
Finding Wasserstein Ball Center: Efficient Algorithm and The Applications in Fairness

Yuntao Wang^{*1} Yuxuan li^{*2} Qingyuan Yang¹ Hu Ding¹

Abstract

Wasserstein Barycenter (WB) is a fundamental geometric optimization problem in machine learning, whose objective is to find a representative probability measure that minimizes the sum of Wasserstein distances to given distributions. WB has a number of applications in various areas. However, WB may lead to unfair outcome towards underrepresented groups in some applications (e.g., a “minority” distribution may be far away from the obtained WB under Wasserstein distance). To address this issue, we propose an alternative objective called “Wasserstein Ball Center (WBC)”. Specifically, WBC is a distribution that encompasses all input distributions within the minimum Wasserstein distance, which can be formulated as a “minmax” optimization problem. We show that the WBC problem with fixed support is equivalent to solving a large-scale linear programming (LP) instance, which is quite different from the previously studied LP model for WB. By incorporating some novel observations on the induced normal equation, we propose an efficient algorithm that accelerates the interior point method by $O(\min\{N^2m, Nm^2, m^4\})$ times (“ N ” is the number of distributions and “ m ” is the support size). Finally, we conduct a set of experiments on both synthetic and real-world datasets, demonstrating the computational efficiency of our algorithm, and showing its ability to provide more fairness for input distributions.

1. Introduction

To find a representative of several given probability distributions is a natural problem in machine learning. One popular approach is to compute the geometric center on probability space with induced distances between probabilities, such as the seminal *Wasserstein distance* in the field of optimal transport (Villani, 2021). Given a non-negative weight vector $(\omega_1, \omega_2, \dots, \omega_N)$ with $N \geq 2$, the **Wasserstein barycenter** (WB) of N probability measures $\{\mu_k\}_{k=1}^N$ under norm p is defined as the weighted Fréchet Mean mean under Wasserstein distance. Namely, it is the solution of the following problem

$$\min_{\mu \in \mathcal{P}_p(\Omega)} \sum_{t=1}^N \omega_t W_p^p(\mu, \mu_t), \quad (1)$$

where $\mathcal{P}_p(\Omega)$ is the set of Borel probability measure on Ω with finite p -th moment, and W_p is the Wasserstein distance under norm p , which will be formally defined in Section 2. WB has found various applications in many fields, such as economics (Carlier & Ekeland, 2010; Chiappori, 2017), physics (Benamou et al., 2014; Koehl et al., 2019), statistics (Kroshnin et al., 2021; Backhoff-Veraguas et al., 2022; Goldfeld et al., 2024), medical imaging (Gramfort et al., 2015; Nadeem et al., 2020), and machine learning (Qin et al., 2021; Cheng et al., 2021; Zhuang et al., 2022).

As the Fréchet mean under Wasserstein distance, WB tends to assign more measure to the region where the input density functions “cluster”. In other words, to minimize the average distance from the barycenter to the input probabilities, if the support of most distribution is concentrated with high probability in a region, then the WB should also have measure concentrated in that region. But this property may behave “unfairly” to “minority”, *i.e.*, the distributions with support deviated from the majority of others could be too far away from the WB. Fig. 1 gives an intuitive demonstration for this issue.

The unfairness could cause negative impact in some scenarios (Mehrabi et al., 2021; Caton & Haas, 2024). In medical applications, datasets usually reflect societal biases or historical inequities in healthcare access. If a deep learning model is trained on biased data, it can propagate or even amplify these biases, leading to unequal treatment outcomes. For

^{*}Equal contribution ¹School of Computer Science and Technology, University of Science and Technology of China, Hefei, China ²School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei, China. Correspondence to: Hu Ding <huding@ustc.edu.cn>.

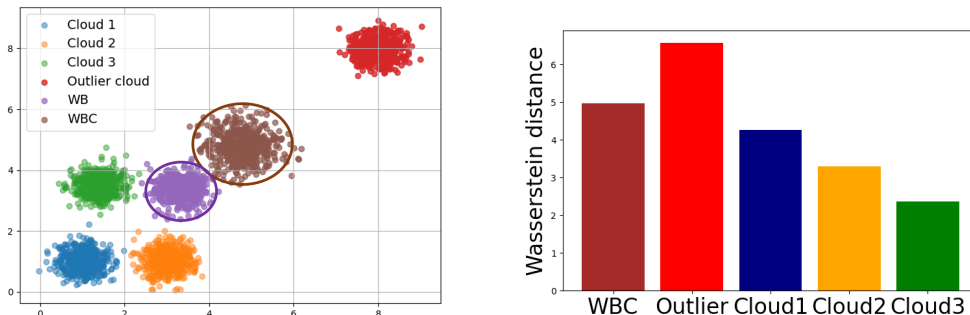


Figure 1. Left: four probability measures (cloud 1-3, and an outlier cloud), with their WB (computed by (1)) enclosed in purple ellipse, WBC (computed by (2)) enclosed in brown ellipse. The red cloud (*i.e.*, the “outlier cloud”) has measure distributed distinctly from the other three clouds. In the histogram on the right, the “WBC” bin denotes the maximal Wasserstein distance from WBC to four clouds, and each of the other four bins denotes the Wasserstein distance from WB to the corresponding cloud. From this histogram, we can see that WBC treats the outlier more equally (for example, the “WBC” bin is significantly lower than the “Outlier” bin).

instance, many skin lesion datasets predominantly contain images from lighter-skinned individuals (Bhardwaj & Rege, 2021; Adegun & Viriri, 2021). As a result, AI systems trained on such data often perform poorly on darker-skinned patients, leading to underdiagnosis or misdiagnosis in certain populations (Montoya et al., 2025). In addition, the medical data may come from different sources, therefore a fair alignment can also enhance model’s accuracy (Aayushman et al., 2024; Lin et al., 2023).

To address this unfair issue, we propose a different objective function. Rather than minimizing the summation of Wasserstein distances, we try to find a distribution that is of minimal distance from the farthest input distribution:

$$\min_{\mu \in \mathcal{P}_p(\Omega)} \max_{1 \leq t \leq N} W_p(\mu, \mu_t). \quad (2)$$

From a geometric perspective, we can think of it as the center of the ball in Wasserstein space, who covers all input distributions with minimum radius. In this setting, the output distribution does not put extra measure to the region where input distributions cluster with high density. Please see Fig. 1 for an illustrative comparison. We call the solution for Problem (2) the **Wasserstein Ball Center (WBC)**, and aim to design an efficient algorithm to solve it. It should be noted that “Wasserstein ball” is not a new concept and actually has been studied by several works before (Yue et al., 2022; Pesenti & Jaimungal, 2023; Chen et al., 2024), yet these previous works usually assume **the ball center is given** and take the ball as a feasible region for constraining some optimization objective. In this paper, we focus on how to compute an optimal center so that the induced radius (under Wasserstein distance) is minimized.

The WBC model (2) also shares the similar “**minmax**” property with the recently proposed clustering problem under the notion of group fairness, called *social fairness* or *equitable group representation* (Abbasi et al., 2021; Ghadiri

et al., 2021). In this framework, the data set for clustering consists of a set of groups, where each group contains a set of data items. Standard clustering algorithms could incur higher clustering costs for certain protected groups (*e.g.*, the groups that are defined by a sensitive attribute such as gender or race). This lack of fairness has motivated the study of clustering that minimizes the maximum clustering cost across different demographic groups, such as gender, race, high socioeconomic status, *etc.* The sensitive groups thus can obtain more equitable attention in the algorithm. Comparing with the traditional clustering models (*e.g.*, *k*-means or *k*-median clustering s), the main difference (which is also the main challenge) lies in this “minmax” objective.

1.1. Our Main Contributions

Similar with the aforementioned socially fair clustering problems, solving Problem (2) is also not easy due to its inherent “minmax” nature. When all distributions are of discrete support, the problem can be formulated as a linear programming (LP) problem, where the details are shown in Section 2. Partly inspired by the recent developments of *interior point method (IPM)* and their various applications (Gondzio, 2012) (*e.g.*, the IPM algorithm for solving WB (Ge et al., 2019)), we also consider developing an efficient IPM based algorithm for the WBC problem, though its formulation is much more complicated than WB due to the minimax issue.

Technically, there is a major challenge for directly applying IPM to the WBC problem, *viz.*, the computational cost and space complexity are both very large. The linear programming formulation of WBC has $m \sum_{i=1}^N m_i + m + N + 1$ variables and $Nm + \sum_{i=1}^N m_i + N + 1$ constraints, where the integer N denotes number of distributions, m_i and m denote the support sizes for the i -th distribution and WBC respectively. This brings the challenge for computing the in-

ner loop of IPM, which requires a time complexity as large as $O((Nm + \sum_{i=1}^N m_i + N)^2(m \sum_{i=1}^N m_i + m + N))$. To tackle this difficulty, we grind the intrinsic information of constraint matrix to simplify the Newton normal equation, which is a linear system with a large positive definite constraint matrix, and is also the most expensive part in each inner loop of IPM. Specifically, we simplify the matrix inverse occurred in the solution of *Newton path*, based on an important observation:

After performing some carefully designed transformations, the seemingly dense coefficient matrix of normal equation (a core part in our IPM framework) can be converted into a block-diagonal matrix in a particularly efficient way, where the inverse of each block is also easy to obtain.

Through this observation, we obtain an $O(Nm^3 + N^2m^2 + N^3)$ time complexity for each iteration, whereas the vanilla IPM requires $O(N^3m^4)$ by straight matrix inversion (for simplicity we just assume $m_i = O(m)$ here). The latter one is often beyond acceptable scope in real-world scenarios. For example, assume we are given an instance that $N = 1000$ and $m = 100$. In each loop of IPM, the complexity of our algorithm is $O(10^{10})$, while the vanilla IPM requires $O(10^{17})$, which is $O(10^7)$ times higher. The strict description on this result is presented in Theorem 3.2. As for the practical effectiveness, our algorithm is significantly faster than the popular commercial solver ‘‘Gurobi’’ (Gurobi Optimization, LLC, 2024), which provides a concurrent optimizer that run different state-of-the-art optimization methods simultaneously. In particular for LP models, Gurobi simultaneously runs the ‘‘dual simplex’’, ‘‘parallel barrier’’, and ‘‘primal simplex’’ algorithms in independent threads. In other words, our proposed algorithm runs faster than all these commonly used LP algorithms. In Section 4, we provide a set of experiments on numerical performance and applications.

1.2. Related works

We present an overview on several important related works, and the detailed introduction is placed in Section A.

Wasserstein distance Wasserstein distance is a classic topic in machine learning (Rüschendorf, 1985). Cuturi (2013) introduced the ‘‘Sinkhorn Distance’’, which incorporates an entropic regularization term to smooth the transportation problem, offering paralleled and significantly faster solutions than exact computation of the discrete Wasserstein distance. Following Cuturi’s work, recent years have seen the development of several improved Sinkhorn algorithms (Lin et al., 2019; Altschuler et al., 2019; Benamou et al., 2015; Altschuler et al., 2017). The discrete Wasserstein distance is also closely related to the *min-cost max flow*

problem (Ahuja et al., 1994). In the past years, a number of elegant algorithms have been proposed in theoretical computer science, such as (Orlin, 1988; Lee & Sidford, 2014; Sherman, 2017), and the ones focusing on Euclidean space (Khesin et al., 2021; Fox & Lu, 2023) or parallelism (Lahn et al., 2023).

Wasserstein barycenter. The computation for Wasserstein barycenter is an NP-hard problem (Altschuler & Boix-Adsera, 2022). Cuturi & Doucet (2014) showed that the computation for WB can be improved by using an entropic regularization. Further progress includes iterative Bregman projection (IBP) algorithm (Benamou et al., 2015), the semi-dual gradient descent (Cuturi & Peyré, 2018), accelerated primal-dual gradient descent (APDAGD) (Kroshnin et al., 2019), alternating direction method of multipliers (ADMM) (Ye et al., 2017), deterministic IBP (Lin et al., 2020), and the IPM algorithm MAAIPM (Ge et al., 2019). In Euclidean space, WB can be solved in polynomial time with fixed dimension (Altschuler & Boix-Adsera, 2021; Agarwal et al., 2025).

Interior Point Method. Dikin (1967) proposed the first interior point method. Further, Karmarkar (1984) proposed the first polynomial time method for linear programming called ‘‘Karmarkar’s algorithm’’. Mehrotra’s *predictor-corrector* algorithm (Mehrotra, 1992) provides the basis for most implementations of this class of methods, which is also the type of IPM applied in this paper (the details of predictor-corrector IPM are presented in Section 3.2). The seminal Mizuno-Todd-Ye predictor-corrector method has quadratic convergence (Mizuno et al., 1993; Ye et al., 1993). For more details on IPM, we refer the reader to the survey paper (Gondzio, 2012). Recently, there are also several new studies on reducing the IPM complexity (Jiang et al., 2021; Cohen et al., 2021), which rely on a technique called ‘‘matrix maintenance’’ to reduce the update time of each iteration.

2. Preliminaries

Some notations. Throughout this paper, we always use low case letter to denote scalar value, and bold low case letter to denote vector; also, we use ‘‘ $[n]$ ’’ to denote the set $\{1, 2, \dots, n\}$. Let ‘‘ $\mathbf{1}_m$ ’’ be the all-one vector in \mathbb{R}^m . For two discrete probability vectors $\mathbf{u} \in \mathbb{R}^{m_1}$ and $\mathbf{v} \in \mathbb{R}^{m_2}$ with m_1 and m_2 being their support sizes, define the set of matrices $\mathcal{M}(\mathbf{u}, \mathbf{v}) = \{\Pi \in \mathbb{R}_+^{m_1 \times m_2} : \Pi \mathbf{1}_{m_2} = \mathbf{u}, \Pi^\top \mathbf{1}_{m_1} = \mathbf{v}\}$ as the **coupling matrices**, which consists of all joint distributions of margin \mathbf{u} and \mathbf{v} . Let $\mathcal{Q} = \{(a_i, \mathbf{q}_i) : i \in [m_1]\}$ denote the discrete probability measure supported on m_1 points $\mathbf{q}_1, \dots, \mathbf{q}_{m_1}$ in \mathbb{R}^d with weights a_1, \dots, a_{m_1} respectively. Another probability measure $\mathcal{P} = \{(b_j, \mathbf{p}_j) : j \in [m_2]\}$ is defined similarly. The **p -norm Wasserstein distance** of these two discrete proba-

bility measures \mathcal{Q} and \mathcal{P} is

$$W_p(\mathcal{Q}, \mathcal{P}) := \min \left\{ \left(\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \pi_{ij} \| \mathbf{q}_i - \mathbf{p}_j \|_p^p \right)^{\frac{1}{p}} : \right. \\ \left. \Pi = (\pi_{ij})_{(m_1 \times m_2)} \in \mathcal{M}(\mathbf{a}, \mathbf{b}) \right\} \quad (\text{OT})$$

where $\mathbf{a} = (a_1, \dots, a_{m_1})^\top$ and $\mathbf{b} = (b_1, \dots, b_{m_2})^\top$. The coupling matrix achieving the minimum value can be seen as an alignment between the two distributions, which provides the optimal transportation (OT) plan between their supports.

Now, we consider the scenario where there are multiple probability measures in the same space, $\{\mathcal{P}^{(t)}, t = 1, \dots, N\}$ ($N \geq 2$), where each $\mathcal{P}^{(t)} = \{\{a_i^{(t)}, \mathbf{q}_i^{(t)} : i \in [m_t]\}$ with probability vector $\mathbf{a}^{(t)} = (a_1^{(t)}, \dots, a_{m_t}^{(t)})^\top$. The optimal Wasserstein ball center (WBC) $\mathcal{P}_{\text{opt}} = \{(w_i, \mathbf{x}_i) : i \in [m]\}$ is another probability measure such that the maximum Wasserstein distance to these given N probability measures is minimized, as defined in the objective function (2) when $\Omega = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$. The probability $\mathbf{w} = (w_1, \dots, w_m)^\top$ of \mathcal{P}_{opt} and its coupling matrices with $\{\mathbf{a}^{(t)} : t \in [N]\}$ must be in the solution set $\mathcal{S} =$

$$\{(\mathbf{w}, \Pi^{(1)}, \dots, \Pi^{(N)}) \in \mathbb{R}_+^m \times \mathbb{R}_+^{m \times m_1} \times \dots \times \mathbb{R}_+^{m \times m_N} : \\ \underbrace{\mathbf{1}_m^\top \mathbf{w} = 1, \mathbf{w} \geq 0}_{\text{constraint (i)}}; \\ \underbrace{\Pi^{(t)} \mathbf{1}_{m_t} = \mathbf{w}, (\Pi^{(t)})^\top \mathbf{1}_m = \mathbf{a}^{(t)}, \Pi^{(t)} \geq 0, t \in [N]}_{\text{constraint (ii)}}\}, \quad (3)$$

where “ \geq ” means each entry of the vector/matrix is non-negative, the constraint (i) guarantees \mathbf{w} to be a feasible probability distribution, and the constraint (ii) follows the coupling condition in (OT). For a given support Ω , the **distance matrices** are defined as $D^{(t)}(\Omega) = (\|\mathbf{x}_i - \mathbf{q}_j^{(t)}\|_p)_{(i,j)} \in \mathbb{R}^{m \times m_t}$ for $t = 1, \dots, N$. Since in most practical applications, the measures in $\{\mathcal{P}^{(t)}\}_{t=1}^N$ have finite support points, and we can assume the barycenter is supported on a given finite set Ω (in general, Ω does not need to be the same as the supports of $\mathcal{P}^{(t)}$ s), the problem WBC in (2) can be re-written as:

$$\min_{\mathbf{w}} \max_{t \in [N]} \min_{\Pi^{(t)}} \langle D^{(t)}, \Pi^{(t)} \rangle \quad \text{s.t. } (\mathbf{w}, \Pi^{(1)}, \dots, \Pi^{(N)}) \in \mathcal{S} \quad (4)$$

where “ $\langle \cdot, \cdot \rangle$ ” denotes the inner product in Euclidean space, and $D^{(t)}$ denotes $D^t(\Omega)$ for simplicity.

3. Our Optimization Framework for WBC

In this section, we introduce our optimization framework for solving the Problem (4). Specifically, in Section 3.1 we formalize Problem (4) to be the standard LP form, ensuring that the constraint matrix is full row-rank so that the

IPM framework introduced in Section 3.2 can run properly. In Section 3.3 we illustrate how to eliminate unnecessary computations to speed up IPM, as the key part of our contribution in the current article. We take advantage of some critical observation on the symmetry behind the seemingly dense constraint matrix. With that observation, we can significantly reduce the complexity of the most time consuming part in each loop of IPM, which is the inverting for a large matrix.

3.1. From Minmax to LP

First, we apply a slack variable $\gamma \in \mathbb{R}$, turning Problem (4) into the following LP problem:

$$\min_{\mathbf{w}, \Pi^{(t)}, \gamma} \quad \gamma \\ \text{s.t. } (\mathbf{w}, \Pi^{(1)}, \dots, \Pi^{(N)}) \in \mathcal{S}, \mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n \quad (5) \\ \langle D^{(t)}(X), \Pi^{(t)} \rangle \leq \gamma, 1 \leq t \leq N.$$

In (5), the value γ indicates the maximal Wasserstein distance from the WBC to the given N probability measures.

Then, we conduct the following transformation on Problem (5). We use “ $\text{vec}(U)$ ” to denote the vectorization of a matrix U , which is a vector obtained by concatenating every column of U to its predecessor. Let $\text{diag}(U_1, \dots, U_m)$ denote the block diagonal matrix with the i -th block U_i , and “ \otimes ” denote tensor product. Let “ I_m ” be the identity matrix of order m . To transform the problem to a standard LP form, we vectorize the constraint (ii) in (3), “ $\Pi^{(t)} \mathbf{1}_{m_t} = \mathbf{w}$ ” and “ $(\Pi^{(t)})^\top \mathbf{1}_m = \mathbf{a}^{(t)}$ ”, to be:

$$(\mathbf{1}_{m_t}^\top \otimes I_m) \text{vec}(\Pi^{(t)}) = \mathbf{w}, (I_{m_t} \otimes \mathbf{1}_m^\top) \text{vec}(\Pi^{(t)}) = \mathbf{a}^{(t)},$$

for $t = 1, \dots, N$. We further define the following matrices: $E_1 = \text{diag}(I_{m_1} \otimes \mathbf{1}_m^\top, \dots, I_{m_N} \otimes \mathbf{1}_m^\top)$, $E_2 = \text{diag}(\mathbf{1}_{m_1}^\top \otimes I_m, \dots, \mathbf{1}_{m_N}^\top \otimes I_m)$, $E_3 = -\mathbf{1}_N \otimes I_m$ and $D = \text{diag}(\text{vec}(D^{(1)}), \dots, \text{vec}(D^{(N)}))$. Through some simple calculations, Problem (5) can be formulated as a standard LP model:

$$\min \mathbf{c}^\top \mathbf{x} \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0 \quad (\text{WBC-LP})$$

where $\mathbf{x} = (\text{vec}(\Pi^{(1)}); \dots; \text{vec}(\Pi^{(N)}); \mathbf{w}; \gamma_1; \dots; \gamma_N, \gamma)$, $\gamma_i = \gamma - \langle D^{(i)}, \Pi^{(i)} \rangle$, $\mathbf{b} = (\mathbf{a}^{(1)}; \dots; \mathbf{a}^{(N)}; \underbrace{\mathbf{0}_m; \dots; \mathbf{0}_m; 1}_N)$,

$\mathbf{c} = (0; 0; \dots; 0; 1)$ and

$$\mathbf{A} = \begin{bmatrix} E_1 & & & & & \\ E_2 & E_3 & & & & \\ & \mathbf{1}_m^\top & & & & \\ D & & I_N & & -\mathbf{1}_N & \end{bmatrix}.$$

Let $M := \sum_{i=1}^N m_i$, and then the numbers of constraints and variables are $n_c := Nm + M + N + 1$ and $n_v :=$

$Mm + m + N + 1$, respectively. For a visual demonstration of these matrices, please see Eq. (12).

Preprocessing on (WBC-LP). To implement IPM, it is essential that the constraint matrix A should be of full row-rank, so that the problem’s complexity is reduced and IPM is also guaranteed to work more efficiently. In particular, the “full row-rank” property can simplify the procedure for solving the “normal equation” in IPM (which is discussed in next section). The following lemma eliminates all redundant constraints, turning A into a full row-rank matrix \bar{A} . Specifically, $\bar{A} \in \mathbb{R}^{(n_c - N) \times n_v}$ is the matrix obtained from A by removing the $(M + 1)$ -th, $(M + m + 1)$ -th, \dots , $(M + (N - 1)m + 1)$ -th rows of A , and $\bar{\mathbf{b}} \in \mathbb{R}^{n_c - N}$ is the vector obtained from \mathbf{b} by removing the $(M + 1)$ -th, $(M + m + 1)$ -th, \dots , $(M + (N - 1)m + 1)$ -th entries of \mathbf{b} .

Proposition 3.1. 1) \bar{A} has full row-rank; 2) solving the equation $A\mathbf{x} = \mathbf{b}$ in (WBC-LP) is equivalent to solving the equation $\bar{A}\mathbf{x} = \bar{\mathbf{b}}$.

We leave the proof to Section B. Due to Theorem 3.1, we will focus on \bar{A} instead of A in the following sections.

3.2. Predictor-Corrector IPM on WBC-LP

We choose the classic predictor-corrector scheme (Mehrotra, 1992), which is a variant of primal-dual IPM, that applies the block coordinate descent on the Lagrangian of the linear program smoothed by logarithm regularization. As a second order method, it is proved to have quadratic convergence rate (Ye et al., 1993), which surpasses first-order methods in general (for a detailed analysis, please see (Wright, 1997)).

The high level idea. When we deal with a primal-dual system of linear programming, from the *Karush–Kuhn–Tucker* theory (Wright, 1997), we can update the primal and dual variables step by step using a Newton-like method to the Lagrangian of LP. In particular, the algorithm takes a logarithm barrier function, which is determined by a coefficient $\mu_+ \geq 0$, to measure how close the current solution is to that of the original problem described in WBC-LP.

Let $\mathbf{y} \in \mathbb{R}^{n_c - N}$ and $\mathbf{s} \in \mathbb{R}^{n_v}$ be the dual variables corresponding to the constraints “ $\bar{A}\mathbf{x} = \bar{\mathbf{b}}$ ” and “ $\mathbf{x} \geq 0$ ”, respectively. Writing in matrix form, the search direction at the current primal-dual tuple $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ should be the solution of the following nonlinear system of equations:

$$\begin{bmatrix} 0 & \bar{A}^\top & I_{n_v} \\ \bar{A} & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{bmatrix} = - \begin{bmatrix} \bar{A}^\top \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \bar{A} \mathbf{x} - \bar{\mathbf{b}} \\ X \mathbf{s} + \Delta X \Delta \mathbf{s} + \mu_+ \mathbf{1}_{n_v} \end{bmatrix} \quad (6)$$

where the notation “ Δ ” denotes the change when updating some variable or matrix. For the vector \mathbf{x} , in Eq. (6) we use the capitalized letter “ X ” to denote the diagonal matrix with $X_{ii} = x_i$, where x_i is the i -th term of \mathbf{x} . Similarly, we also have the diagonal matrix S for the vector \mathbf{s} in (6).

Because of the quadratic term “ $\Delta X \Delta \mathbf{s}$ ” on the right hand-side (RHS), Eq. (6) does not have an explicit solution. To reduce this nonlinear system to be a linear case, first we obtain a **predictor step** by removing the “ $\Delta X \Delta \mathbf{s} + \mu_+ \mathbf{1}_{n_v}$ ” term on the RHS of Eq. (6), then compute the **corrector step** by assigning the solution of predictor step to the quadratic term in the RHS of Eq. (6). Through combining them together, we obtain the final update step, hence the name “predictor-corrector”. The value μ_+ is updated after each loop, descending as the primal-dual gap narrows. When μ_+ approaches to 0, the current position converges to the optimal solution of WBC-LP.

Built upon the above idea, both the predictor and corrector steps can be obtained by sequentially computing $\Delta \mathbf{y}$, $\Delta \mathbf{s}$ and $\Delta \mathbf{x}$, where the complete details are shown in Algorithm 2 of Appendix C. Here, we only focus on the most time consuming part, computing $\Delta \mathbf{y}$, as the solution of **normal equation** (Wright, 1997). Namely, for both the predictor and corrector steps, we need to solve a system of linear equations as the following form:

$$(\bar{A}R\bar{A}^\top)\Delta \mathbf{y} = \mathbf{f}, \quad (7)$$

where $R = \text{diag}(\mathbf{s})^{-1}X$, and the symbol “ \mathbf{f} ” represents the vector obtained in either the predictor or corrector steps, as shown in Eqs. (13) and (14) of Algorithm 2 (here, we can temporarily ignore the detailed form of \mathbf{f} since it does not affect designing our algorithm for solving Eq. (7)).

Other parts in one loop of Algorithm 2 are just simple operations like matrix-vector multiplications, and thus the whole time complexity is dominated by the cost for solving Eq. (7). We can solve Eq. (7) by inverting the matrix applied to $\Delta \mathbf{y}$ (this requires \bar{A} to be of full row-rank, and this is also the reason why we need Theorem 3.1). However, ordinary inverse to such a large matrix will take excessively high complexity, as shown in Remark 3.3 later. This motivates us to propose a new method to further improve the complexity in next section.

3.3. Solving the Normal Equation (7) Efficiently

We introduce an efficient algorithm for solving the normal equation $(\bar{A}R\bar{A}^\top)\Delta \mathbf{y} = \mathbf{f}$ of Eq. (7), whose complexity is summarized in the following theorem.

Theorem 3.2. In each inner iteration of IPM, the time complexity in terms of flops is $O(m^2 \sum_{i=1}^N m_i + Nm^3 + N^2m^2 + N^3)$, and the memory usage in terms of doubles is $O(m \sum_{i=1}^N m_i + Nm^2 + N^2)$.

Remark 3.3. For comparison, a vanilla IPM takes $O((Nm + \sum_{i=1}^N m_i + N)^2(m \sum_{i=1}^N m_i + m + N))$ time and $O((Nm + \sum_{i=1}^N m_i + N)(m \sum_{i=1}^N m_i + m + N))$ storage.

Roadmap of the proof for Theorem 3.2. It is worth noting that we do not need to really compute the matrix $(\bar{A}R\bar{A}^\top)^{-1}$

for solving Eq. (7); instead, we only need to find an efficient way to simplify the structure of $\bar{A}\bar{R}\bar{A}^\top$, so that we can easily compute the matrix-vector multiplication “ $(\bar{A}\bar{R}\bar{A}^\top)^{-1}\mathbf{f}$ ”. Proposition 3.4 illustrates the detailed structure of $\bar{A}\bar{R}\bar{A}^\top$, which paves the way for designing the following transformations. Our method contains three steps. First, we transform $\bar{A}\bar{R}\bar{A}^\top$ to be a nearly block-diagonal matrix. Then, we resolve the most challenging part that hinders it from being block-diagonal, as shown in Theorem 3.6. Finally, we obtain the desired block-diagonal form for $\bar{A}\bar{R}\bar{A}^\top$. Namely, to compute the product $(\bar{A}\bar{R}\bar{A}^\top)^{-1}\mathbf{f}$, we only need to care about the inverses of several small matrices with sizes no bigger than $\max\{m_i, m, N\}$. As a consequence, we save the massive time and storage for inverting a matrix of size $O(Nm + N \max_i(m_i))$.

Let \mathbf{r} be the n_v -dimensional vector with its i -th entry $r_i = R_{ii}$ (recall that n_v denotes the number of variables as defined in Section 3.1). Also recall that $M = \sum_{i=1}^N m_i$, and we further define $M' = N(m-1)$. Let $\mathbf{z} = \mathbf{r}(Mm+2 : Mm+m)$, the sub-vector of \mathbf{r} including from the $(Mm+2)$ -th to $(Mm+m)$ -th entries.

Proposition 3.4. $\bar{A}\bar{R}\bar{A}^\top$ has the following form:

$$\bar{A}\bar{R}\bar{A}^\top = \begin{bmatrix} B_1 & B_2 & \mathbf{0} & K_1 \\ B_2^\top & B_3 + B_4 & \boldsymbol{\alpha} & K_2 \\ \mathbf{0} & \boldsymbol{\alpha}^\top & l & 0 \\ K_1^\top & K_2^\top & 0 & W \end{bmatrix}$$

where the sub-matrices are defined as follows (their illustrations are shown in Section D):

- $B_1 \in \mathbb{R}^{M \times M}$ is a diagonal matrix with positive diagonal entries; $B_2 \in \mathbb{R}^{M \times M'}$ is a block-diagonal matrix with N blocks, where each i -th block is of size $m_i \times (m-1)$; $B_3 \in \mathbb{R}^{M' \times M'}$ is a diagonal matrix with positive diagonal entries; $B_4 = (\mathbf{1}_N \mathbf{1}_N^\top) \otimes \text{diag}(\mathbf{z})$;
- $\boldsymbol{\alpha} = -\mathbf{1}_N \otimes \mathbf{z}$; $l = \mathbf{1}_m^\top \mathbf{r}(n_v - m + 1 : n_v - N)$;
- $K_1 \in \mathbb{R}^{M \times N}$ is a block-diagonal matrix with N blocks, with the i -th block being of size $m_i \times 1$; $K_2 \in \mathbb{R}^{M' \times N}$ is a block-diagonal matrix with N blocks, with the i -th block being of size $(m-1) \times 1$;
- $W = W_1 + r_{n_v} \mathbf{1}_N \mathbf{1}_N^\top$, where r_{n_v} is the n_v -th term of \mathbf{r} , and $W_1 \in \mathbb{R}^{N \times N}$ is a diagonal matrix with positive diagonal entries.

We left the proof for Proposition 3.4 to Appendix D. Then, we illustrate our idea for efficiently solving the normal equation (7) via three steps.

Step 1: transforming $\bar{A}\bar{R}\bar{A}^\top$ to be nearly block-diagonal. We simplify the coefficient matrix $\bar{A}\bar{R}\bar{A}^\top$ by performing

several transformations. Relying on Proposition 3.4 we design the transformation matrix $Q :=$

$$Q := \begin{bmatrix} I_M & & & \\ & I_{M'} & & \\ & & 1 & \\ & & & I_N \end{bmatrix} \cdot \begin{bmatrix} I_M & & & \\ & -B_2^\top B_1^{-1} & & \\ & & 1 & \\ & & & I_N \end{bmatrix}, \text{ such that}$$

$$Q \cdot \bar{A}\bar{R}\bar{A}^\top \cdot Q^\top = \begin{bmatrix} B_1 & & & \\ & G_1 + G_2 & & \bar{K}_2 \\ & & l & \\ & & & \bar{W} \end{bmatrix}, \quad (8)$$

where $G_1 := B_3 - B_2^\top B_1^{-1} B_2$, $G_2 := B_4 - \frac{1}{l} \boldsymbol{\alpha} \boldsymbol{\alpha}^\top$, $\bar{W} = W - K_1^\top B_1^{-1} K_1$. See Section E for computation.

Step 2: dealing with $(G_1 + G_2)^{-1}$. In the RHS of Eq. (8), we can see that the transformed matrix is still not block-diagonal due to the existence of “ \bar{K}_2 ”. To further eliminate \bar{K}_2 , we need to design another transformation matrix that involves the computation of $(G_1 + G_2)^{-1}$, the inverse of a large matrix of size $M' \times M'$. To find a way to compute $(G_1 + G_2)^{-1}$ more efficiently, we have a useful observation on the matrices G_1 and G_2 , which indicates the specificity of their structures (the proof of Proposition 3.5 is placed in Section F).

Proposition 3.5. There exist positive definite matrices $Y, G_{ii} \in \mathbb{R}^{(m-1) \times (m-1)}$, $i = 1, \dots, N$, such that $G_1 = \text{diag}(G_{11}, \dots, G_{NN})$ and $G_2 = (\mathbf{1}_N \mathbf{1}_N^\top) \otimes Y$.

With the above observation, we achieve the following key lemma.

Lemma 3.6. Let $\bar{Z} = \text{diag}(\mathbf{z}) - \frac{1}{l} \mathbf{z} \mathbf{z}^\top$ (\mathbf{z} is defined before Theorem 3.4), and $\tilde{G} = (\mathbf{1}_N \mathbf{1}_N^\top) \otimes (\bar{Z}^{-1} + \sum_{i=1}^N G_{ii}^{-1})^{-1}$, then we have

$$(G_1 + G_2)^{-1} = G_1^{-1} - G_1^{-1} \tilde{G} G_1^{-1}. \quad (9)$$

Further, the flops needed for applying a vector to the RHS of Eq. (9) is $O(Nm^2)$.

Proof. Since Y is positive definite, there should exist a matrix $U \in \mathbb{R}^{(m-1) \times (m-1)}$, such that $Y = U^\top U$. Then, from Proposition 3.5 we have $G_2 = (\mathbf{1}_N \mathbf{1}_N^\top) \otimes Y = (\mathbf{1}_N \otimes U^\top)(\mathbf{1}_N^\top \otimes U)$, and it implies

$$(G_1 + G_2)^{-1} = (G_1 + (\mathbf{1}_N \otimes U^\top) \cdot (\mathbf{1}_N^\top \otimes U))^{-1} \\ = G_1^{-1} - G_1^{-1} \cdot (\mathbf{1}_N \otimes U^\top).$$

$$(I_{m-1} + (\mathbf{1}_N^\top \otimes U) G_1^{-1} (\mathbf{1}_N \otimes U^\top))^{-1} \cdot (\mathbf{1}_N^\top \otimes U) \cdot G_1^{-1},$$

where the second equality is obtained by *Woodbury matrix identity* (Hager, 1989), a commonly used equation in linear algebra (see Section H). Noticing that both G_1 and $\mathbf{1}_N^\top \otimes U$ are divided into $N \times N$ sub-matrices of size

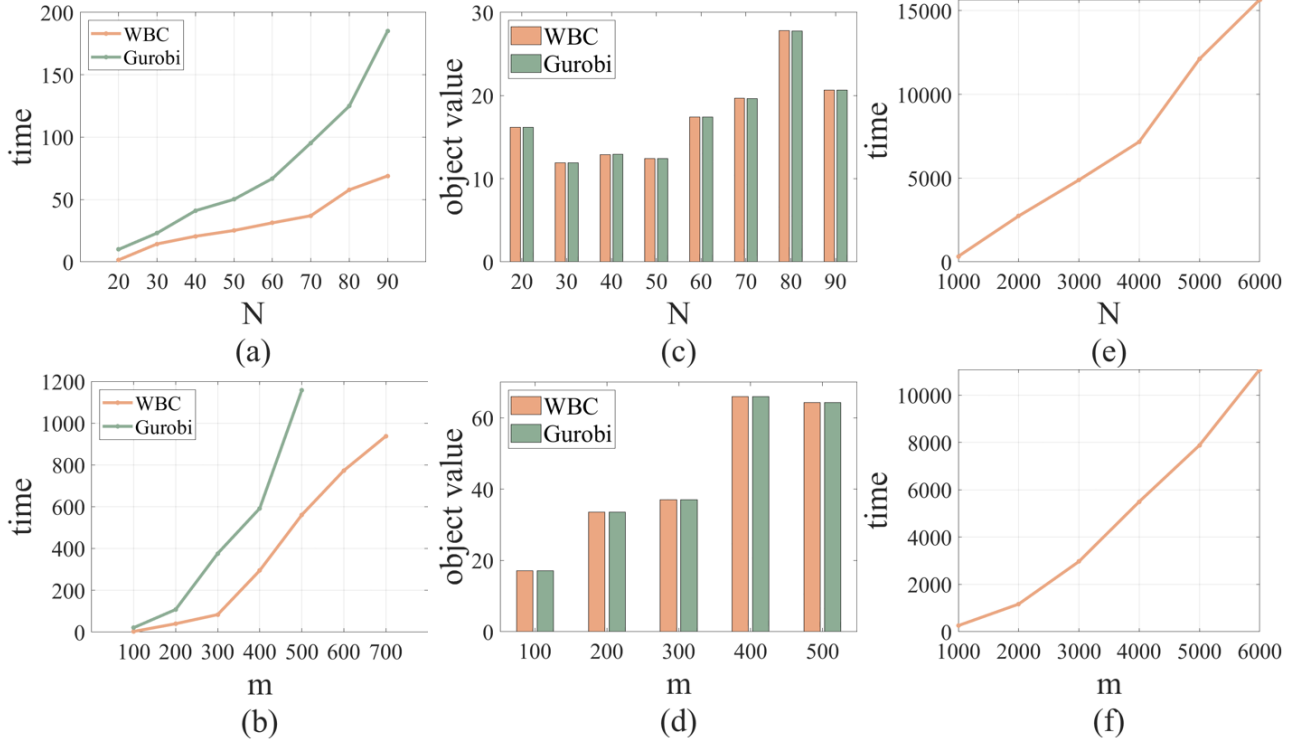


Figure 2. The first two column figures are the computational time (s) and obtained objective value of Gurobi and our method. For (a) and (c), we fix $m = 100$ and vary N . For (b) and (d), we fix $N = 30$ and vary m . The third column figures are the computational times of our method when the problem scale is very large. For (e), we fix $m = 50$. For (f), we fix $N = 10$.

$(m-1) \times (m-1)$, together with Proposition 3.5, we can simplify the RHS of the above equality to be

$$\begin{aligned} & G_1^{-1} - G_1^{-1} \left((\mathbf{1}_N \mathbf{1}_N^\top) \otimes \right. \\ & \quad \left. (U^\top (I_{m-1} + \sum_{i=1}^N U G_{ii}^{-1} U^\top)^{-1} U) \right) G_1^{-1} \\ & = G_1^{-1} - G_1^{-1} \left((\mathbf{1}_N \mathbf{1}_N^\top) \otimes (Y^{-1} + \sum_{i=1}^N G_{ii}^{-1})^{-1} \right) G_1^{-1}. \end{aligned}$$

Therefore, we obtain the equality (9).

As for the complexity, we notice the following two facts. **(1)** G_1^{-1} is a block diagonal matrix with only $N(m-1)^2$ non-zero terms, thus the operation of multiplying a vector to it takes no more than $O(Nm^2)$ time. **(2)** The matrix \tilde{G} can be viewed as an $N \times N$ matrix with every term being $(Z^{-1} + \sum_{i=1}^N G_{ii}^{-1})^{-1}$. That is, we only have to store $(Z^{-1} + \sum_{i=1}^N G_{ii}^{-1})^{-1}$, a square matrix of size $m-1$ instead of \tilde{G} with size $N(m-1)$. Immediately, if we multiply a vector on the right of \tilde{G} , the i -th coordinate of the resulting vector would be the same as the $(i+m-1)$ -th coordinate for any $i \leq M' - (m-1)$. Therefore we only have to compute

the first $m-1$ rows, then copy them N times down to obtain the multiplication. This part also takes $O(Nm^2)$ time. \square

Step 3: from nearly block-diagonal to block-diagonal, and achieve the final solution $\Delta \mathbf{y}$. We design another transformation matrix Q' to eliminate \bar{K}_2 in the nearly block diagonal matrix obtained in (8). Specifically, we let $Q' =$

$$\begin{bmatrix} I_{M'} & & & \\ & I_{M'} & & \\ & & 1 & \\ & -\bar{K}_2^\top (G_1 + G_2)^{-1} & & I_N \end{bmatrix}. \text{ Through combining with the transformation } Q \text{ introduced in (8), we have}$$

$$Q' \cdot Q \bar{A} \bar{R} \bar{A}^\top Q'^\top \cdot Q'^\top = \text{diag}(B_1, G_1 + G_2, l, \tilde{W}), \quad (10)$$

where $\tilde{W} = \bar{W} - \bar{K}_2^\top (G_1 + G_2)^{-1} \bar{K}_2$. Based on Lemma 3.6, the construction of \tilde{W} can be done in $O(N^2 m^2)$ flops, since \bar{K}_2 has N columns.

Now, we can solve the normal equation $(\bar{A} \bar{R} \bar{A}^\top) \Delta \mathbf{y} = \mathbf{f}$ of Eq. (7). Set $\tilde{\mathbf{y}}$ as the solution to

$$\text{diag}(B_1, G_1 + G_2, l, \tilde{W}) \tilde{\mathbf{y}} = Q' Q \mathbf{f}, \quad (11)$$

where the left coefficient matrix is from the obtained block-diagonal matrix in (10). Similar with the method for computing \tilde{W} , due to the term “ $G_1 + G_2$ ” in the coefficient matrix

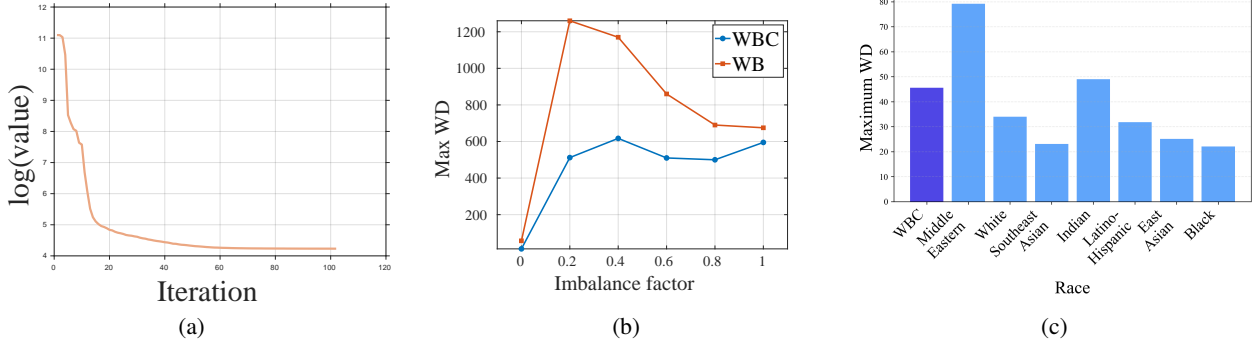


Figure 3. (a) $N = 90, m = 2000$. The objective value of our algorithm becomes stable after 80 iterations. For other settings of m and N , see Section J. (b) The maximum Wasserstein distance between the barycenter (WBC or WB) and input distributions, with fixing $N = 20, m = 500$. (c) The dark bin marked “WBC” is the maximum WD between WBC and all the faces, while the rest represent the maximum WD from WB to the faces in different races.

of $\bar{\mathbf{y}}$, we also need to apply Theorem 3.6 to obtain $\bar{\mathbf{y}}$. After that, from Eq. (10) it is easy to verify that $\Delta\mathbf{y} = Q^\top Q'^\top \bar{\mathbf{y}}$.

Algorithm 1 summarizes the above steps. Due to the space limit, we leave the detailed proof for the overall complexities to Section G.

Algorithm 1 Solver for $(\bar{A}R\bar{A}^\top)\Delta\mathbf{y} = \mathbf{f}$

Input: $R \in \mathbb{R}^{n_v \times n_v}, \mathbf{f} \in \mathbb{R}^{n_c - N}$ as described in Eq. (7)

Output: The solution $\Delta\mathbf{y}$

Step 1: Compute blocks of $\bar{A}R\bar{A}^\top$

- 1: Compute $B_1, B_2, B_3, K_1, K_2, W$

Step 2: Simplify the constraint matrix into block diagonal

- 1: Compute Q
- 2: $\bar{K}_2 \leftarrow K_2 - B_2^\top B_1^{-1} K_1$
- 3: $\bar{W} \leftarrow W - K_1^\top B_1^{-1} K_1$ //Eliminate K_1
- 4: Compute G_1, G_2
- 5: Decompose $(G_1 + G_2)^{-1}$ according to Lemma 3.6
- 6: Compute Q'
- 7: $\bar{W} \leftarrow \bar{W} - \bar{K}_2^\top (G_1 + G_2)^{-1} \bar{K}_2$ //Eliminate \bar{K}_2

Step 3: Solve $\Delta\mathbf{y}$

- 1: $\mathbf{z}_1 \leftarrow Q\mathbf{f}, \mathbf{z}_2 \leftarrow Q'\mathbf{z}_1$
- 2: $\mathbf{z}_3(1 : M) \leftarrow B_1^{-1}\mathbf{z}_2(1 : M)$
- 3: $\mathbf{z}_3(n_c - N + 1 : n_c) \leftarrow \bar{W}^{-1}\mathbf{z}_2(n_c - N + 1 : n_c)$
- 4: $\mathbf{z}_3(n_c - N) \leftarrow c^{-1}\mathbf{z}_2(n_c - N)$
- 5: Solve $(G_1 + G_2)\mathbf{z}_4(M + 1 : n_c - N - 1) = \mathbf{z}_3(M + 1 : n_c - N - 1)$
- 6: $\mathbf{z}_5 \leftarrow Q'^\top \mathbf{z}_4, \Delta\mathbf{y} \leftarrow Q^\top \mathbf{z}_5$

Return: $\Delta\mathbf{y}$

4. Experiments

We conduct four experiments to investigate the real performance of our algorithm. (1) The first experiment demonstrates our advantages on computational speed and memory

usage over the commercial solver Gurobi, a powerful optimization solver widely used across various fields such as operations research, finance, and data science. The baseline we choose is *Gurobi Optimizer version 11.0.0* (academic license). (2) The second experiment reflects the fairness of WBC over the standard WB. For these two experiments, we generate random datasets in Euclidean space, and the weights of $(q_1^{(t)}, \dots, q_m^{(t)})$ in each distribution $\mathcal{P}^{(t)}$ are generated uniformly at random. (3) The third experiment further illustrates the performance on a real-world dataset FairFace (Karkkainen & Joo, 2021) with diverse racial representation. We choose 700 (100 for each race) images including seven racial groups of “Black”, “East Asian”, “Indian”, “Latino-Hispanic”, “Middle Eastern”, “Southeast Asian” and “White” as 700 distributions, where each image actually can be regarded as a distribution of size 50×50 in \mathbb{R}^2 . (4) The fourth experiment on 3D point-cloud gives a visual example to the fairness of WBC as a barycenter. All the experiments are implemented on a workstation, Intel(R) Core(TM) i5-9400 CPU @ 2.90GHz and 8GB for RAM, equipped with win64 - Windows 11+.0.

(1) **Comparison with Gurobi.** We set m of all distributions to be equal for brevity. As shown in Fig. 2, our algorithm is always faster than Gurobi, and the gap between the two methods is expanding as the scale increases. Moreover, Gurobi cannot solve the instance with $m > 500$ due to memory limitation, which showcases the advantage of our space complexity in Theorem 3.2. We also illustrate the convergence speed of our algorithm. From Fig. 3(a) (and Figs. 5 to 8 in appendix), we can see that our algorithm converges in a super-linear rate, which is consistent with the theoretical result of (Ye et al., 1993).

(2) **Fairness of WBC.** To evaluate the performance on fairness, we first compare the maximum Wasserstein distance (Max WD) between the obtained centers and input distri-

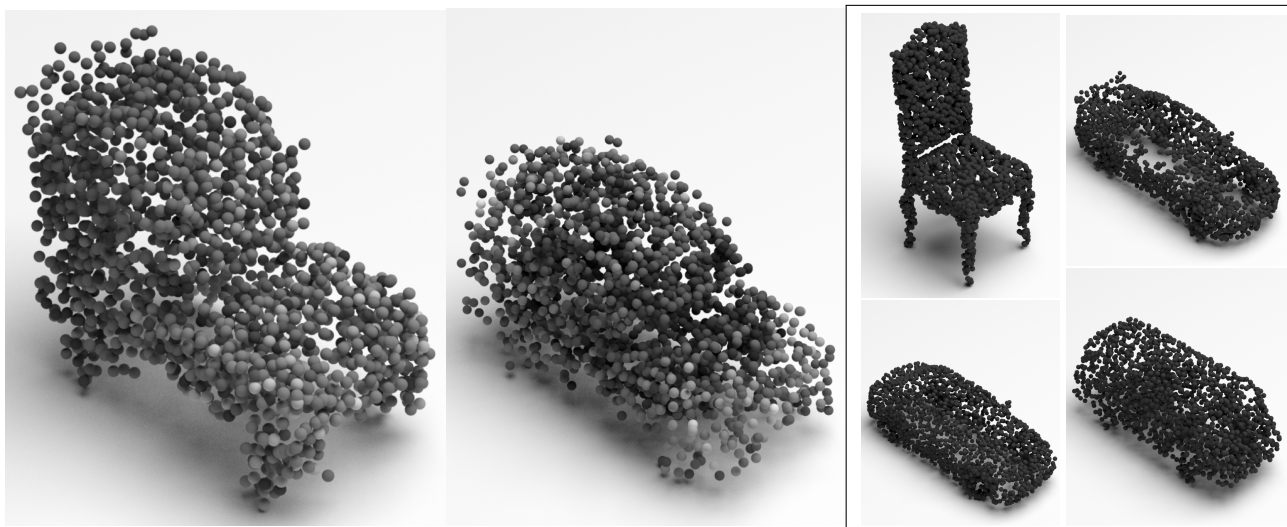


Figure 4. WBC (left) and WB (middle) of 3 cars and one chair. WB looks just like a car while our proposed WBC preserves characteristics of a chair, such as backrest and legs, indicating that WBC indeed places greater emphasis on fairness for the minority than WB.

butions. The experimental distributions are partitioned into two families, with each family’s measure concentrated in distinct regions that are quite distant from each other. To quantify the disparity between these two families, we define the *imbalanced factor* (“imf”) as the ratio of the smaller family’s size to the larger one. For instance, “imf = 1” indicates that both families contain an equal number of distributions. As is shown in Fig. 3(b), the smaller the imf, the bigger the difference of the Max WD between WB and WBC (except for the case “imf=0”, since the smaller family is empty in this case), indicating WBC’s potential as the alternative to WB when fairness is concerned. Please see Section J.2 for the details of this part.

Another fairness measurement is the variance of Wasserstein distances from the obtained center to the input distributions: $\text{var}(W(\bar{\mu}, \mu_i)) := \frac{1}{n-1} \sum_1^n (W(\bar{\mu}, \mu_i) - \bar{W})^2$, denoted as $\text{var}(\text{WD})$, where $\bar{W} = \frac{1}{n} \sum_{i=1}^n W(\bar{\mu}, \mu_i)$. Smaller $\text{var}(\text{WD})$ indicates reduced representative disparity among all inputs. Across all 2,700 experiments (detailed in Section J.2), WBC consistently achieves lower $\text{var}(\text{WD})$ compared to WB. For example, in L_1 space of dimension 100, we sample the 30 supports uniformly in a cube with side length 3, then translate 3 supports by adding $-\mathbf{1}_{100}$, and translate 5 supports by adding $2 * \mathbf{1}_{100}$, we have $\text{var}(\text{WD})=58.35$ for WB, while $\text{var}(\text{WD})=418.61$ for WBC. Those observations suggest that WBC not only improves fairness for the minority distributions, but also could enhance fairness across all distributions by limiting disparities between transport plans.

(3) Experiments on FairFace Dataset. From Fig. 3(c), we can observe that the standard WB has a significant gap between the object values in different races. The object value of WB corresponding to “Middle Eastern” is 78.30,

which is far greater than the object value of our algorithm (45.37). Meanwhile, our algorithm achieves the object value only slightly higher than the average of WD to WB (38.70).

(4) 3D Point-cloud Averaging. WBC provides a means for computing the average shape of point-cloud data by casting a point cloud into an empirical probability measure. We take the ShapeNet Core-55 dataset (Chang et al., 2015), with each cloud containing 2048 points in \mathbb{R}^3 . As an example, in Fig. 4 we show that compared with WB, the WBC preserves more characteristics of the “chair” (which can be regarded as an outlier distribution from the other “car” clouds).

5. Conclusion

In this paper, we present an efficient algorithm to compute Wasserstein ball center, an alternative of Wasserstein barycenter with emphasis on minority distributions, which makes it more suitable for the tasks that are sensitive to fairness. For future work, it would be interesting to explore further properties and applications of WBC. Moreover, it is also deserved to study other models for fair ensembling different distributions in the field of machine learning.

Acknowledgements

We want to thank the anonymous reviewers for their constructive comments. This work was supported in part by the National Key Research and Development Program of China (NO.2021YFA1000900), the National Natural Science Foundation of China (NO.62272432, NO.62432016) and the Natural Science Foundation of Anhui Province (NO.2208085MF163).

Impact Statement

The proposed algorithm provides an effective alternative to the Wasserstein barycenter by offering a trade-off between transport efficiency and fairness. However, practitioners should be cautious of potential outliers arising from noise or data errors. To ensure robust performance, data cleansing or selection techniques should be applied prior to implementing the WBC.

References

- Aayushman, Gaddey, H., Mittal, V., Chawla, M., and Gupta, G. R. Fair and accurate skin disease image classification by alignment with clinical labels. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 394–404. Springer, 2024.
- Abbasi, M., Bhaskara, A., and Venkatasubramanian, S. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 504–514, 2021.
- Adegun, A. and Viriri, S. Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art. *Artificial Intelligence Review*, 54(2): 811–841, 2021.
- Agarwal, A., Dudík, M., and Wu, Z. S. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019.
- Agarwal, P. K., Raghvendra, S., Shirzadian, P., and Yao, K. Efficient approximation algorithm for computing wasserstein barycenter under euclidean metric. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4809–4826. SIAM, 2025.
- Ahuja, R. K., Orlin, J. B., Stein, C., and Tarjan, R. E. Improved algorithms for bipartite network flow. *SIAM Journal on Computing*, 23(5):906–933, 1994.
- Altschuler, J., Niles-Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30, 2017.
- Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. Massively scalable sinkhorn distances via the nyström method. *Advances in neural information processing systems*, 32, 2019.
- Altschuler, J. M. and Boix-Adsera, E. Wasserstein barycenters can be computed in polynomial time in fixed dimension. *Journal of Machine Learning Research*, 22(44): 1–19, 2021.
- Altschuler, J. M. and Boix-Adsera, E. Wasserstein barycenters are np-hard to compute. *SIAM Journal on Mathematics of Data Science*, 4(1):179–203, 2022.
- Backhoff-Veraguas, J., Fontbona, J., Rios, G., and Tobar, F. Bayesian learning with wasserstein barycenters. *ESAIM: Probability and Statistics*, 26:436–472, 2022.
- Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. Scalable fair clustering. In *International Conference on Machine Learning*, pp. 405–413. PMLR, 2019.
- Benamou, J.-D., Froese, B. D., and Oberman, A. M. Numerical solution of the optimal transportation problem using the monge–ampère equation. *Journal of Computational Physics*, 260:107–126, 2014.
- Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.
- Bhardwaj, A. and Rege, P. P. Skin lesion classification using deep learning. In *Advances in Signal and Data Processing: Select Proceedings of ICSDP 2019*, pp. 575–589. Springer, 2021.
- Borgwardt, S. An lp-based, strongly-polynomial 2-approximation algorithm for sparse wasserstein barycenters. *Operational Research*, 22(2):1511–1551, 2022.
- Carlier, G. and Ekeland, I. Matching for teams. *Economic theory*, 42:397–418, 2010.
- Caton, S. and Haas, C. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7):1–38, 2024.
- Chai, J. and Wang, X. Fairness with adaptive weights. In *International Conference on Machine Learning*, pp. 2853–2866. PMLR, 2022.
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- Chen, X., Fain, B., Lyu, L., and Munagala, K. Proportionally fair clustering. In *International Conference on Machine Learning*, pp. 1032–1041. PMLR, 2019.
- Chen, Z., Kuhn, D., and Wiesemann, W. Data-driven chance constrained programs over wasserstein balls. *Operations Research*, 72(1):410–424, 2024.
- Cheng, K., Aeron, S., Hughes, M. C., and Miller, E. L. Dynamical wasserstein barycenters for time-series modeling. *Advances in Neural Information Processing Systems*, 34: 27991–28003, 2021.

- Chiappori, P.-A. *Matching with transfers: The economics of love and marriage*. Princeton University Press, 2017.
- Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.
- Cohen, M. B., Lee, Y. T., and Song, Z. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021.
- Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.
- Cuturi, M. and Doucet, A. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pp. 685–693. PMLR, 2014.
- Cuturi, M. and Peyré, G. Semidual regularized optimal transport. *SIAM Review*, 60(4):941–965, 2018.
- Dikin, I. Iterative solution of problems of linear and quadratic programming. In *Doklady Akademii Nauk*, volume 174, pp. 747–748. Russian Academy of Sciences, 1967.
- Dognin, P., Melnyk, I., Mroueh, Y., Ross, J., Santos, C. D., and Sercu, T. Wasserstein barycenter model ensembling. In *International Conference on Learning Representations*, 2019.
- Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.
- Fox, E. and Lu, J. A deterministic near-linear time approximation scheme for geometric transportation. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1301–1315. IEEE, 2023.
- Ge, D., Wang, H., Xiong, Z., and Ye, Y. Interior-point methods strike back: Solving the wasserstein barycenter problem. *Advances in neural information processing systems*, 32, 2019.
- Ghadiri, M., Samadi, S., and Vempala, S. Socially fair k -means clustering. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 438–448, 2021.
- Goldfeld, Z., Kato, K., Rioux, G., and Sadhu, R. Statistical inference with regularized optimal transport. *Information and Inference: A Journal of the IMA*, 13(1):iaad056, 2024.
- Gondzio, J. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.
- Gong, J. and Kim, H. Rhsboost: Improving classification performance in imbalance data. *Computational Statistics & Data Analysis*, 111:1–13, 2017.
- Gramfort, A., Peyré, G., and Cuturi, M. Fast optimal transport averaging of neuroimaging data. In *Information Processing in Medical Imaging: 24th International Conference, IPMI 2015, Sabhal Mor Ostaig, Isle of Skye, UK, June 28–July 3, 2015, Proceedings 24*, pp. 261–272. Springer, 2015.
- Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL <https://www.gurobi.com>.
- Hager, W. W. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.
- Huang, M., Ma, S., and Lai, L. Projection robust wasserstein barycenters. In *International Conference on Machine Learning*, pp. 4456–4465. PMLR, 2021.
- Jang, T., Zheng, F., and Wang, X. Constructing a fair classifier with generated fair data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7908–7916, 2021.
- Jiang, H. and Nachum, O. Identifying and correcting label bias in machine learning. In *International conference on artificial intelligence and statistics*, pp. 702–712. PMLR, 2020.
- Jiang, S., Song, Z., Weinstein, O., and Zhang, H. A faster algorithm for solving general lps. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 823–832, 2021.
- Joseph, M., Kearns, M., Morgenstern, J. H., and Roth, A. Fairness in learning: Classic and contextual bandits. *Advances in neural information processing systems*, 29, 2016.
- Karkkainen, K. and Joo, J. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1548–1558, 2021.
- Karmarkar, N. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, 1984.
- Khesin, A. B., Nikolov, A., and Paramonov, D. Preconditioning for the geometric transportation problem. *Journal of Computational Geometry*, 11(2):234–259, 2021.
- Kim, K., Zhou, B., Zhu, C., and Chen, X. Sobolev gradient ascent for optimal transport: Barycenter optimization and

- convergence analysis. *arXiv preprint arXiv:2505.13660*, 2025.
- Koehl, P., Delarue, M., and Orland, H. Statistical physics approach to the optimal transport problem. *Physical review letters*, 123(4):040603, 2019.
- Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.
- Krasanakis, E., Spyromitros-Xioufis, E., Papadopoulos, S., and Kompatsiaris, Y. Adaptive sensitive reweighting to mitigate bias in fairness-aware classification. In *Proceedings of the 2018 world wide web conference*, pp. 853–862, 2018.
- Kroshnin, A., Tupitsa, N., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., and Uribe, C. On the complexity of approximating wasserstein barycenters. In *International conference on machine learning*, pp. 3530–3540. PMLR, 2019.
- Kroshnin, A., Spokoiny, V., and Suvorikova, A. Statistical inference for bures–wasserstein barycenters. *The Annals of Applied Probability*, 31(3):1264–1298, 2021.
- Lahn, N., Raghvendra, S., and Zhang, K. A combinatorial algorithm for approximating the optimal transport in the parallel and mpc settings. *Advances in Neural Information Processing Systems*, 36:21675–21686, 2023.
- Lee, Y. T. and Sidford, A. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 424–433. IEEE, 2014.
- Lin, L., Shi, W., Ye, J., and Li, J. Multisource single-cell data integration by maw barycenter for gaussian mixture models. *Biometrics*, 79(2):866–877, 2023.
- Lin, T., Ho, N., and Jordan, M. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning*, pp. 3982–3991. PMLR, 2019.
- Lin, T., Ho, N., Chen, X., Cuturi, M., and Jordan, M. Fixed-support wasserstein barycenters: Computational hardness and fast algorithm. *Advances in neural information processing systems*, 33:5368–5380, 2020.
- Lohaus, M., Perrot, M., and Von Luxburg, U. Too relaxed to be fair. In *International Conference on Machine Learning*, pp. 6360–6369. PMLR, 2020.
- Makarychev, Y. and Vakilian, A. Approximation algorithms for socially fair clustering. In *Conference on Learning Theory*, pp. 3246–3264. PMLR, 2021.
- Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International conference on machine learning*, pp. 6755–6764. PMLR, 2020.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.
- Mehrotra, S. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4): 575–601, 1992.
- Mizuno, S., Todd, M. J., and Ye, Y. On adaptive-step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations research*, 18(4):964–981, 1993.
- Montoya, L. N., Roberts, J. S., and Hidalgo, B. S. Towards fairness in ai for melanoma detection: Systemic review and recommendations. In *Future of Information and Communication Conference*, pp. 320–341. Springer, 2025.
- Nadeem, S., Hollmann, T., and Tannenbaum, A. Multi-marginal wasserstein barycenter for stain normalization and augmentation. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*, pp. 362–371. Springer, 2020.
- Nietert, S., Goldfeld, Z., Sadhu, R., and Kato, K. Statistical, robustness, and computational guarantees for sliced wasserstein distances. *Advances in Neural Information Processing Systems*, 35:28179–28193, 2022.
- Orlin, J. A faster strongly polynomial minimum cost flow algorithm. In *Proceedings of the Twentieth annual ACM symposium on Theory of Computing*, pp. 377–387, 1988.
- Pele, O. and Werman, M. Fast and robust earth mover’s distances. In *2009 IEEE 12th international conference on computer vision*, pp. 460–467. IEEE, 2009.
- Pesenti, S. M. and Jaimungal, S. Portfolio optimization within a wasserstein ball. *SIAM Journal on Financial Mathematics*, 14(4):1175–1214, 2023.
- Qin, R., Li, M., and Ding, H. Solving soft clustering ensemble via k -sparse discrete wasserstein barycenter. *Advances in Neural Information Processing Systems*, 34: 900–913, 2021.

- Rüschendorf, L. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70 (1):117–129, 1985.
- Sherman, J. Generalized preconditioning and undirected minimum-cost flow. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 772–780. SIAM, 2017.
- Song, S., Mo, G., and Ding, H. Relax and merge: A simple yet effective framework for solving fair k -means and k -sparse wasserstein barycenter problems. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Vaskevicius, T. and Chizat, L. Computational guarantees for doubly entropic wasserstein barycenters. *Advances in Neural Information Processing Systems*, 36, 2024.
- Villani, C. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.
- Wang, F., Cheng, L., Guo, R., Liu, K., and Yu, P. S. Equal opportunity of coverage in fair regression. *Advances in Neural Information Processing Systems*, 36, 2024.
- Wright, S. J. *Primal-dual interior-point methods*. SIAM, 1997.
- Yang, Q. and Ding, H. Approximate algorithms for k -sparse wasserstein barycenter with outliers. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 5316–5325, 2024.
- Ye, J., Wu, P., Wang, J. Z., and Li, J. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9): 2317–2332, 2017.
- Ye, Y., Güler, O., Tapia, R. A., and Zhang, Y. A quadratically convergent $o(1)$ -iteration algorithm for linear programming. *Mathematical programming*, 59(1):151–162, 1993.
- Yue, M.-C., Kuhn, D., and Wiesemann, W. On linear optimization over wasserstein balls. *Mathematical Programming*, 195(1):1107–1122, 2022.
- Zhuang, Y., Chen, X., and Yang, Y. Wasserstein k -means for clustering probability distributions. *Advances in Neural Information Processing Systems*, 35:11382–11395, 2022.

A. Related Works

Wasserstein distance The Wasserstein distance, also known as the Earth Mover’s distance when $p = 1$, quantifies the disparity between two probability distributions, particularly when their supports are discrete sets. Computing the discrete Wasserstein metric reduces to solving a *minimum-cost maximum-flow* optimization problem (Ahuja et al., 1994; Orlin, 1988). Consider a bipartite graph with n vertices, m edges, and maximum edge weight U . Lee & Sidford (2014) designed an efficient general LP solver that can solve the minimum cost flow problem in $O(n^{2.5} \text{poly}(\log U))$. Lahn et al. (2023) designed a combinatorial algorithm that computes an ϵ -approximate transport plan in $O(\frac{n^2}{\epsilon^2})$ sequential time and $O(\frac{\log n}{\epsilon^2})$ parallel time. Recently, Fox & Lu (2023) presented the first deterministic algorithm that computes optimal transportation in Euclidean space with near-linear time.

Wasserstein distance is also a classic topic in machine learning (Rüschendorf, 1985; Pele & Werman, 2009; Dognin et al., 2019). By using matrix scaling technique, Cuturi (2013) introduced the “Sinkhorn Distance”, which incorporates an entropic regularization term to smooth the transportation problem, offering paralleled and significantly faster solutions than exact computation of the discrete Wasserstein distance. Following Cuturi’s work, recent years have seen the development of several improved Sinkhorn algorithms (Lin et al., 2019; Altschuler et al., 2019; Benamou et al., 2015; Altschuler et al., 2017). Another practical variant is “sliced Wasserstein distance”, which saves computation time by taking expectation of Wasserstein distance of the distributions projected from high dimension to one-dimensional space (Kolouri et al., 2019; Nietert et al., 2022).

Wasserstein barycenter. Different from Wasserstein distance, the computation for Wasserstein barycenter is an NP-hard problem (Altschuler & Boix-Adsera, 2022). Cuturi & Doucet (2014) showed that the computation for WB can be improved by using an entropic regularization, leading to a simple gradient-descent scheme that was later improved and generalized under the iterative Bregman projection (IBP) algorithm (Benamou et al., 2015). Further progress includes the semi-dual gradient descent (Cuturi & Peyré, 2018), accelerated primal-dual gradient descent (APDAGD) (Kim et al., 2025; Kroshnin et al., 2019), alternating direction method of multipliers (ADMM) (Ye et al., 2017), deterministic IBP (Lin et al., 2020), and the IPM algorithm MAAIPM (Ge et al., 2019). In Euclidean space, WB can be solved in polynomial time with fixed dimension (Altschuler & Boix-Adsera, 2021). Vaskevicius & Chizat (2024) offers the first non-asymptotic convergence guarantees for approximating Wasserstein barycenter between discrete point clouds in the free-support/grid-free setting. Agarwal et al. (2025) gives the first polynomial algorithm for the case where the cost is L_1 -norm. In applications, we sometimes need the support of WB to be sparse (Ye et al., 2017). Borgwardt (2022) proposed a strongly-polynomial 2-approximation algorithm for sparse Wasserstein barycenters. Yang & Ding (2024) investigated the case with outliers, connecting this problem with k-means clustering.

Interior Point Method. Dikin (1967) proposed the first interior point method, which was reinvented in Karmarkar (1984) which also proposed the first poly-time method for linear programming called “Karmarkar’s algorithm”. Since then IPM has attracted a great amount of attention, where one of the most successful IPM methods is the class of *primal-dual* approaches, which iteratively optimize the primal and the dual variable, until the dual gap converges to zero. In particular, Mehrotra’s *predictor-corrector* algorithm (Mehrotra, 1992) provides the basis for most implementations of this class of methods, which is also the type of IPM applied in this paper (the details of predictor-corrector IPM are presented in Section 3.2). The seminal Mizuno-Todd-Ye predictor-corrector method has quadratic convergence (Mizuno et al., 1993; Ye et al., 1993). For more details on IPM, we refer the reader to the survey paper (Gondzio, 2012). Recently, there are also some new studies on reducing the exponent of IPM complexity in theoretical computer science (Jiang et al., 2021; Cohen et al., 2021), which relies on a technique called “matrix maintenance” to reduce the update time of each iteration.

Fairness. The fairness issue has attracted a great amount of attention in computer science (Joseph et al., 2016; Mehrabi et al., 2021; Caton & Haas, 2024). The proposed solutions include adjusting labels from sensitive groups to reconstruct unbiased mapping (Dwork et al., 2012; Jiang & Nachum, 2020), and removing sensitive attributions (Krasanakis et al., 2018). As is mentioned before, the WBC problem is also related with socially fair clustering (Abbasi et al., 2021; Ghadiri et al., 2021). An $(e^{O(p)} \frac{\log \ell}{\log \log \ell})$ -approximation algorithm for socially fair clustering with the ℓ_p -objective is given by Makarychev & Vakilian (2021). Unfairness can result from the issue of representation bias, which arises due to insufficient amount of data in certain groups or subgroups (Lohaus et al., 2020; Chai & Wang, 2022). Existing methods include fair data generation (Jang et al., 2021), multi-objective optimization (Martinez et al., 2020) and boosting (Gong & Kim, 2017). There are other concepts on fairness studied in computer science, for example the problems of fair k-means (Chierichetti et al., 2017; Chen et al., 2019; Backurs et al., 2019; Song et al., 2025) and fair regression (Agarwal et al., 2019; Wang et al., 2024).

Then

$$L_2 L_1 A_0 = \begin{bmatrix} F_1 & & & & & \\ & F_2 & & & & \\ & & \ddots & & & \\ & & & F_N & & \\ G_1^{(1)} & & & & H^{(1)} & \\ & G_2^{(1)} & & & H^{(1)} & \\ & & \ddots & & \vdots & \\ & & & G_N^{(1)} & H^{(1)} & \\ & & & & \mathbf{1}_m^T & \end{bmatrix},$$

where $G_i^{(1)} = S_i G_i^{(0)} - S_i T_i F_i$, $H^{(1)} = \mathbf{e}_1 \mathbf{1}_m^T - S_i$. Note that elements in the first rows of $H^{(1)}$ and $G_i^{(1)}$, $i = 1, \dots, N$ are zeros. We have proved the claims in a).

b). \bar{A}_0 is obtained by removing the $(M+1)$ -th, $(M+m+1)$ -th, \dots , $(M+(N-1)m+1)$ -th rows of A_0 . That is,

$$\bar{A}_0 = \begin{bmatrix} F_1 & & & & & \\ & F_2 & & & & \\ & & \ddots & & & \\ & & & F_N & & \\ G_1^{(2)} & & & & H^{(2)} & \\ & G_2^{(2)} & & & H^{(2)} & \\ & & \ddots & & \vdots & \\ & & & G_N^{(2)} & H^{(2)} & \\ & & & & \mathbf{1}_m^T & \end{bmatrix}$$

where $G_i^{(2)} = G_i^{(1)}(2:m, :) = \mathbf{1}_{m_i}^T \otimes [\mathbf{0}_{m-1}, I_{m-1}]$, $H^{(2)} = H^{(1)}(2:m, :) = [\mathbf{0}_{m-1}, -I_{m-1}]$ and $F_i = I_{m_i} \otimes \mathbf{1}_m^T$. Let $n'_{row} = M + N(m-1) + 1$.

For $i = 1, \dots, N$, let

$$U_i = I_{m_i} \otimes \begin{bmatrix} 1 & -1 & \cdots & -1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{m \times m}, \quad U_{N+1} = \begin{bmatrix} 1 & -1 & \cdots & -1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{m \times m}, \quad R_1 = \begin{bmatrix} U_1 & & & \\ & U_2 & & \\ & & \ddots & \\ & & & U_{N+1} \end{bmatrix}$$

then

$$\bar{A}_0 R_1 = \begin{bmatrix} F_1^{(3)} & & & & & \\ & F_2^{(3)} & & & & \\ & & \ddots & & & \\ & & & F_N^{(3)} & & \\ G_1^{(3)} & & & & H^{(3)} & \\ & G_2^{(3)} & & & H^{(3)} & \\ & & \ddots & & \vdots & \\ & & & G_N^{(3)} & H^{(3)} & \\ & & & & \boldsymbol{\alpha}^T & \end{bmatrix}$$

where $F_i^{(3)} = F_i U_i = I_{m_i} \otimes [1, \mathbf{0}_{m-1}^T]$, $G_i^{(3)} = G_i^{(2)} U_i = G_i^{(2)} = \mathbf{1}_{m_i}^T \otimes [\mathbf{0}_{m-1}, I_{m-1}]$, $i = 1, \dots, N$, $H^{(3)} = H^{(2)} U_{N+1} = H^{(2)} = [\mathbf{0}_{m-1}, -I_{m-1}]$ and $\boldsymbol{\alpha}^T = \mathbf{1}_m^T U_{N+1} = [1, \mathbf{0}_{m-1}^T]$.

Let

$$\tilde{K} = \begin{bmatrix} 0 & & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \end{bmatrix}_{m \times m}, \quad K_i = \begin{bmatrix} I_m & \tilde{K} & \cdots & \tilde{K} \\ & I_m & & \\ & & \ddots & \\ & & & I_m \end{bmatrix}_{mm_i \times mm_i}, \quad R_2 = \begin{bmatrix} K_1 & & \\ & \ddots & \\ & & K_N \\ & & & I_m \end{bmatrix}$$

then

$$\bar{A}_0 R_1 R_2 = \begin{bmatrix} F_1^{(4)} & & & & \\ & F_2^{(4)} & & & \\ & & \ddots & & \\ & & & F_N^{(4)} & \\ G_1^{(4)} & & & & H^{(3)} \\ & G_2^{(4)} & & & H^{(3)} \\ & & \ddots & & \vdots \\ & & & G_N^{(4)} & H^{(3)} \\ & & & & \alpha^\top \end{bmatrix}$$

where $F_i^{(4)} = F_i^{(3)} K_i = F_i^{(3)} = I_{m_i} \otimes [1, \mathbf{0}_{m-1}^\top]$, $G_i^{(4)} = G_i^{(3)} K_i = [\mathbf{0}_{m-1}, I_{m-1}, \mathbf{0}_{(m-1) \times (mm_i - m)}]$, $i = 1, \dots, N$,

Let \tilde{A}_0 be the matrix consisting of the first $(mM + 1)$ columns of $\bar{A}_0 R_1 R_2$. That is,

$$\tilde{A}_0 = \begin{bmatrix} F_1^{(4)} & & & & \\ & F_2^{(4)} & & & \\ & & \ddots & & \\ & & & F_N^{(4)} & \\ G_1^{(4)} & & & & \\ & G_2^{(4)} & & & \\ & & \ddots & & \\ & & & G_N^{(4)} & \\ & & & & 1 \end{bmatrix}$$

The matrix \tilde{A}_0 possesses the following two properties:

1. Every row of \tilde{A}_0 contains exactly one nonzero entry equal to 1, while all remaining entries are zero;
2. Every column of \tilde{A}_0 contains no more than one nonzero entry.

Given these properties, we can establish the existence of permutation matrices $P_1 \in \mathbb{R}^{M+M'}$ and $Q_1 \in \mathbb{R}^{mM+1}$ such that the product $P_1 \tilde{A}_0 Q_1$ yields the form $[I_{(M+M')}, \mathbf{0}, \dots, \mathbf{0}]$. Consequently, we have $\text{rank}(\tilde{A}_0) = \text{rank}(P_1 \tilde{A}_0 Q_1) = M + M'$, which implies $\text{rank}(\bar{A}_0) = M + M'$. This demonstrates that \bar{A}_0 achieves full row-rank.

C. Algorithm for (WBC-LP): Predictor-Corrector Inner Point Method

Algorithm 2 Predictor-Corrector Inner Point Method for Linear Programming (WBC-LP)

1: **Input:** The standard LP form of Problem (WBC-LP)

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \bar{A}x = b, \quad x \geq 0 \end{aligned}$$

2: **Initialization:** Set initial feasible point (x_0, y_0, s_0) , where $x_0 > 0, s_0 > 0$ (dual variables). Choose tolerance $\epsilon > 0$ and set iteration counter $k = 0$.

3: **while** $\|r_b\| > \epsilon$ and $\|r_c\| > \epsilon$ **do**

4: Compute residuals:

$$\begin{aligned} r_b &= \bar{A}x - b \quad (\text{primal residual}) \\ r_c &= \bar{A}^T y + s - c \quad (\text{dual residual}) \\ r_s &= Xs \quad (\text{complementarity residual}) \end{aligned}$$

where $X = \text{diag}(x)$, $S = \text{diag}(s)$, and $\mu = \frac{x^T s}{n}$ is the duality measure.

5: **Predictor Step:** Solve the linear system for affine scaling direction $(\Delta x^{\text{aff}}, \Delta y^{\text{aff}}, \Delta s^{\text{aff}})$:

$$\begin{bmatrix} 0 & \bar{A}^T & I \\ \bar{A} & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta y^{\text{aff}} \\ \Delta s^{\text{aff}} \end{bmatrix} = - \begin{bmatrix} r_c \\ r_b \\ r_s \end{bmatrix}$$

We have

$$\begin{aligned} \Delta y^{\text{aff}} &= -(\bar{A}R\bar{A}^T)^{-1}(r_c - \bar{A}R(s - r_b)), \text{ solved by calling Algorithm 1.} \\ \Delta s^{\text{aff}} &= r_b - \bar{A}^T \Delta y^{\text{aff}}, \quad \Delta x^{\text{aff}} = R(s - \Delta s). \end{aligned} \quad (13)$$

6: Compute the step size α_{aff} by finding the maximum step length that maintains $x + \alpha_{\text{aff}}\Delta x^{\text{aff}} \geq 0$ and $s + \alpha_{\text{aff}}\Delta s^{\text{aff}} \geq 0$.

7: **Corrector Step:** Compute the corrector directions using central path perturbation with updated r_s :

$$r_s = r_s + \Delta X^{\text{aff}} \Delta s^{\text{aff}} - \mu \mathbf{1}$$

and solve the system again to get $(\Delta x^{\text{corr}}, \Delta y^{\text{corr}}, \Delta s^{\text{corr}})$:

$$\begin{aligned} \Delta y^{\text{corr}} &= -(\bar{A}R\bar{A}^T)^{-1} \bar{A}(S^{-1} \Delta X^{\text{aff}} \Delta s^{\text{aff}} - \mu S^{-1} \mathbf{1}), \text{ solved by calling Algorithm 1.} \\ \Delta s^{\text{corr}} &= -\bar{A}^T \Delta y^{\text{corr}}, \quad \Delta x^{\text{corr}} = S^{-1} \Delta X^{\text{aff}} \Delta s^{\text{aff}} - \mu S^{-1} \mathbf{1}. \end{aligned} \quad (14)$$

8: Compute the total search direction:

$$\Delta x = \Delta x^{\text{aff}} + \Delta x^{\text{corr}}, \quad \Delta y = \Delta y^{\text{corr}}, \quad \Delta s = \Delta s^{\text{aff}} + \Delta s^{\text{corr}}$$

9: Compute the step size α by updating with both predictor and corrector directions.

10: Update variables:

$$x_{k+1} = x_k + \alpha \Delta x, \quad y_{k+1} = y_k + \alpha \Delta y, \quad s_{k+1} = s_k + \alpha \Delta s.$$

11: Update the duality measure μ and increment the iteration counter $k = k + 1$.

12: **end while**

13: **Output:** Optimal solution (x^*, y^*, s^*) .

Each $F_i R_i F_i^\top$ is a diagonal matrix with positive diagonal entries.

$$\begin{aligned} \bar{G}_2 R \bar{G}_1^\top &= \begin{bmatrix} G_1^{(2)} R_1 F_1^\top & & \\ & \ddots & \\ & & G_N^{(2)} R_N F_N^\top \end{bmatrix} := B_2^\top, \\ \bar{G}_2 R \bar{G}_2^\top &= \begin{bmatrix} G_1^{(2)} R_1 G_1^{(2)\top} & & \\ & \ddots & \\ & & G_N^{(2)} R_N G_N^{(2)\top} \end{bmatrix} + \begin{bmatrix} H^{(2)} R_{N+1} H^{(2)\top} & \cdots & H^{(2)} R_{N+1} H^{(2)\top} \\ & \vdots & \vdots \\ H^{(2)} R_{N+1} H^{(2)\top} & \cdots & H^{(2)} R_{N+1} H^{(2)\top} \end{bmatrix}. \end{aligned} \quad (15)$$

where $H^{(2)} R_{N+1} H^{(2)\top}$ and each $G_i^{(2)} R_i G_i^{(2)\top}$ is a diagonal matrix with positive diagonal entries. We use B_3 to denote the first matrix in the right hand side of (15) and B_4 to denote the second. In addition, other blocks of $\bar{A} R \bar{A}^\top$ are

$$\bar{A}_3 R \bar{G}_1^\top = 0,$$

$$\bar{A}_3 R \bar{G}_2^\top = [\mathbf{1}_m^\top R_{N+1} H^{(2)\top} \quad \cdots \quad \mathbf{1}_m^\top R_{N+1} H^{(2)\top}] := \boldsymbol{\alpha}^\top,$$

$$\bar{A}_3 R \bar{A}_3^\top = \mathbf{1}_m^\top R_{N+1} \mathbf{1}_m := l.$$

$$\bar{A}_4 R \bar{G}_1^\top = \begin{bmatrix} D_1 R_1 F_1^\top & & \\ & \ddots & \\ & & D_N^{(2)} R_N F_N^\top \end{bmatrix} := K_1^\top,$$

$$\bar{A}_4 R \bar{G}_2^\top = \begin{bmatrix} D_1^{(2)} R_1 F_1^\top & & \\ & \ddots & \\ & & D_N^{(2)} R_N F_N^\top \end{bmatrix} := K_2^\top,$$

$$\bar{A}_4 R \bar{A}_4^\top = \text{diag}(D_1 R_1 D_1^\top, D_2 R_2 D_2^\top, \dots, D_N R_N D_N^\top) + R_{N+2} + r_{n_v} \mathbf{1}_N \mathbf{1}_n^\top := W,$$

here we have $W_1 := \text{diag}(D_1 R_1 D_1^\top, D_2 R_2 D_2^\top, \dots, D_N R_N D_N^\top) + R_{N+2}$.

With all notations above, we have

$$\bar{A} R \bar{A}^\top = \begin{bmatrix} B_1 & B_2 & \mathbf{0} & K_1 \\ B_2^\top & B_3 + B_4 & \boldsymbol{\alpha} & K_2 \\ \mathbf{0} & \boldsymbol{\alpha}^\top & l & 0 \\ K_1^\top & K_2^\top & 0 & W \end{bmatrix}$$

□

E. Derivation of Eq. (8)

$$\begin{aligned} & \begin{bmatrix} I_M & & & \\ -B_2^\top B_1^{-1} & I_{M'} & & \\ & & \mathbf{1} & \\ & & & I_N \end{bmatrix} \cdot \bar{A} R \bar{A}^\top \cdot \begin{bmatrix} I_M & & & \\ -B_2^\top B_1^{-1} & I_{M'} & & \\ & & \mathbf{1} & \\ & & & I_N \end{bmatrix}^\top = \begin{bmatrix} B_1 & & \mathbf{0} & K_1 \\ & G_1 + B_4 & \boldsymbol{\alpha} & \bar{K}_2 \\ \mathbf{0} & \boldsymbol{\alpha}^\top & l & 0 \\ K_1^\top & \bar{K}_2^\top & 0 & W \end{bmatrix} \\ & \begin{bmatrix} I_M & & & \\ & I_{M'} & -\boldsymbol{\alpha}/l & \\ & & \mathbf{1} & \\ & & & I_N \end{bmatrix} \begin{bmatrix} B_1 & & \mathbf{0} & K_1 \\ & G_1 + B_4 & \boldsymbol{\alpha} & \bar{K}_2 \\ \mathbf{0} & \boldsymbol{\alpha}^\top & l & 0 \\ K_1^\top & \bar{K}_2^\top & 0 & W \end{bmatrix} \begin{bmatrix} I_M & & & \\ & I_{M'} & -\boldsymbol{\alpha}/l & \\ & & \mathbf{1} & \\ & & & I_N \end{bmatrix}^\top = \begin{bmatrix} B_1 & & \mathbf{0} & K_1 \\ & G_1 + G_2 & \mathbf{0} & \bar{K}_2 \\ \mathbf{0} & \mathbf{0}^\top & l & 0 \\ K_1^\top & \bar{K}_2^\top & 0 & W \end{bmatrix} \\ & \begin{bmatrix} I_M & & & \\ & I_{M'} & & \\ -K_1^\top B_1^{-1} & & \mathbf{1} & \\ & & & I_N \end{bmatrix} \begin{bmatrix} B_1 & & \mathbf{0} & K_1 \\ & G_1 + G_2 & \mathbf{0} & \bar{K}_2 \\ \mathbf{0} & \mathbf{0}^\top & l & 0 \\ K_1^\top & \bar{K}_2^\top & 0 & W \end{bmatrix} \begin{bmatrix} I_M & & & \\ & I_{M'} & & \\ -K_1^\top B_1^{-1} & & \mathbf{1} & \\ & & & I_N \end{bmatrix}^\top = \begin{bmatrix} B_1 & & \mathbf{0} & K_1 \\ & G_1 + G_2 & \mathbf{0} & \bar{K}_2 \\ & & l & \\ & \bar{K}_2^\top & & \bar{W} \end{bmatrix}. \end{aligned}$$

F. Proof for Theorem 3.5

The proof follows the same approach of Lemma 4.2 in Ge et al. (2019), which is to prove positive definite property by confirming strict diagonal dominance. This approach requires several foundational results about the underlying matrix structure.

Lemma F.1. *The block matrices B_1, B_2, B_3, B_4 possess the following fundamental properties:*

- (i) *All entries are non-negative.*
- (ii) *The identity $B_3 \mathbf{1}_{M_2} = B_2^\top \mathbf{1}_M$ holds.*
- (iii) *The strict inequality $B_1 \mathbf{1}_M - B_2 \mathbf{1}_{M_2} > \mathbf{0}$ is satisfied.*

Proof. Property (i) follows directly from the construction using non-negative matrices \bar{A} and R .

For property (ii), we compute:

$$B_3 \mathbf{1}_{M_2} = \begin{bmatrix} G_1^{(2)} R_1 G_1^{(2)\top} \mathbf{1}_{m-1} \\ \vdots \\ G_N^{(2)} R_N G_N^{(2)\top} \mathbf{1}_{m-1} \end{bmatrix}$$

and

$$B_2^\top \mathbf{1}_M = \begin{bmatrix} G_1^{(2)} R_1 F_1^\top \mathbf{1}_{m_1} \\ \vdots \\ G_N^{(2)} R_N F_N^\top \mathbf{1}_{m_N} \end{bmatrix}$$

Using the definitions $G_i^{(2)} = \mathbf{1}_{m_i}^\top \otimes [\mathbf{0}_{m-1}, I_{m-1}]$ and $F_i = I_{m_i} \otimes \mathbf{1}_m^\top$, along with the diagonal structure of R_i , we establish that $G_i^{(2)} R_i F_i^\top \mathbf{1}_{m_i} = G_i^{(2)} R_i G_i^{(2)\top} \mathbf{1}_{m-1}$, yielding the desired identity.

For property (iii), observe that:

$$B_1 \mathbf{1}_M = \begin{bmatrix} F_1 R_1 F_1^\top \mathbf{1}_{m_1} \\ \vdots \\ F_N R_N F_N^\top \mathbf{1}_{m_N} \end{bmatrix}, \quad B_2 \mathbf{1}_{M_2} = \begin{bmatrix} F_1 R_1 G_1^{(2)\top} \mathbf{1}_{m-1} \\ \vdots \\ F_N R_N G_N^{(2)\top} \mathbf{1}_{m-1} \end{bmatrix}$$

Since $F_i R_i F_i^\top \mathbf{1}_{m_i} > F_i R_i G_i^{(2)\top} \mathbf{1}_{m-1}$ for each block i , the component-wise strict inequality follows. \square

Now we proceed to establish Theorem 3.5.

Proof of Theorem 3.5. The block-diagonal structure of G_1 is immediate from its construction. We establish positive definiteness via contradiction.

Assume G_1 is not positive definite. Then there exists an eigenvalue $-\lambda \leq 0$, making the matrix $\lambda I_{M_2} + G_1$ singular.

Consider the vector equation:

$$(\lambda I_{M_2} + G_1) \mathbf{1}_{M_2} = \lambda \mathbf{1}_{M_2} + B_3 \mathbf{1}_{M_2} - (B_2^\top B_1^{-1} B_2) \mathbf{1}_{M_2} \quad (16)$$

Applying property (ii) from Lemma F.1:

$$\begin{aligned} &= \lambda \mathbf{1}_{M_2} + B_2^\top \mathbf{1}_M - (B_2^\top B_1^{-1} B_2) \mathbf{1}_{M_2} \\ &= \lambda \mathbf{1}_{M_2} + B_2^\top B_1^{-1} B_1 \mathbf{1}_M - (B_2^\top B_1^{-1} B_2) \mathbf{1}_{M_2} \\ &= \lambda \mathbf{1}_{M_2} + B_2^\top B_1^{-1} (B_1 \mathbf{1}_M - B_2 \mathbf{1}_{M_2}) \end{aligned} \quad (17)$$

Since $B_2^\top B_1^{-1} \geq 0$ with each row containing at least one strictly positive element, and property (iii) ensures $B_1 \mathbf{1}_M - B_2 \mathbf{1}_{M_2} > \mathbf{0}$, we conclude:

$$(\lambda I_{M_2} + G_1) \mathbf{1}_{M_2} > \mathbf{0}_{M_2}$$

Next, we analyze the structure of $\lambda I_{M_2} + G_1 = \lambda I_{M_2} + B_3 - B_2^\top B_1^{-1} B_2$. Since $B_1, B_2, B_3 \geq 0$ and B_3 is diagonal, this matrix has positive diagonal entries and non-positive off-diagonal entries.

Define the projection operators:

$$D_{M_2} := I_{M_2} \quad (\text{diagonal mask}) \quad (18)$$

$$O_{M_2} := \mathbf{1}_{M_2} \mathbf{1}_{M_2}^\top - I_{M_2} \quad (\text{off-diagonal mask}) \quad (19)$$

Then:

$$D_{M_2} \circ |\lambda I_{M_2} + G_1| = D_{M_2} \circ (\lambda I_{M_2} + G_1) \quad (20)$$

$$O_{M_2} \circ |\lambda I_{M_2} + G_1| = -O_{M_2} \circ (\lambda I_{M_2} + G_1) \quad (21)$$

where \circ denotes the Hadamard product. This yields:

$$[D_{M_2} \circ |\lambda I_{M_2} + G_1| - O_{M_2} \circ |\lambda I_{M_2} + G_1|] \mathbf{1}_{M_2} = (\lambda I_{M_2} + G_1) \mathbf{1}_{M_2} > \mathbf{0}_{M_2}$$

This establishes strict diagonal dominance of $\lambda I_{M_2} + G_1$, implying non-singularity and contradicting our assumption. Therefore, G_1 is positive definite.

As for G_2 , direct computation shows that:

$$G_2 = (\mathbf{1}_N \mathbf{1}_N^\top) \otimes \left(\text{diag}(z) - \frac{1}{l} z z^\top \right)$$

Given that $l > \mathbf{1}_{m-1}^\top z$ and $z \geq \mathbf{0}$ by construction, set $Y = \text{diag}(z) - \frac{1}{l} z z^\top$, we know that Y is positive definite by similar proof as that of G_1 . \square

G. Algorithm Complexity

Proof of Theorem 3.2. We analyze the time and storage complexities of Algorithm 1 separately.

Time complexity. We list the most time consuming part of each step corresponding to Algorithm 1.

In step 1, computing of B_1, B_3, W, Q takes most flops of $O(\sum_{i=1}^N m^2 m_i)$, achieved when computing B_1 and B_3 .

In step 2, computing of $(G_1 + G_2)^{-1}$ takes most flops of $O(Nm^3)$, due to the need to invert $G_{ii}, i = 1, \dots, N$; computing of \tilde{W} takes $O(N^2 m^2)$ flops, which may exceed other procedures on time complexity when N is sufficiently large.

In step 3, computing \tilde{W}^{-1} takes $O(N^3)$ flops, while other part takes most $O(Nm^2)$.

To sum up, the total time complexity is $O(m^2 \sum_{i=1}^N m_i + Nm^3 + N^2 m^2 + N^3)$.

Space complexity. For the implementation of the whole interior-point methods, the major data that should be kept in the memory include the following four parts:

(a) Compute primal variables or dual variables. Note that the size of a primal variable is $O(m(\sum_{i=1}^N m_i))$, and a dual variable is of size $O(M + M')$.

(b) Matrix \bar{A} . According to the structure of \bar{A} shown in appendix D, the matrix contains several sparse blocks, and we only need to store the non-zero entries of each block. Overall, the space for storing \bar{A} is bounded by $2m \sum_{i=1}^N m_i + N(m-1) + m$. Since each column of E_1 and E_2 has at most one non-zero term, the total number of non-zero entries in E_1 and E_2 is bounded by $2m \sum_{i=1}^N m_i$. In addition, E_3 has $m-1$ non-zero terms, D has $m \sum_{i=1}^N m_i$ non-zero terms as the i -th row of it contains mm_i non-zero terms. So the total number of non-zero terms in \bar{A} is bounded by $O(M)$.

(c) Diagonals of matrices B_1 and B_3 , and diagonal blocks of matrices K_1, K_2, B_2 and G_1 . B_1 and B_3 takes $n_c - N = O(M + M')$ doubles. The diagonal blocks of matrices B_2 and G_1 have $m \sum_{i=1}^N m_i$ elements and $N(m-1)^2$ elements, respectively. Noticing that every row of D' only has one non-zero term, we have K_1 of size smaller than $nnz(E_1) = O(M)$, the number of nonzero terms of E_1 , and K_2 smaller than $nnz(E_2) = O(M)$.

(d) The matrix W, \bar{W} and \tilde{W} . Takes no more than N^2 to store.

(e) Other intermediate vectors or matrices, whose data scale is bounded by a constant time of the data scale in (a), (b), (c) and (d).

With the analysis in (a) to (d), we know the memory usage of algorithm 1 is bounded by $O(m \sum_{i=1}^N m_i + Nm^2 + N^2)$. \square

H. Woodbury identity

Given the matrices A, T, C , and V , the standard formula for Woodbury identity (Hager, 1989) is

$$(A + TCV)^{-1} = A^{-1} - A^{-1}T(C^{-1} + VA^{-1}T)^{-1}VA^{-1}.$$

In the proof of Lemma 3.6, the corresponding A, T, C , and V are $G_1, \mathbf{1}_N \otimes U^\top, I_{M'}, \mathbf{1}_N^\top \otimes U$, respectively.

I. The free support WBC

For the free support WB, many previous researches (Cuturi & Doucet, 2014; Ge et al., 2019; Huang et al., 2021) apply block coordinate descent, which requires minimizing the

$$\begin{aligned} \min_{\mathbf{w}, X, \{\Pi^{(t)}\}} \quad & \sum_{t=1}^N \langle D(X, Q^{(t)}), \Pi^{(t)} \rangle \\ \text{s.t.} \quad & \Pi^{(t)} \mathbf{1}_{m_t} = \mathbf{w}, (\Pi^{(t)})^\top \mathbf{1}_m = \mathbf{a}^{(t)}, \Pi^{(t)} \geq 0, \forall t = 1, \dots, N \\ & \mathbf{1}_m^\top \mathbf{w} = 1, \mathbf{w} \geq 0 \end{aligned} \quad (22)$$

where $\mathbf{w} := (w_1, \dots, w_m)^\top \in \mathbb{R}_+^m$, $X := [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{d \times m_t}$, $\Pi^{(t)} \in \mathbb{R}_+^{m \times m_t}$ and $D(X, Q^{(t)}) := [\|\mathbf{x}_i - \mathbf{q}_j^{(t)}\|^2] \in \mathbb{R}^{m \times m_t}$ for $t = 1, \dots, N$. Problem (22) is a nonconvex problem, where one needs to find the optimal support points X and the optimal weight vector \mathbf{w} of a barycenter simultaneously. But it is specified from the support points of $\{\mathcal{P}^{(t)}\}_{t=1}^N$.

For the free support Wasserstein barycenter, many previous researches (Cuturi & Doucet, 2014; Ge et al., 2019; Huang et al., 2021) apply block coordinate descent. For our Wasserstein ball center, the objective becomes:

$$\begin{aligned} \min_{\mathbf{w}, X, \{\Pi^{(t)}\}_{t \in [N]}} \quad & \max_{t \in [N]} \langle D(X, Q^{(t)}), \Pi^{(t)} \rangle \\ \text{s.t.} \quad & \Pi^{(t)} \mathbf{1}_{m_t} = \mathbf{w}, (\Pi^{(t)})^\top \mathbf{1}_m = \mathbf{a}^{(t)}, \Pi^{(t)} \geq 0, \forall t = 1, \dots, N \\ & \mathbf{1}_m^\top \mathbf{w} = 1, \mathbf{w} \geq 0 \end{aligned} \quad (23)$$

where $\mathbf{w} := (w_1, \dots, w_m)^\top \in \mathbb{R}_+^m$, $X := [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{d \times m_t}$, $\Pi^{(t)} \in \mathbb{R}_+^{m \times m_t}$ and $D(X, Q^{(t)}) := [\|\mathbf{x}_i - \mathbf{q}_j^{(t)}\|^2] \in \mathbb{R}^{m \times m_t}$ for $t = 1, \dots, N$. Problem (23) is a nonconvex problem. By block coordinate descent, one optimizes the support set X , and then the fixed support WBC to obtain the weight \mathbf{w} of WBC and coupling matrices $\Pi^{(t)}$ alternately. The algorithm will converge to a local minima. For instance, in l_2 space, the minimization of X is $\min_X \max_{t \in [N]} \sum_{k=1}^m \sum_{j=1}^{m_t} \|\mathbf{x}_k - \mathbf{q}_j^{(t)}\|^2 \pi_{kj}^{(t)}$.

This is a quadratically constrained quadratic program (QCQP) in that it can be reformulated as the following problem:

$$\begin{aligned} \min_{X, \zeta} \quad & \zeta \\ \text{s.t.} \quad & \sum_{k=1}^m \sum_{j=1}^{m_t} \|\mathbf{x}_k - \mathbf{q}_j^{(t)}\|^2 \pi_{kj}^{(t)} \leq \zeta \end{aligned} \quad (24)$$

Since the quadratic forms of X in the constraints is positive semidefinite, the problem is convex [1], thus can be efficiently solved with convex programming, such as interior point method. Note that the size of variable X, ζ is $m + 1$, the number of constraints is N , the scale of solving X is much smaller than solving the fixed support WBC.

J. Supplementary Experiment

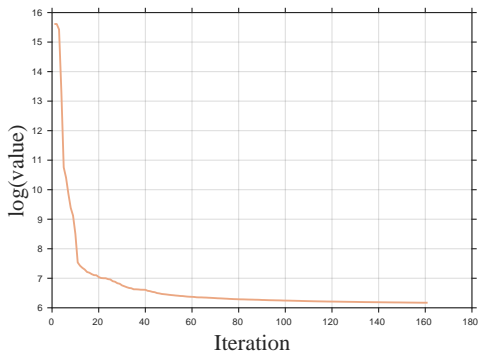


Figure 5. $N=10, m=6000$

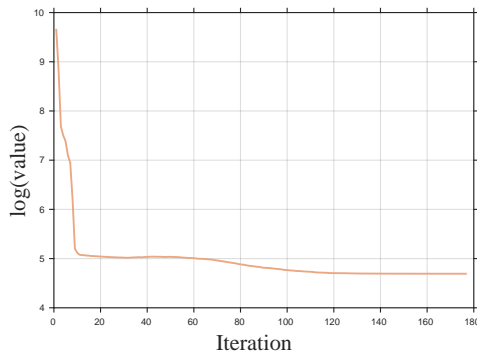


Figure 6. $N=2000, m=300$

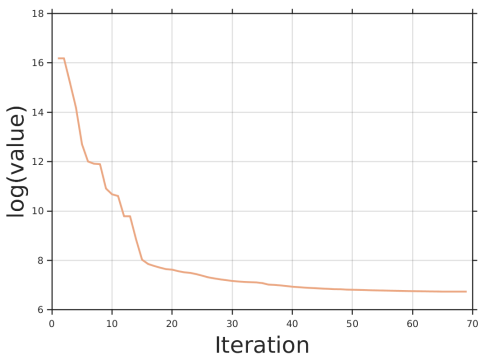


Figure 7. $N=3, m=30000$

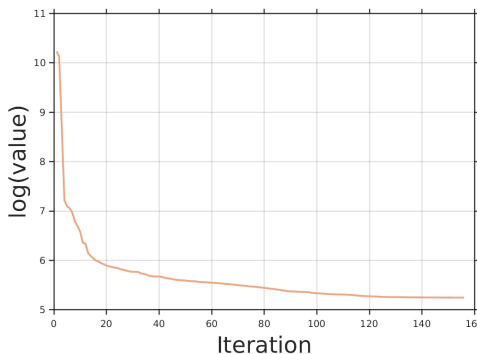


Figure 8. $N=500, m=500$

J.1. Super-linearity of our algorithm

A sequence $\{x_n\}$ is said to converge *super-linearly* to a limit x^* if:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = 0$$

This means that the error $|x_n - x^*|$ decreases at a rate faster than linear convergence. With our logarithmic coordinate of objective value, we have shown that our algorithm converges in super-linearly speed,

In both case when $m \gg N$ and $m \ll N$, the super-linear convergence speed maintains.

J.2. Fairness of WBC

To compare the imf, we set all distributions to be on support $[m] \subset \mathbb{R}^1$. For distributions in family 1, we assign the points in $\{1, 2, \dots, 10\}$ with probability proportional to independent samples from $U([3, 4])$, the uniform distribution on $[3, 4]$, and other points with probability proportional to samples from $U([0, 1])$. For a distribution in family 2, we assign the points in $\{m - 9, \dots, m\}$ with probability proportional to independent samples $U([3, 4])$, and other points with probability proportional to samples from $U([0, 1])$. See Fig. 9 for the instance tested in Fig. 3(b): We also conduct several other experiments with different settings, see Figs. 10 and 11.

To evaluate and compare the var(WD), we consider a comprehensive set of experimental configurations. The experiments are conducted across the following settings:

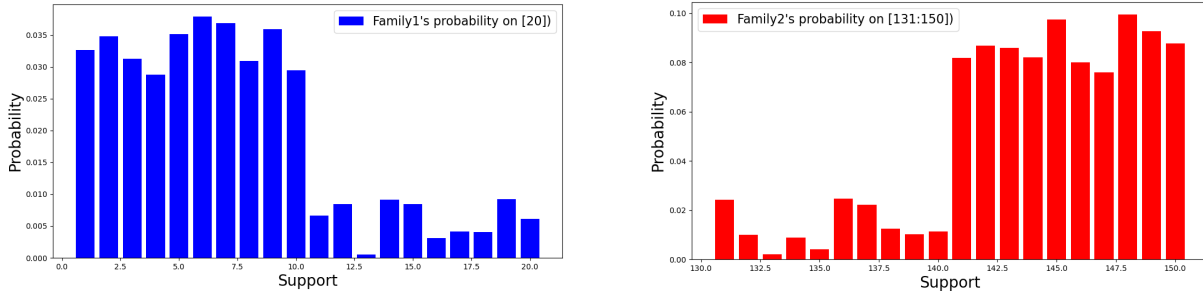


Figure 9. The left histogram shows a probability in family 1 on $[20]$, whose density is concentrated in $[10]$. The right is a probability in family 2, concentrated in $[m - 9, m]$.

- Ground space dimensions: 3, 10, and 100.
- Ground metrics: $l_1, l_2,$ and l_∞ .
- Support selection regimes:
 - Uniform sampling within a cube;
 - Gaussian sampling;

Each sampling regime is further augmented with cluster-diversifying transformations, which divide the support points into 1, 2, or 3 clusters.

For every combination of the above three parameter categories, we generate 50 independent instances, resulting in a total of 2700 experimental runs. Every instance consists of 30 probability distributions, each of which has weights sampled uniformly and support size of 200.

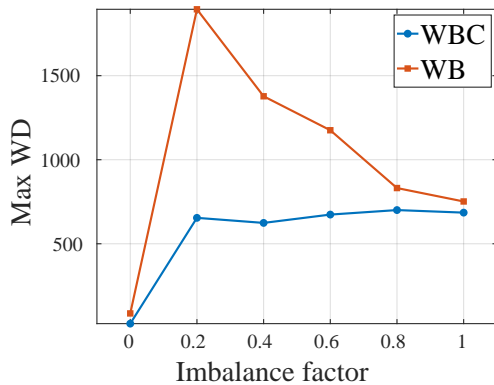


Figure 10. $N=40, m=150$

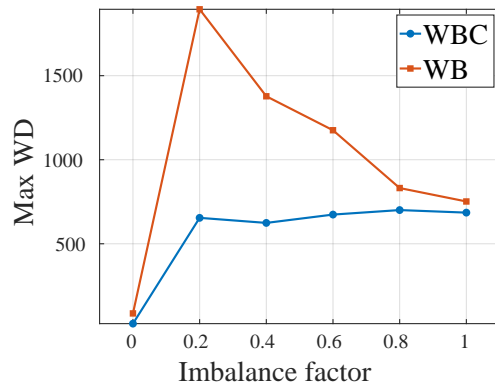


Figure 11. $N=15, m=600$

J.3. The Point cloud data

This is a free-support problem. We select the support with a two-step strategy: firstly, choose the support from a combination of a dense grid and samples from a normal distribution with large variance, computing the weight of WBC; then update the support of WBC by adding the support in last step with samples near the points who have particular large weights, computing the final WBC. Both WB and WBC are given grey scales proportional to the weight of the point, so more weight means paler color.

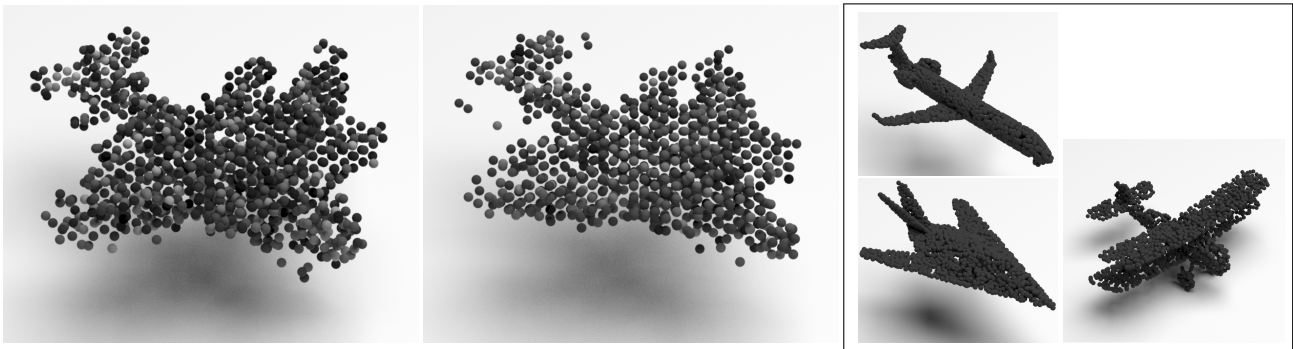


Figure 12. WBC (left) and WB (middle) of 3 airplane, one of which is a biplane (with a double-decker wing structure), while the other two are both monoplanes. Compared to the WB, the WBC retains more characteristics of the biplane.