$ formed as follows. For any pair of points $(a,b) \in A \times B$, if $\sigma'(a,b) > 0$, then $\mathcal{G}$ contains an edge from b to a with a capacity $\sigma'(a,b)$, which we refer to by a *forward edge*. On the other hand, if $\sigma'(a,b) < 0$, then $\mathcal{G}$ contains an edge from a to b with a capacity $|\sigma'(a,b)|$, which we refer to by a *backward edge*. For any demand point $a \in A$, we define the *capacity* of a, denoted by $c(a)$, to be $c(a) = \sum_{b' \in B} \sigma'(a,b')$. Similarly, for any supply point $b \in B$, we define the capacity $c(b)$ of b to be $c(b) = \sum_{a' \in A} \sigma'(a',b)$.

An *augmenting path* P with respect to σ_1 is any simple directed path in $\mathcal{G}$ from a free supply point $b_P \in B_F^{\sigma_1}$ to a free demand point $a_P \in A_F^{\sigma_1}$. Note that the edges of any augmenting path alternates between forward edges and backward edges. We define the capacity of P, denoted by $\beta(P)$, as

$$\beta(P) = \min \left\{ c(a_P), c(b_P), \min_{(u,v) \in P} \{|\sigma'(u,v)|\} \right\}. \tag{2}$$

One can *augment* the transport plan σ_1 along the augmenting path P by setting $\sigma_1(a,b) \leftarrow \sigma_1(a,b) + \beta(P)$ for any forward edge $(b,a) \in P$ and $\sigma_1(a,b) \leftarrow \sigma_1(a,b) - \beta(P)$ for any backward edge $(a,b) \in P$. As a result of the augmentation, the excess weights of the endpoints a_P and b_P reduce by $\beta(P)$. Equivalently, one can update the function σ' using P by setting $\sigma'(a,b) \leftarrow \sigma'(a,b) - \beta(P)$ for any forward edge $(b,a) \in P$ and $\sigma'(a,b) \leftarrow \sigma'(a,b) + \beta(P)$ for any backward edge $(a,b) \in P$.

Similar to augmenting paths, one can define an *alternating cycle* as a directed cycle C on the graph $\mathcal{G}$. For any cycle C, we define the capacity of the cycle as $\beta(C) = \min_{(a,b) \in C} \{|\sigma'(a,b)|\}$. One can *cancel* by setting $\sigma_1(a,b) \leftarrow \sigma_1(a,b) + \beta(C)$ for any forward edge $(b,a) \in C$ and $\sigma_1(a,b) \leftarrow \sigma_1(a,b) - \beta(C)$ for any backward edge $(a,b) \in C$. Equivalently, one can update σ' by setting $\sigma'(a,b) \leftarrow \sigma'(a,b) - \beta(C)$ for any forward edge $(b,a) \in C$ and $\sigma'(a,b) \leftarrow \sigma'(a,b) + \beta(C)$ for any backward edge $(a,b) \in C$. Cancelling a cycle does not affect the capacity of the points; however, it reduces the capacity on the edges of the cycle.

Computing the transport plan σ_F: Recall that σ_1 is a partial transport plan and σ^* is an optimal transport plan on $A \cup B$. In this part, we show that one can obtain a transport plan σ_F on the point sets $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ such that

(i) σ_F transports all excess weights $\eta_{\sigma_1}(\cdot)$ on the points in $A_F^{\sigma_1} \cup B_F^{\sigma_1}$, and

(ii) $w(\sigma_F) \leq w(\sigma^*) + w(\sigma_1)$.

Consider the transport σ_F on $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ constructed iteratively as follows. In each iteration, we find an augmenting path P from a free supply point $b \in B_F^{\sigma_1}$ to a free demand point $a \in A_F^{\sigma_1}$ in $\mathcal{G}$. We update the function σ' using P and assign a transportation of $\beta(P)$ supplies from b to a to the transport plan σ_F. After enough iterations, there will be no remaining free points (the capacity of all points in $A \cup B$ will be 0) and the transport plan σ_F would be a complete transport plan on $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ with the weight function $\eta_{\sigma_1}(\cdot)$. Figure 1(a) shows an example of σ_F.

To find an augmenting path, one can execute a DFS procedure from a free supply point $b \in B_F^{\sigma_1}$ on the graph $\mathcal{G}$ to find any free demand point $a \in A_F^{\sigma_1}$. During the execution of the DFS procedure, if we find an alternating cycle (which happens when the DFS finds an edge to an already visited point), we can update σ' along the cycle right away, remove the points on the cycle from the search tree, and continue our search. For any point $u \in A \cup B$ visited by the DFS procedure, if u is not a free point (i.e, $c(u) = 0$), then the DFS has visited u using a directed edge to u. Since $c(u) = 0$, there also exists an outgoing edge incident on u in $\mathcal{G}$; thus, the search cannot stop at a point that is not free. In other words, the DFS procedure, possibly after updating σ' along a set of cycles, finds an augmenting path.

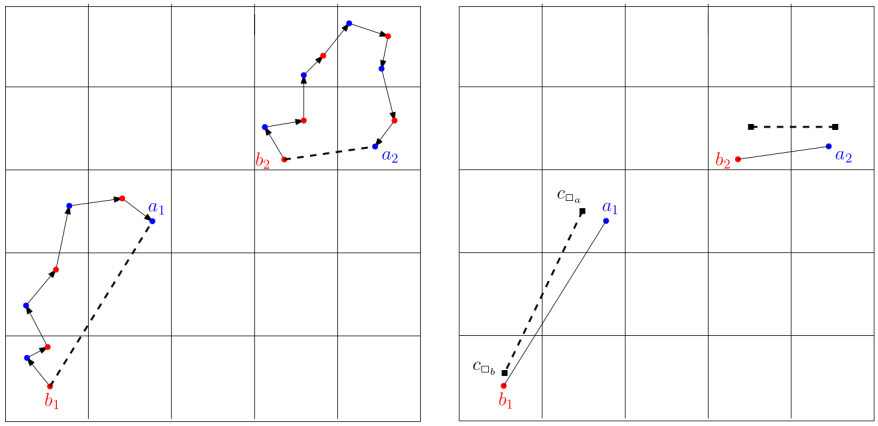


Figure 1: (a) Augmenting paths (solid lines) and the obtain transport plan σ_F (dashed lines), and, (b) Obtaining σ_2^* (dashed lines) from σ_f (solid lines).

Analyzing the cost of σ_F: Next, we show that the cost of σ_F is bounded by $w(\sigma^*) + w(\sigma_1)$. For any augmenting path P from $b_P \in B_F^{\sigma_1}$ to $a_P \in A_F^{\sigma_1}$, by the triangle inequality, $\|b_P - a_P\| \leq \sum_{(u,v)\in P} \|u - v\|$. For any pair of points $(a, b) \in A \times B$, let $\mathcal{P}_{ab}$ denote the set of all augmenting paths containing the edge (a, b).

$$\begin{aligned} w(\sigma_F) &= \sum_{P\in\mathcal{P}} \beta(P)\|a_P - b_P\| \\ &\leq \sum_{P\in\mathcal{P}} \beta(P) \sum_{(u,v)\in P} \|u - v\| \\ &= \sum_{(a,b)\in A\times B} \sum_{P\in\mathcal{P}_{ab}} \beta(P)\|a - b\|. \end{aligned}$$

For any forward edge (b, a) in $\mathcal{G}$, during the process of finding augmenting paths and updating σ', we have iteratively decreased the value of $\sigma'(b, a)$ while guaranteeing (by Equation 2) that $\sigma'(b, a) \geq 0$. Thus, $\sum_{P\in\mathcal{P}_{ba}} \beta(P) \leq \sigma'(b, a) = \sigma^*(b, a) - \sigma_1(b, a) \leq \sigma^*(b, a)$. Similarly, for any backward edge (a, b) in $\mathcal{G}$, we iteratively increase the value of $\sigma'(a, b)$ while guaranteeing that $\sigma'(a, b) \leq 0$. As a result, $\sum_{P\in\mathcal{P}_{ab}} \beta(P) \leq -\sigma'(a, b) \leq \sigma_1(a, b)$. Therefore,

$$\begin{aligned} w(\sigma_F) &\leq \sum_{(a,b)\in A\times B} \sum_{P\in\mathcal{P}_{ab}} \beta(P)\|a - b\| \\ &= \sum_{(a,b)\in A\times B:\sigma'(a,b)>0} \|a - b\| \sum_{P\in\mathcal{P}_{ab}} \beta(P) + \sum_{(a,b)\in A\times B:\sigma'(a,b)<0} \|a - b\| \sum_{P\in\mathcal{P}_{ab}} \beta(P) \\ &\leq \sum_{(a,b)\in A\times B:\sigma'(a,b)>0} \|a - b\|\sigma^*(a, b) + \sum_{(a,b)\in A\times B:\sigma'(a,b)<0} \|a - b\|\sigma_1(a, b) \\ &\leq w(\sigma^*) + w(\sigma_1). \end{aligned}$$

Lemma A.1. *There exists a complete transport plan σ_F on $A_F^\sigma \cup B_F^\sigma$ with the weight function $\eta_\sigma(\cdot)$ such that $w(\sigma_F) \leq w(\sigma^*) + w(\sigma_1)$.*

Computing the transport plan σ_2^*: Using the transport plan σ_F obtained in the previous step, in this step, we construct a transport plan σ_2^* on $\mathcal{A} \cup \mathcal{B}$ as follows. For any pair of points $(a, b) \in A_F^{\sigma_1} \times B_F^{\sigma_1}$ with $\sigma_F(a, b) > 0$, let $\square_a$ and $\square_b$ denote the cells of the grid containing a and b, respectively. Note that $\square_a$ (resp. $\square_b$) is a deficit (resp. surplus) cell, since $\square_a$ (resp. $\square_b$) contains a free demand point (resp. supply point) with respect to σ_1. We assign a transportation of $\sigma_F(a, b)$ from $c_{\square_b}$ to $c_{\square_a}$ to the transport plan σ_2^*. Note that σ_2^* is a complete transport plan for $\mathcal{A} \cup \mathcal{B}$. Figure 1(b) shows an example of σ_2^*. This completes the construction of σ_2^*.

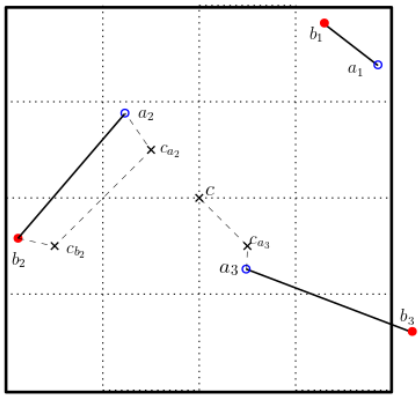


Figure 2: An example showing edges of first case (a_1, b_1), second case (a_2, b_2), and third case (a_3, b_3). The budget assigned to them are shown as dashed lines.

Next, we bound the cost of σ_2^*. Since the diameter of each cell is $\varepsilon/2$, by the triangle inequality,

$$\begin{aligned} w(\sigma_2^*) &= \sum_{(a,b)\in A_F^{\sigma_1}\times B_F^{\sigma_1}} \sigma_F(a,b)\|c_{\square_a} - c_{\square_b}\| \\ &\le \sum_{(a,b)\in A_F^{\sigma_1}\times B_F^{\sigma_1}} \sigma_F(a,b)(\|a-b\| + \frac{\varepsilon}{2}) \\ &\le w(\sigma_F) + \frac{\varepsilon}{2}(U - U_1). \end{aligned}$$

Lemma A.2. *Given a complete transport plan σ_F on the set of free points $A_F^{\sigma_1} \cup B_F^{\sigma_1}$ with the weight function $\eta_F(\cdot)$, there exists a transport plan σ_2^* on $\mathcal{A} \cup \mathcal{B}$ such that $w(\sigma_2^*) \le w(\sigma_F) + \frac{\varepsilon}{2}(U - U_1)$.*

This completes the description of the steps of our analysis.

B Missing details of Section 3

In this section, we provide the details of the discussion on the quality of approximation in Section 3 and show that the cost computed by our 1-Wasserstein algorithm is an $O(d \log_{\sqrt{d}/\varepsilon} n)$-approximation of the optimal transport cost on the point set $A \cup B$.

For any level i of the tree T, recall that C_i denotes the set of all cells at level i, the grid $\mathbb{G}_i$ defines level i of T, and ℓ_i is the cell-side-length of $\mathbb{G}_i$. In the following, first, we describe the two steps of our analysis to show that for each level $i < h$, the expected cost of the transport plans computed at all cells of level i is $O(d)w(\sigma^*)$.

Assigning Budgets: For a level $i < h$, consider the grids $\mathbb{G}_i$ and $\mathbb{G}_{i+1}$ (this is defined since $i < h$). For any edge $(a, b) \in A \times B$ with $\sigma^*(a, b) > 0$, we define its *budget* $\phi_i(a, b)$ as follows:

- *Case 1: both a and b are inside the same cell of $\mathbb{G}_{i+1}$*: We set $\phi_i(a, b) = 0$.
- *Case 2: both a and b are inside the same cell of $\mathbb{G}_i$ but in different cells $\square_a$ and $\square_b$ of $\mathbb{G}_{i+1}$*: We set $\phi_i(a, b) = (\sqrt{d}\ell_{i+1} + \|c_{\square_a} - c_{\square_b}\|)\sigma^*(a, b)$. Intuitively, this is the cost of moving from b to a via $c_{\square_b}$ and $c_{\square_a}$.
- *Case 3: a and b lie inside different cells $\square_a$ and $\square_b$ of $\mathbb{G}_i$*: We set $\phi_i(a, b) = (\sqrt{d}\ell_i)\sigma^*(a, b)$. Intuitively, this is the cost of moving a to $c_{\square_a}$ and b to $c_{\square_b}$.

Figure 2 shows an example of each of the above three cases. The following lemma relates the expected budget on any edge to its Euclidean cost.

Lemma B.1. *For any edge $(a, b) \in A \times B$, $\mathbb{E}[\phi_i(a, b)] = O(d)\|a - b\|\sigma^*(a, b)$.*

Proof. The probability of a and b lying inside the same cell of $\mathbb{G}_i$ but in different cells $\square_a$ and $\square_b$ of $\mathbb{G}_{i+1}$ is no more than the probability of (a,b) crossing $\mathbb{G}_{i+1}$. Furthermore, the probability that a and b lie inside different cells of $\mathbb{G}_i$ is equal to the probability of (a,b) crossing the grid $\mathbb{G}_i$. Therefore,

$$\begin{aligned}\mathbb{E}\left[\phi_i(a,b)\right] \leq{}& \Pr((a,b) \text{ crosses } \mathbb{G}_{i+1})(\sqrt{d}\ell_{i+1} + \|c_{\square_a} - c_{\square_b}\|)\sigma^*(a,b)\\ &+ \Pr((a,b) \text{ crosses } \mathbb{G}_i)\sqrt{d}\ell_i\sigma^*(a,b).\end{aligned}$$

By the triangle inequality,

$$\|c_{\square_a} - c_{\square_b}\| \leq \|c_a - a\| + \|a-b\| + \|b - c_b\| \leq \sqrt{d}\ell_{i+1} + \|a-b\|.$$

In addition, for any $j \leq h$, the probability that the edge (a,b) crosses the grid $\mathbb{G}_j$ is $\Pr((a,b) \text{ crosses } \mathbb{G}_j) \leq \min\{1, \frac{\sqrt{d}\|a-b\|}{\ell_j}\}$. Thus,

$$\begin{aligned}\mathbb{E}\left[\phi_i(a,b)\right] &\leq \Pr((a,b) \text{ crosses } \mathbb{G}_{i+1})(2\sqrt{d}\ell_{i+1} + \|a-b\|)\sigma^*(a,b)\\ &\quad + \Pr((a,b) \text{ crosses } \mathbb{G}_i)\sqrt{d}\ell_i\sigma^*(a,b)\\ &\leq \|a-b\|\sigma^*(a,b) + \left(\frac{\sqrt{d}\|a-b\|}{\ell_{i+1}}2\sqrt{d}\ell_{i+1} + \frac{\sqrt{d}\|a-b\|}{\ell_i}\sqrt{d}\ell_i\right)\sigma^*(a,b)\\ &\leq (1+3d)\|a-b\|\sigma^*(a,b).\end{aligned}$$

□

Using the linearity of expectation, the following is a straightforward corollary of Lemma B.1.

Corollary B.2. $\mathbb{E}\left[\sum_{(a,b)\in A\times B}\phi_i(a,b)\right] = O(d)w(\sigma^*)$.

Redistribution of Budgets to cells: In this section, we distribute the total budget on all edges of σ^* to non-empty cells of $\mathbb{G}_i$ and show that the budget assigned to any cell $\square \in \mathbb{G}_i$ is at least $w(\sigma_\square)/2$. For any cell $\square$ at level i of T, we define a budget $\phi_\square$ as follows. For any edge $(a,b) \in A \times B$ with $\sigma^*(a,b) > 0$, if

- *both a and b lie inside $\square$:* We assign the entire budget of (a,b) (i.e, $\phi_i(a,b)$) to $\phi_\square$.
- *only a or b is inside $\square$:* We assign half the budget of (a,b) (i.e, $\phi_i(a,b)/2$) to $\phi_\square$.

It is easy to observe that the total budget of all cells at level i of T is equal to the total budget of all edges carrying a positive amount of supplies in σ^*.

For any cell $\square$, the following lemma relates the cost of $\sigma_\square$ to the budget assigned to $\square$.

Lemma B.3. *For any cell $\square$ at a level $i < h$ of T, $w(\sigma_\square) \leq 2\phi_\square$.*

Proof. To prove the lemma, we show that there exists a transport plan σ' on the instance $\mathcal{I}_\square$ that has a cost no more than $\phi_\square$. Assuming this and considering that $\sigma_\square$ is a 2-approximate transport plan on $\mathcal{I}_\square$, we get $w(\sigma_\square) \leq 2w(\sigma') \leq 2\phi_\square$, as desired. Recall that $V_\square = (A \cup B) \cap \square$. For any point $u \in V_\square$, let $\square_u$ be the child of $\square$ that contains u. For any edge (u,v) with $\sigma^*(u,v) > 0$, consider the following cases:

- If $u, v \in V_\square$ and $\square_u \neq \square_v$, then σ' transports $\sigma^*(u,v)$ supplies from $c_{\square_u}$ to $c_{\square_v}$. Note that the edge (u,v) contributes to $\phi_\square$ by $\sigma^*(u,v)(\sqrt{d}\ell_{i+1} + \|c_{\square_u} - c_{\square_v}\|)$, which is greater than the cost of transporting $\sigma^*(u,v)$ supplies from $c_{\square_u}$ to $c_{\square_v}$.
- If $u \in V_\square$ and $v \notin V_\square$, then if u is a supply point, then σ' transports $\sigma^*(u,v)$ supplies from $c_{\square_u}$ to $c_\square$. Otherwise, u is a demand point and σ' transports $\sigma^*(u,v)$ supplies from $c_\square$ to $c_{\square_u}$. Note that (u,v) contributes to $\phi_\square$ by $\sigma^*(u,v)\frac{\sqrt{d}\ell_i}{2}$, which is greater than the cost of transporting $\sigma^*(u,v)$ supplies between $c_{\square_u}$ and $c_\square$.
- If $u, v \in V_\square$ and $\square_u = \square_v$ or if $u, v \notin V_\square$, then σ' do not transport any supplies to/from the points u and v.

By the discussion above, the cost of σ' is no more than $\phi_{\square}$. However, σ' might not be a feasible transport plan for the instance $\mathcal{I}_{\square}$. In that case, we can improve σ' as follows. For any points $a, b \in V_{\square}$ and $a', b' \notin V_{\square}$ such that $\sigma^*(a, b'), \sigma^*(a', b) > 0$, the transport plan σ' transports supplies from $c_{\square}$ to $c_{\square_a}$ and from $c_{\square_b}$ to $c_{\square}$, which we replace with a direct transportation from $c_{\square_b}$ to $c_{\square_a}$ without increasing the cost of σ'. Similarly, for any pair of cell centers c_u and c_v in $\mathcal{I}_{\square}$, where σ' transports supplies both from c_u to c_v and from c_v to c_u, we replace them with a direct transportation of supplies from one to the other without increasing the cost of σ'. The obtained transport plan is a feasible solution for the instance $\mathcal{I}_{\square}$ with a cost $w(\sigma') \leq \phi_{\square}$. □

Combining Lemma B.4, Corollary B.2, and Lemma B.3, we get the following.

$$\mathbb{E}\left[\sum_{i=0}^{h-1}\sum_{\square\in\mathsf{C}_i} w(\sigma_{\square})\right] \leq 2\sum_{i=0}^{h-1}\mathbb{E}\left[\sum_{\square\in\mathsf{C}_i}\phi_{\square}\right] = 2\sum_{i=0}^{h-1}\mathbb{E}\left[\sum_{(a,b)\in A\times B}\phi_i(a,b)\right]$$
$$= O(d\log_{\sqrt{d}/\varepsilon} n)w(\sigma^*). \tag{3}$$

Next, we show that the expected distortion cause by moving the points to the centers of the cells of $\mathbb{G}_h$ is also bounded by $O(d\log_{\sqrt{d}/\varepsilon} n)w(\sigma^*)$. For any non-empty cell $\square \in \mathsf{C}_h$, $\sigma_{\square}$ moves the supplies or demands of the only point in $V_{\square}$ to $c_{\square}$. Since the diameter of the cells at level h is $\sqrt{d}\ell_h$,

$$\mathbb{E}\left[\sum_{\square\in\mathsf{C}_h} w(\sigma_{\square})\right] \leq \mathbb{E}\left[\sum_{\square\in\mathsf{C}_h}\sqrt{d}\eta(\square)\ell_h\right] = 2\sqrt{d}U\mathbb{E}\left[\ell_h\right]. \tag{4}$$

Thus, it remains to bound the expected value of ℓ_h.

Lemma B.4. $\mathbb{E}\left[\ell_h\right] = O(\sqrt{d}C_{\min}\log_{\sqrt{d}/\varepsilon} n)$.

Proof. For any level i, recall that $\ell_i = 2(\frac{\varepsilon}{4\sqrt{d}})^i$. We bound the probability that $h = i$ as follows. Let (u, v) be the closest pair of points in $A \cup B$. Then, the probability that $h = i$ is no more than the probability that u and v lie inside different cells of $\mathbb{G}_i$ (i.e, the edge (u, v) crosses the grid $\mathbb{G}_i$), which is at most $\frac{\sqrt{d}\|u-v\|}{\ell_i}$.

Define $h_{\max} = \lceil\log_{\frac{4\sqrt{d}}{\varepsilon}}\frac{2\sqrt{d}}{C_{\min}}\rceil$, where $C_{\min}$ denotes the distance of the closest pair of points in $A \cup B$. Note that $h \leq h_{\max}$, since the diameter of any cell with side-length $\ell_{h_{\max}} = 2(\frac{\varepsilon}{4\sqrt{d}})^{h_{\max}}$ is less than $C_{\min}$ and no two points of $A \cup B$ can lie inside the same cell of a grid with cell-side-length $\ell_{h_{\max}}$. Since the spread of $A \cup B$ is $n^{O(1)}$, $h_{\max} = O(\log_{\sqrt{d}/\varepsilon} n)$. Thus,

$$\mathbb{E}\left[\ell_h\right] = \sum_{i=1}^{h_{\max}}\ell_i\Pr(h = i) \leq \sum_{i=1}^{h_{\max}}\ell_i\frac{\sqrt{d}C_{\min}}{\ell_i} = O(\sqrt{d}C_{\min}\log_{\frac{\sqrt{d}}{\varepsilon}} n).$$

□

Plugging the expected value of ℓ_h from Lemma B.4 into Equation 4,

$$\mathbb{E}\left[\sum_{\square\in\mathsf{C}_h} w(\sigma_{\square})\right] \leq 2\sqrt{d}U\mathbb{E}\left[\ell_h\right] = O(dUC_{\min}\log_{\sqrt{d}/\varepsilon} n) = O(d\log_{\sqrt{d}/\varepsilon} n)w(\sigma^*), \tag{5}$$

where the last equality holds true since $w(\sigma^*) \geq UC_{\min}$. The following is a straight-forward corollary from the combination of Equation 3 and Equation 5.

Corollary B.5. $\mathbb{E}\left[\sum_{\square\in T} w(\sigma_{\square})\right] = O(d\log_{\sqrt{d}/\varepsilon} n)w(\sigma^*)$.

C MISSING DETAILS OF SECTION 4

C.1 INPUT TRANSFORMATION

Let $A' \cup B'$ be the input point set. Similar to Agarwal et al. (2022), we transform $A' \cup B'$ into another point set $A \cup B$ such that

(T1) The coordinates of the points in $A \cup B$ are positive integers bounded by $n^{O(1)}$,

(T2) Any optimal matching with respect to $A \cup B$ is a $(1+\varepsilon)$ approximation for $A' \cup B'$, and,

(T3) The cost of the optimal matching of A and B is at least $5\sqrt{d}n/\varepsilon$ and at most $5d^{3/2}n \log n/\varepsilon$.

The details of this transformation are as follows. Recall that the hierarchical greedy algorithm described in Section 1 computes, in expectation, an $O(d \log n)$-approximation of the cost of the optimal matching. Suppose θ is the minimum value returned by $O(\log n)$ independent executions of the hierarchical greedy algorithm on $A' \cup B'$. Note that θ is an $O(d \log n)$-approximation of the optimal matching cost with a high probability. Using θ, we can transform any point $p' = (p_{x_1}, \ldots, p_{x_d}) \in A' \cup B'$ into another point $p = (\lceil \frac{5d^{3/2}n \log n}{\varepsilon\theta} p_{x_1} \rceil, \ldots, \lceil \frac{5d^{3/2}n \log n}{\varepsilon\theta} p_{x_d} \rceil)$ to obtain a set of points $A \cup B$. If any pair of points $(a, b) \in A \times B$ map to the same location, we match the two and remove them from $A \cup B$. The resulting point set satisfies (T1)–(T3).

C.2 RETRIEVING AN APPROXIMATE BIPARTITE MATCHING

In this section, we show how to retrieve a matching M on the point set $A \cup B$ using an identical approach as described in Section 3. We process the grids $\langle \mathbb{G}_h, \ldots, \mathbb{G}_1 \rangle$ in the decreasing order of their level. For any cell $\square$ of $\mathbb{G}_h$, we map the unique point in $V_\square$ to the center point $c_\square$. Inductively, assume that for a level $i < h$, after processing all non-empty cells of $\mathbb{G}_{i+1}$, conditions (i)–(iii) defined in Section 3 hold for each cell of that grid. For each non-empty cell $\square$ of $\mathbb{G}_i$, we show how to process $\square$ so that conditions (i)–(iii) hold for $\square$ as well. For any two children $\square_1, \square_2 \in \mathsf{C}[\square]$, we match $\sigma_\square(c_{\square_1}, c_{\square_2})$ many unmatched points that are mapped to $c_{\square_1}$ to the unmatched points that are mapped to $c_{\square_2}$. Additionally, for any child $\square_1 \in \mathsf{C}[\square]$ with $\sigma_\square(c_{\square_1}, c_\square) > 0$, we map $\sigma_\square(c_{\square_1}, c_\square)$ unmatched points from $c_{\square_1}$ to $c_\square$.

C.3 QUALITY OF APPROXIMATION

In this section, we show that the cost returned by the algorithm is an $O(d \log \log n)$-approximate matching cost. To do so, using an argument identical to Section 3, we can show that the total cost of the transport plans computed at all cells of level i (in expectation) is at most $O(d)w(M^*)$.

Lemma C.1. $\mathbb{E}\left[\sum_{\square \in \mathsf{C}_i} w(\sigma_\square)\right] \leq O(d)w(M^*)$.

Summing over all $O(\log \log n)$ levels of T, we get the following corollary.

Corollary C.2. $\sum_{i=1}^{h-1} \mathbb{E}\left[\sum_{\square \in \mathsf{C}_i} w(\sigma_\square)\right] = O(d \log \log n)w(M^*)$.

Furthermore, using an argument identical to Section 3, we can show that the expected cost of moving the points to the centers of the cells of the grid $\mathbb{G}_h$ is $O(d \log \log n)w(M^*)$.

Lemma C.3. $\mathbb{E}\left[\sum_{\square \in \mathsf{C}_h} w(\sigma_\square)\right] = O(d \log \log n)w(M^*)$.

Thus, the computed cost would be an $O(d \log \log n)$-approximation of the optimal matching cost, in expectation.

C.4 EFFICIENCY OF OUR ALGORITHM

In this section, we analyze the efficiency of our EBM algorithm and show that the running time of our algorithm is $O(T(n, \varepsilon/d) \log \log n)$. To do so, first, in Lemma C.4, we show that for each non-zero level $0 < i \leq h$, the algorithm requires $O(T(n, \varepsilon/d))$ time to process all cells at level i. Summing over all levels $0 < i \leq h$, the total running time of our algorithm on all non-root cells is $O(T(n, \varepsilon/d) \log \log n)$. Second, in Lemma C.7, we show that, with probability at least $1 - \frac{\varepsilon}{\sqrt{d}}$, no

edges of an optimal matching will cross $\mathbb{G}_1$. In this case, all children of the root cell $\square^*$ are neutral cells and the problem instance $\mathcal{I}_{\square^*}$ is an empty instance. By constructing $O(\log_{\sqrt{d}/\varepsilon} n)$ randomly-shifted hierarchical partitions, with a high probability, in at least one partition, all children of the root are neutral cells. Thus, with a high probability, we obtain a randomly shifted partition where no computation is needed for the root cell and the total running time of our algorithm would be $O(T(n, \varepsilon/d) \log\log n)$. We describe the details of each step in the following.

Lemma C.4. *For any level $0 < i \leq h$, our EBM algorithm processes all cells of level i in at most $T(n, \varepsilon/d)$ time.*

Proof. For any cell $\square$, recall that $n_\square$ denotes the number of points in $\mathcal{I}_\square$. Define $n_i = \sum_{\square \in \mathsf{C}_i} n_\square$. For any cell $\square \in \mathsf{C}_i$, the spread of the points in $\mathcal{I}_\square$ is $O\left(\frac{d}{\varepsilon} n^{1/2^i}\right)$. Therefore, since $T(n, \varepsilon) = \tilde{O}(n^k/\varepsilon)$ with $k \geq 1$, the execution time of our algorithm over all cells of level i is

$$\sum_{\square \in \mathsf{C}_i} T\left(n_\square, \frac{\varepsilon}{dn^{1/2^i}}\right) \leq T\left(n_i, \frac{\varepsilon}{dn^{1/2^i}}\right).$$

Additionally, since $n_i \leq n$, we have $\mathbb{E}\left[n_i^k\right] \leq n^{k-1}\mathbb{E}\left[n_i\right] = \left(\frac{\mathbb{E}[n_i]}{n}\right) n^k$. Therefore,

$$\sum_{\square \in \mathsf{C}_i} T\left(n_\square, \frac{\varepsilon}{dn^{1/2^i}}\right) \leq T\left(n_i, \frac{\varepsilon}{dn^{1/2^i}}\right) \leq \frac{n^{1/2^i}\mathbb{E}\left[n_i\right]}{n} T\left(n, \frac{\varepsilon}{d}\right). \tag{6}$$

In Corollary C.6 below, we show that $\mathbb{E}\left[n_i\right] \leq n^{1-\frac{1}{2^i}}$. Plugging into Equation 6, the time taken to process all cells of level i is

$$\frac{\mathbb{E}\left[n_i\right]}{n^{1-\frac{1}{2^i}}} T\left(n, \frac{\varepsilon}{d}\right) \leq T\left(n, \frac{\varepsilon}{d}\right),$$

as desired. □

Next, we bound the expected value of n_i. For any cell $\square$ and any child $\square' \in \mathsf{C}[\square]$, the weight of $c_{\square'}$ in $\mathcal{I}_\square$ is the minimum number of matching edges connecting a point inside $\square'$ to a point out of it in any perfect matching. Summing over all cells of level i, n_i is upper-bounded by the number of edge of M^* crossing the grid $\mathbb{G}_{i+1}$. The following lemma bounds this number as a function of $w(M^*)$ and ℓ_{i+1}.

Lemma C.5. *For any $0 < j \leq h$, the expected number of matching edges in M^* crossing $\mathbb{G}_j$ is at most $\sqrt{d}w(M^*)/\ell_j$.*

Proof. For any edge $(a, b) \in M^*$, the probability that the edge (a, b) crosses the grid $\mathbb{G}_j$ is $\Pr((a, b) \text{ crosses } \mathbb{G}_j) \leq \frac{\sqrt{d}\|a-b\|}{\ell_j}$. Let $X_j(a, b)$ be an indicator random variable such that $X_j(a, b) = 1$ if (a, b) crosses $\mathbb{G}_j$ and $X_j(a, b) = 0$ otherwise. Thus, $\mathbb{E}\left[X_j(a, b)\right] \leq \frac{\sqrt{d}\|a-b\|}{\ell_j}$. Define X_j to be a random variable indicating the number of edges of M^* crossing $\mathbb{G}_j$; i.e, $X_j = \sum_{(a,b)\in M^*} X_j(a, b)$. Using the linearity of expectation, the expected value of X_j is

$$\mathbb{E}\left[X_j\right] = \sum_{(a,b)\in M^*} \mathbb{E}\left[X_j(a, b)\right] \leq \frac{\sqrt{d}w(M^*)}{\ell_j}.$$

□

By invoking Lemma C.5 on level $i + 1$, as discussed above, $\mathbb{E}\left[n_i\right] \leq \sqrt{d}w(M^*)/\ell_{i+1}$. By definition, $\ell_{i+1} = \delta n^{1/2^i}$ and from property (3) of the input transformation, $w(M^*) \leq 5d^{3/2} n \log n/\varepsilon$. Therefore,

$$\mathbb{E}\left[n_i\right] \leq \frac{\sqrt{d}w(M^*)}{\ell_{i+1}} \leq \frac{5d^2 n \log n/\varepsilon}{5d^2 n^{\frac{1}{2^i}} \log n/\varepsilon} \leq n^{1-\frac{1}{2^i}}.$$

Thus, we get the following corollary.

Corollary C.6. *For any* $0 < j \leq h$, $\mathbb{E}[n_i] \leq n^{1-\frac{1}{2^i}}$.

Finally, we show that with probability at least $1 - \varepsilon/\sqrt{d}$, no edges of an optimal matching M^* cross the grid $\mathbb{G}_1$.

Lemma C.7. *Let M^* be an optimal matching on the point set $A \cup B$. Then, with probability at least $1 - \varepsilon/\sqrt{d}$, no edge of M^* crosses $\mathbb{G}_1$.*

Proof. Recall that $\ell_1 = 5d^{5/2}n \log n/\varepsilon^2$. For any pair of points $(a, b) \in A \times B$, the probability that the edge (a, b) crosses the grid $\mathbb{G}_1$ is upper-bounded as follows.

$$\Pr((a,b) \text{ crosses } \mathbb{G}_1) \leq \frac{\sqrt{d}\|a-b\|}{5d^{5/2}n\log n/\varepsilon^2}. \tag{7}$$

Let $X_{(a,b)}$ be an indicator random variable where $X_{(a,b)} = 1$ if (a, b) crosses $\mathbb{G}_1$ and $X_{(a,b)} = 0$ otherwise. Clearly, $\mathbb{E}\left[X_{(a,b)}\right] = \Pr((a, b) \text{ crosses } \mathbb{G}_1)$. Define $X = \sum_{(a,b)\in M^*} X_{(a,b)}$ to be the number of edges of M^* crossing the grid $\mathbb{G}_1$. By the linearity of expectation and Equation 7,

$$\mathbb{E}[X] = \sum_{(a,b)\in M^*} \mathbb{E}\left[X_{(a,b)}\right] \leq \sum_{(a,b)\in M^*} \frac{\sqrt{d}\|a-b\|}{5d^{5/2}n\log n/\varepsilon^2} = \frac{w(M^*)}{5d^2 n\log n/\varepsilon^2} \leq \frac{\varepsilon}{\sqrt{d}},$$

where the last inequality follows from property (T3) in the input transformation. Therefore, with probability at least $1 - \varepsilon/\sqrt{d}$, $X = 0$ and no edge of M^* crosses the grid $\mathbb{G}_1$. □

D A Faster Relative Approximation Algorithm via Approximate Nearest Neighbor

Both of our algorithms in Section 3 and Section 4 build a hierarchical structure and create an instance of the optimal transport at each cell of this hierarchical structure. Our algorithms then use an additive approximation algorithm for each such instance to obtain a 2-approximate transport plan (Lemma 2.2). In this section, we show that, instead of boosting the additive approximation, if we are only interested in achieving a relative approximation, it suffices if we find a certain bi-criteria approximate transport plan (defined below). In Euclidean setting, this bi-criteria approximate transport plan can be found significantly faster than additive approximate transport plans and therefore, we obtain faster execution times for our relative approximation algorithms (Theorems 1.3 and 1.4).

Given μ (with support A) and ν (with support B), let w^* denote the cost of the optimal transport plan from μ to ν. For parameter $\varepsilon > 0$ and $\alpha > 1$, let (α, ε)-*approximate transport plan* on $A \cup B$ be a transport plan σ satisfying $w(\sigma) \leq \alpha w^* + \varepsilon$. Using an identical discussion as in Section 2.2, when points have a unit diameter, by setting $\varepsilon = 1/\Delta$, the additive error of the transport plan will be at most w^*, and as a result, any $(\alpha, 1/\Delta)$-approximate transport plan is also an $(\alpha + 2)$-relative approximate transport plan. In the following, we show that (α, ε)-approximate transport plans can be computed efficiently in $O\left(\frac{nC^2\Phi(\alpha,n)}{\varepsilon^2}\right)$ time. We get the algorithms of Theorems 1.3 and 1.4 by simply replacing additive approximation algorithms used within the algorithms of Theorems 1.1 and 1.2, with a (α, ε)-approximation algorithm.

In this section, we will adapt the additive approximation algorithm of Lahn et al. (2019) (we refer to this algorithm by the LMR-Algorithm) to obtain an (α, ε)-approximate transport plan. The execution time of this adaptation is $\tilde{O}(n\Phi(n, \alpha)/\varepsilon^2)$ time which is significantly faster than the execution time of $O(n^2/\varepsilon + n/\varepsilon^2)$ achieved by the LMR-algorithm; here, $\tilde{O}$ hides factors of $\mathrm{poly}(\log n)$.

Overview of the LMR-Algorithm: For input points $A \cup B$ of size n and diameter C, the LMR-Algorithm computes an ε-close transport plan on $A \cup B$ as follows. Initially, the distances are scaled by $4/\varepsilon$ and rounded down so that the distance between any two points is an integer bounded by $O(C/\varepsilon)$. Furthermore, all demands and supplies are scaled and rounded so that the total supplies and demands are integers and $O(n/\varepsilon)$. The LMR-algorithm then iteratively executes phases of the Gabow and Tarjan's matching algorithm to match the demands to supplies. Within each phase, the dual weights of the free supply nodes increase by at least 1. Using the fact that the largest edge cost

is C/ε, the authors show that the algorithm transports all supplies in $O(C/\varepsilon)$ phases. The algorithm then rescales the supplies and costs and maps the solution computed by this algorithm to an ε-close transport plan.

As shown by Lahn & Raghvendra (2019), one can execute each phase of Gabow and Tarjan's algorithm by simply implementing a partial-DFS procedure and adjusting the dual weights during the procedure. Such a partial-DFS based implementation raises the dual weight of each free supply node increases by exactly 1. Using the arugments in Lahn et al. (2019), it follows that the algorithm transports all the supplies in $O(C/\varepsilon)$ phases. In order to speed up a phase of Gabow and Tarjan's algorithm, Agarwal and Sharathkumar Agarwal & Sharathkumar (2014) relaxed the feasibility conditions to the following.

$$\begin{aligned} y(b) - y(a) &\leq \alpha d(a,b), && \text{if } \sigma(a,b) < \min\{\eta(a), \eta(b)\}, \\ y(b) - y(a) &\geq d(a,b), && \text{if } \sigma(a,b) > 0. \end{aligned} \tag{8}$$

They showed how one can maintain these relaxed feasibility conditions and implement a partial-DFS in $O(n\Phi(n,\varepsilon))$ time. Similar to this, we can relax the feasibility conditions for the LMR algorithm and then execute the phases of the LMR algorithm using an approximate nearest neighbor data structure. Consequently, we find an (α,ε)-approximate transport plan in $O(nC^2\Phi(n,\varepsilon)/\varepsilon^2)$ time leading to Theorems 1.3 and 1.4

E Additional Experiments

In this section, we show the results of our additional experiments. In addition to the two datasets (15D and Real) introduced in Section 5, we use three more datasets in our experiments: (i) a uniform distribution inside a unit square (Uniform), (ii) 3-dimensional Gaussian mixture with 2 mean points centered at $(0.24, 0.24, 0.24)$ and $(0.72, 0.72, 0.72)$ inside a unit cube (Bimodal), and (iii) a uniform distribution from inside a 2-dimensional unit square placed on a random plane in 6-dimensional space (6D).

E.1 Comparison with Sinkhorn algorithm

In this experiment, we want to determine, for any distribution and any fixed sample set drawn from that distribution, how long it takes for the Sinkhorn algorithm to compute a transport plan with a comparable cost to ours. Using a binary search, we identify, for each sample set, the value of the regularization parameter η in the Sinkhorn algorithm which results in a cost that is nearly identical to the cost computed by our algorithm. Figure 3 depicts the results of this experiment.

E.2 Comparison with the LMR algorithm

In this experiment, we use the LMR algorithm as the blackbox solver in our 1-Wasserstein algorithm. We compare our result with the result of applying the LMR algorithm on the entire pointset. In this experiment, we compared the execution time of our 1-Wasserstein algorithm with the LMR algorithm when the two algorithms produce comparable costs. Figure 4 shows the cost and running time plots of the executions of our algorithm and that of the LMR algorithm for four data sets we use in our experiments, i.e. Uniform, Bimodal, 15D-Uniform and the Real dataset.

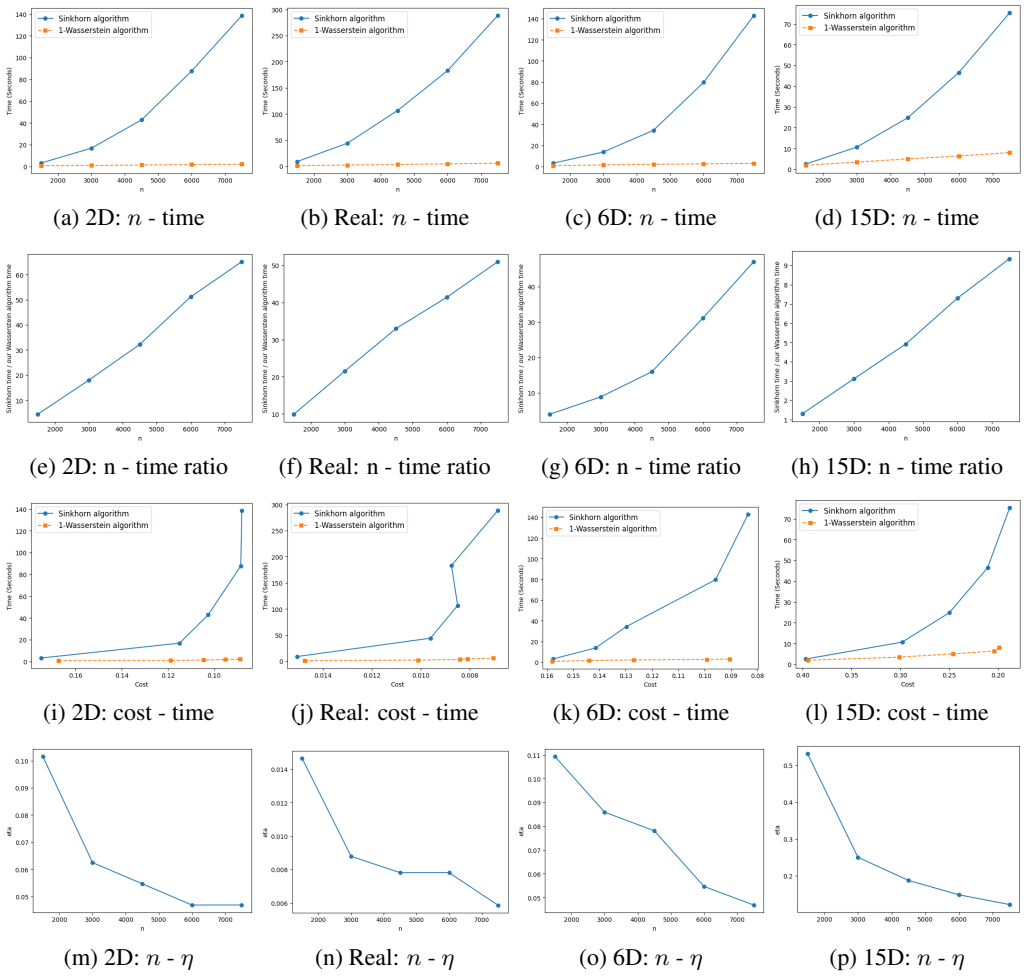
We can see in Figure 4 that while computing similar costs, the running time of our algorithm grows drastically slower than the LMR algorithm in all the cases.

E.3 Comparison with the Geometric-Additive algorithm

In this experiment, we compared the performance of our 1-Wasserstein and EBM algorithms to the Geometric Additive algorithm from Section 2.1. The results are illustrated in Figure 5.

References

Pankaj K. Agarwal and R. Sharathkumar. Approximation algorithms for bipartite matching with metric and geometric costs. In *Proc. ACM Symposium on Theory of Computing*, pp. 555–564,



(a) 2D: n - time (b) Real: n - time (c) 6D: n - time (d) 15D: n - time

(e) 2D: n - time ratio (f) Real: n - time ratio (g) 6D: n - time ratio (h) 15D: n - time ratio

(i) 2D: cost - time (j) Real: cost - time (k) 6D: cost - time (l) 15D: cost - time

(m) 2D: n - η (n) Real: n - η (o) 6D: n - η (p) 15D: n - η

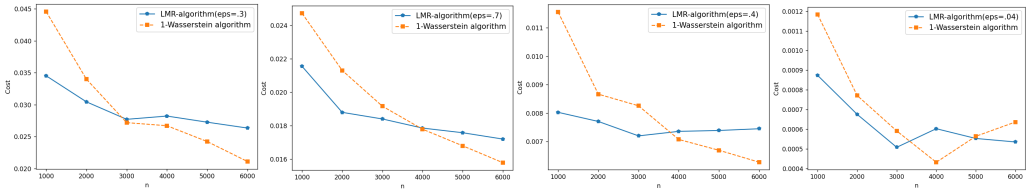
Figure 3: Comparison of our 1-Wasserstein algorithm with Sinkhorn algorithm while producing solutions of similar quality. (a)-(d) Sample size and execution time, (e)-(h) Sample size and the ratio of the execution time of Sinkhorn and our Wasserstein algorithm, (i)-(l) Cost and execution time, (m)-(n) Sample size and value of the regularization parameter η in Sinkhorn algorithm.

2014. doi: 10.1145/2591796.2591844. URL https://doi.org/10.1145/2591796.2591844.

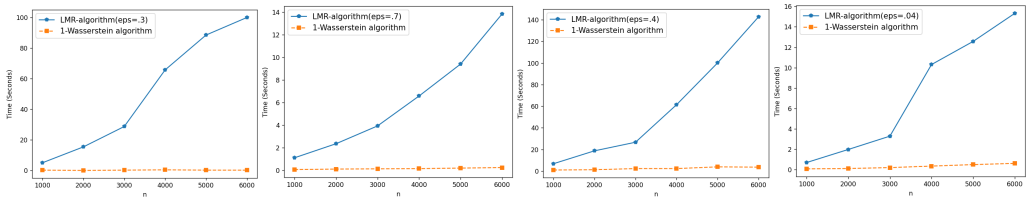
Pankaj K Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Rachita Sowle. An improved ε-approximation algorithm for geometric bipartite matching. In *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

Nathaniel Lahn and Sharath Raghvendra. A faster algorithm for minimum-cost bipartite matching in minor-free graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 569–588. SIAM, 2019.

Nathaniel Lahn, Deepika Mulchandani, and Sharath Raghvendra. A graph theoretic additive approximation of optimal transport. *In Advances in Neural Information Processing Systems 32*, pp. 13813–13823, 2019.

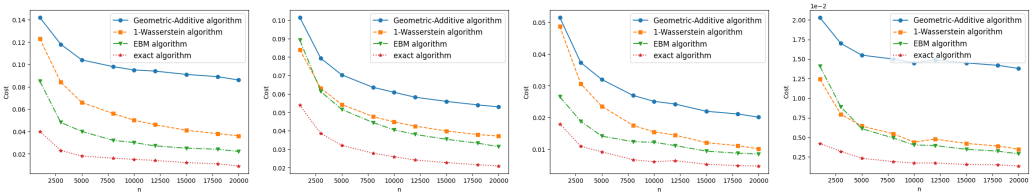


(a) 2D: n - cost (b) Bimodal: n - cost (c) 15D: n - cost (d) Real: n - cost

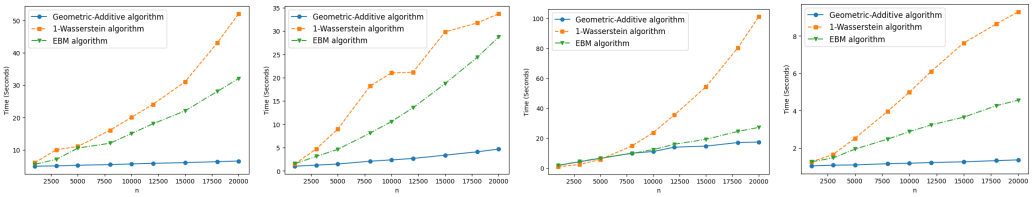


(a) 2D: n - time (b) Bimodal: n - time (c) 15D: n - time (d) Real: n - time

Figure 4: (a)-(d) comparing the cost computed by our 1-Wasserstein algorithm vs cost returned by LMR algorithm on Uniform, Bimodal, 15-D Uniform, and Real dataset. (e)-(h) comparing the running time of our 1-Wasserstein algorithm vs the running time of LMR algorithm on Uniform, Bimodal, 15-D Uniform, and Real dataset.



(a) 2D: n-cost (b) Bimodal: n-cost (c) 15D: n-cost (d) Real: n-cost



(e) 2D: n-time (f) Bimodal: n-time (g) 15D: n-time (h) Real: n-time

Figure 5: (a)–(d) estimated 1-Wasserstein distance, and (e)–(h) execution times of the algorithms