Journal of Machine Learning Research 27 (2026) 1-44 Submitted 11/22; Revised 10/26; Published 10/26

An Efficient HPR Algorithm for the Wasserstein Barycenter Problem with $O(\mathrm{Dim(P)}/\varepsilon)$ Computational Complexity

Guojun Zhang GUOJUN.ZHANG@CONNECT.POLYU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Yancheng Yuan* YANCHENG.YUAN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Defeng Sun DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Editor: Peter Richtarik

Abstract

In this paper, we study and implement an efficient Halpern–Peaceman–Rachford (HPR) algorithm for solving the Wasserstein barycenter problem (WBP) with fixed supports. While the Peaceman–Rachford (PR) splitting method itself may not converge for the WBP, applying the Halpern iteration to the PR operator yields an $O(1/\varepsilon)$ non-ergodic iteration complexity with respect to the Karush–Kuhn–Tucker (KKT) residual. The structured linear solve in the HPR subproblem corresponds to an affine projection that can be viewed as a coupled extension of Romero's classical OT projection formula. This observation yields a closed-form solution procedure with cost linear in the size of the linear system. Consequently, HPR achieves an $O(\mathrm{Dim(P)}/\varepsilon)$ computational complexity in terms of flops for obtaining an ε-optimal solution measured by the KKT residual of the unregularized WBP, where Dim(P) denotes the number of variables in the corresponding LP formulation. This bound matches the order of the best-known bound for the WBP, while requiring no entropic regularization and measuring accuracy by the KKT residual. Numerical experiments on synthetic and real data sets demonstrate the practical efficiency of the restarted HPR implementation for large-scale WBPs.

Keywords: Wasserstein barycenter problem, optimal transport, Halpern iteration, non-ergodic complexity, Peaceman–Rachford splitting

1. Introduction

Optimal transport (OT) (Monge, 1781; Kantorovich, 1942) defines a Wasserstein distance between two probability distributions through the minimal transportation cost. As an important application, the Wasserstein distance naturally leads to the concept of Wasserstein barycenter, which defines a mean of a set of complex objects (i.e., images, videos, texts,

*. Corresponding author.

©2026 Guojun Zhang, Yancheng Yuan, and Defeng Sun.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v27/22-1363.html.

and so on) that can preserve their geometric structure (Agueh and Carlier, 2011). The Wasserstein barycenter problem (WBP) has made a significant impact in a variety of fields, including machine learning (Li and Wang, 2008; Ye and Li, 2014; Peyré and Cuturi, 2019), physics (Cotar et al., 2013), statistics (Bigot and Klein, 2018), economics (Chiappori et al., 2010; Galichon, 2016), brain imaging (Gramfort et al., 2015), and so on. However, the computational cost for computing the Wasserstein distance and finding the Wasserstein barycenters is expensive, especially for modern applications with an immense amount of data.

To address this challenge, various computational approaches have been developed for WBPs. Existing methods for broader barycenter models on continuous spaces (Agueh and Carlier, 2011) include stochastic optimization approaches (Staib et al., 2017; Claici et al., 2018; Dvurechenskii et al., 2018) and Sinkhorn-divergence-based methods (Luise et al., 2019). In practice, probability measures are often approximated by finite supports, leading to discrete WBPs. In the discrete setting, Altschuler and Boix-Adsera (2021) developed a polynomial-time algorithm for fixed support dimension d by solving an exponential-size linear programming (LP) via an efficient separation oracle; however, the running time still contains a factor that grows exponentially with d. A common practical strategy for high-dimensional free-support WBPs is alternating minimization (Cuturi and Doucet, 2014; Ye and Li, 2014; Ye et al., 2017; Yang et al., 2021; Ge et al., 2019), where fixing the barycenter support locations leads to a fixed-support WBP that often becomes the main computational bottleneck. Hence, in this paper, we focus on the fixed-support discrete WBP, where the barycenter support is prescribed or selected in advance. This setting is natural when the barycenter is required on a common grid and also serves as a key building block for free-support barycenter algorithms.

For the fixed-support discrete WBP, Anderes et al. (2016) formulated the problem as an LP. Nonetheless, state-of-the-art LP solvers such as Gurobi face challenges in solving it even with a moderate number of objects and supports.

To overcome the computational challenges, Cuturi (2013) proposed an entropic regularization for OT, such that Sinkhorn's algorithm is applicable to solving the entropy-regularized OT problem and computing an approximate Wasserstein distance. Cuturi and Doucet (2014) further applied the entropic regularization idea to study the WBP. Later, Benamou et al. (2015) proposed an iterative Bregman projection (IBP) method, which generalized Sinkhorn's algorithm, to solve the entropy-regularized WBP. Along this direction, plenty of algorithms have been proposed, including gradient-type methods (Cuturi and Peyré, 2016), accelerated primal-dual gradient descent (Dvurechenskii et al., 2018; Krawtschenko et al., 2025), the fast IBP (Lin et al., 2020), stochastic gradient descent (Claici et al., 2018; Tiapkin et al., 2022), and distributed and parallel gradient descent (Staib et al., 2017; Uribe et al., 2018; Rogozin et al., 2025).

When the regularization parameter in the entropy-regularized WBP is moderate, say no less than 10^{-2}, the aforementioned algorithms, such as the IBP, can be effective in finding an approximate solution to the entropy-regularized WBP. However, it is known that one needs to choose a small regularization parameter in the regularized WBP to obtain a high-quality solution in real applications. This has also been observed in our numerical experiments. A small regularization parameter will usually cause numerical issues and slower convergence for these algorithms, such as the IBP. Some stabilized and scaling techniques have been

proposed to improve the robustness of entropic-regularization-based algorithms (Schmitzer, 2019), but the efficiency of the stabilized algorithms is still not satisfactory for solving the large-scale WBP. These issues may restrict the role of the entropic regularization approach in applications with high requirements on solution quality (Bonneel et al., 2016).

Algorithm	$\mathrm{Obj_P}$	$\mathrm{D_{gap}}$	$\mathrm{R_{KKT}}$	Complexity	Iterates
IBP (Benamou et al., 2015) (Kroshnin et al., 2019)	✓			$\widetilde{O}(Tm^2/\varepsilon^2)$	Non-ergodic
PDAGD (Kroshnin et al., 2019)	✓			$\widetilde{O}(Tm^{5/2}/\varepsilon)$	Ergodic
FastIBP (Lin et al., 2020)	✓			$\widetilde{O}(Tm^{7/3}/\varepsilon^{4/3})$	Non-ergodic
Dual extrapolation with area-convexity (Dvinskikh and Tiapkin, 2021)		✓		$\widetilde{O}(Tm^2/\varepsilon)$	Ergodic
Mirror-Prox (Dvinskikh and Tiapkin, 2021)		✓		$\widetilde{O}(Tm^{5/2}/\varepsilon)$	Ergodic
Accelerated alternating minimization (Guminov et al., 2021)	✓			$\widetilde{O}(Tm^{5/2}/\varepsilon)$	Non-ergodic
Accelerated Bregman primal-dual method (Chambolle and Contreras, 2022)	✓			$\widetilde{O}(Tm^{5/2}/\varepsilon)$	Ergodic

Table 1: A summary of the known complexity of entropic regularization type algorithms for the WBP with T sample distributions and m supports. In this table, $\mathbf{Obj_P}$, $\mathbf{D_{gap}}$ and $\mathbf{R_{KKT}}$ mean that the error $\varepsilon > 0$ is measured by the primal objective function value gap, the duality gap, and the KKT residual, respectively.

The instability and unsatisfactory efficiency of entropic-regularization-based algorithms for solving the WBP in the relatively high accuracy regime motivate some researchers to return to designing efficient algorithms for solving the WBP without entropic regularization. In this direction, first-order splitting algorithms, in particular, the alternating direction method of multipliers (ADMM) (Glowinski and Marroco, 1975; Gabay and Mercier, 1976) and its variants, are very popular. When applying the ADMM directly to solve the WBP, one needs to solve a huge-scale linear system in each iteration, which is computationally challenging and may be prohibitive. The specific definition of this linear system can be found later in Section 2.1. To avoid solving this huge linear system, Ye et al. (2017) proposed a modified Bregman ADMM (mBADMM), where each step of the mBADMM has a closed-form solution for solving the WBP. Later, Yang et al. (2021) proposed a symmetric Gauss-Seidel ADMM (sGS-ADMM) for solving the WBP, which is more efficient and robust than the mBADMM. Regarding the WBP as a multi-block problem and partially using the special structure of the WBP, the sGS-ADMM enjoys a low computational complexity in each iteration. It has also been demonstrated in Yang et al. (2021) that the sGS-ADMM is more stable compared to algorithms based on entropic regularization, such as IBP. The quality of the solution obtained by the sGS-ADMM can also outperform the one obtained by the IBP. However, we observed in our numerical experiments that the sGS-ADMM generally needs more iterations than the ADMM if the huge linear systems involved can be solved. Moreover, it has been demonstrated in Yang et al. (2021) that the sGS-ADMM is much

more efficient than the mBADMM. These observations motivate us to carefully investigate the linear system involved in the ADMM for solving the WBP. This structured linear solve is a dual representation of an affine projection that can be viewed as a coupled extension of the classical OT projection formula of Romero (1990). Exploiting this structure yields an explicit solution of the linear system with linear computational cost, enabling a fast-ADMM for solving the WBP with a cheap per-iteration computational complexity.

The proposed efficient procedure for solving the linear system further motivates us to investigate the computational complexity for solving the WBP. The computational complexity of finding an approximate solution to the OT problem and the WBP with ε-precision has been extensively studied in recent years, in particular for the entropic regularization type algorithms. For the rest of this section, we consider the WBP with T sample distributions and m supports. Kroshnin et al. (2019) have established an $\widetilde{O}(Tm^2/\varepsilon^2)$ complexity bound for the IBP method and an $\widetilde{O}(Tm^{2.5}/\varepsilon)$ complexity bound for a variant of the primal-dual accelerated gradient (PDAGD) method (both in the sense of the primal objective function value gap). Here, the notation $\widetilde{O}(\cdot)$ hides only absolute constants and polylogarithmic factors. So far, the best computational complexity bound for the WBP is $\widetilde{O}(Tm^2/\varepsilon)$ (in the sense of the duality gap), which is achieved by the dual extrapolation algorithm proposed by Dvinskikh and Tiapkin (2021). A summary of the complexity bounds of the entropic regularization type algorithms for solving the WBP is in Table 1.

Algorithm	$\mathrm{Obj_P}$	$\mathrm{Obj_D}$	$\mathrm{R_{KKT}}$	Complexity	Iterates
sGS-ADMM (for dual WBP) (Yang et al., 2021) + (Cui et al., 2016)			✓	$O(Tm^2/\varepsilon^2)$	Non-ergodic
sGS-ADMM (for dual WBP) (Yang et al., 2021) + (Cui et al., 2016)		✓		$O(Tm^2/\varepsilon)$	Ergodic
HPR (for primal or dual WBP) (This paper)			✓	$O(Tm^2/\varepsilon)$	Non-ergodic
HPR (for primal WBP) (This paper)	✓			$O(Tm^2/\varepsilon)$	Non-ergodic
HPR (for dual WBP) (This paper)		✓		$O(Tm^2/\varepsilon)$	Non-ergodic

Table 2: A summary of the complexity of the HPR algorithm and the sGS-ADMM algorithm for the WBP with T sample distributions and m supports. In the table, $\mathbf{Obj_P}$, $\mathbf{Obj_D}$, and $\mathbf{R_{KKT}}$ mean that the error $\varepsilon > 0$ is measured by the primal objective function value gap, the dual objective function value gap, and the KKT residual, respectively. We regard the LP reformulation of the WBP and its dual problem as the primal WBP and the dual WBP, respectively.

On the other hand, the discussion of the computational complexity of operator splitting methods, such as ADMM-type algorithms for solving the WBP, is not as extensive. In this direction, we can obtain an $O(Tm^2/\varepsilon^2)$ computational complexity in terms of the KKT residual by combining the per-iteration computational complexity of the sGS-ADMM (Yang et al., 2021) and the non-ergodic $O(1/\sqrt{k})$ iteration convergence rate of the majorized ADMM (Cui et al., 2016). This is perhaps the only known complexity bound so far that is comparable to the best-known complexity bound of entropy-regularized algorithms. There are also some known iteration complexity results for the ADMM. Monteiro and Svaiter (2013) first proved the $O(1/k)$ ergodic convergence rate (regarding the KKT-type residual)

of the ADMM with unit dual step size for a class of linearly constrained convex programming problems with a separable objective function. Later, Davis and Yin (2016) established an $O(1/\sqrt{k})$ non-ergodic iteration complexity bound of the ADMM with unit dual step size with respect to the primal feasibility and the primal objective function value gap. The key bottleneck for establishing an attractive complexity bound comes from the expensive Cholesky decomposition, which is $O(T^3m^3)$, although it only needs to be performed once. Thus, for the WBP, the ergodic computational complexity and the non-ergodic computational complexity of the ADMM method can be known to be $O((T^3m^3) + (Tm^2/\varepsilon))$ and $O((T^3m^3) + (Tm^2/\varepsilon^2))$, respectively. Instead, the fast-ADMM can benefit from the linear-time procedure for solving the involved linear system. Thus, the ergodic computational complexity and the non-ergodic computational complexity of the fast-ADMM method can be improved to $O(Tm^2/\varepsilon)$ and $O(Tm^2/\varepsilon^2)$, respectively. In real applications, the non-ergodic complexity is more important since the non-ergodic sequence can preserve sparsity. In this paper, we study the HPR method that enjoys the stronger $O(Tm^2/\varepsilon)$ non-ergodic complexity. This bound matches, up to logarithmic factors, the order of the best-known bound $\widetilde{O}(Tm^2/\varepsilon)$ for the WBP (Dvinskikh and Tiapkin, 2021), and is obtained without entropic regularization, with accuracy measured by the KKT residual of the LP formulation of the WBP.

The HPR algorithm applies the Halpern iteration (Halpern, 1967) to the PR splitting method (Lions and Mercier, 1979). While it is not clear to us when the PR applied to the WBP converges, the convergence of the HPR follows directly from Wittmann (1992). Starting from any initial point, the HPR enjoys an $O(1/k)$ iteration complexity with respect to the fixed point residual of the corresponding PR operator (Lieder, 2021). When the HPR algorithm is applied to solving the two-block convex optimization problem with linear constraints, we establish an $O(1/k)$ non-ergodic iteration complexity in terms of the KKT residual and the primal objective function value gap. Here, we mention that Kim (2021) proposed an accelerated ADMM and proved an $O(1/k)$ convergence rate with respect to the primal feasibility only. To this end, we prove that the HPR algorithm enjoys an $O(Tm^2/\varepsilon)$ computational complexity guarantee for obtaining an ε-optimal solution with respect to the KKT residual of the WBP with T distributions and m supports. We briefly compare the computational complexity bounds of the HPR algorithm and the sGS-ADMM algorithm and summarize the results in Table 2. As we can see in the table, in terms of the non-ergodic complexity, the HPR improves the dependence on ε by a factor of $O(1/\varepsilon)$ compared to the sGS-ADMM, which is substantial. We highlight the main contributions of this paper as follows:

1. We propose a linear-time procedure, in terms of the linear-system size, for solving the structured linear systems arising in the ADMM for the WBP. The structured linear system can be viewed as a dual representation of an affine projection that is a coupled extension of Romero's classical OT projection formula. This linear-time procedure yields a fast-ADMM implementation for large-scale WBP with low per-iteration computational cost.

2. We study the HPR algorithm, which combines classical Peaceman–Rachford splitting with Halpern iteration. For two-block convex programming problems with linear con-

straints, the HPR algorithm enjoys $O(1/\varepsilon)$ non-ergodic iteration complexity bounds in terms of the KKT residual and the objective function value gap.

3. We apply the HPR algorithm to the WBP with T distributions and m supports, and establish an $O(Tm^2/\varepsilon)$ non-ergodic computational complexity bound for obtaining an ε-optimal solution in terms of the KKT residual. This KKT-residual-based complexity bound is particularly relevant for primal-dual optimization algorithms, since the KKT residual is widely used as a practical stopping criterion.

4. We demonstrate the promising numerical performance of the restarted HPR implementation in obtaining high-quality solutions to the WBP on both synthetic and real data.

The initial version of this manuscript, released in 2022, represents an early application of the HPR method to the structured LP arising from the fixed-support WBP, a special case of the convex optimization framework considered here without additional proximal terms. Subsequent work extended this idea to proximal settings, thereby broadening the applicability of the HPR framework to large-scale convex optimization. In particular, Yang et al. (2025) developed a proximal HPR method via a classical proximal point method reformulation with a positive definite preconditioner and established an $O(1/k)$ rate for the weighted fixed point residual. Further, Sun et al. (2025) proposed a semi-proximal HPR method and, through a degenerate proximal point method (dPPM) reformulation, established an $O(1/k)$ iteration complexity for both the KKT residual and the objective error.

Building on these theoretical developments, HPR-based methods have also led to the development of efficient solvers for large-scale optimization problems. For example, HPR-LP (Chen et al., 2026), an HPR implementation for linear programming, has demonstrated strong performance on large-scale LP instances and outperformed the GPU implementation of the award-winning PDLP solver in several benchmark tests (Applegate et al., 2021, 2023; Lu and Yang, 2025). Subsequently, HPR-LP (Chen et al., 2026) has led to the development of the reflected restarted Halpern-PDHG method (r2HPDHG) (Lu and Yang, 2024), which is implemented in cuPDLPx (Lu et al., 2025). Its underlying iteration is a special case of the semi-proximal HPR method (Sun et al., 2025; Chen et al., 2026, 2027). Moreover, the HPR framework has shown strong empirical performance in optimal transport (HOT) (Zhang et al., 2025) and convex quadratic composite programming (HPR-QP) (Chen et al., 2025).

Organization. The rest of the paper is organized as follows. In Section 2, we introduce the model of the WBP. To make the ADMM computationally affordable for solving the WBP, we propose a linear-time procedure for solving the involved linear system. In Section 3, we first introduce the HPR for solving the maximal monotone inclusion problem. When the HPR algorithm is applied to solving the two-block convex optimization problem with linear constraints, we establish an $O(1/\varepsilon)$ non-ergodic iteration complexity in terms of the KKT residual and the primal objective function value gap. Then, we prove an $O(Tm^2/\varepsilon)$ computational complexity in terms of flops for obtaining an ε-optimal solution measured by the KKT residual of the WBP with T distributions and m supports. Section 4 is devoted to demonstrating the efficiency and robustness of the HPR for solving the WBP with extensive numerical experiments on both synthetic and real data sets. We conclude this paper in Section 5.

Notation. We use $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ to denote finite-dimensional real Euclidean spaces equipped with the inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. In particular, we denote the n-dimensional real Euclidean space as $\mathbb{R}^n$. For any $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, we define $\langle x, y \rangle := \sum_{i=1}^n x_i y_i$ and $\|x\| := \sqrt{\sum_{i=1}^n x_i^2}$, respectively. We also denote the nonnegative orthant of $\mathbb{R}^n$ as $\mathbb{R}^n_+$. A vector $x \in \mathbb{R}^n$ is a column vector by default. For a given linear mapping $A : \mathbb{X} \to \mathbb{Y}$, $A^* : \mathbb{Y} \to \mathbb{X}$ is the adjoint of A. We denote $\|A\| := \sup_{\|x\| \le 1} \|Ax\|$ as the spectral norm of A. We denote the transpose of a matrix $B \in \mathbb{R}^{m \times n}$ as $B^\top \in \mathbb{R}^{n \times m}$. $\|B\|_F := \sqrt{\mathrm{trace}(BB^\top)}$ and $\|B\|_\infty := \max_{1 \le i \le m, 1 \le j \le n} |B_{ij}|$ are the Frobenius norm of B and the infinity norm of B, respectively. Here, $\mathrm{trace}(BB^\top)$ is the trace of $BB^\top$. Let $\mathbf{1}_m$ (resp. $\mathbf{0}_m$) denote the m-dimensional vector with all entries being 1 (resp. 0). $\otimes$ stands for Kronecker product. We denote the vectorization of a matrix $X \in \mathbb{R}^{m \times n}$ as $\mathrm{vec}(X) \in \mathbb{R}^{mn}$. For a collection of matrices $\{A_1, \ldots, A_m\}$, we denote the block diagonal matrix with diagonal blocks A_i as $\mathrm{diag}(A_1, \ldots, A_m)$. For a nonempty closed convex set C, we denote the indicator function of C and the Euclidean projector over C as δ_C and $\Pi_C(x) := \arg\min_{s \in C} \|x - s\|$, respectively. Let $f : \mathbb{X} \to (-\infty, +\infty]$ be a proper closed and convex function. We denote the effective domain of f and the proximal mapping of f at $x \in \mathbb{X}$ as $\mathrm{dom}(f) := \{x \in \mathbb{X} \mid f(x) < +\infty\}$ and $\mathrm{Prox}_f(x) := \arg\min_{y \in \mathbb{X}} \{f(y) + \frac{1}{2}\|y - x\|^2\}$, respectively. The Fenchel conjugate function of f is defined as $f^*(x) := \sup_{y \in \mathbb{X}} \langle y, x \rangle - f(y)$. Let $D \subseteq \mathbb{X}$ be a nonempty closed convex set. We call $\boldsymbol{T} : D \to D$ a nonexpansive operator if $\|\boldsymbol{T}(x) - \boldsymbol{T}(y)\| \le \|x - y\|$ for any $x, y \in D$. For a nonexpansive operator $\boldsymbol{T} : D \to D$, we denote the set of its fixed points as $\mathrm{Fix}(\boldsymbol{T}) := \{x \in D \mid x = \boldsymbol{T}(x)\}$.

2. Preliminaries

In this section, we first introduce the model of the Wasserstein barycenter problem. Then we propose a linear-time procedure for the linear system involved in solving the Wasserstein barycenter problem.

2.1 Wasserstein Barycenter Problem

Consider the following discrete probability distribution with finite support points:

$$\mathcal{P} := \left\{ (a_i, \boldsymbol{q}_i) \in \mathbb{R}_+ \times \mathbb{R}^d : i = 1, \cdots, m \right\},$$

where $\{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_m\}$ are the support points and $\mathbf{a} := (a_1, \cdots, a_m)$ is the associated probability vector satisfying $\sum_{i=1}^m a_i = 1$. Given two discrete distributions

$$\mathcal{P}^u = \{(a_i^u, \boldsymbol{q}_i^u) : i = 1, \ldots, m_u\}, \qquad \mathcal{P}^v = \{(a_i^v, \boldsymbol{q}_i^v) : i = 1, \ldots, m_v\},$$

the p-th power of the p-Wasserstein distance between $\mathcal{P}^u$ and $\mathcal{P}^v$ is defined as the optimal objective function value of the following optimal transport problem:

$$\begin{array}{rll} (\mathcal{W}_p(\mathcal{P}^u, \mathcal{P}^v))^p := & \min\limits_{X \in \mathbb{R}^{m_u \times m_v}} & \langle X, \mathcal{D}(\mathcal{P}^u, \mathcal{P}^v) \rangle \\ \text{s.t.} & X^\top \mathbf{1}_{m_u} = \mathbf{a}^v, & \\ & X \mathbf{1}_{m_v} = \mathbf{a}^u, & \\ & X \ge \mathbf{0}, & \end{array} \tag{1}$$

where $\mathcal{D}(\mathcal{P}^u, \mathcal{P}^v) \in \mathbb{R}^{m_u \times m_v}$ is the distance matrix with $\mathcal{D}(\mathcal{P}^u, \mathcal{P}^v)_{ij} = \|\boldsymbol{q}_i^u - \boldsymbol{q}_j^v\|_p^p$ and $p \geq 1$. Based on the Wasserstein distance, Agueh and Carlier (2011) proposed the Wasserstein barycenter problem. Specifically, given a collection of discrete probability distributions $\left\{\mathcal{P}^t\right\}_{t=1}^T$ with $\mathcal{P}^t = \left\{\left(a_i^t, \boldsymbol{q}_i^t\right) : i = 1, \cdots, m_t\right\}$, a p-Wasserstein barycenter $\mathcal{P}^c := \{(a_i^c, \boldsymbol{q}_i^c) : i = 1, \cdots, m\}$ with m support points is a minimizer of the following optimization problem:

$$\min_{\mathcal{P}^c \in \Xi_m^p\left(\mathbb{R}^d\right)} \sum_{t=1}^T \omega_t \left(\mathcal{W}_p(\mathcal{P}^c, \mathcal{P}^t)\right)^p, \tag{2}$$

where $\Xi_m^p\left(\mathbb{R}^d\right)$ denotes the set of all discrete probability distributions on $\mathbb{R}^d$ with m support points and finite p-th moment, and the weight vector $(\omega_1, \cdots, \omega_T)$ satisfies $\sum_{t=1}^T \omega_t = 1$ and $\omega_t > 0, t = 1, \cdots, T$. Note that problem (2) is generally nonconvex in the support locations and weights. It requires finding the optimal support $\mathcal{Q}^c := \{\boldsymbol{q}_i^c : i = 1, \ldots, m\}$ and the optimal weight vector $\mathbf{a}^c$ simultaneously.

In many real applications, the support $\mathcal{Q}^c$ can be specified empirically. As a result, one only needs to compute the optimal weight vector $\mathbf{a}^c$ via solving the optimization problem (2) with finite specified supports. In this paper, we focus on the WBP with specified supports. From now on, we assume that the support $\mathcal{Q}^c$ is given. Under this setting, the WBP can be formulated as the following linear program:

$$\begin{array}{ll} \min\limits_{\substack{\mathbf{a}^c \in \mathbb{R}^m,\ X^t \in \mathbb{R}^{m \times m_t} \\ t=1,\ldots,T}} & \sum_{t=1}^T \left\langle D^t, X^t \right\rangle \\ \text{s.t.} & \left(X^t\right)^\top \mathbf{1}_m = \mathbf{a}^t, \quad t = 1, \cdots, T, \\ & X^t \mathbf{1}_{m_t} = \mathbf{a}^c, \quad t = 1, \cdots, T, \\ & X^t \geq 0, \quad t = 1, \cdots, T, \\ & \langle \mathbf{a}^c, \mathbf{1}_m \rangle = 1, \end{array} \tag{3}$$

where $D^t := \omega_t \mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right)$. We can write (3) in the following standard-form linear program:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^N} & \langle c, x \rangle + \delta_K(x) \\ \text{s.t.} & Ax = b, \end{array} \tag{4}$$

where

$$x = \left(\text{vec}(X^1); \ldots; \text{vec}(X^T); \mathbf{a}^c\right), \qquad c = \left(\text{vec}(D^1); \ldots; \text{vec}(D^T); \mathbf{0}_m\right),$$

and K is the nonnegative orthant with compatible dimension. Moreover,

$$M := \sum_{t=1}^T m_t + T(m-1) + 1, \qquad N := m \sum_{t=1}^T m_t + m, \tag{5}$$

and

$$A := \begin{bmatrix} A_1 & \mathbf{0} \\ A_2 & A_3 \\ \mathbf{0} & \mathbf{1}_m^\top \end{bmatrix} \in \mathbb{R}^{M \times N}, \qquad b := \left(\mathbf{a}^1; \ldots; \mathbf{a}^T; \mathbf{0}_{m-1}; \ldots; \mathbf{0}_{m-1}; 1\right) \in \mathbb{R}^M, \tag{6}$$

with

$$A_1 = \text{diag}\left(I_{m_1} \otimes \mathbf{1}_m^\top, \ldots, I_{m_T} \otimes \mathbf{1}_m^\top\right),$$

$$A_2 = \mathrm{diag}\left(\mathbf{1}_{m_1}^\top \otimes [\mathbf{0}_{m-1}, I_{m-1}], \ldots, \mathbf{1}_{m_T}^\top \otimes [\mathbf{0}_{m-1}, I_{m-1}]\right), \qquad A_3 = -\mathbf{1}_T \otimes [\mathbf{0}_{m-1}, I_{m-1}].$$

Here A_2 and A_3 are obtained by removing one redundant row from each block of the barycenter marginal constraints. By Ge et al. (2019, Lemma 3.1), A has full row rank, and the feasible region of (4) is equivalent to that of (3).

The dual problem of (4) is

$$\min_{y\in\mathbb{R}^M, s\in\mathbb{R}^N} \left\{-\langle b, y\rangle + \delta_K^*(-s) \mid A^*y + s = c\right\}, \tag{7}$$

where $\delta_K^*(\cdot)$ is the support function of K. The KKT conditions associated with (4) and (7) are given by

$$A^*y + s = c, \quad Ax = b, \quad K \ni x \perp s \in K, \tag{8}$$

where $x \perp s$ means x is perpendicular to s, i.e., $\langle x, s\rangle = 0$.

Operator-splitting algorithms such as ADMM can be applied to the dual problem (7) to compute the Wasserstein barycenter. The main computational bottleneck is solving the linear system $AA^*y = R$ for a given $R \in \mathbb{R}^M$. Since AA^* is an $M \times M$ matrix and M can be very large for the WBP, even a single Cholesky factorization may be computationally prohibitive. In practice, iterative methods such as the conjugate gradient method are often used for such large linear systems. In the next subsection, we derive a linear-time procedure for solving this structured system exactly.

2.2 A Linear-Time Complexity Procedure for Solving $AA^*y = R$

We now derive an efficient procedure for solving $AA^*y = R$, where A is defined in (6) and $R \in \mathbb{R}^M$ is given. The resulting procedure has complexity $O\left(Tm + \sum_{t=1}^T m_t\right)$. Let $M_1 := \sum_{t=1}^T m_t$ and $M_2 := T(m-1)$. By direct calculation,

$$AA^* = \begin{bmatrix} A_1 & \mathbf{0} \\ A_2 & A_3 \\ \mathbf{0} & \mathbf{1}_m^\top \end{bmatrix} \begin{bmatrix} A_1^* & A_2^* & \mathbf{0} \\ \mathbf{0} & A_3^* & \mathbf{1}_m \end{bmatrix} = \begin{bmatrix} E_1 & E_2 & \mathbf{0} \\ E_2^* & E_3 + E_4 & E_5 \\ \mathbf{0} & E_5^* & m \end{bmatrix}, \tag{9}$$

where

1. $E_1 := A_1A_1^* = \mathrm{diag}\left(mI_{m_1}, \ldots, mI_{m_T}\right) \in \mathbb{R}^{M_1\times M_1}$,
2. $E_2 := A_1A_2^* = \mathrm{diag}\left(\mathbf{1}_{m_1}\mathbf{1}_{m-1}^\top, \ldots, \mathbf{1}_{m_T}\mathbf{1}_{m-1}^\top\right) \in \mathbb{R}^{M_1\times M_2}$,
3. $E_3 := A_2A_2^* = \mathrm{diag}\left(m_1I_{m-1}, \ldots, m_TI_{m-1}\right) \in \mathbb{R}^{M_2\times M_2}$,
4. $E_4 := A_3A_3^* = (\mathbf{1}_T\mathbf{1}_T^\top) \otimes I_{m-1} \in \mathbb{R}^{M_2\times M_2}$,
5. $E_5 := A_3\mathbf{1}_m = -\mathbf{1}_T \otimes \mathbf{1}_{m-1} \in \mathbb{R}^{M_2}$.

Writing

$$y = (y_1; y_2; y_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R}, \qquad R = (R_1; R_2; R_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R},$$

the linear system $AA^*y = R$ becomes

$$\begin{bmatrix} E_1 & E_2 & \mathbf{0} \\ E_2^* & E_3 + E_4 & E_5 \\ \mathbf{0} & E_5^* & m \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}. \tag{10}$$

For the block representation, let

$$y_1 = (y_1^1; \ldots; y_1^T), \quad R_1 = (R_1^1; \ldots; R_1^T), \qquad y_2 = (y_2^1; \ldots; y_2^T), \quad R_2 = (R_2^1; \ldots; R_2^T),$$

where $y_1^t, R_1^t \in \mathbb{R}^{m_t}$ and $y_2^t, R_2^t \in \mathbb{R}^{m-1}$ for $t = 1, \ldots, T$. The next proposition provides an explicit solution to (10). This is the dual form of a coupled extension of Romero's projection formula for the OT problem (Romero, 1990). A short proof of Proposition 1 is provided in Appendix A, while the corresponding primal projection formula is given in Appendix B.

Proposition 1 *Consider $A \in \mathbb{R}^{M \times N}$ defined in (6). Given $R \in \mathbb{R}^M$, the solution y to $AA^*y = R$ in the form (10) is given by*

$$y_2^t = \frac{1}{m_t}\left(\hat{y}_2^t - \hat{y}_2^a\right), \quad t = 1, \ldots, T, \tag{11}$$

$$y_1^t = \frac{R_1^t}{m} - \frac{\mathbf{1}_{m-1}^\top y_2^t}{m}\mathbf{1}_{m_t}, \quad t = 1, \ldots, T, \tag{12}$$

$$y_3 = \frac{1}{m}\left(R_3 + \mathbf{1}_{M_2}^\top y_2\right), \tag{13}$$

where

$$\hat{y}_2^t := R_2^t + \left(\mathbf{1}_{m-1}^\top R_2^t - \mathbf{1}_{m_t}^\top R_1^t + R_3\right)\mathbf{1}_{m-1}, \quad t = 1, \ldots, T,$$

and

$$\hat{y}_2^a := \sum_{t=1}^T \frac{\bar{m}}{m_t}\hat{y}_2^t, \qquad \bar{m} := \left(1 + \sum_{t=1}^T \frac{1}{m_t}\right)^{-1}.$$

The explicit formula in Proposition 1 leads to Algorithm 1.

Algorithm 1 A linear-time solver for $AA^*y = R$

Input: $R \in \mathbb{R}^M$.
Step 1. Compute y_2 by (11).
Step 2. Compute y_1 by (12).
Step 3. Compute y_3 by (13).
Output: $y = (y_1; y_2; y_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R}$.

Proposition 2 *The computational complexity of Algorithm 1 in terms of flops is $7Tm + 3\sum_{t=1}^T m_t + O(T)$.*

Proof The complexity of each step in Algorithm 1 can be summarized as follows:

$$\begin{aligned} &\text{Step1} : 6Tm + \textstyle\sum_{t=1}^T m_t + O(T), \\ &\text{Step2} : Tm + 2\textstyle\sum_{t=1}^T m_t + O(T), \\ &\text{Step3} : O(T). \end{aligned}$$

Summing them up, we obtain that the overall computational complexity of Algorithm 1 is $7Tm + 3\sum_{t=1}^{T} m_t + O(T)$. ■

Remark 3 *The structured linear system studied here is closely related to that in Ge et al. (2019). In the interior-point method of Ge et al. (2019), each Newton step requires solving $ADA^*y = R$, where D is a positive diagonal matrix that changes across iterations. By exploiting the block structure of A, Ge et al. (2019) solves such systems with flop complexity $O(m^2\sum_{t=1}^{T} m_t + Tm^3)$ or $O(m\sum_{t=1}^{T} m_t^2 + \sum_{t=1}^{T} m_t^3)$, depending on the relative sizes of m and m_t. Since D is general, the resulting block matrices are iteration-dependent and require additional blockwise matrix inversions. In contrast, the ADMM/HPR framework considered in this paper gives rise to the fixed system $AA^*y = R$. This fixed structure allows a further simplification of the block decomposition and yields the explicit formula in Proposition 1. Consequently, the system can be solved exactly by a reusable linear-time procedure with complexity $O(Tm + \sum_{t=1}^{T} m_t)$.*

Remark 4 *For the standard OT problem* (1)*, the explicit primal formula for the Euclidean projection onto the affine set of matrices with prescribed marginals is classical and was given by Romero (1990); related results and applications appear in Khoury (1998); Glunt et al. (1998); Calvillo and Romero (2016); Zass and Shashua (2006); Bauschke et al. (2021). The corresponding structured linear system considered here is a dual representation of this known projection after removing one redundant marginal equality. For completeness, Appendix C recalls the classical primal projection formula and discusses its relation to the $O(m_u + m_v)$ solution procedure for the resulting dual linear system.*

3. A Halpern-Peaceman-Rachford Algorithm

We start this section by introducing the HPR algorithm, which applies the Halpern iteration to the PR splitting method for finding a solution $w^* \in \mathbb{X}$ to the following inclusion problem:

$$0 \in \boldsymbol{M}_1(w) + \boldsymbol{M}_2(w), \tag{14}$$

where $\boldsymbol{M}_1 : \mathbb{X} \rightrightarrows \mathbb{X}$, $\boldsymbol{M}_2 : \mathbb{X} \rightrightarrows \mathbb{X}$, and $\boldsymbol{M}_1 + \boldsymbol{M}_2$ are all maximal monotone operators. We denote the zeros of $\boldsymbol{M}_1 + \boldsymbol{M}_2$ as $\mathrm{Zer}(\boldsymbol{M}_1 + \boldsymbol{M}_2)$.

For a maximal monotone operator $\boldsymbol{M} : \mathbb{X} \rightrightarrows \mathbb{X}$, the resolvent $\boldsymbol{J}_{\boldsymbol{M}} := (\boldsymbol{I} + \boldsymbol{M})^{-1}$ and reflected resolvent $\boldsymbol{R}_{\boldsymbol{M}} := 2\boldsymbol{J}_{\boldsymbol{M}} - \boldsymbol{I}$ are single-valued and, respectively, firmly nonexpansive and nonexpansive (Minty, 1962; Bauschke and Combettes, 2011), where $\boldsymbol{I}$ is the identity operator. For $\sigma > 0$ and an initial point $\eta^0 \in \mathbb{X}$, the PR iteration is

$$\eta^{k+1} = \boldsymbol{T}_{\sigma}^{\mathrm{PR}}(\eta^k) := \boldsymbol{R}_{\sigma\boldsymbol{M}_1} \circ \boldsymbol{R}_{\sigma\boldsymbol{M}_2}(\eta^k), \quad \forall k \geq 0, \tag{15}$$

where $\boldsymbol{T}_{\sigma}^{\mathrm{PR}}$ is nonexpansive and each fixed point $\eta^* \in \mathrm{Fix}(\boldsymbol{T}_{\sigma}^{\mathrm{PR}})$ yields a solution $w^* = \boldsymbol{J}_{\sigma\boldsymbol{M}_2}(\eta^*)$ of (14) (Lions and Mercier, 1979). Applying the Halpern iteration (Halpern, 1967) to the PR operator with weight $1/(k+2)$ gives the HPR method in Algorithm 2:

Algorithm 2 An HPR algorithm for the problem (14)

Input: $\eta^0 \in \mathbb{X}$. For $k = 0, 1, \ldots$

$$\begin{aligned} w^{k+1} &= \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^k), \\ x^{k+1} &= \boldsymbol{J}_{\sigma \boldsymbol{M}_1}\left(2w^{k+1} - \eta^k\right), \\ v^{k+1} &= 2x^{k+1} - \left(2w^{k+1} - \eta^k\right), \\ \eta^{k+1} &= \frac{1}{k+2}\eta^0 + \frac{k+1}{k+2}v^{k+1}. \end{aligned}$$

If $\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}}) \neq \varnothing$, then $\eta^k \to \Pi_{\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}})}(\eta^0)$ by (Wittmann, 1992, Theorem 2). Moreover, Lieder (2021) established the following residual bound for the Halpern iteration in Algorithm 2:

$$\|\eta^k - \boldsymbol{T}_\sigma^{\mathrm{PR}}(\eta^k)\| \leq \frac{2\|\eta^0 - \bar{\eta}\|}{k+1}, \quad \forall k \geq 0, \quad \forall \bar{\eta} \in \mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}}).$$

The convergence of the iterates in Algorithm 2 is summarized in Corollary 5.

Corollary 5 *Let $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ and $\boldsymbol{M}_1 + \boldsymbol{M}_2$ be maximal monotone operators on $\mathbb{X}$ satisfying $\mathrm{Zer}(\boldsymbol{M}_1 + \boldsymbol{M}_2) \neq \emptyset$. Let $\{w^k, x^k, v^k, \eta^k\}_{k=1}^\infty$ be the sequence generated by Algorithm 2. Let $\eta^* = \Pi_{\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}})}(\eta^0)$ and $w^* = \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^*)$. Then we have*

$$\eta^k \to \eta^*, \quad v^k \to \eta^*, \quad x^k \to w^*, \text{ and } w^k \to w^*,$$

and w^* is a solution to the problem (14)*.*

Proof The convergence of the sequence $\{\eta^k\}_{k=0}^\infty$ follows from (Wittmann, 1992, Theorem 2). Since for all $k \geq 0$,

$$\eta^{k+1} = \frac{1}{k+2}\eta^0 + \frac{k+1}{k+2}v^{k+1},$$

we can directly obtain $v^k \to \eta^*$. It follows from $x^{k+1} - w^{k+1} = 1/2(v^{k+1} - \eta^k), \forall k \geq 0$ in Algorithm 2 that

$$x^k - w^k \to 0. \tag{16}$$

Since $\boldsymbol{J}_{\sigma \boldsymbol{M}_2}$ is nonexpansive and $w^{k+1} = \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^k)$, we directly obtain $\lim_{k\to\infty} x^k = \lim_{k\to\infty} w^k = \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^*) = w^*$. This completes the proof. ■

Remark 6 *We use the Halpern iteration for its simplicity and optimal $O(1/k)$ global residual bound (Lieder, 2021; Park and Ryu, 2022). Other acceleration techniques, such as the fast Krasnosel'skii–Mann iteration (Boţ and Nguyen, 2023), may also be incorporated into the PR framework; see Sun et al. (2025) for related developments. Halpern-type acceleration for other operator-splitting methods has also been studied; see, e.g., Tran-Dinh and Luo (2021).*

3.1 An HPR Algorithm for the Two-Block Convex Programming Problems

Now, we consider the following two-block convex optimization problem with linear constraints:

$$\begin{array}{ll} \min_{y\in\mathbb{Y},s\in\mathbb{Z}} & f_1(y)+f_2(s) \\ \text{subject to} & B_1y+B_2s=c, \end{array} \tag{17}$$

where $f_1:\mathbb{Y}\to(-\infty,+\infty]$ and $f_2:\mathbb{Z}\to(-\infty,+\infty]$ are proper closed convex functions, which may take extended values; $B_1:\mathbb{Y}\to\mathbb{X}$ and $B_2:\mathbb{Z}\to\mathbb{X}$ are two given linear mappings, and $c\in\mathbb{X}$ is a given vector. It is clear that the optimization problem (7), which is the dual problem of the LP, is a special case of (17). Given $\sigma>0$, the augmented Lagrange function corresponding to (17) is

$$L_\sigma(y,s;x):=f_1(y)+f_2(s)+\langle x,B_1y+B_2s-c\rangle+\frac{\sigma}{2}\|B_1y+B_2s-c\|^2,$$

where $x\in\mathbb{X}$ is the multiplier. The dual problem of (17) is

$$\max_{x\in\mathbb{X}}\left\{-f_1^*(-B_1^*x)-f_2^*(-B_2^*x)-\langle c,x\rangle\right\}, \tag{18}$$

where f_1^* and f_2^* are the Fenchel conjugates of f_1 and f_2, respectively; $B_1^*:\mathbb{X}\to\mathbb{Y}$ and $B_2^*:\mathbb{X}\to\mathbb{Z}$ are the adjoints of B_1 and B_2, respectively.

Since f_1 and f_2 are proper closed convex functions, there exist two self-adjoint and positive semidefinite operators Σ_{f_1} and Σ_{f_2} such that for all $y,\hat{y}\in\mathrm{dom}(f_1),\phi\in\partial f_1(y)$, and $\hat{\phi}\in\partial f_1(\hat{y})$,

$$f_1(y)\geq f_1(\hat{y})+\langle\hat{\phi},y-\hat{y}\rangle+\frac{1}{2}\|y-\hat{y}\|^2_{\Sigma_{f_1}}\text{ and }\langle\phi-\hat{\phi},y-\hat{y}\rangle\geq\|y-\hat{y}\|^2_{\Sigma_{f_1}},$$

and for all $s,\hat{s}\in\mathrm{dom}(f_2),\varphi\in\partial f_2(s)$, and $\hat{\varphi}\in\partial f_2(\hat{s})$

$$f_2(s)\geq f_2(\hat{s})+\langle\hat{\varphi},s-\hat{s}\rangle+\frac{1}{2}\|s-\hat{s}\|^2_{\Sigma_{f_2}}\text{ and }\langle\varphi-\hat{\varphi},s-\hat{s}\rangle\geq\|s-\hat{s}\|^2_{\Sigma_{f_2}}.$$

We make the following assumptions:

Assumption 1 *For $i=1,2$, the following conditions hold:*

(A1.1) $\mathrm{ri}(\mathrm{dom}\,f_i^*)\bigcap\mathrm{Range}(B_i^*)\neq\varnothing$,

(A1.2) $\mathrm{ri}\left(\mathrm{dom}(f_1^*\circ(-B_1^*))\right)\cap\mathrm{ri}\left(\mathrm{dom}(f_2^*\circ(-B_2^*))\right)\neq\varnothing$.

(A1.3) The solution set of the optimization problem (18) *is nonempty.*

Assumption 2 *Both $\Sigma_{f_1}+B_1^*B_1$ and $\Sigma_{f_2}+B_2^*B_2$ are positive definite.*

Under Assumption 1, $(y^*,s^*)\in\mathbb{Y}\times\mathbb{Z}$ is a solution to the optimization problem (17) and $x^*\in\mathbb{X}$ is a solution to the optimization problem (18) if and only if the following KKT system is satisfied:

$$B_1y^*+B_2s^*=c,\quad -B_2^*x^*\in\partial f_2(s^*),\quad -B_1^*x^*\in\partial f_1(y^*). \tag{19}$$

The residual mapping associated with the KKT system is

$$\mathcal{R}(y,s,x)=\begin{pmatrix} y-\mathrm{Prox}_{f_1}(y-B_1^*x)\\ s-\mathrm{Prox}_{f_2}(s-B_2^*x)\\ c-B_1y-B_2s\end{pmatrix},\quad (y,s,x)\in\mathbb{Y}\times\mathbb{Z}\times\mathbb{X}.$$

Then (y^*,s^*,x^*) satisfies the KKT system (19) if and only if $\mathcal{R}(y^*,s^*,x^*)=0$.

If we take $\boldsymbol{M}_1(w):=\partial(f_1^*\circ(-B_1^*))(w)+c$ and $\boldsymbol{M}_2(w):=\partial(f_2^*\circ(-B_2^*))(w)$, then the inclusion problem (14) is equivalent to the optimization problem (18). As a result, we can apply Algorithm 2 to solve (18), which is presented in Algorithm 3.

Algorithm 3 An HPR algorithm for solving the convex optimization problem (17)

1: Input: $y^0\in\mathrm{dom}(f_1)$, $x^0\in\mathbb{X}$, and $\sigma>0$.
2: Initialization: $\hat{x}^0:=x^0$, $\eta^0:=\hat{x}^0+\sigma(B_1y^0-c)$.
3: For $k=0,1,\ldots$
4: Step 1. $s^{k+1}=\arg\min_{s\in\mathbb{Z}}\{f_2(s)+\langle\eta^k,B_2s\rangle+\frac{\sigma}{2}\|B_2s\|^2\}$.
5: Step 2. $w^{k+1}=\eta^k+\sigma B_2s^{k+1}$.
6: Step 3. $y^{k+1}=\arg\min_{y\in\mathbb{Y}}\{f_1(y)+\langle\eta^k+2\sigma B_2s^{k+1},B_1y-c\rangle+\frac{\sigma}{2}\|B_1y-c\|^2\}$.
7: Step 4. $x^{k+1}=\eta^k+\sigma(B_1y^{k+1}-c)+2\sigma B_2s^{k+1}$.
8: Step 5. $v^{k+1}=\eta^k+2\sigma\left(B_1y^{k+1}+B_2s^{k+1}-c\right)$.
9: Step 6. $\eta^{k+1}=\frac{1}{k+2}\eta^0+\frac{k+1}{k+2}v^{k+1}$.

In Algorithm 3, if we define

$$\hat{x}^{k+1}:=\eta^{k+1}-\sigma(B_1y^{k+1}-c),\quad\forall k\geq0,\tag{20}$$

we can discard the sequences $\{\eta^k\}_{k=0}^\infty$, $\{w^k\}_{k=1}^\infty$, and $\{v^k\}_{k=1}^\infty$. This leads to Algorithm 4. We show the equivalence of Algorithm 2, Algorithm 3, and Algorithm 4 in Proposition 7.

Algorithm 4 An HPR algorithm for solving two-block convex optimization problem (17)

1: Input: $y^0\in\mathrm{dom}(f_1)$, $x^0\in\mathbb{X}$, and $\sigma>0$.
2: Initialization: $\hat{x}^0:=x^0$.
3: For $k=0,1,\ldots$
4: Step 1. $s^{k+1}=\arg\min_{s\in\mathbb{Z}}\{L_\sigma(y^k,s;\hat{x}^k)\}$.
5: Step 2. $x^{k+\frac{1}{2}}=\hat{x}^k+\sigma(B_1y^k+B_2s^{k+1}-c)$.
6: Step 3. $y^{k+1}=\arg\min_{y\in\mathbb{Y}}\left\{L_\sigma(y,s^{k+1};x^{k+\frac{1}{2}})\right\}$.
7: Step 4. $x^{k+1}=x^{k+\frac{1}{2}}+\sigma(B_1y^{k+1}+B_2s^{k+1}-c)$.
8: Step 5. $\hat{x}^{k+1}=\left(\frac{1}{k+2}\hat{x}^0+\frac{k+1}{k+2}x^{k+1}\right)+\frac{\sigma}{k+2}\left[(B_1y^0-c)-(B_1y^{k+1}-c)\right]$.

Proposition 7 *Suppose that Assumption 1 and Assumption 2 hold. Then all the subproblems in Algorithm 3 and Algorithm 4 are solvable. Take* $\boldsymbol{M}_1(w)=\partial(f_1^*\circ(-B_1^*))(w)+c$ *and* $\boldsymbol{M}_2(w)=\partial(f_2^*\circ(-B_2^*))(w)$. *The following statements hold:*

1. *With the same initial point η^0 and penalty parameter σ, Algorithms 2 and 3 generate the same sequence $\{w^k, x^k, v^k, \eta^k\}_{k=1}^{\infty}$.*

2. *With the same initial points $y^0 \in \mathrm{dom}(f_1)$ and $x^0 \in \mathbb{X}$ and penalty parameter σ, Algorithms 3 and 4 generate the same sequence $\{s^k, y^k, x^k\}_{k=1}^{\infty}$.*

Proof See Appendix D. ■

Now, we prove the convergence of Algorithm 4 for solving the optimization problem (17).

Corollary 8 *Assume that Assumption 1 and Assumption 2 hold. Let $\{y^k, s^k, x^k\}_{k=1}^{\infty}$ and $\{\hat{x}^k, x^{k+\frac{1}{2}}\}_{k=0}^{\infty}$ be the sequences generated by Algorithm 4. Then, we have*

$$y^k \to y^*, \quad s^k \to s^*, \quad x^k \to x^*, \quad x^{k+\frac{1}{2}} \to x^*, \text{ and } \hat{x}^k \to x^*,$$

where (y^*, s^*) is a solution to the problem (17) and x^* is a solution to the problem (18).

Proof

Assumption 1 ensures that the KKT system (19) has a solution (Rockafellar, 1970, Corollary 28.2.2). By Proposition 7 and Corollary 5, the corresponding sequences in Algorithm 3, initialized with $\eta^0 = x^0 + \sigma(B_1 y^0 - c)$, satisfy $\eta^k, v^k \to \eta^*$ and $w^k, x^k \to x^*$.

Denote $\widehat{f}_2(s) := f_2(s) + \frac{\sigma}{2}\|B_2 s\|^2$, which is strongly convex by Assumption 2. Its conjugate $\widehat{f}_2^*$ has a continuous gradient (Rockafellar, 1970, Theorems 25.5 and 26.3), and the s-subproblem gives (Rockafellar, 1970, Theorem 23.5)

$$s^{k+1} = \nabla \widehat{f}_2^*(-B_2^* \eta^k), \quad \forall k \geq 0.$$

Hence $s^k \to s^*$. The same argument applied to the y-subproblem yields $y^k \to y^*$, since $\eta^k + 2\sigma B_2 s^{k+1}$ converges. Algorithm 3 also gives

$$B_1 y^{k+1} + B_2 s^{k+1} - c = \frac{v^{k+1} - \eta^k}{2\sigma} \to 0, \quad -B_2^* w^{k+1} \in \partial f_2(s^{k+1}), \quad -B_1^* x^{k+1} \in \partial f_1(y^{k+1}).$$

Taking limits and using the closedness of the subdifferential graphs (Rockafellar, 1970, Theorem 24.4) shows that (y^*, s^*, x^*) satisfies the KKT system (19). Finally, the multiplier updates in Algorithm 4 imply $\hat{x}^k, x^{k+\frac{1}{2}} \to x^*$, completing the proof.

■

Remark 9 *Assumption 2 ensures the solvability of the subproblems of Algorithm 4 and is sufficient for the convergence of the sequence $\{(y^k, s^k)\}_{k\geq 1}$. Assumption 2 holds automatically if either B_i is injective or f_i is strongly convex (for $i = 1, 2$). Han et al. (2018) gives an example where the sufficient conditions just mentioned fail to hold, but Assumption 2 can still hold.*

3.2 Iteration Complexity Analysis

In this subsection, we analyze the iteration complexity of Algorithm 4 for solving the optimization problem (17). Given the sequence $\{y^k, s^k\}_{k=1}^{\infty}$ generated by Algorithm 4, define

$$h(y^{k+1}, s^{k+1}) := f_1(y^{k+1}) + f_2(s^{k+1}) - f_1(y^*) - f_2(s^*), \forall k \geq 0, \tag{21}$$

where (y^*, s^*) is the limit point of the sequence $\{y^k, s^k\}_{k=1}^{\infty}$. Define

$$d_{f_1}(w) := f_1^*(-B_1^* w) + \langle w, c\rangle \quad \text{and} \quad d_{f_2}(w) := f_2^*(-B_2^* w), \quad \forall w \in \mathbb{X}.$$

We first prove the following lemma before we establish the iteration complexity of the HPR.

Lemma 10 *Suppose that Assumption 1 and Assumption 2 hold. Take $\boldsymbol{M}_1(w) = \partial(f_1^* \circ (-B_1^*))(w) + c$ and $\boldsymbol{M}_2(w) = \partial(f_2^* \circ (-B_2^*))(w)$. Let $\{y^k, s^k\}_{k=1}^{\infty}$ be the sequence generated by Algorithm 4 and the sequence $\{w^k, x^k, v^k, \eta^k\}_{k=1}^{\infty}$ be generated by Algorithm 3 with $\eta^0 := \hat{x}^0 + \sigma(B_1 y^0 - c)$. Then, we have for all $k \geq 0$,*

$$\begin{aligned} 4\sigma h(y^{k+1}, s^{k+1}) = & -4\sigma\left(d_{f_1}(x^{k+1}) + d_{f_2}(w^{k+1}) - d_{f_1}(w^*) - d_{f_2}(w^*)\right) \\ & + \|\eta^k - v^{k+1}\|^2 + 2\langle \eta^k - v^{k+1}, v^{k+1}\rangle. \end{aligned} \tag{22}$$

Moreover, the following bounds hold for all $k \geq 0$:

$$4\sigma h(y^{k+1}, s^{k+1}) \leq \|\eta^k - (\eta^* - w^*)\|^2 - \|v^{k+1} - (\eta^* - w^*)\|^2, \tag{23}$$

and

$$h(y^{k+1}, s^{k+1}) \geq \langle B_1 y^{k+1} + B_2 s^{k+1} - c, -w^*\rangle, \tag{24}$$

where $\eta^* = \Pi_{\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}})}(\eta^0)$ and $w^* = \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^*)$.

Proof See Appendix E. ■

Now, we are ready to prove the convergence rate of the HPR algorithm for solving the optimization problem (17).

Theorem 11 *Suppose that Assumption 1 and Assumption 2 hold. Take $\boldsymbol{M}_1(w) = \partial(f_1^* \circ (-B_1^*))(w) + c$ and $\boldsymbol{M}_2(w) = \partial(f_2^* \circ (-B_2^*))(w)$. Let $\{y^{k+1}, s^{k+1}, x^{k+1}\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 4. Let (y^*, s^*, x^*) be the limit point of $\{(y^{k+1}, s^{k+1}, x^{k+1})\}_{k=0}^{\infty}$. Then for all $k \geq 0$, we have the following bounds:*

$$\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \leq \frac{1}{k+1}\left(\frac{\sigma\|B_2^*\| + 1}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|B_1 y^0 - B_1 y^*\right\|\right)\right). \tag{25}$$

Moreover, we have for $k \geq 0$ that

$$h(y^{k+1}, s^{k+1}) \geq -\frac{1}{k+1}\left(\frac{\|x^*\|}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|B_1 y^0 - B_1 y^*\right\|\right)\right)$$

and

$$\begin{aligned} h(y^{k+1}, s^{k+1}) \leq & \frac{1}{\sigma(k+1)}\left(\|x^0 - x^*\| + \sigma\|B_1 y^0 - B_1 y^*\|\right) \\ & \times \left(\|x^0 - x^*\| + \sigma\|B_1 y^0 - B_1 y^*\| + \|x^*\|\right). \end{aligned}$$

Proof Consider the corresponding sequence $\{w^k, v^k, \eta^k\}_{k=1}^{\infty}$ generated by Algorithm 3 with $\eta^0 = x^0 + \sigma(B_1 y^0 - c)$. By Corollaries 5 and 8 and Proposition 7,

$$\eta^* = \Pi_{\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}})}(\eta^0) = x^* + \sigma(B_1 y^* - c), \qquad w^* = x^*.$$

By the Halpern residual bound (Lieder, 2021, Theorem 2.1), for every $k \geq 0$,

$$\|v^{k+1} - \eta^k\| \leq \frac{2\|\eta^0 - \eta^*\|}{k+1}. \tag{26}$$

Hence, Algorithm 3 yields

$$\|B_1 y^{k+1} + B_2 s^{k+1} - c\| = \frac{\|v^{k+1} - \eta^k\|}{2\sigma} \leq \frac{\|\eta^0 - \eta^*\|}{\sigma(k+1)}. \tag{27}$$

The subproblem optimality conditions and multiplier updates also give

$$\begin{aligned} -B_1^* x^{k+1} \in \partial f_1(y^{k+1}), \qquad -B_2^* w^{k+1} \in \partial f_2(s^{k+1}), \\ x^{k+1} - w^{k+1} = \sigma(B_1 y^{k+1} + B_2 s^{k+1} - c). \end{aligned}$$

Therefore, by the nonexpansiveness of Prox_{f_2} and the definition of $\mathcal{R}$,

$$\begin{aligned} \|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| &\leq (\sigma\|B_2^*\| + 1)\|B_1 y^{k+1} + B_2 s^{k+1} - c\| \\ &\leq \frac{(\sigma\|B_2^*\| + 1)\|\eta^0 - \eta^*\|}{\sigma(k+1)}. \end{aligned}$$

For the objective bounds, nonexpansiveness of $\boldsymbol{T}_\sigma^{\mathrm{PR}}$ and the convex-combination update for η^k imply, by induction, that

$$\|v^{k+1} - \eta^*\| \leq \|\eta^k - \eta^*\| \leq \|\eta^0 - \eta^*\|, \qquad k \geq 0. \tag{28}$$

Using (23), the difference-of-squares identity, and (26), we obtain

$$\begin{aligned} h(y^{k+1}, s^{k+1}) &\leq \frac{1}{4\sigma}\left\langle \eta^k + v^{k+1} - 2(\eta^* - w^*), \eta^k - v^{k+1} \right\rangle \\ &\leq \frac{(\|\eta^0 - \eta^*\| + \|x^*\|)\|\eta^0 - \eta^*\|}{\sigma(k+1)}. \end{aligned}$$

Similarly, (24) and (27) give

$$h(y^{k+1}, s^{k+1}) \geq -\frac{\|x^*\|\|\eta^0 - \eta^*\|}{\sigma(k+1)}.$$

The stated bounds then follow from

$$\|\eta^0 - \eta^*\| = \|x^0 - x^* + \sigma(B_1 y^0 - B_1 y^*)\| \leq \|x^0 - x^*\| + \sigma\|B_1 y^0 - B_1 y^*\|.$$

■

Remark 12 *The non-ergodic $O(1/k)$ convergence result stated in Theorem 11 was established in the initial version of this manuscript in 2022. This result further motivated the subsequent development of the more general Halpern-accelerated pADMM framework in Sun et al. (2025). In particular, Sun et al. (2025) extends the HPR scheme considered here to a more general semi-proximal framework. Specifically, in the notation of Sun et al. (2025), setting $\mathcal{T}_1 = 0$, $\mathcal{T}_2 = 0$, $\alpha = 2$, and $\rho = 2$ in Algorithm 3.1 recovers the HPR scheme considered in this paper. Accordingly, the non-ergodic convergence rates in Theorem 11 are a specialization of (Sun et al., 2025, Theorem 3.7(a)) to the case without proximal terms.*

Remark 13 *Here, we make some remarks.*

1. *We emphasize that $\|B_2^*\|$ is the spectral norm of B_2^* and $\|B_2^*\| = 1$ for the WBP.*
2. *Under Assumption 1, if all the subproblems in Algorithm 4 are solvable, the $O(1/k)$ bounds in Theorem 11 remain valid without Assumption 2. Indeed, for*

$$\eta^* = \Pi_{\mathrm{Fix}(\boldsymbol{T}_\sigma^{\mathrm{PR}})}(\eta^0), \qquad x^* = \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^*),$$

the subdifferential chain rule under (A1.1) yields a primal solution (y^*, s^*) satisfying

$$\eta^* = x^* + \sigma(B_1 y^* - c).$$

Thus, the proof of Theorem 11 applies to this primal–dual solution without requiring convergence of $\{(y^k, s^k)\}$.

3.3 A Fast Implementation of the HPR for Solving the WBP

In this subsection, we present a fast implementation of the HPR for solving the WBP. In particular, we show that, for the WBP, all subproblems of the HPR have a closed-form solution. We also analyze the per-iteration computational complexity of the HPR for solving the WBP. An HPR algorithm for solving the linear programming problem (7) is presented in Algorithm 5, which is a direct application of Algorithm 4.

Algorithm 5 An HPR algorithm for solving the dual linear program (7)

Input: $x^0 \in \mathbb{R}^N$, $y^0 \in \mathbb{R}^M$, and $\sigma > 0$.
Initialization: $\hat{x}^0 := x^0$, $s^0 := c - A^* y^0$.
For $k = 0, 1, \ldots$
Step 1. Compute $s^{k+1} = \Pi_K\left(c - A^* y^k - \hat{x}^k/\sigma\right)$.
Step 2. Compute $x^{k+\frac{1}{2}} = \hat{x}^k + \sigma\left(s^{k+1} + A^* y^k - c\right)$.
Step 3. Compute y^{k+1} by applying Algorithm 1 to solve the following linear system:

$$AA^* y = b/\sigma - A(x^{k+\frac{1}{2}}/\sigma + s^{k+1} - c). \tag{29}$$

Step 4. Compute $x^{k+1} = x^{k+\frac{1}{2}} + \sigma\left(s^{k+1} + A^* y^{k+1} - c\right)$.
Step 5. Compute $\hat{x}^{k+1} = \left(\frac{1}{k+2}\hat{x}^0 + \frac{k+1}{k+2}x^{k+1}\right) + \frac{\sigma}{k+2}\left[(A^* y^0 - c) - (A^* y^{k+1} - c)\right]$.

Now, we are ready to present the per-iteration computational complexity $O(m\sum_{t=1}^{T} m_t)$ of Algorithm 5 in the following lemma. This implies that the per-iteration computational complexity of the HPR algorithm is comparable to that of the IBP algorithm (Benamou et al., 2015) for solving the WBP.

Lemma 14 *The per-iteration computational complexity of Algorithm 5 in terms of flops is* $26m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$.

Proof The computational complexity of iteration k (for any $k \geq 0$) of Algorithm 5 can be summarized as follows:

Step 1. Since $s^{k+1} := \Pi_K(c - A^*y^k - \hat{x}^k/\sigma) = \max(0, c - A^*y^k - \hat{x}^k/\sigma)$, and the matrix A has $(2m-1)\sum_{t=1}^{T} m_t + T(m-1) + m$ non-zero entries, the complexity for updating s^{k+1} is $7m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$.

Step 3. The complexity of forming $R = \left(b/\sigma - A(x^{k+\frac{1}{2}}/\sigma + s^{k+1} - c)\right)$ is $6m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$ due to the structure of A. From Proposition 2, we know that the complexity for solving the equation $AA^*y = R$ is $O(Tm + \sum_{t=1}^{T} m_t)$. Therefore, the complexity of Step 3 is $6m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$.

Steps 2, 4, and 5. A direct calculation shows that the complexity of these three steps is $13m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$.

Summing them up, we know the per-iteration computational complexity of Algorithm 5 in terms of flops is $26m\sum_{t=1}^{T} m_t + O(Tm + \sum_{t=1}^{T} m_t)$. ■

Now, we can obtain the overall computational complexity of the HPR for solving the WBP.

Theorem 15 *Let* $\{y^k, s^k, x^k\}_{k=0}^{\infty}$ *be the sequence generated by Algorithm 5. For any given tolerance* $\varepsilon > 0$*, any integer* $k \geq 0$ *satisfying*

$$k \geq \left\lceil \frac{1}{\varepsilon}\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|s^0 - s^*\right\|\right)\right)\right\rceil - 1$$

is sufficient for HPR to return an iterate for the WBP such that

$$\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \leq \varepsilon,$$

where (x^*, s^*) *is the limit point of the sequence* $\{x^k, s^k\}_{k=0}^{\infty}$*. In particular, the overall computational complexity of HPR to achieve this accuracy in terms of flops is*

$$O\left(\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|s^0 - s^*\right\|\right)\right)\frac{m\sum_{t=1}^{T} m_t}{\varepsilon}\right).$$

Proof Since $B_2 = I$, it follows from Theorem 11 that, for the linear program (7), we have

$$\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \leq \frac{1}{k+1}\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|s^0 - s^*\right\|\right)\right), \quad \forall k \geq 0.$$

Therefore, $\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \leq \varepsilon$ if

$$k \geq \left\lceil \frac{1}{\varepsilon} \left(\frac{1+\sigma}{\sigma} \left(\left\| x^0 - x^* \right\| + \sigma \left\| s^0 - s^* \right\| \right) \right) \right\rceil - 1.$$

According to Lemma 14, the per-iteration computational complexity in terms of the flops of the HPR algorithm for solving the WBP is $O(m \sum_{t=1}^{T} m_t)$. The overall computational complexity of HPR to achieve this accuracy in terms of flops is

$$O\left(\left(\frac{1+\sigma}{\sigma} \left(\left\| x^0 - x^* \right\| + \sigma \left\| s^0 - s^* \right\| \right) \right) \frac{m \sum_{t=1}^{T} m_t}{\varepsilon} \right).$$

This completes the proof. ■

Remark 16 *Actually, we can reformulate the primal LP model of the WBP* (4) *equivalently as*

$$\begin{array}{ll} \min_{x \in \mathbb{R}^N, z \in \mathbb{R}^N} & (\langle c, x \rangle + \delta_K(x)) + \delta_\Omega(z) \\ \text{s.t.} & x - z = 0, \end{array} \tag{30}$$

where δ_Ω is the indicator function of the set $\Omega := \{z \in \mathbb{R}^N \mid Az - b = 0\}$. Since the optimization problem (30) *is in the form of* (17)*, the HPR algorithm can be applied to solve it. A key step in the HPR algorithm for solving* (30) *is to compute the projection of a given point $\widehat{z} \in \mathbb{R}^N$ onto the set Ω, which is given by*

$$\Pi_\Omega(\widehat{z}) = \widehat{z} - A^*(AA^*)^{-1}(A\widehat{z} - b).$$

The projection $\Pi_\Omega(\widehat{z})$ can be efficiently computed by applying Algorithm 1. As a consequence, we can also obtain the $O((m \sum_{t=1}^{T} m_t)/\varepsilon)$ computational complexity of the HPR algorithm for solving the WBP in terms of the primal objective function value gap. In this paper, we prefer to apply the HPR algorithm to solve the dual problem (7) *partially because it can be more memory efficient.*

Remark 17 (Extension to the free-support case) *For the free-support WBP* (2)*, the barycenter support locations $\mathcal{Q}^c$ are also decision variables, and the resulting problem is generally nonconvex. A standard strategy is alternating minimization (Cuturi and Doucet, 2014; Ye and Li, 2014; Ye et al., 2017; Yang et al., 2021; Ge et al., 2019): with $\mathcal{Q}^c$ fixed, one solves the fixed-support WBP* (3)*; with the transport plans $\{X^t\}_{t=1}^T$ and barycentric weights $\mathbf{a}^c$ fixed, the support locations are updated by solving a simple quadratic minimization problem. The main computational bottleneck is the repeated solution of the fixed-support LP subproblem, which can be expensive for large-scale instances. The HPR method developed in this paper is designed for this challenging subproblem and can serve as an efficient subroutine in free-support barycenter algorithms.*

4. Numerical Experiments

In this section, we present the numerical performance of the HPR algorithm for solving the WBP with fixed supports on both synthetic and real data sets. We use the 2-Wasserstein

distance in all our experiments. We compare the performance of the restarted implementation of HPR algorithm with the fast-ADMM, IBP (Benamou et al., 2015; Schmitzer, 2019), the commercial software Gurobi, and the tailored interior-point method MAAIPM (Ge et al., 2019). All the numerical experiments in this paper are obtained by running MATLAB R2022a on a desktop with Intel(R) Core i7-10700HQ CPU @2.90GHz and 32GB of RAM.

4.1 Implementation Details

We use Algorithm 1 to solve the linear systems arising in ADMM and refer to the resulting method as fast-ADMM. For HPR and fast-ADMM, we use the following stopping criterion based on the relative KKT residual:

$$\mathrm{KKT}_{\mathrm{res}} = \max\left\{\frac{\|b - Ax\|}{1 + \|b\|}, \frac{\|\min(x, 0)\|}{1 + \|x\|}, \frac{\|A^*y + s - c\|}{1 + \|c\| + \|s\|}, \frac{\|s - \Pi_K(s - x)\|}{1 + \|x\| + \|s\|}\right\} \leq 10^{-5}. \quad (31)$$

In our implementation, this residual is checked every 50 iterations. For ADMM, we follow Chen et al. (2021), where the dual step size γ can be chosen from $(0, 2)$. Based on our numerical tests, we set $\gamma = 1.9$ as the default value for fast-ADMM.

In the numerical experiments below, we use a restarted implementation of HPR, referred to as restarted HPR. The restart strategy resets the Halpern anchor to the latest iterate and is described, together with an ablation study and further discussion, in Appendix F. We also consider a hybrid method, called restarted HPR-hybrid, which uses fast-ADMM in the early stage and switches to restarted HPR afterward:

$$\begin{cases} \text{Run fast-ADMM}, & \text{if } k \leq 800 \text{ and } \mathrm{KKT}_{\mathrm{res}} \geq 2 \times 10^{-4}, \\ \text{Run restarted HPR}, & \text{otherwise}, \end{cases}$$

where k denotes the iteration number.

For IBP, the regularization parameters are chosen from $\{0.01, 0.001\}$ for the synthetic data sets and $\{0.01, 0.001, 0.0005\}$ for real data sets, respectively. If sample distributions share the same ground cost $\{\mathcal{D}(\mathcal{P}^c, \mathcal{P}^t)\}_{t=1}^T$, then we run the IBP implemented in the POT toolbox,[1] which is a standard and highly efficient numerical solver for the WBP. Otherwise, we use the MATLAB code of the IBP implemented by Yang et al. (2021). For convenience, we call them POT and IBP, respectively. When the regularization parameter is less than 0.01, we will run both the IBP and its stabilized version simultaneously, and will only report the faster results in terms of running time. We adopt the default stopping criteria in the POT and IBP and set $\mathrm{Tol}_{\mathrm{IBP}} = 10^{-6}$.

In this paper, we compare with the interior-point method (IPM) in Gurobi v9.50 with an academic license. We use the default stopping criteria with tolerance 10^{-6}, and disable presolve and crossover, since IPM was the most efficient Gurobi method in our tests, presolve did not improve performance, and no basic solution was needed. For MAAIPM, we use the fixed-support implementation in Ge et al. (2019) with the primal-dual gap tolerance 5×10^{-5} adopted in that work. We also tested using the relative KKT residual (31) as the stopping criterion for MAAIPM, but this can terminate the method after only a few iterations while

1. The POT toolbox is available at `https://pythonot.github.io`.

the primal-dual gap remains large. Therefore, for MAAIPM, the KKT residual is reported only as an evaluation metric for the returned solution, not as the stopping rule.

We set the maximum number of iterations as 10000 for all algorithms. The maximum running time (for each run) of all algorithms is set to be **one hour**. For the evaluation of the quality of the solution, we report the $\mathrm{KKT}_{\mathrm{res}}$ for Gurobi, MAAIPM, fast-ADMM, restarted HPR, and restarted HPR-hybrid. To compare with IBP, we will use '**relative obj gap**' and '**relative primal feasibility error**'. Specifically, '**relative obj gap**' stands for the relative objective value gap which is defined by

$$\textbf{relative obj gap:} = \frac{\left|\sum_{t=1}^{T}\left\langle D^t, X^t\right\rangle - \sum_{t=1}^{T}\left\langle D^t, X_{\mathrm{g}}^t\right\rangle\right|}{\left|\sum_{t=1}^{T}\left\langle D^t, X_{\mathrm{g}}^t\right\rangle\right| + 1}, \tag{32}$$

where $\left(X^1, \ldots, X^T\right)$ is the solution obtained by the algorithm, $\left(X_{\mathrm{g}}^1, \ldots, X_{\mathrm{g}}^T\right)$ denotes the solution obtained by Gurobi, and $D^t = \omega_t \mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right), \quad t = 1, \ldots, T$. "**relative primal feasibility error**" denotes the value of

$$\max\left\{\frac{\|b - Ax\|}{1 + \|b\|}, \frac{\|\min(x, 0)\|}{1 + \|x\|}\right\}.$$

4.2 Experiments on Synthetic Data

We generate synthetic fixed-support WBP instances following Yang et al. (2021). The support points of $\{\mathcal{P}^t\}_{t=1}^T$ are sampled from a Gaussian mixture distribution, while both the distribution weights and the barycentric weights are sampled from the uniform distribution on $(0, 1)$ and then normalized. The barycenter support is obtained by applying k-means to all generated support points. The cost matrices are computed using squared Euclidean distances and normalized so that their maximum infinity norm equals one. Further details are provided in Appendix G. Since these random instances generally do not satisfy $\mathcal{D}(\mathcal{P}^c, \mathcal{P}^1) = \cdots = \mathcal{D}(\mathcal{P}^c, \mathcal{P}^T)$, POT is not applicable and is therefore omitted. In all experiments of this subsection, we set $d = 3$ and $m_1 = \cdots = m_T = m_t$, and report the average results over 10 independent trials for each triple (T, m, m_t).

The moderate-scale synthetic-data results are summarized in Table 3. We highlight several observations. First, both fast-ADMM and the HPR implementations employ the exact linear-time solve in Algorithm 1. Relative to fast-ADMM, the HPR-type methods generally require fewer iterations and less running time while attaining comparable KKT residuals. Compared with fast-ADMM,[2] restarted HPR reduces the average iteration number by about 47% and the average running time by about 44% over the tested instances. These results indicate a constant-factor improvement in empirical performance over the tested instances, while the difference in the worst-case convergence rates does not directly translate into a corresponding runtime advantage. In particular, fast-ADMM is slightly faster for $(m, m_t, T) = (800, 100, 100)$, taking 25.96 s compared with 27.76 s for restarted HPR. This observation provides further motivation for the restarted HPR-hybrid strategy. Indeed,

2. We also tested sGS-ADMM (Yang et al., 2021). In our tests, to obtain solutions of comparable quality, sGS-ADMM requires about 20%–30% more iterations than fast-ADMM, and its per-iteration time is about 70% longer.

restarted HPR-hybrid further improves the practical performance by using fast-ADMM as a warm start before switching to restarted HPR; on average, it reduces the running time of restarted HPR by about 16%. Second, compared with IBP, fast-ADMM, restarted HPR, and restarted HPR-hybrid obtain much smaller relative objective gaps under comparable primal feasibility errors, although IBP can be faster when a larger regularization parameter is used. Finally, MAAIPM is competitive on several moderate-size instances, but appears less robust on some randomly generated cases. In particular, among the ten independent trials, MAAIPM produces four runs with KKT residuals exceeding 5×10^{-5}, with the largest residual being 1.705×10^{-4} for $(m, m_t, T) = (100, 100, 800)$.

Note that Table 3 reports the restarted HPR and restarted HPR-hybrid implementations, which are not directly covered by Theorem 11. The theorem establishes a non-ergodic $O(1/k)$ rate for unrestarted HPR with a fixed anchor, which provides a theoretical basis for HPR between restarts. Further discussion of the restart implementation, its numerical effect, and its connection to linear convergence is provided in Appendix F.

In addition, Figure 1 reports the wall-clock histories of the relative objective gap and relative primal feasibility error on the synthetic instance $(m, m_t, T) = (100, 100, 100)$. Each method uses the stopping settings specified for it in Section 4.1. Fast-ADMM, restarted HPR, and restarted HPR-hybrid steadily reduce both measures, with restarted HPR-hybrid showing the best overall performance. In contrast, IBP is sensitive to the regularization parameter and yields larger objective gaps. Moreover, MAAIPM effectively reduces the objective gap, but its relative primal feasibility error is nonmonotone and temporarily deteriorates. A possible explanation is the accumulation of finite-precision errors when solving the linear systems via matrix inversions; such errors may be more visible in the reported feasibility residual because the MAAIPM implementation, based on a feasible interior-point framework, does not include an additional refinement procedure.

To further evaluate scalability and illustrate the practical impact of the complexity advantage discussed in Remark 3, we conduct two additional experiments in Figure 2, focusing on MAAIPM and the two HPR-type methods. In Figure 2(a), we fix $T = 100$ and vary $m = m_t \in \{200, 400, 800, 1600\}$. The advantage of the HPR-type methods becomes increasingly clear as the support size grows; when $m = m_t = 1600$, restarted HPR and restarted HPR-hybrid are about 3.59 and 4.59 times faster than MAAIPM, respectively. In Figure 2(b), we fix $m = 50$ and $m_t = 10$ and vary $T \in \{10000, 20000, 40000, 80000\}$. The HPR-type methods scale much better with respect to T; when $T = 80000$, restarted HPR and restarted HPR-hybrid are about 12.97 and 13.00 times faster than MAAIPM, respectively. These results are consistent with the complexity discussion in Remark 3: by exploiting the special structure of the linear system arising in our formulation, the HPR-type methods admit a linear-time solution procedure with complexity $O(Tm+\sum_{t=1}^{T} m_t)$, whereas MAAIPM solves more general iteration-dependent linear systems with higher per-iteration complexity. Moreover, restarted HPR and restarted HPR-hybrid consistently attain KKT residuals around 10^{-5}, whereas MAAIPM yields noticeably larger residuals in several large-scale cases.

m	m_t	T	Gurobi	MAAIPM	fast-ADMM	restarted HPR	restarted HPR-hybrid	IBP (0.01)	IBP (0.001)
					KKT$_{\mathbf{res}}$				
100	100	100	1.73E-07	1.82E-06	9.87E-06	9.88E-06	9.86E-06	-	-
100	100	200	1.40E-07	2.46E-06	9.91E-06	9.87E-06	9.89E-06	-	-
100	100	400	1.50E-07	2.10E-06	9.93E-06	9.87E-06	9.91E-06	-	-
100	100	800	1.72E-07	1.97E-05	9.94E-06	9.82E-06	9.90E-06	-	-
200	100	100	2.22E-07	8.44E-06	9.87E-06	9.87E-06	9.76E-06	-	-
400	100	100	1.43E-07	2.02E-06	9.88E-06	9.93E-06	9.85E-06	-	-
800	100	100	1.23E-07	1.52E-05	9.89E-06	9.90E-06	9.80E-06	-	-
100	200	100	2.12E-07	6.97E-06	9.89E-06	9.90E-06	9.84E-06	-	-
100	400	100	6.00E-08	2.19E-06	9.92E-06	9.84E-06	9.81E-06	-	-
100	800	100	3.16E-08	3.01E-06	9.90E-06	9.80E-06	9.39E-06	-	-
					Relative objective gap				
100	100	100	0	4.98E-05	8.41E-06	3.54E-05	6.37E-05	3.29E-01	1.60E-02
100	100	200	0	5.64E-05	1.62E-05	4.85E-05	7.22E-05	4.11E-01	1.92E-02
100	100	400	0	5.57E-05	1.29E-05	4.69E-05	7.24E-05	4.81E-01	2.23E-02
100	100	800	0	6.33E-05	1.77E-05	5.24E-05	6.98E-05	5.34E-01	2.40E-02
200	100	100	0	4.79E-05	1.71E-05	5.89E-05	8.68E-05	3.61E-01	2.32E-02
400	100	100	0	5.37E-05	3.23E-05	6.09E-05	1.13E-04	3.80E-01	3.10E-02
800	100	100	0	5.65E-05	3.94E-05	8.24E-05	1.82E-04	4.10E-01	4.06E-02
100	200	100	0	4.31E-05	1.39E-05	7.28E-05	1.10E-04	3.77E-01	1.56E-02
100	400	100	0	4.72E-05	1.78E-05	1.10E-04	1.66E-04	4.20E-01	1.77E-02
100	800	100	0	5.21E-05	2.52E-05	1.43E-04	2.04E-04	4.37E-01	1.75E-02
					Iterations				
100	100	100	39	30	4635	2110	1580	190	3060
100	100	200	54	31	5208	2292	1670	200	4220
100	100	400	48	37	5785	2530	1855	210	6120
100	100	800	45	40	6318	2612	1975	210	7530
200	100	100	49	38	4122	2375	1640	200	2110
400	100	100	51	48	3838	2725	1745	200	1490
800	100	100	55	67	3688	3080	2000	170	1270
100	200	100	36	25	4488	1978	1585	200	2440
100	400	100	47	27	3905	1765	1470	200	2400
100	800	100	49	27	3260	1648	1448	210	2070
					Relative primal feasibility error				
100	100	100	1.76E-09	8.25E-07	9.87E-06	9.88E-06	9.07E-06	5.89E-08	5.71E-07
100	100	200	1.70E-09	1.45E-06	9.90E-06	9.87E-06	8.74E-06	3.57E-08	6.65E-07
100	100	400	1.07E-09	1.81E-06	9.92E-06	9.87E-06	9.00E-06	3.50E-08	6.84E-07
100	100	800	3.88E-09	1.85E-05	9.94E-06	9.76E-06	8.98E-06	2.09E-08	1.16E-06
200	100	100	1.54E-09	8.11E-06	9.87E-06	9.86E-06	8.77E-06	1.41E-09	4.88E-07
400	100	100	1.57E-09	2.02E-06	9.88E-06	9.93E-06	9.08E-06	1.78E-09	3.27E-07
800	100	100	1.57E-09	1.52E-05	9.89E-06	9.90E-06	8.92E-06	6.39E-08	2.70E-07
100	200	100	1.52E-09	6.85E-06	9.89E-06	9.86E-06	8.99E-06	2.69E-09	5.67E-07
100	400	100	1.37E-09	2.02E-06	9.92E-06	9.61E-06	8.86E-06	4.19E-09	5.22E-07
100	800	100	2.62E-12	2.98E-06	9.90E-06	9.77E-06	8.39E-06	7.86E-10	4.20E-07
					Time (s)				
100	100	100	7.33	2.72	3.78	1.72	1.50	0.99	16.28
100	100	200	23.23	5.57	10.80	4.85	4.02	2.20	46.62
100	100	400	40.71	13.01	26.57	11.78	9.89	4.59	130.67
100	100	800	70.43	28.49	60.29	25.07	21.77	8.71	310.59
200	100	100	20.20	7.86	8.47	4.94	3.93	2.17	22.91
400	100	100	44.74	18.24	17.07	12.21	8.92	4.20	31.32
800	100	100	202.25	56.62	25.96	27.76	16.96	6.97	51.68
100	200	100	12.77	3.92	9.05	4.07	3.77	2.05	24.66
100	400	100	34.27	7.49	17.20	7.99	7.53	4.16	49.90
100	800	100	224.91	14.15	29.89	15.31	15.23	8.68	85.93

Table 3: Numerical results on Gaussian mixture distributions. The stopping criteria for all methods are specified in Section 4.1. In particular, MAAIPM is terminated with a primal-dual gap tolerance of 5×10^{-5}, whereas fast-ADMM, restarted HPR, and restarted HPR-hybrid are terminated with a relative KKT residual tolerance of 10^{-5}.

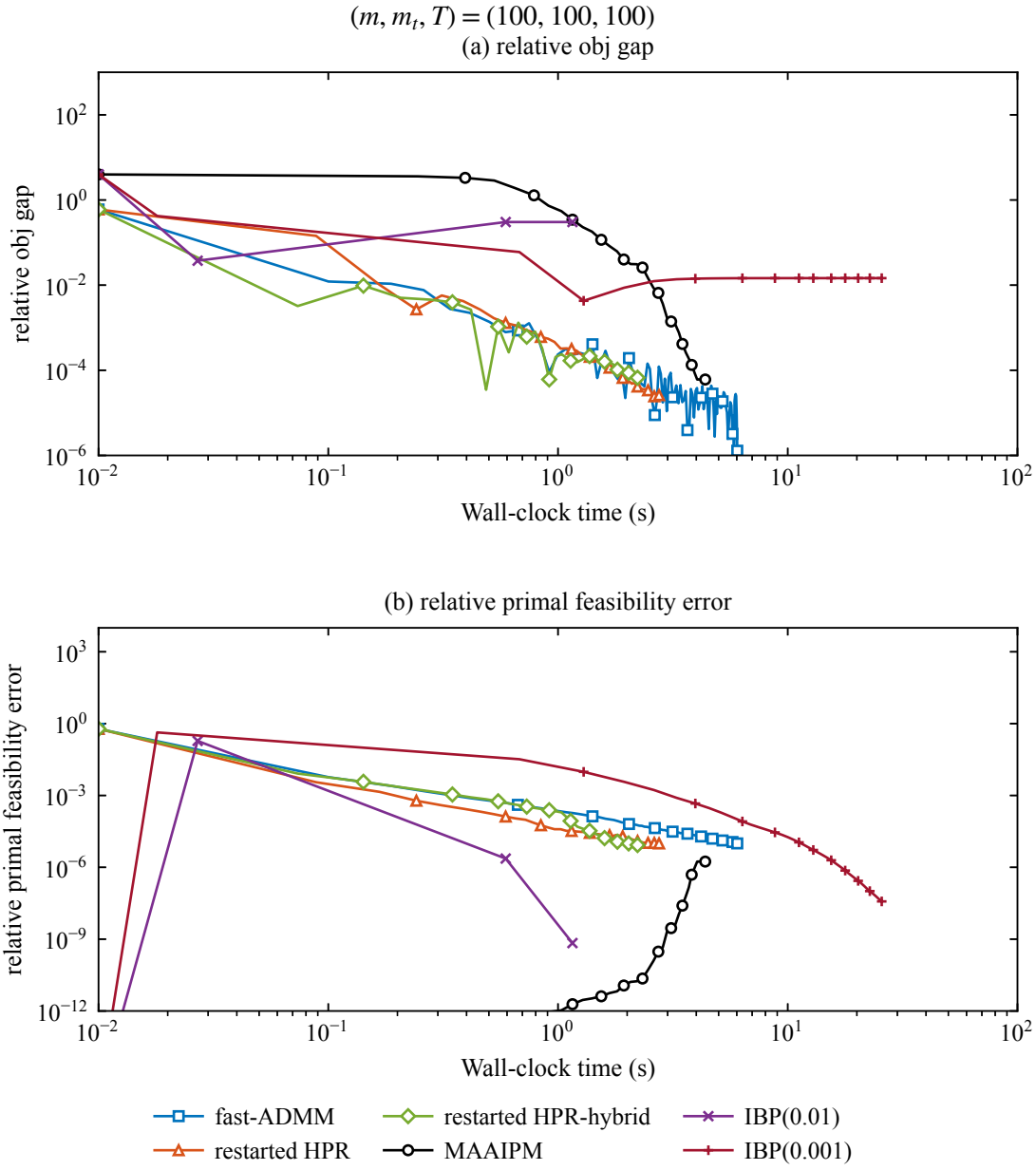
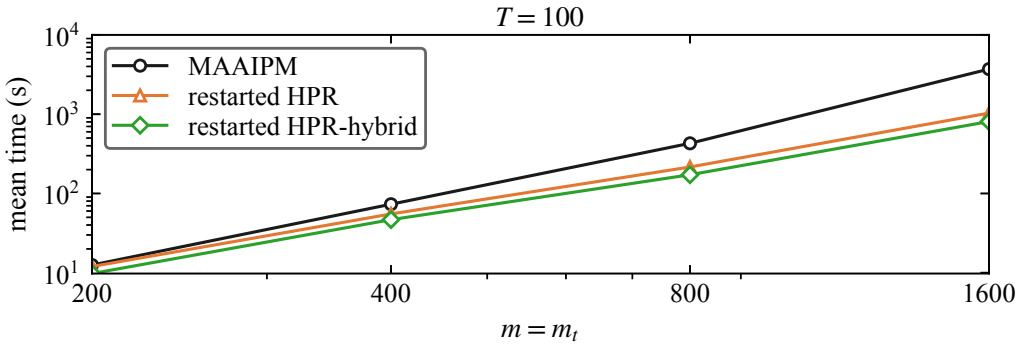
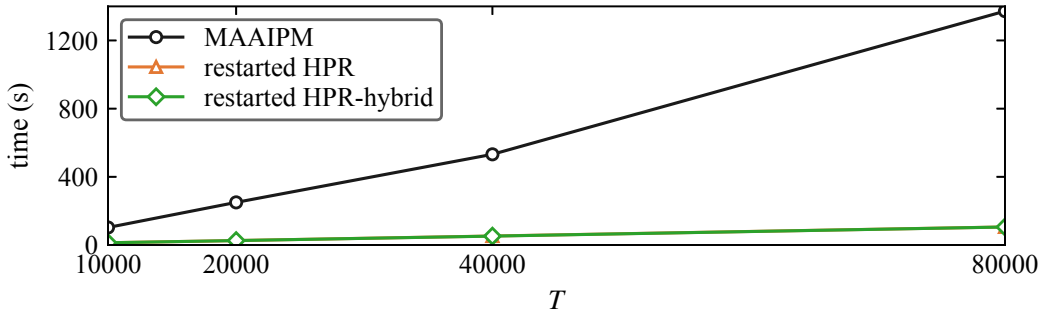


Figure 1: Wall-clock histories of the relative objective gap and relative primal feasibility error on the synthetic instance $(m, m_t, T) = (100, 100, 100)$. The stopping criteria for all methods are specified in Section 4.1. In particular, MAAIPM is terminated with a primal-dual gap tolerance of 5×10^{-5}, whereas fast-ADMM, restarted HPR, and restarted HPR-hybrid are terminated with a relative KKT residual tolerance of 10^{-5}.



$m = m_t$	$\mathbf{KKT_{res}}$ MAAIPM	restarted HPR	restarted HPR-hybrid
200	2.47E-06	9.89E-06	9.88E-06
400	2.07E-06	9.90E-06	9.91E-06
800	7.80E-05	9.94E-06	9.83E-06
1600	3.48E-06	9.93E-06	9.85E-06

(a) $m = m_t$ varies with $T = 100$



T	$\mathbf{KKT_{res}}$ MAAIPM	restarted HPR	restarted HPR-hybrid
10000	2.75E-04	9.88E-06	9.83E-06
20000	3.13E-03	9.84E-06	9.81E-06
40000	2.62E-03	9.86E-06	9.82E-06
80000	1.98E-04	9.89E-06	9.81E-06

(b) T varies with $m = 50$, $m_t = 10$

Figure 2: Large-support and large-T scalability comparisons on Gaussian mixture distributions. MAAIPM uses a primal-dual gap tolerance of 5×10^{-5}, whereas restarted HPR and restarted HPR-hybrid use a relative KKT residual tolerance of 10^{-5}.

4.3 Experiments on Real Data

We further evaluate restarted HPR and restarted HPR-hybrid on three real data sets: MNIST (LeCun et al., 1998), COIL20 (Nene et al., 1996), and Yale Face B (Georghiades et al., 2001). For MNIST, we construct one WBP instance for each digit $0, \ldots, 9$ by randomly selecting 50 images and resizing them to 56×56. For COIL20, we use three representative objects, Car, Duck, and Pig, each with 10 images resized to 64×64. For Yale Face B, we use two subjects, YaleB01 and YaleB02, and randomly select 5 images for each subject, resized to 96×84. All images are normalized so that their pixel values sum to one. The cost matrices are generated in the same way as in Subsection 4.2, and the barycentric weights are set uniformly, i.e., $\omega_t = 1/T$ for $t = 1, \ldots, T$. Table 4 summarizes the resulting problem dimensions.

Since Gurobi runs out of memory on these real-data instances, a Gurobi-based relative objective gap is not available. POT is included only as an entropy-regularized visual benchmark, since it does not directly provide the primal-dual variables needed to compute the KKT residual of the original unregularized LP. We omit fast-ADMM and MAAIPM from the real-data comparison because the synthetic experiments already provide controlled residual-based comparisons: restarted HPR-hybrid is consistently faster than fast-ADMM, and Figure 2(a) shows that MAAIPM becomes less competitive as the support size grows. Thus, the real-data experiments focus on the practical barycenter quality and running time of the proposed HPR-type methods on large image instances.

For MNIST, Figure 3 visualizes the Wasserstein barycenters obtained by POT and the HPR-type methods after 50 s and 100 s, respectively. The time-budgeted comparison shows that restarted HPR and restarted HPR-hybrid produce clear and smooth barycenters within a fixed computational budget. In particular, their visual quality is comparable to that of POT with a regularization parameter 0.001, while avoiding the regularization bias inherent in entropy-based methods.

For the COIL20 and Yale Face B data sets, we first use POT to compute benchmark barycenters. Among the tested POT settings, the regularization parameter 0.0005 yields the best visual results, whereas smaller parameters lead to numerical errors in the IBP implementation even with stabilization. We therefore focus on restarted HPR and restarted HPR-hybrid, and examine how these unregularized methods improve the barycenter quality as the iteration budget increases. The results are shown in Figures 4 and 5, and the corresponding computational times for fixed iteration budgets of 100, 200, and 300 are reported in Table 5. The figures show that restarted HPR and restarted HPR-hybrid progressively improve the visual quality and produce barycenters that are visually competitive with the POT benchmark within 300 iterations. Table 5 further shows that the HPR-type methods achieve these results within practical running times. In particular, restarted HPR-hybrid with 300 iterations is comparable to or faster than POT on most instances.

Data set	m	m_t	T
MNIST	3136	3136	50
Car	4096	4096	10
Duck	4096	4096	10
Pig	4096	4096	10
YaleB01	8064	8064	5
YaleB02	8064	8064	5

Table 4: Summary of real data sets.

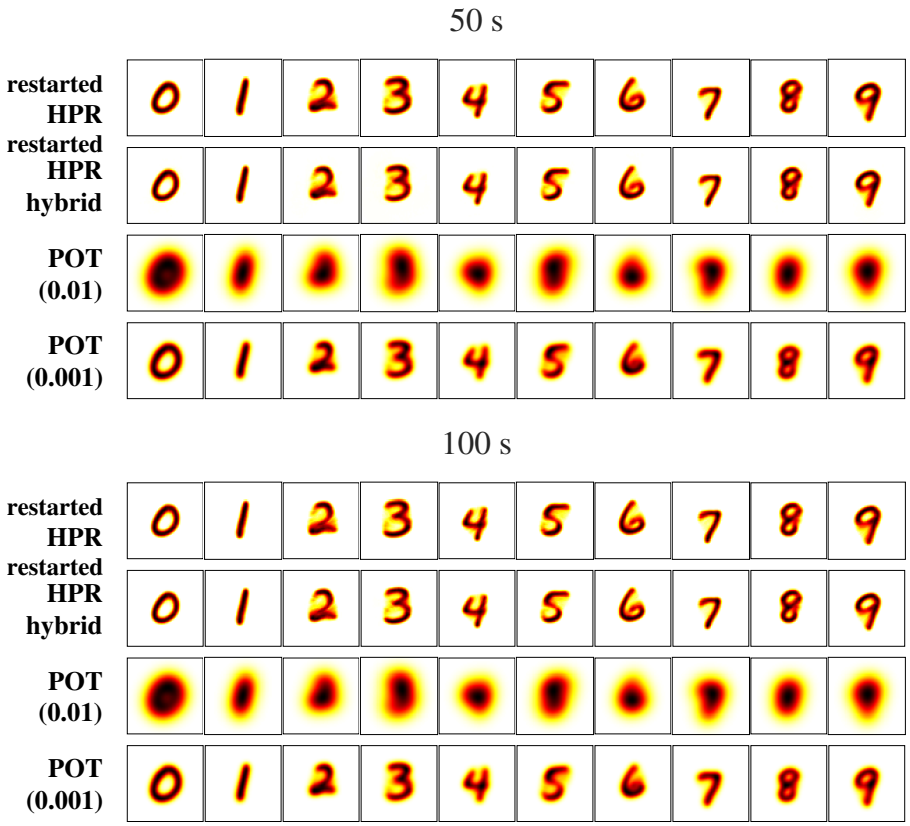


Figure 3: Barycenters obtained by POT and the HPR-type methods for 50 s and 100 s, respectively.

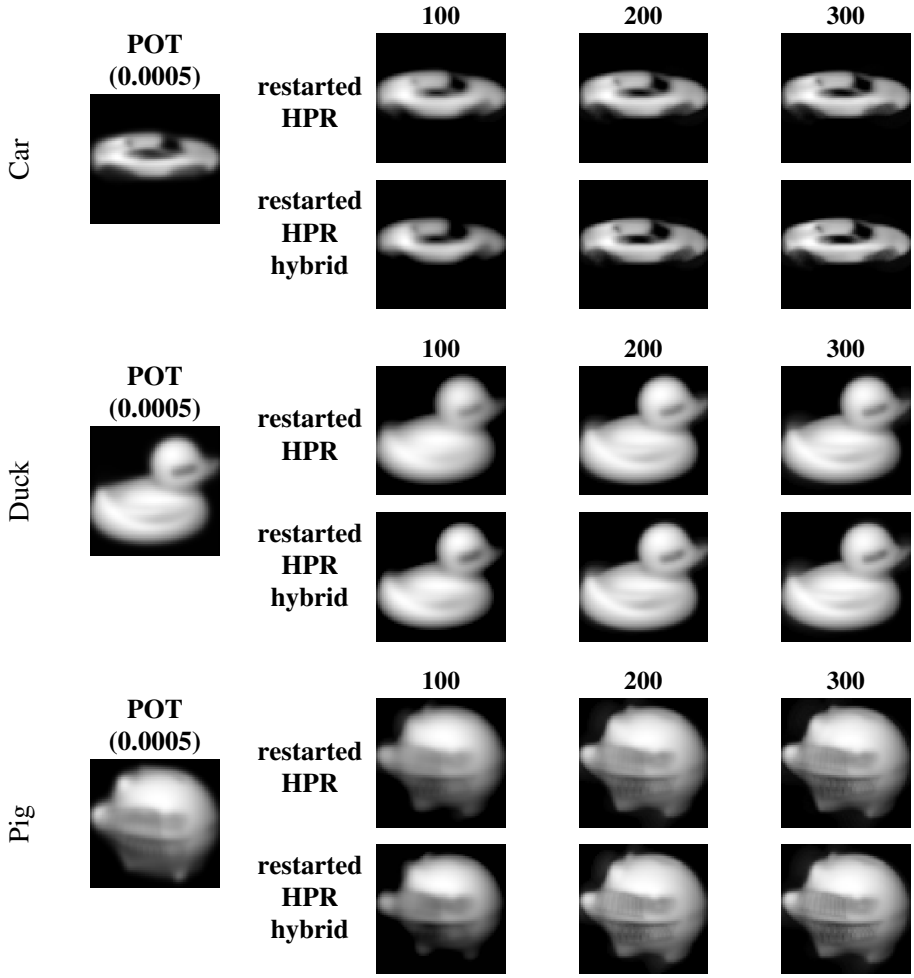


Figure 4: Barycenters obtained by POT and the HPR-type methods on the COIL20 data set for the 100th, 200th, and 300th iterations.

	POT (0.0005)	restarted HPR			restarted HPR-hybrid		
iter	-	100	200	300	100	200	300
Car	44.59	5.48	10.81	16.09	4.14	9.17	14.65
Duck	48.89	18.87	37.69	56.18	14.30	34.28	52.75
Pig	33.25	11.93	23.80	35.61	9.02	21.79	34.03
YaleB01	164.62	36.74	72.73	107.86	33.47	71.07	107.37
YaleB02	153.86	36.74	73.03	109.23	33.87	70.14	109.73

Table 5: Computational time of POT and the HPR-type methods for computing Wasserstein barycenters on the COIL20 and Yale Face B data sets under fixed iteration budgets. Unit: seconds.

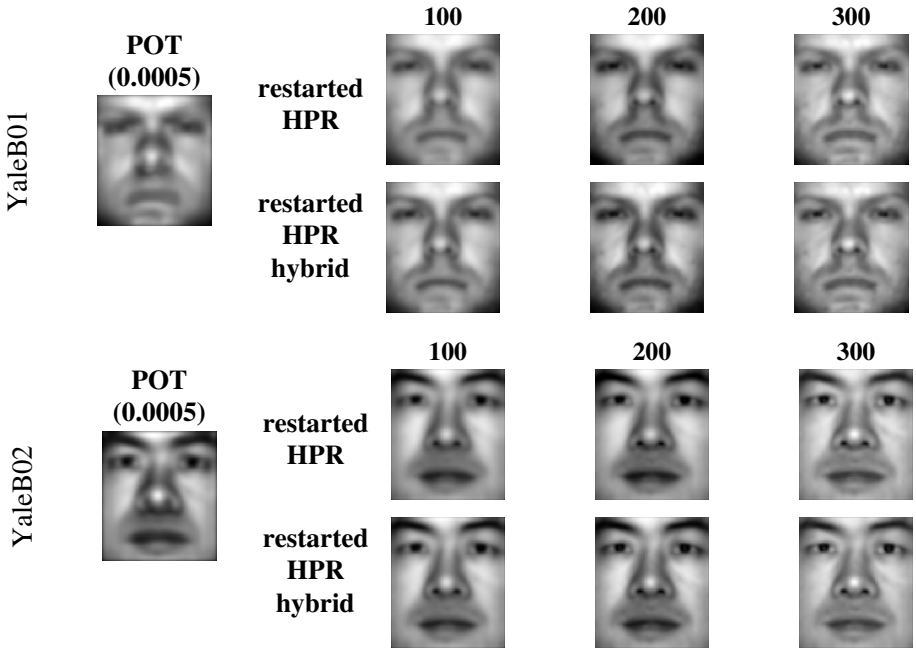


Figure 5: Barycenters obtained by POT and the HPR-type methods on the Yale Face B data set for the 100th, 200th, and 300th iterations.

4.4 Summary of Experiments

The numerical results show that the HPR-type methods are effective for the large-scale LPs arising from WBPs. On synthetic data, restarted HPR-hybrid is substantially faster than Gurobi on larger instances and scales favorably compared with MAAIPM. On real data, the HPR-type methods avoid the regularization sensitivity of IBP-type methods and produce high-quality barycenters within practical time or iteration budgets. Overall, restarted HPR-hybrid is a competitive choice when high accuracy and stable visual quality are desired.

5. Concluding Remark

In this paper, we study and implement an efficient HPR algorithm for solving the WBP. The HPR algorithm enjoys an $O(1/\varepsilon)$ non-ergodic iteration complexity with respect to the KKT residual, which is widely used as a practical stopping criterion for primal-dual algorithms. We also present a linear-time procedure, in terms of the linear-system size, for solving the structured linear system arising in the HPR subproblem. This linear system can be viewed as a dual representation of an affine projection that is a coupled extension of Romero's classical OT projection formula. Consequently, HPR achieves an $O(\mathrm{Dim(P)}/\varepsilon)$ computational complexity in terms of flops for obtaining an ε-optimal solution measured by the KKT residual, where Dim(P) denotes the number of variables in the LP formulation. Numerical experiments on synthetic data sets and real image data sets demonstrate the practical efficiency of the restarted HPR implementation for computing Wasserstein barycenters.

In the future, we will develop a highly efficient GPU solver based on the HPR algorithm for solving the WBP and OT. It has been shown that the proposed efficient procedure for solving the linear system can directly benefit the ADMM algorithm. We note that Gonçalves et al. (2017) proposed a dynamic regularized ADMM algorithm that enjoys a non-ergodic iteration complexity of $O(\varepsilon^{-1}\log(1/\varepsilon))$ in terms of the KKT residual. We will further study how to extend the proposed procedure to solve the subproblems of the dynamic regularized ADMM algorithms for solving the WBP. As an open research question, we will further investigate other acceleration techniques to see whether it is possible to design a new algorithm to solve the WBP with a better computational complexity than $O(\mathrm{Dim(P)}/\varepsilon)$.

Acknowledgments

The research of Yancheng Yuan is supported by the Hong Kong Polytechnic University under grant P0038284. The research of Defeng Sun is supported in part by the Hong Kong Research Grants Council under grant 15303720. The authors declare no competing interests.

Appendix A. The Proof of Proposition 1

Proof The first and last block rows of (10) give (12) and (13). If $m = 1$, the y_2 blocks are empty and these two formulas complete the proof. For $m > 1$, substituting them into the tth middle block row yields

$$\Big(I_{m-1} - \frac{1}{m}\mathbf{1}_{m-1}\mathbf{1}_{m-1}^\top\Big)\Big(m_t y_2^t + \sum_{s=1}^{T} y_2^s\Big) = R_2^t - \frac{\mathbf{1}_{m_t}^\top R_1^t - R_3}{m}\mathbf{1}_{m-1}.$$

Since

$$\Big(I_{m-1} - \frac{1}{m}\mathbf{1}_{m-1}\mathbf{1}_{m-1}^\top\Big)^{-1} = I_{m-1} + \mathbf{1}_{m-1}\mathbf{1}_{m-1}^\top,$$

applying this inverse and using $\mathbf{1}_{m-1}^\top\mathbf{1}_{m-1} = m-1$ gives

$$m_t y_2^t + \sum_{s=1}^{T} y_2^s = R_2^t + \Big(\mathbf{1}_{m-1}^\top R_2^t - \mathbf{1}_{m_t}^\top R_1^t + R_3\Big)\mathbf{1}_{m-1} = \hat{y}_2^t, \qquad t = 1, \ldots, T.$$

Dividing these identities by m_t and summing over t gives

$$\Big(1 + \sum_{t=1}^{T}\frac{1}{m_t}\Big)\sum_{s=1}^{T} y_2^s = \sum_{t=1}^{T}\frac{\hat{y}_2^t}{m_t}.$$

Hence

$$\sum_{s=1}^{T} y_2^s = \bar{m}\sum_{t=1}^{T}\frac{\hat{y}_2^t}{m_t} = \hat{y}_2^a,$$

and substitution into $m_t y_2^t + \sum_{s=1}^{T} y_2^s = \hat{y}_2^t$ yields (11). Together with (12) and (13), this proves the stated formula. Uniqueness follows from the full row rank of A. ■

Appendix B. Primal Projection Formula for the WBP

The primal projection associated with Proposition 1 is a coupled extension of Romero's classical projection formula for the OT problem (Romero, 1990). Let

$$\Omega := \{x \in \mathbb{R}^N : Ax = b\}, \qquad \widehat{z} = \big(\mathrm{vec}(Y^1); \ldots; \mathrm{vec}(Y^T); \widehat{\mathbf{a}}\big),$$

where $Y^t \in \mathbb{R}^{m \times m_t}$ and $\widehat{\mathbf{a}} \in \mathbb{R}^m$. The Euclidean projection of $\widehat{z}$ onto Ω is equivalently obtained by solving

$$\begin{aligned} \min_{\{X^t\}, \mathbf{a}^c} \quad & \frac{1}{2}\sum_{t=1}^{T} \|X^t - Y^t\|_F^2 + \frac{1}{2}\|\mathbf{a}^c - \widehat{\mathbf{a}}\|^2 \\ \text{s.t.} \quad & (X^t)^\top \mathbf{1}_m = \mathbf{a}^t, \qquad X^t \mathbf{1}_{m_t} = \mathbf{a}^c, \qquad t = 1, \ldots, T, \\ & \mathbf{1}_m^\top \mathbf{a}^c = 1. \end{aligned}$$

For a fixed shared marginal $\mathbf{a}^c$, each transport block is exactly the classical OT projection of Romero (1990); see Appendix C. Hence, for $t = 1, \ldots, T$,

$$X^t = Y^t + \frac{1}{m_t}(\mathbf{a}^c - Y^t \mathbf{1}_{m_t})\mathbf{1}_{m_t}^\top + \frac{1}{m}\mathbf{1}_m\big(\mathbf{a}^t - (Y^t)^\top \mathbf{1}_m\big)^\top + \frac{\mathbf{1}_m^\top Y^t \mathbf{1}_{m_t} - 1}{m m_t}\mathbf{1}_m \mathbf{1}_{m_t}^\top. \tag{33}$$

Moreover, Romero's formula gives

$$\|X^t - Y^t\|_F^2 = \frac{\|\mathbf{a}^c - Y^t\mathbf{1}_{m_t}\|^2}{m_t} + \frac{\|\mathbf{a}^t - (Y^t)^\top \mathbf{1}_m\|^2}{m} - \frac{(\mathbf{1}_m^\top Y^t \mathbf{1}_{m_t} - 1)^2}{m m_t}.$$

Since the last two terms are independent of $\mathbf{a}^c$, determining the common marginal reduces to

$$\min_{\substack{\mathbf{a} \in \mathbb{R}^m \\ \mathbf{1}_m^\top \mathbf{a} = 1}} \left\{ \|\mathbf{a} - \widehat{\mathbf{a}}\|^2 + \sum_{t=1}^{T} \frac{\|\mathbf{a} - Y^t \mathbf{1}_{m_t}\|^2}{m_t} \right\}.$$

Define

$$\bar{m} := \big(1 + \sum_{t=1}^{T} \frac{1}{m_t}\big)^{-1}, \qquad \bar{\mathbf{a}} := \bar{m}\big(\widehat{\mathbf{a}} + \sum_{t=1}^{T} \frac{Y^t \mathbf{1}_{m_t}}{m_t}\big). \tag{34}$$

Completing the square shows that the objective above is $\bar{m}^{-1}\|\mathbf{a} - \bar{\mathbf{a}}\|^2$ plus a constant independent of $\mathbf{a}$. Therefore, its unique minimizer is

$$\mathbf{a}^c = \bar{\mathbf{a}} + \frac{1 - \mathbf{1}_m^\top \bar{\mathbf{a}}}{m}\mathbf{1}_m. \tag{35}$$

Combining (33) and (35), we obtain

$$\Pi_\Omega(\widehat{z}) = \big(\mathrm{vec}(X^1); \ldots; \mathrm{vec}(X^T); \mathbf{a}^c\big). \tag{36}$$

Thus, the WBP projection is obtained by applying Romero's classical projection formula to each transport block and coupling these blockwise projections through the common marginal $\mathbf{a}^c$. In this sense, (36) is a coupled extension of Romero's classical OT projection formula. Finally,

$$\Pi_\Omega(\widehat{z}) = \widehat{z} - A^*(AA^*)^{-1}(A\widehat{z} - b),$$

so Proposition 1 gives the corresponding dual linear-system representation of this primal projection.

Appendix C. The Classical OT Projection and Its Dual Linear System

Using the probability vectors in (1), define the affine set

$$\Omega_{\mathrm{OT}} := \left\{ X \in \mathbb{R}^{m_u \times m_v} : X\mathbf{1}_{m_v} = \mathbf{a}^u, \quad X^\top \mathbf{1}_{m_u} = \mathbf{a}^v \right\}.$$

Since $\mathbf{1}_{m_u}^\top \mathbf{a}^u = \mathbf{1}_{m_v}^\top \mathbf{a}^v = 1$, the classical projection formula in Romero (1990, Corollary 2.1) gives, for any $Y \in \mathbb{R}^{m_u \times m_v}$,

$$\begin{aligned} \Pi_{\Omega_{\mathrm{OT}}}(Y) = Y + \frac{1}{m_v}(\mathbf{a}^u - Y\mathbf{1}_{m_v})\mathbf{1}_{m_v}^\top + \frac{1}{m_u}\mathbf{1}_{m_u}(\mathbf{a}^v - Y^\top \mathbf{1}_{m_u})^\top \\ + \frac{\mathbf{1}_{m_u}^\top Y \mathbf{1}_{m_v} - 1}{m_u m_v}\mathbf{1}_{m_u}\mathbf{1}_{m_v}^\top. \end{aligned} \tag{37}$$

Related results include the square case with unit row and column sums (Khoury, 1998; Glunt et al., 1998), projection methods for transportation polytopes (Calvillo and Romero, 2016), doubly stochastic normalization for spectral clustering (Zass and Shashua, 2006), and a Moore–Penrose inverse treatment that also covers inconsistent prescribed marginals (Bauschke et al., 2021); see also Bauschke et al. (2021, Remark 3.5) for the above formula.

To relate this classical primal projection formula to the dual linear system used here, we write the standard OT problem (1) in the form

$$\begin{aligned} \min_{x} \quad & \langle c, x \rangle + \delta_K(x) \\ \text{s.t.} \quad & Ax = b, \end{aligned} \tag{38}$$

where

$$x = \mathrm{vec}(X), \qquad c = \mathrm{vec}\left(\mathcal{D}(\mathcal{P}^u, \mathcal{P}^v)\right),$$

and K is the nonnegative orthant with compatible dimension. Moreover,

$$b = \left(\mathbf{a}^v; [\mathbf{0}_{m_u-1}, I_{m_u-1}]\mathbf{a}^u\right),$$

and

$$A = \begin{bmatrix} I_{m_v} \otimes \mathbf{1}_{m_u}^\top \\ \mathbf{1}_{m_v}^\top \otimes [\mathbf{0}_{m_u-1}, I_{m_u-1}] \end{bmatrix}.$$

By Dantzig and Thapa (2003, Lemma 7.1), A has full row rank. Here one row-sum equality has been removed because it is implied by the remaining row-sum equalities, all column-sum equalities, and the equality of the total masses. Thus $\{x : Ax = b\} = \{\mathrm{vec}(X) : X \in \Omega_{\mathrm{OT}}\}$. For $R = A\,\mathrm{vec}(Y) - b$, the projection can therefore be expressed in dual variables as

$$\mathrm{vec}\big(\Pi_{\Omega_{\mathrm{OT}}}(Y)\big) = \mathrm{vec}(Y) - A^* y, \qquad AA^* y = R. \tag{39}$$

For any given

$$R = (R_1; R_2) \in \mathbb{R}^{m_v} \times \mathbb{R}^{m_u - 1},$$

consider the linear system

$$AA^* y = R, \qquad y = (y_1; y_2) \in \mathbb{R}^{m_v} \times \mathbb{R}^{m_u - 1}.$$

Then its solution is given by

$$\begin{cases} y_1 = \dfrac{R_1}{m_u} + \dfrac{1}{m_v}\left(\dfrac{m_u - 1}{m_u}\mathbf{1}_{m_v}^\top R_1 - \mathbf{1}_{m_u-1}^\top R_2\right)\mathbf{1}_{m_v}, \\ y_2 = \dfrac{R_2}{m_v} + \dfrac{1}{m_v}\left(\mathbf{1}_{m_u-1}^\top R_2 - \mathbf{1}_{m_v}^\top R_1\right)\mathbf{1}_{m_u-1}. \end{cases} \tag{40}$$

Therefore, the system $AA^*y = R$ can be solved in $O(m_u + m_v)$ time. Substituting (40) into (39) recovers (37). Hence (39)–(40) provide a dual linear-system representation of the classical OT projection. The $O(m_u + m_v)$ cost refers to solving for y once R is available; forming the marginals of a dense Y and the projected matrix requires $O(m_u m_v)$ operations.

Appendix D. The Proof of Proposition 7

Proof Assumption 2 ensures that each subproblem objective is proper, closed, and strongly convex for every $\sigma > 0$, so each subproblem has a unique minimizer. We prove the first statement in the proposition by induction. Given $\eta^0 \in \mathbb{X}$, for $k = 0$, by the definition of s^1 in Algorithm 3, we have

$$0 \in \partial f_2(s^1) + B_2^*\eta^0 + \sigma B_2^* B_2 s^1.$$

Thus,

$$-B_2^*(\eta^0 + \sigma B_2 s^1) \in \partial f_2(s^1).$$

It follows from Rockafellar (1970, Theorem 23.5) that

$$s^1 \in \partial f_2^*(-B_2^*(\eta^0 + \sigma B_2 s^1)).$$

Hence,

$$-\sigma B_2 s^1 \in -\sigma B_2 \partial f_2^*(-B_2^*(\eta^0 + \sigma B_2 s^1)).$$

This means

$$\eta^0 \in \eta^0 + \sigma B_2 s^1 + \sigma\partial(f_2^* \circ (-B_2^*))(\eta^0 + \sigma B_2 s^1).$$

Since $\boldsymbol{M}_2 = \partial(f_2^* \circ (-B_2^*))$, we have

$$\eta^0 \in \eta^0 + \sigma B_2 s^1 + \sigma \boldsymbol{M}_2(\eta^0 + \sigma B_2 s^1),$$

This implies

$$w^1 := \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^0) = \eta^0 + \sigma B_2 s^1. \tag{41}$$

Similarly, by the definition of y^1, we have

$$0 \in \partial f_1(y^1) + B_1^*\left(\eta^0 + 2\sigma B_2 s^1 + \sigma(B_1 y^1 - c)\right).$$

It follows from Rockafellar (1970, Theorem 23.5) that

$$y^1 \in \partial f_1^*\left(-B_1^*\left(\eta^0 + 2\sigma B_2 s^1 + \sigma(B_1 y^1 - c)\right)\right).$$

Hence,

$$-\sigma B_1 y^1 \in -\sigma B_1 \partial f_1^*\left(-B_1^*\left(\eta^0 + 2\sigma B_2 s^1 + \sigma(B_1 y^1 - c)\right)\right).$$

This implies that

$$2(\eta^0+\sigma B_2 s^1)-\eta^0 \in 2(\eta^0+\sigma B_2 s^1)-\eta^0+\sigma B_1 y^1 \\ -\sigma B_1 \partial f_1^*\left(-B_1^*\left(\eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c)\right)\right).$$

That is

$$2(\eta^0+\sigma B_2 s^1)-\eta^0 \in \quad \eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c) \\ +\sigma\left(\partial(f_1^*\circ(-B_1^*))\left(\eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c)\right)+c\right).$$

Since $\boldsymbol{M}_1=\partial(f_1^*\circ(-B_1^*))+c$, we have

$$2(\eta^0+\sigma B_2 s^1)-\eta^0 \in \eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c)+\sigma \boldsymbol{M}_1\left(\eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c)\right).$$

This implies

$$x^1 := \boldsymbol{J}_{\sigma \boldsymbol{M}_1}\left(2w^1-\eta^0\right)=\eta^0+2\sigma B_2 s^1+\sigma(B_1 y^1-c). \tag{42}$$

From (41) and (42) we have

$$x^1-w^1=\sigma(B_1 y^1+B_2 s^1-c).$$

Hence

$$v^1 := \eta^0+2\left(x^1-w^1\right)=\eta^0+2\sigma(B_1 y^1+B_2 s^1-c). \tag{43}$$

It follows that the update of η^1 is the same in Algorithm 2 as in Algorithm 3. Hence, we prove the statement for $k=0$. Assume that the statement holds for some $k\geq 0$. Then the statement for $k+1$ follows similarly to the case $k=0$. Thus, the statement holds for any $k\geq 0$ by induction. This completes the first part.

Now we show the proof of the second part by induction. Let $\eta^0 := \hat{x}^0+\sigma(B_1 y^0-c)$. For $k=0$, from Algorithm 3, we have

$$\begin{aligned} s^1 &:= \arg\min_{s\in\mathbb{Z}}\left\{f_2(s)+\langle \eta^0, B_2 s\rangle+\tfrac{\sigma}{2}\|B_2 s\|^2\right\} \\ &= \arg\min_{s\in\mathbb{Z}}\left\{f_2(s)+\langle \hat{x}^0+\sigma(B_1 y^0-c), B_2 s\rangle+\tfrac{\sigma}{2}\|B_2 s\|^2\right\} \\ &= \arg\min_{s\in\mathbb{Z}}\left\{f_2(s)+\langle \hat{x}^0, B_1 y^0+B_2 s-c\rangle+\tfrac{\sigma}{2}\|B_1 y^0+B_2 s-c\|^2\right\}. \end{aligned}$$

For y^1, from Algorithm 3, we have

$$\begin{aligned} y^1 &:= \arg\min_{y\in\mathbb{Y}}\left\{f_1(y)+\langle \eta^0+2\sigma B_2 s^1, B_1 y-c\rangle+\tfrac{\sigma}{2}\|B_1 y-c\|^2\right\} \\ &= \arg\min_{y\in\mathbb{Y}}\left\{\begin{array}{l} f_1(y)+\langle \hat{x}^0+\sigma(B_1 y^0-c)+2\sigma B_2 s^1, B_1 y-c\rangle \\ \qquad +\dfrac{\sigma}{2}\|B_1 y-c\|^2 \end{array}\right\} \\ &= \arg\min_{y\in\mathbb{Y}}\left\{\begin{array}{l} f_1(y)+\langle \hat{x}^0+\sigma(B_1 y^0+B_2 s^1-c), B_1 y+B_2 s^1-c\rangle \\ \qquad +\dfrac{\sigma}{2}\|B_1 y+B_2 s^1-c\|^2 \end{array}\right\}. \end{aligned}$$

Define $x^{\frac{1}{2}} := \hat{x}^0+\sigma\left(B_1 y^0+B_2 s^1-c\right)$. We have

$$y^1=\arg\min_{y\in\mathbb{Y}}\left\{f_1(y)+\langle x^{\frac{1}{2}}, B_1 y+B_2 s^1-c\rangle+\tfrac{\sigma}{2}\|B_1 y+B_2 s^1-c\|^2\right\}.$$

For x^1, from Algorithm 3, we have

$$\begin{aligned} x^1 &:= \eta^0 + \sigma(B_1 y^1 - c) + 2\sigma B_2 s^1 \\ &= \hat{x}^0 + \sigma\left(B_1 y^0 - c\right) + \sigma(B_1 y^1 - c) + 2\sigma B_2 s^1 \\ &= \hat{x}^0 + \sigma\left(B_1 y^0 + B_2 s^1 - c\right) + \sigma(B_1 y^1 + B_2 s^1 - c) \\ &= x^{\frac{1}{2}} + \sigma(B_1 y^1 + B_2 s^1 - c). \end{aligned}$$

For $\hat{x}^1$ defined in (20), we have

$$\begin{aligned} \hat{x}^1 &:= \eta^1 - \sigma\left(B_1 y^1 - c\right) \\ &= \tfrac{1}{2}\eta^0 + \tfrac{1}{2}v^1 - \sigma\left(B_1 y^1 - c\right) \\ &= \tfrac{1}{2}(\hat{x}^0 + \sigma(B_1 y^0 - c)) - \sigma(B_1 y^1 - c) \\ &\quad + \tfrac{1}{2}\left(\hat{x}^0 + \sigma(B_1 y^0 - c) + 2\sigma(B_1 y^1 + B_2 s^1 - c)\right) \\ &= \tfrac{1}{2}\hat{x}^0 + \tfrac{1}{2}x^1 + \tfrac{\sigma}{2}\left[(B_1 y^0 - c) - \left(B_1 y^1 - c\right)\right]. \end{aligned}$$

Hence, we prove the statement for $k = 0$. Assume that the statement holds for some $k \geq 0$. Then the statement for $k + 1$ follows similarly to the case $k = 0$. Thus, by induction, we have completed the proof. ■

Appendix E. The Proof of Lemma 10

Proof We first prove the equation (22). For all $k \geq 0$, from Algorithm 3 we have

$$-B_2^* w^{k+1} \in \partial f_2(s^{k+1}), \quad -B_1^* x^{k+1} \in \partial f_1(y^{k+1}). \tag{44}$$

It follows from Rockafellar (1970, Theorem 23.5) that for all $k \geq 0$,

$$\begin{aligned} d_{f_1}(x^{k+1}) &= \langle x^{k+1}, c\rangle - \langle B_1^* x^{k+1}, y^{k+1}\rangle - f_1(y^{k+1}), \\ d_{f_2}(w^{k+1}) &= -\langle B_2^* w^{k+1}, s^{k+1}\rangle - f_2(s^{k+1}). \end{aligned}$$

Therefore, for all $k \geq 0$,

$$\begin{aligned} & -d_{f_1}(x^{k+1}) - d_{f_2}(w^{k+1}) \\ = \;& f_1(y^{k+1}) + f_2(s^{k+1}) - \langle x^{k+1}, c\rangle + \langle B_1^* x^{k+1}, y^{k+1}\rangle + \langle B_2^* w^{k+1}, s^{k+1}\rangle \\ = \;& f_1(y^{k+1}) + f_2(s^{k+1}) + \langle x^{k+1}, B_1 y^{k+1} + B_2 s^{k+1} - c\rangle - \langle x^{k+1} - w^{k+1}, B_2 s^{k+1}\rangle. \end{aligned}$$

From Algorithm 3, we also have for all $k \geq 0$, $x^{k+1} - w^{k+1} = \sigma\left(B_1 y^{k+1} + B_2 s^{k+1} - c\right) = \frac{1}{2}(v^{k+1} - \eta^k)$. Thus, for all $k \geq 0$,

$$-d_{f_1}(x^{k+1}) - d_{f_2}(w^{k+1}) = f_1(y^{k+1}) + f_2(s^{k+1}) + \frac{1}{\sigma}\langle x^{k+1} - \sigma B_2 s^{k+1}, x^{k+1} - w^{k+1}\rangle. \tag{45}$$

Recall that $x^{k+1} - \sigma B_2 s^{k+1} = \eta^k + \sigma(B_1 y^{k+1} + B_2 s^{k+1} - c) = \eta^k + x^{k+1} - w^{k+1}$. Therefore, for all $k \geq 0$,

$$x^{k+1} - \sigma B_2 s^{k+1} = \eta^k + \frac{1}{2}(v^{k+1} - \eta^k) = \frac{1}{2}(\eta^k - v^{k+1}) + v^{k+1},$$

and

$$\begin{aligned}\frac{1}{\sigma}\langle x^{k+1}-\sigma B_2 s^{k+1}, x^{k+1}-w^{k+1}\rangle &= \frac{1}{\sigma}\left\langle \frac{1}{2}(\eta^k - v^{k+1}) + v^{k+1}, \frac{1}{2}(v^{k+1}-\eta^k)\right\rangle \\ &= -\frac{1}{4\sigma}\|v^{k+1}-\eta^k\|^2 - \frac{1}{2\sigma}\langle \eta^k - v^{k+1}, v^{k+1}\rangle.\end{aligned} \tag{46}$$

According to Corollary 5, w^* is a solution to problem (14) and problem (18) under Assumption 1. We also have that the limit point (y^*, s^*) of the sequence $\{y^k, s^k\}_{k=1}^{\infty}$ is a solution to problem (17) from Corollary 8. It follows from Rockafellar (1970, Theorem 28.4) that

$$-d_{f_1}(w^*) - d_{f_2}(w^*) = f_1(y^*) + f_2(s^*). \tag{47}$$

Therefore, it follows from (45), (46), and (47) that the equation (22) holds.

Next, we prove the upper bound of $h(y^{k+1}, s^{k+1})$ for all $k \geq 0$. From Algorithm 2 and Corollary 5, we have that $(\eta^* - w^*)/\sigma \in \partial d_{f_2}(w^*)$ and $(w^* - \eta^*)/\sigma \in \partial d_{f_1}(w^*)$. Hence,

$$\begin{aligned} d_{f_1}(x^{k+1}) - d_{f_1}(w^*) &\geq \langle (w^* - \eta^*)/\sigma, x^{k+1} - w^*\rangle, \\ d_{f_2}(w^{k+1}) - d_{f_2}(w^*) &\geq \langle (\eta^* - w^*)/\sigma, w^{k+1} - w^*\rangle. \end{aligned}$$

Thus,

$$\begin{aligned} &-4\sigma\left(d_{f_1}(x^{k+1}) + d_{f_2}(w^{k+1}) - d_{f_1}(w^*) - d_{f_2}(w^*)\right) \\ &\quad \leq 4\langle x^{k+1} - w^{k+1}, \eta^* - w^*\rangle = 2\langle v^{k+1} - \eta^k, \eta^* - w^*\rangle. \end{aligned}$$

Together with (22), we obtain the inequality (23).

Finally, using (19) and $w^* = x^*$, we have for all $k \geq 0$,

$$f_1(y^{k+1}) - f_1(y^*) \geq \langle -B_1^* w^*, y^{k+1} - y^*\rangle, \quad f_2(s^{k+1}) - f_2(s^*) \geq \langle -B_2^* w^*, s^{k+1} - s^*\rangle.$$

Thus, the inequality (24) holds since $B_1 y^* + B_2 s^* = c$. This completes the proof of the lemma. ■

Appendix F. Restart Strategy for the HPR Method

For completeness, we describe the restart rule used in our implementation. This rule is a practical heuristic used in the numerical experiments. Let k_{in} denote the number of iterations since the last restart. The relative KKT residual is checked every 50 iterations. During the first 500 iterations, we restart the method every 50 iterations. After the first 500 iterations, a restart is triggered if either $k_{\text{in}} \geq 500$, or the relative KKT residual does not decrease sufficiently between two consecutive checkpoints. Once a restart is triggered, the Halpern anchor point is reset to the latest iterate and the local counter k_{in} is reset to zero. Subsequent works also investigate restart strategies for Halpern-type methods (Lu and Yang, 2024; Chen et al., 2026, 2025). In contrast, the algorithm in Applegate et al. (2023) restarts from averaged iterates.

We also compare the performance of the HPR method with and without restart for the WBP on Gaussian mixture distributions with $m = m_t = 100$ and $T = 100$ generated as in Section 4.2. As shown in Figure 6, the restarted HPR method achieves a faster decrease in the KKT residual than the unrestarted HPR method. This suggests that the restart strategy can significantly improve the practical performance of HPR. For further recent developments on restart strategies for HPR methods that appeared after the initial version of this work, we refer the reader to Chen et al. (2026, 2025).

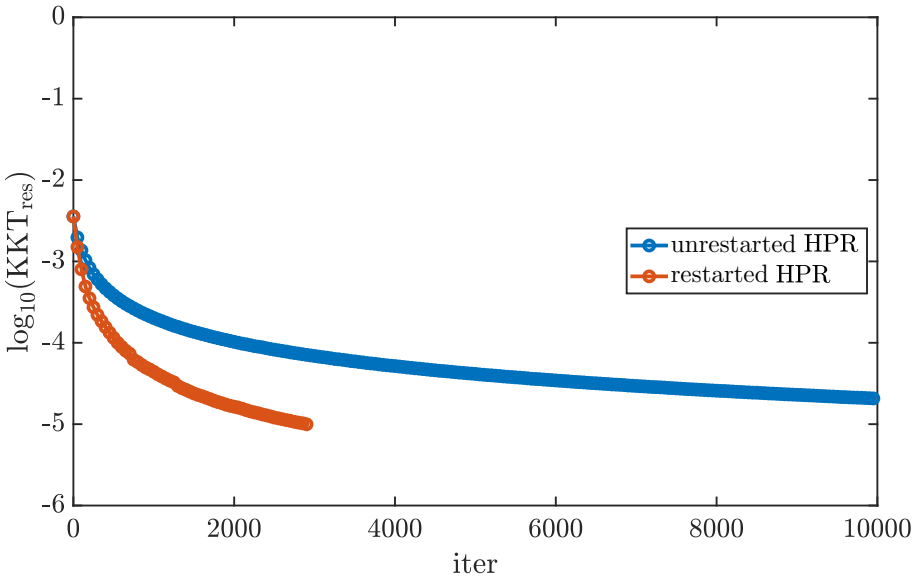


Figure 6: Convergence behavior of HPR methods with and without restart for Gaussian mixture distributions with $m = m_t = 100$ and $T = 100$.

From a theoretical perspective, for solvable LPs, including the WBP, an error-bound condition holds, under which ADMM can achieve linear convergence with suitable algorithmic parameters (Han et al., 2018). For unrestarted HPR with a fixed Halpern anchor, Theorem 11 establishes a non-ergodic $O(1/k)$ convergence rate with respect to the KKT residual. Although the weight $1/(k+2)$ attached to the fixed initial anchor η^0 decays to zero, its slow decay may hinder faster local convergence. This observation motivates restarting HPR by resetting the Halpern anchor to the latest iterate and resetting the local iteration counter to zero. Related restart strategies have also been studied for first-order methods for LPs. For example, based on sharpness analysis, Applegate et al. (2023) develop a restarted scheme using averaged iterates as restart points, while Lu and Yang (2024) study restarted Halpern PDHG and establish linear convergence across restart points under appropriate conditions. These results illustrate how, under suitable sharpness or error-bound conditions, an $O(1/k)$ bound for the underlying method can be combined with an appropriate restart rule to achieve faster linear convergence at the restart points than the original method. It would be interesting to develop and analyze restart schemes for HPR that establish linear convergence of the full sequence of iterates while retaining practical efficiency.

Appendix G. Synthetic Data Generation

Following Yang et al. (2021), the support points $\{\boldsymbol{q}_i^t : i = 1, \ldots, m_t,\ t = 1, \ldots, T\}$ are generated coordinatewise from a univariate Gaussian mixture. For each input distribution, its mixture weights are generated independently, and the mixture is constructed in MATLAB by

$$\begin{aligned}
&\text{gm_num} = 5; \quad \text{gm_mean} = [-20; -10; 0; 10; 20];\\
&\text{sigma} = \text{zeros}(1, 1, \text{gm_num}); \quad \text{sigma}(1, 1, :) = 5 * \text{ones}(\text{gm_num}, 1);\\
&\text{gm_weights} = \text{rand}(\text{gm_num}, 1);\\
&\text{gm_weights} = \text{gm_weights}/\text{sum}(\text{gm_weights});\\
&\text{distrib} = \text{gmdistribution}(\text{gm_mean}, \text{sigma}, \text{gm_weights}).
\end{aligned}$$

For each distribution $\mathcal{P}^t$, the weight vector $(a_1^t, \ldots, a_{m_t}^t)$ is sampled from the uniform distribution on $(0, 1)$ and normalized so that $\sum_{i=1}^{m_t} a_i^t = 1$. The barycentric weights $(\omega_1, \ldots, \omega_T)$ are generated in the same way and normalized so that $\sum_{t=1}^{T} \omega_t = 1$.

The barycenter support $\{\boldsymbol{q}_i^c\}_{i=1}^m$ is selected by applying k-means to all generated support points $\{\boldsymbol{q}_i^t : i = 1, \ldots, m_t,\ t = 1, \ldots, T\}$. For each $t = 1, \ldots, T$, the cost matrix between $\{\boldsymbol{q}_i^c\}_{i=1}^m$ and the support of $\mathcal{P}^t$ is defined by

$$\mathcal{D}(\mathcal{P}^c, \mathcal{P}^t)_{ij} = \|\boldsymbol{q}_i^c - \boldsymbol{q}_j^t\|_2^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m_t.$$

All cost matrices are then normalized so that $\max_{t=1,\ldots,T} \|\mathcal{D}(\mathcal{P}^c, \mathcal{P}^t)\|_\infty = 1$.

References

M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

J. M. Altschuler and E. Boix-Adsera. Wasserstein barycenters can be computed in polynomial time in fixed dimension. *Journal of Machine Learning Research*, 22(44):1–19, 2021.

E. Anderes, S. Borgwardt, and J. Miller. Discrete Wasserstein barycenters: Optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84(2):389–409, 2016.

D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34:20243–20257, 2021.

D. Applegate, O. Hinder, H. Lu, and M. Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, 201(1): 133–184, 2023.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2011.

H. H. Bauschke, S. Singh, and X. Wang. Projecting onto rectangular matrices with prescribed row and column sums. *Fixed Point Theory and Algorithms for Sciences and Engineering*, 2021:23, 2021. doi: 10.1186/s13663-021-00708-1.

J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

J. Bigot and T. Klein. Characterization of barycenters in the Wasserstein space by averaging optimal transport maps. *ESAIM: Probability and Statistics*, 22:35–57, 2018.

N. Bonneel, G. Peyré, and M. Cuturi. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Transactions on Graphics*, 35(4):71:1–71:10, 2016. doi: 10.1145/2897824.2925918.

R. I. Boţ and D.-K. Nguyen. Fast Krasnosel'skiĭ-Mann algorithm with a convergence rate of the fixed point iteration of $o(\frac{1}{k})$. *SIAM Journal on Numerical Analysis*, 61(6):2813–2843, 2023.

G. Calvillo and D. Romero. On the closest point to the origin in transportation polytopes. *Discrete Applied Mathematics*, 210:88–102, 2016. doi: 10.1016/j.dam.2015.01.027.

A. Chambolle and J. P. Contreras. Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein barycenter problems. *SIAM Journal on Mathematics of Data Science*, 4(4):1369–1395, 2022. doi: 10.1137/22M1481865.

K. Chen, D. Sun, Y. Yuan, G. Zhang, and X. Zhao. HPR-QP: A dual Halpern Peaceman–Rachford method for solving large-scale convex composite quadratic programming. *arXiv preprint arXiv:2507.02470*, 2025.

K. Chen, D. Sun, Y. Yuan, G. Zhang, and X. Zhao. HPR-LP: An implementation of an HPR method for solving linear programming. *Mathematical Programming Computation*, 18(1):183–210, 2026. doi: 10.1007/s12532-025-00292-0.

K. Chen, D. Sun, Y. Yuan, G. Zhang, and X. Zhao. On the relationships among GPU-accelerated first-order methods for solving linear programming. *Journal of Computational Mathematics*, 45(1):30–51, 2027. doi: 10.4208/jcm.2606-m2025-0260. Published online in 2026.

L. Chen, X. Li, D. Sun, and K.-C. Toh. On the equivalence of inexact proximal ALM and ADMM for a class of convex composite programming. *Mathematical Programming*, 185 (1):111–161, 2021.

P.-A. Chiappori, R. J. McCann, and L. P. Nesheim. Hedonic price equilibria, stable matching, and optimal transport: Equivalence, topology, and uniqueness. *Economic Theory*, 42(2):317–354, 2010.

S. Claici, E. Chien, and J. Solomon. Stochastic Wasserstein barycenters. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2018.

C. Cotar, G. Friesecke, and C. Klüppelberg. Density functional theory and optimal transportation with Coulomb cost. *Communications on Pure and Applied Mathematics*, 66(4): 548–599, 2013.

Y. Cui, X. Li, D. Sun, and K.-C. Toh. On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions. *Journal of Optimization Theory and Applications*, 169 (3):1013–1041, 2016.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300, 2013.

M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693. PMLR, 2014.

M. Cuturi and G. Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

G. B. Dantzig and M. N. Thapa. *Linear Programming 2: Theory and Extensions*. Springer, 2003.

D. Davis and W. Yin. Convergence rate analysis of several splitting schemes. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 115–163. Springer, 2016.

D. Dvinskikh and D. Tiapkin. Improved complexity bounds in Wasserstein barycenter problem. In *International Conference on Artificial Intelligence and Statistics*, pages 1738–1746. PMLR, 2021.

P. Dvurechenskii, D. Dvinskikh, A. Gasnikov, C. Uribe, and A. Nedich. Decentralize and randomize: Faster algorithm for Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 31, 2018.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

A. Galichon. *Optimal Transport Methods in Economics*. Princeton University Press, 2016.

D. Ge, H. Wang, Z. Xiong, and Y. Ye. Interior-point methods strike back: Solving the Wasserstein barycenter problem. *Advances in Neural Information Processing Systems*, 32, 2019.

A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue Française D'automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, 9(R2):41–76, 1975.

W. Glunt, T. L. Hayden, and R. Reams. The nearest "doubly stochastic" matrix to a real matrix with the same first moment. *Numerical Linear Algebra with Applications*, 5: 475–482, 1998.

M. L. N. Gonçalves, J. G. Melo, and R. D. Monteiro. Improved pointwise iteration-complexity of a regularized ADMM and of a regularized non-Euclidean HPE framework. *SIAM Journal on Optimization*, 27(1):379–407, 2017.

A. Gramfort, G. Peyré, and M. Cuturi. Fast optimal transport averaging of neuroimaging data. In *International Conference on Information Processing in Medical Imaging*, pages 261–272. Springer, 2015.

S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov. On a combination of alternating minimization and Nesterov's momentum. In *International Conference on Machine Learning*, pages 3886–3898. PMLR, 2021.

B. Halpern. Fixed points of nonexpanding maps. *Bulletin of the American Mathematical Society*, 73(6):957–961, 1967.

D. Han, D. Sun, and L. Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Mathematics of Operations Research*, 43(2):622–637, 2018.

L. Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 37(2): 227–229, 1942.

R. N. Khoury. Closest matrices in the space of generalized doubly stochastic matrices. *Journal of Mathematical Analysis and Applications*, 222(2):561–568, 1998. doi: 10.1006/jmaa.1998.5970.

D. Kim. Accelerated proximal point method for maximally monotone operators. *Mathematical Programming*, 190(1):57–87, 2021.

R. Krawtschenko, C. A. Uribe, A. Gasnikov, and P. Dvurechensky. Distributed optimization with quantization for computing Wasserstein barycenters. In *2025 IEEE 64th Conference on Decision and Control (CDC)*. IEEE, 2025. URL `https://arxiv.org/abs/2010.14325`.

A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. Uribe. On the complexity of approximating Wasserstein barycenters. In *International Conference on Machine Learning*, pages 3530–3540. PMLR, 2019.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

F. Lieder. On the convergence rate of the Halpern-iteration. *Optimization Letters*, 15(2): 405–418, 2021.

T. Lin, N. Ho, X. Chen, M. Cuturi, and M. Jordan. Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm. *Advances in Neural Information Processing Systems*, 33:5368–5380, 2020.

P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

H. Lu and J. Yang. Restarted Halpern PDHG for linear programming. *arXiv preprint arXiv:2407.16144*, 2024.

H. Lu and J. Yang. cuPDLP.jl: A GPU implementation of restarted primal-dual hybrid gradient for linear programming in Julia. *Operations Research*, 73(6):3440–3452, 2025.

H. Lu, Z. Peng, and J. Yang. cuPDLPx: A further enhanced GPU-based first-order solver for linear programming. *arXiv preprint arXiv:2507.14051*, 2025.

G. Luise, S. Salzo, M. Pontil, and C. Ciliberto. Sinkhorn barycenters with free support via Frank–Wolfe algorithm. *Advances in Neural Information Processing Systems*, 32, 2019.

G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal*, 29:341–346, 1962. ISSN 0012-7094.

G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences*, pages 666–704, 1781.

R. D. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23 (1):475–507, 2013.

S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report CUCS-005-96, Department of Computer Science, Columbia University, 1996. URL `https://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php`.

J. Park and E. K. Ryu. Exact optimal accelerated complexity for fixed-point iterations. In *International Conference on Machine Learning*, pages 17420–17457. PMLR, 2022.

G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

R. T. Rockafellar. *Convex Analysis*, volume 18. Princeton University Press, 1970.

A. Rogozin, A. Beznosikov, D. Dvinskikh, D. Kovalev, P. Dvurechensky, and A. Gasnikov. Decentralized saddle point problems via non-Euclidean mirror prox. *Optimization Methods and Software*, 40(5):1127–1152, 2025. doi: 10.1080/10556788.2023.2280062.

D. Romero. Easy transportation-like problems on K-dimensional arrays. *Journal of Optimization Theory and Applications*, 66(1):137–147, 1990. doi: 10.1007/BF00940537.

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

M. Staib, S. Claici, J. M. Solomon, and S. Jegelka. Parallel streaming Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 30, 2017.

D. Sun, Y. Yuan, G. Zhang, and X. Zhao. Accelerating preconditioned ADMM via degenerate proximal point mappings. *SIAM Journal on Optimization*, 35(2):1165–1193, 2025.

D. Tiapkin, A. Gasnikov, and P. Dvurechensky. Stochastic saddle-point optimization for the Wasserstein barycenter problem. *Optimization Letters*, 16:2145–2175, 2022. doi: 10.1007/s11590-021-01834-w.

Q. Tran-Dinh and Y. Luo. Halpern-type accelerated and splitting algorithms for monotone inclusions. *arXiv preprint arXiv:2110.08150*, 2021.

C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedić. Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6544–6549. IEEE, 2018.

R. Wittmann. Approximation of fixed points of nonexpansive mappings. *Archiv der Mathematik*, 58(5):486–491, 1992.

B. Yang, X. Zhao, X. Li, and D. Sun. An accelerated proximal alternating direction method of multipliers for optimal decentralized control of uncertain systems. *Journal of Optimization Theory and Applications*, 204(1):9, 2025.

L. Yang, J. Li, D. Sun, and K.-C. Toh. A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters. *Journal of Machine Learning Research*, 22 (21):1–37, 2021.

J. Ye and J. Li. Scaling up discrete distribution clustering using ADMM. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5267–5271. IEEE, 2014.

J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

R. Zass and A. Shashua. Doubly stochastic normalization for spectral clustering. In *Advances in Neural Information Processing Systems*, volume 19, pages 1569–1576, 2006.

G. Zhang, Z. Gu, Y. Yuan, and D. Sun. HOT: An efficient Halpern accelerating algorithm for optimal transport problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6703–6714, 2025. doi: 10.1109/TPAMI.2025.3564353.