HOT: An Efficient Halpern Accelerating Algorithm for Optimal Transport Problems

Guojun Zhang, Zhexuan Gu, Yancheng Yuan, Defeng Sun

***Abstract*—This paper proposes an efficient HOT algorithm for solving the optimal transport (OT) problems with finite supports. We particularly focus on an efficient implementation of the HOT algorithm for the case where the supports are in $\mathbb{R}^2$ with ground distances calculated by L_2^2-norm. Specifically, we design a Halpern accelerating algorithm to solve the equivalent reduced model of the discrete OT problem. Moreover, we derive a novel procedure to solve the involved linear systems in the HOT algorithm in linear time complexity. Consequently, we can obtain an ε-approximate solution to the optimal transport problem with M supports in $O(M^{1.5}/\varepsilon)$ flops, which significantly improves the best-known computational complexity. We further propose an efficient procedure to recover an optimal transport plan for the original OT problem based on a solution to the reduced model, thereby overcoming the limitations of the reduced OT model in applications that require the transport plan. We implement the HOT algorithm in PyTorch and extensive numerical results show the superior performance of the HOT algorithm compared to existing state-of-the-art algorithms for solving the OT problems.**

***Index Terms*—Optimal transport, Kantorovich-Wasserstein distance, Halpern iteration, Acceleration, Computational complexity.**

I. Introduction

THE Kantorovich-Wasserstein (KW) distance has become a prime choice for measuring the similarity between two probability distributions. It has demonstrated remarkable success in various applications, including color transfer [1]–[3], texture synthesis and mixing [4], registration and warping [5], transport-based morphometry [6], and hypothesis testing [7], among others. Despite its powerful geometric framework for comparing probabilities, the KW distance is computationally expensive in general [8], [9]. Specifically, it requires solving an optimal transport (OT) problem, which is a (large-scale) linear programming (LP) in a discrete setting. Standard methods, such as the simplex method and the interior point method, suffer from high computational complexity relative to the problem size. Furthermore, these methods are difficult to parallelize, which can hardly benefit from the modern powerful graphics processing units (GPUs). Consequently, solving the LP problem of the discrete OT problem remains daunting in modern data-driven applications due to high computational and memory costs. This paper addresses these two challenges by proposing an efficient and easily parallelizable algorithm for solving an equivalent reduced model of the OT problem.

Guojun Zhang and Zhexuan Gu contribute equally to this manuscript.
Corresponding author: Yancheng Yuan.
The authors are with The Hong Kong Polytechnic University. E-mail: guojun.zhang@connect.polyu.hk, zhexuan.gu@connect.polyu.hk, yancheng.yuan@polyu.edu.hk, defeng.sun@polyu.edu.hk.

A. Related work and existing challenges

When it comes to computing the KW distance between two discrete probability distributions with M supports, there are mainly two popular approaches: (i) Computing an approximated KW distance by solving the optimal transport problem with an additional entropy regularization [10]; (ii) Computing the KW distance via solving the corresponding LP problem [11]. The readers can refer to [9] and the references therein for a more detailed discussion of the algorithms for solving OT problems. Before introducing our new algorithm, we briefly discuss the challenges of the aforementioned approaches.

Challenges with the Entropy-regularized approach: Due to its scalability, the Sinkhorn algorithm and its improved versions have been widely adopted to compute an approximation of the KW distance in applications [10], [12]–[14]. In particular, the Sinkhorn algorithm can efficiently solve the regularized OT problem when the regularization parameter is moderate (i.e., no less than 10^{-2}). However, a high-quality approximation of the KW distance is important for better performance in many applications, which requires solving the regularized OT problem with a small regularization parameter. Unfortunately, a small regularization parameter will usually cause numerical issues and a slower convergence for the Sinkhorn algorithm. Some stabilized and rescaling techniques [15] have been proposed to improve the robustness of solving the regularized OT problems, but the efficiency of the stabilized algorithms is unsatisfactory compared to the Sinkhorn algorithm.

Challenges with the LP approach: Along this line, the interior point method [16] and the network simplex method [17], [18] are popular choices for obtaining solutions with high accuracy to the moderate scale LP problem. Recently, a semismooth Newton based inexact proximal augmented Lagrangian method [19] has been proposed for solving linear programming problems which can solve the OT problem as a special case. The semismooth Newton based algorithm can exploit the sparsity of the solution by the generalized Jacobian and show superior numerical performance compared to Gurobi in some examples. However, these solvers are not applicable for solving the problem on a very large scale due to the high computational complexity. The urgent need to solve large-scale OT problems in applications inspires extensive research in designing first-order methods, such as the Douglas-Rachford splitting algorithm [20] and primal-dual hybrid gradient method (PDHG) [21]–[24]. Jambulapati et al. [25] proposed an algorithm based on a dual-extrapolation algorithm to achieve an $\widetilde{O}(M^2/\varepsilon)$ computational complexity

Fig. 1: Selected examples of color transfer based on the reduced optimal transport model with the optimal transport plan recovered by Algorithm 1.

bound, where M is the number of supports, for obtaining an ε-approximate solution (in terms of objective function value)[1]. Recently, Zhang et al. [26] proposed an efficient Halpern-Peaceman-Rachford (HPR) algorithm for solving the OT model and the Wasserstein barycenter problem, which can obtain an ε-approximate solution (in terms of the Karush-Kuhn-Tucker (KKT) residual) of the OT model in $O(M^2/\varepsilon)$ flops. We summarize some known complexity results for solving the OT problem in Table I. Readers can refer to [26]–[28] and the references therein for a more detailed discussion.

TABLE I: Selected known complexity results for solving OT problem (C represents the largest elements of the cost matrix, while R denotes the distance between the initial point and the solution set.)

Algorithm	Time complexity result	Space complexity result
Sinkhorn [13]	$\widetilde{O}(M^2C^2/\varepsilon^2)$	$O(M^2)$
APDAGD [13], [29]	$\widetilde{O}(M^{2.5}C/\varepsilon)$	$O(M^2)$
Greenkhorn [29]	$\widetilde{O}(M^2C^2/\varepsilon^2)$	$O(M^2)$
Accelerated Sinkhorn [29]	$\widetilde{O}(M^{7/3}C^{4/3}/\varepsilon^{4/3})$	$O(M^2)$
AAM [14]	$\widetilde{O}(M^{2.5}C/\varepsilon)$	$O(M^2)$
Dual extrapolation [25]	$\widetilde{O}(M^2C/\varepsilon)$	$O(M^2)$
HPD [30]	$\widetilde{O}(M^{2.5}C/\varepsilon)$	$O(M^2)$
HPR [26][2]	$O(M^2R/\varepsilon)$	$O(M^2)$
HOT (Ours)	$\mathbf{O(M^{1.5}R}/\varepsilon)$	$\mathbf{O(M^{1.5})}$

Beyond the challenges above in computational efficiency, all these algorithms for the original OT problem with M supports require at least a memory cost of $O(M^2)$. This memory cost makes it forbidden to compute the KW distance of two distributions with massive supports (i.e., the OT problem for computing the KW distance of two 512×512 grey images has more than 6.8×10^{10} variables). To address this issue, researchers have proposed some approximate models for the OT problem, such as the linear OT framework [31], the sliced Wasserstein distance [32], and the approximated earth mover's distance [33]–[35]. While these approaches alleviate memory issues and simplify computations, they inevitably introduce a loss of exactness. This limitation has motivated further research into equivalent reduced models for the OT problem. When the ground distances between supports in $\mathbb{R}^d$ are calculated by L_1-norm, Ling and Okada proposed an equivalent reduced model with $O(dM)$ memory cost to calculate the earth mover distance (equivalent to the KW distance) [36]. Recently, Auricchio et al. [37] extended the idea to the case where the ground distances between supports are calculated by L_2^2-norm and proposed an equivalent reduced model with $O(dM^{\frac{d+1}{d}})$ memory cost. The authors in [37] adopted the Network Simplex method to solve the reduced model and demonstrated superior performance in terms of computational and memory efficiency compared to the Sinkhorn and the convolutional Sinkhorn method [3], [38] on examples of moderate scale. Unfortunately, the efficiency becomes unsatisfactory for very large-scale problems (see Section IV for details). Moreover, the transport plan is not available if we solve the reduced model, which is critical for a wide class of applications, such as color transfer [1], [39], texture synthesis [4], and domain adaptation [40].

B. Contributions

Motivated by the recent advancements in the accelerated algorithms based on Halpern iteration [26], [41]–[45], we

[1]Despite its better complexity bound, the empirical performance of this algorithm is not as efficient as other algorithms, such as accelerated gradient method [13], [14].

propose an efficient **H**alpern accelerating method for solving the reduced **OT** problem, which is abbreviated as "HOT" for convenience, to address the challenges in computing the KW distance with a finite number of supports. We particularly focus on an efficient implementation of the HOT algorithm for the case where the supports are in $\mathbb{R}^2$ with ground distances calculated by L_2^2-norm, which includes a wide class of applications as aforementioned. Specifically, HOT adopts a first-order algorithm with Halpern acceleration to solve the equivalent reduced OT model, which can obtain an ε-approximate solution in $O(1/\varepsilon)$ iterations [26], [41]. More importantly, we design a fast procedure for solving the subproblems with linear time complexity by fully exploiting the problem structure. This also makes the popular alternating direction method of multipliers (ADMM) [46], [47] scalable for solving the reduced OT model. Overall, our proposed HOT algorithm can compute an ε-approximation of the KW distance between two histograms supported on $M = m \times n$ bins within $O((m^2n+n^2m)/\varepsilon)$ flops. This is the best-known computational complexity for computing an approximate KW distance to our knowledge. Moreover, we propose an efficient algorithm to recover a transport plan based on the obtained solution of the reduced model, which releases the power of the reduced model in applications. We implement HOT in PyTorch and extensive numerical results will be shown in Section IV to demonstrate the superior and robust performance of HOT for computing the KW distance, compared to state-of-the-art algorithms, including Sinkhorn [10], convolutional Sinkhorn[3], [38], Network Simplex method [17], [18], ADMM [46], [47], interior point method (in Gurobi).

We summarize the main contributions of this paper as follows:

1) We propose an efficient HOT algorithm for solving the reduced model of the OT problem with an attractive $O(1/\varepsilon)$ iteration complexity guarantee with respect to the KKT residual.
2) We designed a highly efficient algorithm for solving the subproblems of the HOT algorithm with linear time complexity.
3) We propose an efficient algorithm to recover a transport plan based on the obtained solution of the reduced model, which removes the restriction of the reduced model in applications requiring a transport plan.
4) We implement the HOT algorithm in PyTorch, which supports both CPU and GPU computation and is user-friendly for researchers in the machine learning community.
5) Extensive numerical testings are conducted and presented to demonstrate the efficiency of the HOT algorithm.

The rest of the paper is organized as follows. We introduce the equivalent reduced OT model in Section II. This section also includes an efficient procedure for recovering the transport plan from a solution to the reduced OT problem. The HOT algorithm and its computational complexity guarantees will be presented in Section III. We present extensive numerical results in Section IV and conclude the paper in Section V.

Notation. We denote the n-dimensional real Euclidean space as $\mathbb{R}^n$ and the nonnegative orthant of $\mathbb{R}^n$ as $\mathbb{R}^n_+$. For any $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, 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. Additionally, let $\mathbf{1}_m$ (resp. $\mathbf{0}_m$) denote the m-dimensional vector with all entries being 1 (resp. 0). For a given matrix $A \in \mathbb{R}^{m\times n}$, we denote $A^\top \in \mathbb{R}^{n\times m}$ as its transpose. For a collection of matrices $\{A_1, \ldots, A_m\}$, we denote the block diagonal matrix with diagonal blocks A_i as $\mathrm{diag}(A_1, \ldots, A_m)$. $A_1 \otimes A_2$ stands for the Kronecker product of matrices A_1 and A_2. Moreover, for a closed convex set C, we denote the indicator function of C and the Euclidean projector over C as δ_C and $\Pi_C(x) := \arg\min_{z\in C} \|x - z\|$, respectively.

II. Kantorovich-Wasserstein distances

In this section, we first introduce an equivalent reduced model of the OT problem for computing the KW distance between two-dimensional histograms. Subsequently, we present a fast and easily implementable algorithm to reconstruct the optimal transport plan of the original OT model from a solution of this reduced model.

A. An equivalent reduced model of the OT problem

In the following discussion, we assume two-dimensional histograms for simplicity. As previously mentioned, these histograms are widely used in applications as shape and image descriptors. Without loss of generality, we adopt the following assumptions and notations:

1) Histograms have supports in $M = m \times n$ bins with m rows and n columns;
2) The index set for bins is defined as $\mathcal{I} = \{(i, j) \mid 1 \le i \le m, 1 \le j \le n\}$. We use (i, j) to denote a bin or a node corresponding to it;
3) μ^1 and μ^2 are the two histograms to be compared, where each histogram μ^k is defined as $\{\mu^k_{i,j} \mid \mu^k_{i,j} \ge 0, (i, j) \in \mathcal{I}, \sum_{(i,j)\in\mathcal{I}} \mu^k_{i,j} = 1\}$ for $k = 1, 2$.

With these notations and assumptions, the discrete OT problem for computing the KW distance between histograms μ^1 and μ^2 can be defined as follows:

$$\begin{aligned} \min_{\pi} \quad & \sum_{(i,j)\in\mathcal{I}} \sum_{(k,l)\in\mathcal{I}} c_{i,j;k,l}\pi_{i,j;k,l} \\ \text{s.t.} \quad & \begin{cases} \sum_{(k,l)\in\mathcal{I}} \pi_{i,j;k,l} = \mu^1_{i,j}, & \forall (i, j) \in \mathcal{I}, \\ \sum_{(i,j)\in\mathcal{I}} \pi_{i,j;k,l} = \mu^2_{k,l}, & \forall (k, l) \in \mathcal{I}, \\ \pi_{i,j;k,l} \ge 0, & \forall (i, j) \in \mathcal{I},\ \forall (k, l) \in \mathcal{I}, \end{cases} \end{aligned} \tag{1}$$

where π is the transport plan between histograms μ^1 and μ^2. The ground distance $c_{i,j;k,l}$ is commonly defined by the L_p^p distance:

$$c_{i,j;k,l} = \|(i, j)^\top - (k, l)^\top\|_p^p = (|i - k|^p + |j - l|^p). \tag{2}$$

In this paper, we focus on the case where $p = 2$. By exploiting the separable structure of the ground distance, Auricchio et al.

[37] proposed the following equivalent model in terms of the optimal objective function value:

$$\min_{f^{(1)},f^{(2)}} \quad \sum_{(i,j)\in\mathcal{I}} \left[\sum_{k=1}^{m}(k-i)^2 f^{(1)}_{i,k,j} + \sum_{l=1}^{n}(j-l)^2 f^{(2)}_{k,j,l}\right]$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^{m} f^{(1)}_{i,k,j} = \sum_{l=1}^{n} f^{(2)}_{k,j,l}, & \forall (k,j)\in\mathcal{I}, \\ \sum_{k=1}^{m} f^{(1)}_{i,k,j} = \mu^1_{i,j}, & \forall (i,j)\in\mathcal{I}, \\ \sum_{j=1}^{n} f^{(2)}_{k,j,l} = \mu^2_{k,l}, & \forall (k,l)\in\mathcal{I}, \\ f^{(1)}_{i,k,j}\geq 0, f^{(2)}_{k,j,l}\geq 0, & \forall (i,j),(k,l)\in\mathcal{I}, \end{cases} \tag{3}$$

where $f^{(1)}_{i,k,j}$ denotes the input flow from bin (i,j) to (k,j) and $f^{(2)}_{k,j,l}$ denotes the output flow from bin (k,j) to (k,l). Compared to formulation (1), the formulation (3) offers substantial computational benefits. Specifically, the reduced problem (3) only has mn^2+m^2n variables, whereas the original model has m^2n^2 variables. Moreover, the reduced model remains an LP problem. Consequently, popular algorithms for LP problems can be applied to solve this reduced model, such as the network-simplex method and the interior point method. Although the computation and memory costs of these mentioned algorithms are lower for the reduced model, it remains a challenge for solving large-scale problems. In this paper, we focus on addressing the challenges by designing a fast algorithm to solve the reduced model (3).

To facilitate the design of the algorithm, we reformulate the model (3) into the following standard form of linear programming:

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & \langle c, x\rangle + \delta_{\mathbb{R}^N_+}(x) \\ \text{s.t.} \quad & \hat{A}x = \hat{b}, \end{aligned} \tag{4}$$

where

1) $M_3 = 3M-1, N = m^2n+mn^2$;
2) $x = [f^{(1)}; f^{(2)}] \in \mathbb{R}^{m^2n}\times\mathbb{R}^{mn^2}$ with
$$\begin{cases} f^{(1)} = \{f^{(1)}_{i,k,j},\ \forall (i,j)\in\mathcal{I},\ k=1,\dots,m\}, \\ f^{(2)} = \{f^{(2)}_{k,j,l},\ \forall (k,l)\in\mathcal{I},\ j=1,\dots,n\}; \end{cases}$$
3) $c = [c^1; c^2] \in \mathbb{R}^{m^2n}\times\mathbb{R}^{mn^2}$ with
$$\begin{cases} c^1 = \{c^{(1)}_{i,k,j} = (k-i)^2,\ \forall (i,j)\in\mathcal{I},\ k=1,\dots,m\}, \\ c^2 = \{c^{(2)}_{k,j,l} = (j-l)^2,\ \forall (k,l)\in\mathcal{I},\ j=1,\dots,n\}; \end{cases}$$
4) $\hat{b} = [\mathbf{0}_M; \mu^1; \mu^2] \in \mathbb{R}^M\times\mathbb{R}^M\times\mathbb{R}^M$;
5) $\hat{A} = \begin{bmatrix} A_1 & A_2 \\ A_3 & \mathbf{0} \\ \mathbf{0} & \hat{A}_4 \end{bmatrix} \in \mathbb{R}^{(M_3+1)\times N}$ with
$$\begin{cases} A_1 = I_M\otimes \mathbf{1}_m^\top \in \mathbb{R}^{M\times m^2n}, \\ A_2 = -\mathbf{1}_n^\top\otimes I_M \in \mathbb{R}^{M\times mn^2}, \\ A_3 = I_n\otimes(\mathbf{1}_m^\top\otimes I_m) \in \mathbb{R}^{M\times m^2n}, \\ \hat{A}_4 = I_n\otimes(\mathbf{1}_n^\top\otimes I_m) \in \mathbb{R}^{M\times mn^2}. \end{cases} \tag{5}$$

For notational convenience, let $\bar{I}_m = [I_{m-1}, \mathbf{0}_{m-1}] \in \mathbb{R}^{(m-1)\times m}$. We define

$$A := \begin{bmatrix} A_1 & A_2 \\ A_3 & \mathbf{0} \\ \mathbf{0} & A_4 \end{bmatrix} \in \mathbb{R}^{M_3\times N}, \quad b := [0_M; \mu^1; \bar{I}_M\mu^2] \in \mathbb{R}^{M_3}. \tag{6}$$

with

$$A_4 = \operatorname{diag}\left(\mathbf{1}_n^\top\otimes I_m, \dots, \mathbf{1}_n^\top\otimes I_m, \mathbf{1}_n^\top\otimes \bar{I}_m\right) \in \mathbb{R}^{(M-1)\times mn^2}. \tag{7}$$

Similar to [48, Lemma 7.1], we can obtain that A defined in (6) has full row rank, and

$$\{x\in\mathbb{R}^N \mid Ax = b\} = \{x\in\mathbb{R}^N \mid \hat{A}x = \hat{b}\}.$$

As a result, the linear programming problem (4) is equivalent to

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & \langle c, x\rangle + \delta_{\mathbb{R}^N_+}(x) \\ \text{s.t.} \quad & Ax = b. \end{aligned} \tag{8}$$

Furthermore, the dual problem of (8) takes the form:

$$\min_{y\in\mathbb{R}^{M_3}, z\in\mathbb{R}^N} \left\{-\langle b, y\rangle + \delta_{\mathbb{R}^N_+}(z) \mid A^\top y + z = c\right\}. \tag{9}$$

The KKT conditions associated with (8) and (9) can be given by

$$A^*y + z = c, \quad Ax = b, \quad \mathbb{R}^N_+ \ni x \perp z \in \mathbb{R}^N_+, \tag{10}$$

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

B. Reconstruct the transport plan from the reduced model

The absence of the transport plan π makes the reduced OT model less favorable in applications where the transport plan is necessary (i.e., color transfer [1]–[3]). We address this issue by proposing a fast algorithm (shown in Algorithm 1) to reconstruct an optimal transport plan of the original model from an optimal solution of the reduced model (3).

Algorithm 1 A fast algorithm for reconstructing transport plan π from the network flows $f^{(1)}$ and $f^{(2)}$.

Input: An optimal flow $(f^{(1)}, f^{(2)})$ of problem (3).
Output: An optimal transport plan π of problem (1).
for $(k,j)\in\mathcal{I}$ **do**
  for $i = 1,\dots,m$ **do**
    for $l = 1,\dots,n$ **do**
      $\pi_{i,j;k,l} = \min\{f^{(1)}_{i,k,j}, f^{(2)}_{k,j,l}\}$
      $f^{(1)}_{i,k,j} = f^{(1)}_{i,k,j} - \pi_{i,j;k,l}$
      $f^{(2)}_{k,j,l} = f^{(2)}_{k,j,l} - \pi_{i,j;k,l}$
    end for
  end for
end for

The following proposition shows that the output of Algorithm 1 is an optimal transport plan for the original OT model.

Proposition 1. *Given an optimal solution $(f^{(1)}, f^{(2)})$ to problem* (3)*, the output π of Algorithm 1 is an optimal solution to the optimal transport problem* (1).

Proof. Since $(f^{(1)}, f^{(2)})$ is an optimal solution to problem (3), we have

$$f^{(1)}_{i,k,j} \geq 0, \quad f^{(2)}_{k,j,l} \geq 0, \quad \forall (i,j) \in \mathcal{I}, \quad \forall (k,l) \in \mathcal{I},$$

which implies

$$\pi_{i,j;k,l} \geq 0, \quad \forall (i,j), \quad \forall (k,l) \in \mathcal{I}. \tag{11}$$

Fix i, j, k. For convenience, let $f^{(1)-l}_{i,k,j}$ denote the value of $f^{(1)}_{i,k,j}$ at the l-th iteration before updating. According to the update formula of $f^{(1)}_{i,k,j}$, there must exist an index $1 \leq l^* \leq n$ such that $f^{(1)-l^*}_{i,k,j} \leq f^{(2)}_{k,j,l^*}$. Otherwise, we would have

$$f^{(1)}_{i,k,j} > \sum_{l=1}^{n} f^{(2)}_{k,j,l},$$

which contradicts the feasibility of $(f^{(1)}, f^{(2)})$:

$$f^{(1)}_{i,k,j} \leq \sum_{i=1}^{m} f^{(1)}_{i,k,j} = \sum_{l=1}^{n} f^{(2)}_{k,j,l}.$$

Therefore, after n iterations, the final value of $f^{(1)}_{i,k,j}$ must be zero, which leads to

$$\sum_{l=1}^{n} \pi_{i,j;k,l} = f^{(1)}_{i,k,j}. \tag{12}$$

Similarly, by fixing j, k, l, we also obtain

$$\sum_{i=1}^{m} \pi_{i,j;k,l} = f^{(2)}_{k,j,l}. \tag{13}$$

Hence, according to the constraints in problem (3), we have

$$\begin{cases} \sum\limits_{(k,l)\in\mathcal{I}} \pi_{i,j;k,l} = \sum\limits_{k=1}^{m} f^{(1)}_{i,k,j} = \mu^1_{i,j}, \\ \sum\limits_{(i,j)\in\mathcal{I}} \pi_{i,j;k,l} = \sum\limits_{j=1}^{n} f^{(2)}_{k,j,l} = \mu^2_{k,l}, \end{cases}$$

which, together with (11), shows that π is a feasible solution to problem (1). Furthermore, from (12), (13), and the definition of c in (2), we can obtain

$$\begin{aligned} &\sum_{((i,j),(k,l))} c_{i,j;k,l}\pi_{i,j;k,l} \\ = &\sum_{(i,j)\in\mathcal{I}} [\sum_{k=1}^{m} (k-i)^2 f^{(1)}_{i,k,j} + \sum_{l=1}^{n} (j-l)^2 f^{(2)}_{k,j,l}]. \end{aligned}$$

According to [37, Theorem 1], we know that the optimal objective function values of problems (1) and (3) are equivalent. Therefore, π is an optimal solution to problem (1). □

Remark 1. *The worst-case computational complexity of reconstructing the transport plan via Algorithm 1 is $3M^2$. In practice, we can efficiently parallelize the (k, j) loop to leverage the significant benefits of GPU acceleration, enabling an efficient reconstruction of the transport plan from a solution to the reduced OT model. Furthermore, in many applications, such as image retrieval [33] and shape matching [36], only the KW distance is required. In these scenarios, the reconstruction of the transport plan is not necessary.*

III. A Halpern accelerating algorithm for solving OT problem

In this section, we first introduce an efficient Halpern accelerating method for solving problem (9), which includes the equivalent reduced OT problem (8) as a special case. Subsequently, we present an efficient implementation of the proposed algorithm by designing a novel procedure to solve the involved linear system in linear time complexity.

A. HOT: A Halpern accelerating method for solving OT problem

Given $\sigma > 0$, the augmented Lagrange function corresponding to the dual problem (9) is defined by, for any $(y, z, x) \in \mathbb{R}^{M_3} \times \mathbb{R}^N \times \mathbb{R}^N$,

$$L_\sigma(y, z; x) := -\langle b, y\rangle + \delta_{\mathbb{R}^N_+}(z) + \frac{\sigma}{2}\|A^\top y + z - c + \frac{1}{\sigma}x\|^2 - \frac{1}{2\sigma}\|x\|^2.$$

For ease of notation, denote $w := (y, z, x)$. A fast Halpern accelerating method [41], [49] for solving OT problems is presented in Algorithm 2. A detailed derivation of the algorithm in its current form and more discussions can be found in [41] and the references therein.

Algorithm 2 HOT: A Halpern accelerating method for solving the OT problem (9).

1: Input: Choose an initial point $w^0 = (y^0, z^0, x^0) \in \mathbb{R}^{M_3} \times \mathbb{R}^N \times \mathbb{R}^N$. Set parameters $\sigma > 0$. For $k = 0, 1, \ldots,$ perform the following steps in each iteration.
2: Step 1. $\bar{y}^k = \arg\min\limits_{y\in\mathbb{Y}} \{L_\sigma(y, z^k; x^k)\}$.
3: Step 2. $\bar{x}^k = x^k + \sigma(A^\top \bar{y}^k + z^k - c)$.
4: Step 3. $\bar{z}^k = \arg\min\limits_{z\in\mathbb{Z}} \{L_\sigma(\bar{y}^k, z; \bar{x}^k)\}$.
5: Step 4. $w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2}(2\bar{w}^k - w^k)$.

Note that Step 4 in Algorithm 2 is from the Halpern iteration with a stepsize of $\frac{1}{k+2}$. Without Step 4, the HOT algorithm reduces to the ADMM with a unit step size. According to [41, Corollary 3.5], we can obtain the global convergence of the HOT algorithm in the following proposition. The proof can be found in Appendix B.

Proposition 2. *The sequence $\{\bar{w}^k\} = \{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ generated by the HOT algorithm in Algorithm 2 converges to the point $w^* = (y^*, z^*, x^*)$, where (y^*, z^*) is a solution to problem (9) and x^* is a solution to problem (8).*

Next, we analyze the iteration complexity of the HOT algorithm for obtaining an ε-approximate solution, where an appropriate measure for the quality of the solution is crucial. In this paper, we consider the residual mapping associated with the KKT system (10):

$$\mathcal{R}(w) = \begin{pmatrix} b - Ax \\ z - \Pi_{\mathbb{R}^N_+}(z - x) \\ c - A^\top y - z \end{pmatrix}$$

for any $w = (y, z, x) \in \mathbb{R}^{M_3} \times \mathbb{R}^N \times \mathbb{R}^N$. Note that $\mathcal{R}(w^*) = 0$ is equivalent to the facts that $x^* \in \mathbb{R}^N$ and

$(y^*, z^*) \in \mathbb{R}^{M_3} \times \mathbb{R}^N$ are the solution to problems (8) and (9), respectively. The KKT residual $\|\mathcal{R}(\cdot)\|$ is a commonly used and practical measure for the quality of the approximation solution to (8). It follows from [41, Theorem 3.7] that the HOT algorithm enjoys an appealing $O(1/k)$ nonergodic convergence rate in terms of the KKT residual for solving (8), which is summarized in the following proposition. The proof can be found in Appendix C.

Proposition 3. *Let $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ be the sequence generated by Algorithm 2, and let $w^* = (y^*, z^*, x^*)$ be the limit point of the sequence $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ and $R_0 = \|x^0 - x^* + \sigma(z^0 - z^*)\|$. For all $k \geq 0$, we have the following bounds:*

$$\|\mathcal{R}(\bar{w}^k)\| \leq \left(\frac{\sigma+1}{\sigma}\right)\frac{R_0}{(k+1)} \tag{14}$$

and

$$-\|z^*\|\frac{R_0}{(k+1)} \leq \langle c, \bar{x}^k - x^*\rangle \leq (\sigma\|z^*\| + R_0)\frac{R_0}{\sigma(k+1)}.$$

Remark 2. *Note that, without acceleration, the ADMM has an $O(1/\sqrt{k})$ non-ergodic rate in terms of both the objective function value gap and feasibility violations [50], [51]. In contrast, the HOT algorithm in Algorithm 2 achieves an $O(1/k)$ non-ergodic convergence rate, offering significant advantages for solving large-scale OT problems.*

B. A fast implementation of the HOT algorithm

In this section, we present a fast implementation of the HOT algorithm. Through direct calculations, we obtain the following updates of $\bar{z}^k$ and $\bar{y}^k$ for any $k \geq 0$:

1) Update of $\bar{z}^k$:

$$\bar{z}^k = \Pi_{\mathbb{R}^N_+}\left(c - A^\top \bar{y}^k - \bar{x}^k/\sigma\right);$$

2) Update of $\bar{y}^k$:

$$AA^\top \bar{y}^k = \frac{b}{\sigma} - A\left(\frac{x^k}{\sigma} + z^k - c\right). \tag{15}$$

Therefore, the main computational bottleneck of the HOT algorithm for solving (8) is solving the linear system (15). Note that the dimension of the matrix $AA^\top$ is $M_3 \times M_3$ with M_3 defined in (4). In applications, M_3 is usually a huge number. As an illustrative example, for an image with 256×256 pixels, $M_3 = 196,607$. As a result, it is not computationally affordable for computing a (sparse) Cholesky decomposition for the matrix $AA^\top$ or solving the linear system (15) with standard direct solvers. Indeed, it is computationally expensive even for computing the matrix $AA^\top$. Instead, in the remaining part of this subsection, we will derive a linear time complexity procedure for solving the linear equation $AA^\top y = R$ with a given vector $R \in \mathbb{R}^{M_3}$. It is worthwhile mentioning that our procedure does not require calculating nor storing the matrix $AA^\top$. Note that $AA^\top$ can be written in the following form:

$$AA^\top = \begin{bmatrix} E_1 & E_2 & E_3 \\ E_2^\top & E_4 & \mathbf{0} \\ E_3^\top & \mathbf{0} & E_5 \end{bmatrix}, \tag{16}$$

where

1) $E_1 = (m+n)I_M \in \mathbb{R}^{M\times M}$;
2) $E_2 = \operatorname{diag}\left(\mathbf{1}_m\mathbf{1}_m^\top, \ldots, \mathbf{1}_m\mathbf{1}_m^\top, \mathbf{1}_m\mathbf{1}_m^\top\right) \in \mathbb{R}^{M\times M}$;
3) $E_3 = -\mathbf{1}_n \otimes (I_m, \ldots, I_m, \bar{I}_m^\top) \in \mathbb{R}^{M\times(M-1)}$;
4) $E_4 = mI_M \in \mathbb{R}^{M\times M}$;
5) $E_5 = A_4A_4^\top = nI_{M-1} \in \mathbb{R}^{(M-1)\times(M-1)}$.

To better explore the structure of the linear system $AA^\top y = R$, we rewrite it equivalently as

$$AA^\top y = \begin{bmatrix} E_1 & E_2 & E_3 \\ E_2^\top & E_4 & \mathbf{0} \\ E_3^\top & \mathbf{0} & E_5 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}, \tag{17}$$

where $y := (y_1; y_2; y_3) \in \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R}^{M-1}$ and $R := (R_1; R_2; R_3) \in \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R}^{M-1}$. To further explore the block structure of the linear system, we can denote $y_i := (y_i^1; \ldots; y_i^n) \in \mathbb{R}^m \times \cdots \times \mathbb{R}^m$ for $i = 1, 2$, and $y_3 = (y_3^1; \ldots; y_3^n) \in \mathbb{R}^m \times \cdots \times \mathbb{R}^m \times \mathbb{R}^{m-1}$. Correspondingly, we write $R_i = (R_i^1; \ldots; R_i^n)$ for $i = 1, 2, 3$. The next proposition gives an explicit formula of the solution to the linear equation in (17).

Proposition 4. *Consider $A \in \mathbb{R}^{M_3\times N}$ defined in (6). Given $R \in \mathbb{R}^{M_3}$, the solution y to $AA^\top y = R$ in the form (17) is given by:*

$$y_2^j = \frac{1}{m}(R_2^j - \mathbf{1}_m^\top y_1^j), \quad j = 1, \ldots, n, \tag{18}$$

$$y_3^j = \frac{1}{n}(R_3^j + \sum_{j=1}^n y_1^j), \quad j = 1, \ldots, n-1, \tag{19}$$

$$y_3^n = \frac{1}{n}(R_3^n + \bar{I}_m \sum_{j=1}^n y_1^j), \tag{20}$$

$$y_1^j = \hat{y}_1^j - \hat{y}_1^a, \quad j = 1, \ldots, n, \tag{21}$$

where

1) $\hat{y}_1^j = \frac{1}{m+n}\left(\tilde{R}_1^j + \tilde{R}_2^j + \tilde{R}_3\right)$, $j = 1, \ldots, n$, with $\tilde{R}_1^j = R_1^j + \frac{1}{n}\mathbf{1}_m^\top R_1^j$, $\tilde{R}_2^j = -\left(\frac{1}{m} + \frac{1}{n}\right)\mathbf{1}_m^\top R_2^j$, and $\tilde{R}_3 = \frac{1}{n}\left(\sum_{j=1}^{n-1} R_3^j + \bar{I}_m^\top R_3^n\right) + \frac{1}{n^2}\mathbf{1}_{M-1}^\top R_3$;

2) $\hat{y}_1^a = \left(I_m + \frac{1}{n}\mathbf{1}_m\mathbf{1}_m^\top\right)\hat{W}\sum_{j=1}^n \hat{y}_1^j$;

3) $\hat{W} = \left(-\operatorname{diag}\left(\frac{1}{m}I_{m-1}, \frac{1}{m+1}\left(1 - \frac{1}{n}\right)\right) - \frac{1}{w}dd^\top\right)$, with $d = \left[\frac{1}{m}\mathbf{1}_{m-1}; \frac{1}{m+1}\left(1 - \frac{1}{n}\right)\right] \in \mathbb{R}^m$ and $w = \frac{1}{m} - \frac{1}{(m+1)}\left(1 - \frac{1}{n}\right)$.

Proof. By some direct calculations, we can solve (17) equivalently as:

$$y_2 = \frac{R_2 - E_2^\top y_1}{m}, \tag{22}$$

$$y_3 = \frac{R_3 - E_3^\top y_1}{n}, \tag{23}$$

$$\tilde{E}_1 y_1 = R_1 - \frac{1}{m}E_2R_2 - \frac{1}{n}E_3R_3, \tag{24}$$

where $\tilde{E}_1 = \left(E_1 - \frac{1}{m}E_2E_2^\top - \frac{1}{n}E_3E_3^\top\right)$. As a result, the key is to obtain y_1 by solving (24). Let $\hat{E}_1 := E_1 - \frac{1}{m}E_2E_2^\top$. By direct calculations, we have:

$$\hat{E}_1 = \operatorname{diag}\left(\hat{E}_1^1, \ldots, \hat{E}_1^n\right) \tag{25}$$

with $\hat{E}_1^j = (m+n)I_m - \mathbf{1}_m\mathbf{1}_m^\top$, $j = 1,\ldots,n$. By the Sherman–Morrison-Woodbury formula, we directly get:

$$\hat{E}_1^{-1} = \operatorname{diag}\left((\hat{E}_1^1)^{-1},\ldots,(\hat{E}_1^n)^{-1}\right) \tag{26}$$

with $(\hat{E}_1^j)^{-1} = \frac{1}{m+n}\left(I_m + \frac{1}{n}\mathbf{1}_m\mathbf{1}_m^\top\right), j = 1,\ldots,n$. On the other hand, let:

$$Q = \frac{n-1}{n}I_m + \frac{1}{n}\bar{I}_m^\top\bar{I}_m \in \mathbb{R}^{m\times m}.$$

Denote $\hat{Q} := Q^{1/2}$ such that $\hat{Q}\hat{Q} = Q$, and $\bar{Q} := \mathbf{1}_n \otimes \hat{Q}$. We can obtain:

$$\tilde{E}_1 = \left(\hat{E}_1 - \frac{1}{n}E_3E_3^\top\right) = \hat{E}_1 - \bar{Q}\bar{Q}^\top.$$

Hence, by the Sherman–Morrison-Woodbury formula, we can derive:

$$\begin{aligned}\tilde{E}_1^{-1} &= \left(\hat{E}_1 - \tfrac{1}{n}E_3E_3^\top\right)^{-1}\\ &= \hat{E}_1^{-1} - \hat{E}_1^{-1}\bar{Q}W^{-1}\bar{Q}^\top\hat{E}_1^{-1}\end{aligned} \tag{27}$$

with $W = -I_m + \bar{Q}^\top\hat{E}_1^{-1}\bar{Q}$. Note that:

$$\begin{aligned}W &= -I_m + \textstyle\sum_{j=1}^n \hat{Q}(E_1^j)^{-1}\hat{Q}\\ &= \tfrac{1}{m+n}\left(\operatorname{diag}\left(-mI_{m-1}, -(m+1)\right) + d_0d_0^\top\right),\end{aligned}$$

where $d_0 = [\mathbf{1}_{m-1}; \sqrt{1-\frac{1}{n}}] \in \mathbb{R}^m$. Applying the Sherman–Morrison-Woodbury formula to W, we can obtain:

$$W^{-1} = (m+n)\left(\operatorname{diag}(-\tfrac{1}{m}I_{m-1}, -\tfrac{1}{m+1}) - \tfrac{1}{w}d_1d_1^\top\right)$$

with $d_1 = [\frac{1}{m}\mathbf{1}_{m-1}; \frac{1}{m+1}\sqrt{1-\frac{1}{n}}]$. It follows that

$$\begin{aligned}\bar{Q}W^{-1}\bar{Q}^\top &= \mathbf{1}_n\mathbf{1}_n^\top \otimes (\hat{Q}W^{-1}\hat{Q})\\ &= \mathbf{1}_n\mathbf{1}_n^\top \otimes (m+n)\hat{W}.\end{aligned} \tag{28}$$

To explore the block structure of R, we denote:

$$\hat{R}_1 = R_1 - \frac{1}{m}E_2R_2 - \frac{1}{n}E_3R_3 = (\hat{R}_1^1;\ldots;\hat{R}_1^n) \in \mathbb{R}^m\times\cdots\times\mathbb{R}^m,$$

which implies

$$\hat{R}_1^j = R_1^j - \frac{1}{m}\mathbf{1}_m^\top R_2^j + \frac{1}{n}\left(\sum_{j=1}^{n-1}R_3^j + \bar{I}_m^\top R_3^n\right), \quad j = 1,\ldots,n.$$

Then, for $j = 1,\ldots,n$,

$$\begin{aligned}\hat{y}_1^j &= (\hat{E}_1^{-1}\hat{R}_1)^j\\ &= \tfrac{1}{m+n}\left(\tilde{R}_1^j + \tilde{R}_2^j + \tilde{R}_3\right).\end{aligned}$$

Define

$$\hat{y}_1^a := \left(I_m + \frac{1}{n}\mathbf{1}_m\mathbf{1}_m^\top\right)\hat{W}\sum_{j=1}^n \hat{y}_1^j.$$

From (24), (27), and (28), we have:

$$y_1^j = \hat{y}_1^j - \hat{y}_1^a, \quad j = 1,\ldots,n.$$

Substituting y_1 into (22) and (23), we can obtain the results for y_2 and y_3. This completes the proof. □

According to the explicit formula in Proposition 4, we can immediately derive the complexity result for solving the linear equation in (17).

Corollary 1. *Consider $A \in \mathbb{R}^{M_3\times N}$ defined in* (6)*. The linear system $AA^\top y = R$ in the form* (17) *can be solved in $O(M_3)$ flops.*

Based on this corollary, we can determine the per-iteration computational cost of the HOT algorithm in Algorithm 2 for each iteration.

Corollary 2. *The per-iteration computational complexity of the HOT algorithm in Algorithm 2 in terms of flops is $O(N)$.*

Proof. Since A has at most $2N$ nonzero elements, the computational cost for calculating Ax and $A^\top y$ is only $O(N)$. Except for solving linear systems, which can be done in $O(M_3)$ from Corollary 1, Algorithm 2 primarily involves matrix-vector multiplications and vector additions. Hence, the computational cost of Algorithm 2 for each iteration is $O(N)$. □

Remark 3. *Leveraging the sparsity of A, we can compute Ax and $A^\top y$ with linear space complexity $O(N)$. Furthermore, according to Proposition 4, the linear system $AA^\top y = R$ can be solved with a memory cost of $O(M_3)$. Hence, Algorithm 2 can be implemented with linear space complexity $O(N)$. When $m = n$, this corresponds to $O(M^{1.5})$.*

Combining the iteration complexity in Proposition 3 and the computational cost for each iteration in Corollary 2, we can derive the following overall computational complexity result of the HOT algorithm for solving problem (3).

Theorem 1. *Let $\{\bar{y}^k, \bar{z}^k, \bar{x}^k\}$ be the sequence generated by the HOT algorithm in Algorithm 2. For any given tolerance $\varepsilon > 0$, the HOT algorithm needs at most*

$$\frac{1}{\varepsilon}\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|z^0 - z^*\right\|\right)\right) - 1$$

iterations to return a solution to the equivalent OT problem (8) *such that the KKT residual $\left\|\mathcal{R}\left(\bar{w}^k\right)\right\| \le \varepsilon$, where (x^*, z^*) is the limit point of the sequence $\{\bar{x}^k, \bar{z}^k\}$. In particular, the overall computational complexity of the HOT algorithm in Algorithm 2 to achieve this accuracy in terms of flops is*

$$O\left(\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|z^0 - z^*\right\|\right)\right)\frac{m^2n + mn^2}{\varepsilon}\right).$$

Remark 4. *Note that the complexity of the Sinkhorn method for solving the original OT problem* (1) *is $O\left(\frac{m^2n^2}{\varepsilon^2}\right)$ to achieve an ε-accuracy solution in terms of objective function value [13]. Even by leveraging the separable structure of the cost function c defined in* (2)*, the convolutional Sinkhorn method, as mentioned in [3] and [37], still requires $O\left(\frac{m^2n+mn^2}{\varepsilon^2} + m^2n^2\right)$ to compute the KW distance. In contrast, the HOT algorithm exhibits lower overall computational complexity, offering a significant advantage in calculating the KW distance between large-scale histograms.*

Remark 5. *Assume that $m = n$. Blanchet et al. [52] demonstrated that a running time of $\widetilde{O}(M^2/\varepsilon)$ is a bottleneck in their approach by reducing an instance of the bipartite matching*

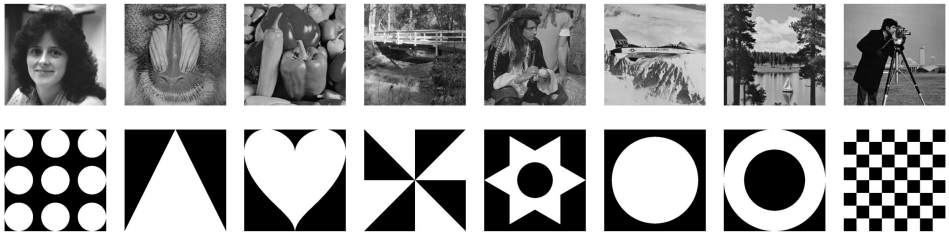


Fig. 2: A visualization of the selected images from the DOTmark Dataset is presented. The upper row features images from the Classic Images category, while the bottom row contains images from the Shapes category.

problem (BMP) to an OT problem. While HOT achieves a complexity of $O(M^{1.5}/\varepsilon)$, it cannot be directly applied to solve the maximum cardinality BMP. This is because, in the reduction from BMP to the OT problem, the ground distance used in [52] differs from the one in our paper. Consequently, the hardness result in [52] does not directly contradict our complexity bound.

IV. Experiments

In this section, we comprehensively compare the HOT algorithm with five other state-of-the-art methods on the popular DOTmark dataset [53]. Additionally, employing the transport plan derived from Algorithm 1, we test the performance of the reduced OT model on the color transfer task. The numerical results reveal that HOT provides significant advantages over state-of-the-art methods in both memory and computational efficiency, particularly for large-scale problems. More numerical results to compare the HOT algorithm to the W2NeuralDual method implemented in the OTT-JAX library [54] can be found in Appendix E. We also present an additional application of HOT for domain adaptation in Appendix F.

A. Numerical comparison on the DOTmark dataset

The DOTmark dataset [53] is a comprehensive collection of benchmark instances for evaluating and comparing algorithms in the field of optimal transport. It consists of a variety of instances categorized into different classes, such as Classic Images, Shapes, and Gaussian Distributions. In this experiment, we selected eight images each from the Classic Images and Shapes categories, as illustrated in Fig. 2. These selected images were resized to four different resolutions: 64×64, 128×128, 256×256, and 512×512. Finally, we randomly selected 10 pairs from each category and computed the KW distance for each pair.

To exhibit the superiority of HOT, we compare it with the following five state-of-the-art methods:

- **Sinkhorn** [10] is a widely used algorithm for computing an approximate KW distance by solving an entropy-regularized OT problem. As the Introduction Section discusses, its performance is sensitive to the regularization parameter, denoted as λ. To achieve varying levels of solution accuracy, we selected the λ to be 1%, 0.1%, and 0.01% of the median transport cost, following the setup from [3]. Due to the potential numerical instability caused by small λ values, we employed the Log-domain Sinkhorn algorithm implemented by POT [55] as the baseline method.
- **Convolutional Sinkhorn** [3], [38] is an improved version of the Sinkhorn algorithm, optimized for computing distances over regular two-dimensional grids. Specifically, since the kernel matrix used by the Sinkhorn algorithm can be constructed using a Kronecker product of two smaller matrices, a matrix-vector product using a matrix of dimension $M \times M$ can be replaced by two matrix-matrix products over matrices of dimension $\sqrt{M} \times \sqrt{M}$, resulting in a significant improvement in computational efficiency. In our experiment, based on the MATLAB implementation of the convolutional Sinkhorn in [37], we further developed the Log-domain convolutional Sinkhorn using PyTorch. The regularization parameter λ is kept the same as that of the original Sinkhorn method.
- **Gurobi (11.0.1)** is a popular optimization solver designed to address a wide range of mathematical programming problems, including linear and quadratic programming. In our experiment, we employ the interior point method (IPM) implemented in Gurobi to solve the reduced model (8). Since it is unnecessary to obtain a basic solution, we disable the cross-over strategy.
- **Network Simplex** implemented in the Lemon C++ graph library [3], is a highly efficient algorithm for solving uncapacitated minimum cost flow problems [56]. It has demonstrated the computational advantages over the Sinkhorn type methods in [37] for small size images by solving the reduced model (8).
- **ADMM** [46], [47] is a popular first-order primal-dual method for solving large-scale optimization problems. It has shown great potential in solving large-scale optimal transport problems (1) in a GPU setting [20]. In this numerical experiment, we use a generalized ADMM [57], [58] to solve the equivalent model (8) (replacing Step 4 in Algorithm 2 with $w^{k+1} = (1-\rho)w^k + \rho\bar{w}^k$ and setting $\rho = 1.7$) as the baseline to evaluate the acceleration effect of the Halpern iteration.

All experiments are conducted on an Ubuntu 22.04 server equipped with an Intel(R) Xeon(R) Platinum 8480C processor and an Nvidia GeForce RTX 4090 GPU with 24 GB of RAM. Due to hardware specifications, we limit the maximum memory usage of each algorithm to 24 GB. Additionally, we

[3] https://lemon.cs.elte.hu/

set the maximum one-call running time of each algorithm to 3600 seconds. For the HOT and ADMM methods, we adopt a stopping criterion based on the relative KKT residual:

$$\mathrm{KKT}_{\mathrm{res}} = \max\left\{\frac{\|A^\top y + z - c\|}{1+\|c\|}, \frac{\|\min(x,z)\|}{1+\|x\|+\|z\|}, \frac{\|Ax-b\|}{1+\|b\|}\right\}. \tag{29}$$

Other methods use their default stopping criteria. We terminate all tested algorithms, except for Network Simplex which is an exact algorithm with a stopping tolerance of 1E-6. Finally, since different methods have varying stopping criteria, we use the following metrics to fairly evaluate the quality of the solutions: the 'relative objective gap' (gap) and the 'relative primal feasibility error' (feaserr). These metrics are defined as follows:

$$\begin{aligned}\mathrm{gap} &= \tfrac{|\langle c,x\rangle - \langle c,x_b\rangle|}{|\langle c,x_b\rangle|+1},\\ \mathrm{feaserr} &= \max\left\{\tfrac{\|\min(x,\mathbf{0})\|}{1+\|x\|}, \tfrac{\|Ax-b\|}{1+\|b\|}\right\},\end{aligned}$$

where x_b is the solution obtained using Gurobi with the tolerance set to 1E-8.

We present the average results of 10 pairs within each category for all tested algorithms in Table II. Since Gurobi runs out of memory for images sized from 256×256 to 512×512, we only report the feaserr for the HOT, Network Simplex, and ADMM. Additionally, Table II only shows the results of Sinkhorn-type methods with $\lambda = 0.01\%$ of the median transport cost, as this parameter returns a solution of comparable quality to other methods. For more results of the convolutional Sinkhorn with different regularization parameters, refer to Table III. Due to space constraints, we omit the results of the Sinkhorn with varying parameters, which exhibit similar performance to the convolutional Sinkhorn in terms of the gap.

We summarized some key findings in Table II from the perspective of computational efficiency and memory cost:

1) **Computational efficiency:** HOT can return a comparable solution in terms of the gap and feaserr in the shortest time. Although the Network Simplex and Gurobi have computational advantages for computing the KW distance for small-scale images, these methods cannot handle large-scale problems effectively. IPM implemented in Gurobi requires solving a linear equation that relies heavily on Cholesky decomposition, causing the computational cost of each iteration to increase rapidly with image size. Additionally, the inherent sequential nature of the Network Simplex algorithm makes it challenging to parallelize effectively. In contrast, HOT and convolutional Sinkhorn benefit from low per-iteration costs and are easily parallelizable. However, the convolutional Sinkhorn needs to recover the solution to the original problem (1) to compute the KW distance, making it unsuitable for large-scale problems. As a result, for images sized 128×128, HOT achieves a 17.44x speedup over Network Simplex, a 15.83x speedup over Gurobi, and a 19.54x speedup over convolutional Sinkhorn. Additionally, compared to ADMM, HOT benefits from a superior $O(1/k)$ iteration complexity and saves 40% of iterations for images sized 512×512.

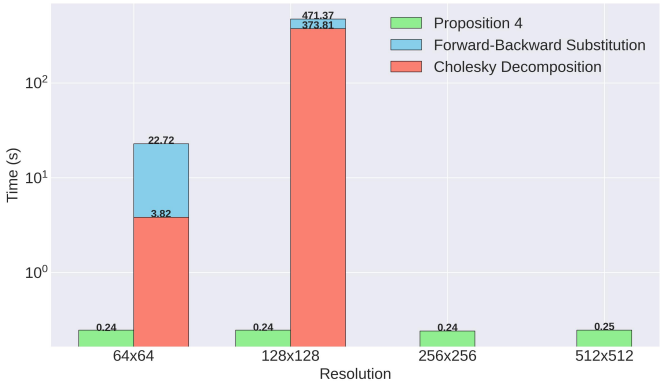


Fig. 3: Comparison of solving the linear system (17) using Proposition 4 and the sparse Cholesky decomposition. The time for solving the linear system using Cholesky decomposition is divided into two parts: the time for Cholesky decomposition (orange part) and the time for forward-backward substitution (blue part). For the 256×256 and 512×512 cases, Cholesky decomposition is out-of-memory in the test.

2) **Memory cost:** HOT demonstrates a significant advantage in memory efficiency by exploiting the sparse structure of A in (5) and utilizing the explicit solution of the linear equation presented in Proposition 4 to avoid sparse Cholesky decomposition. In contrast, Sinkhorn-type methods need to recover the solution to the original OT problem (1) and maintain the transport cost to calculate the KW distance, which involves M^2 variables. Given that these variables are stored using 64-bit floating-point representation, the Sinkhorn-based methods require at least 32 GB of memory for images sized 256×256, which far exceeds the available 24 GB. While the IPM in Gurobi solves the reduced model (8), each iteration requires performing a sparse Cholesky decomposition, making it unsuitable for images sized 256×256 or larger.

To further illustrate the benefit of the explicit solution of the linear system presented in Proposition 4, we conduct a comparison between solving the linear equation using Proposition 4 and sparse Cholesky decomposition. Given that the average iteration number of HOT is around 1500, we solve the linear system (17) 1500 times for different image sizes. The results are shown in Fig. 3. It is clear that the computational costs of Cholesky decomposition increase rapidly as image size increases. Additionally, the forward-backward substitution method used to solve the linear system is also time-consuming because it cannot be efficiently parallelized. In contrast, solving linear systems using Proposition 4 can be easily parallelized. It consumes significantly less time and remains constant as the size varies because it only requires $O(M_3)$ flops, as shown in Corollary 1. Additional numerical results on transport plan recovery time, GPU vs. CPU comparisons, and the acceleration effects of Halpern iteration for high-accuracy solutions can be found in Appendix D.

TABLE II: The numerical results of different algorithms on the DOTmark dataset.

Category	Resolution		**HOT**	Network Simplex	Gurobi	ADMM	Convolutional Sinkhorn	Sinkhorn
Classic	64×64	time(s)	**0.67**	2.73	2.16	1.77	16.18	174.82
		gap	8.26E-04	3.46E-10	1.20E-04	2.67E-04	1.69E-04	2.12E-04
		feaserr	4.58E-07	4.88E-32	2.55E-11	3.09E-07	7.90E-07	9.75E-07
		iter	1700	-	13	3420	64126	62474
	128×128	time(s)	**1.58**	36.18	29.15	3.53	39.40	2632.17
		gap	6.24E-03	8.74E-10	1.07E-04	1.72E-03	6.98E-04	6.35E-04
		feaserr	7.27E-07	9.67E-32	7.24E-12	3.73E-07	8.34E-07	9.87E-07
		iter	1170	-	14	3240	58446	57010
	256×256	time(s)	**12.98**	2562.92	Memory Overflow	20.80	Memory Overflow	Memory Overflow
		feaserr	8.05E-07	1.35E-31		6.04E-07		
		iter	1140	-		2250		
	512×512	time(s)	**81.02**	Over Maximum Running Time	Memory Overflow	116.92	Memory Overflow	Memory Overflow
		feaserr	3.28E-07			4.32E-07		
		iter	900			1610		
Shapes	64×64	time(s)	**0.64**	1.48	1.33	3.92	9.60	103.74
		gap	3.78E-04	1.81E-10	2.28E-05	5.85E-05	4.86E-05	6.07E-05
		feaserr	5.77E-07	7.24E-32	1.88E-10	2.58E-07	7.95E-07	9.68E-07
		iter	1610	-	15	10430	37986	37077
	128×128	time(s)	**1.68**	20.70	22.46	2.32	24.32	1616.34
		gap	2.51E-03	2.46E-09	2.19E-05	4.11E-04	3.28E-04	3.09E-04
		feaserr	1.01E-06	1.16E-31	2.01E-10	7.74E-07	8.01E-07	9.83E-07
		iter	1240	-	18	2130	36080	35009
	256×256	time(s)	**14.87**	959.77	Memory Overflow	23.30	Memory Overflow	Memory Overflow
		feaserr	6.68E-07	1.59E-31		7.17E-07		
		iter	1310	-		2530		
	512×512	time(s)	**87.12**	Over Maximum Running Time	Memory Overflow	118.10	Memory Overflow	Memory Overflow
		feaserr	3.54E-07			5.71E-07		
		iter	970			1630		

TABLE III: The numerical results of the convolutional Sinkhorn method with different λ.

Solver	Category	Resolution		$\lambda = 0.01\%$	$\lambda = 0.1\%$	$\lambda = 1\%$
Convolutional Sinkhorn	Classic	64×64	time(s)	16.18	1.60	0.19
			gap	1.69E-04	2.43E-02	3.29E-01
			feaserr	7.90E-07	8.15E-07	7.10E-07
			iter	64126	6400	650
		128×128	time(s)	39.40	3.94	0.39
			gap	6.98E-04	3.34E-02	3.51E-01
			feaserr	8.34E-07	8.29E-07	7.29E-07
			iter	58446	5850	594

B. An application in color transfer

Color transfer [1]–[3] based on the OT model has found important applications in various fields, including digital image processing, computer graphics, and visual arts. It involves transferring the color characteristics from a target image to a source image to achieve a desired visual effect. Inspired by its convincing performance in color grading and color histogram manipulation [2], [3], we conduct the color transfer over the CIE-Lab domain by applying the optimal transport model to the 1D luminance channel and the 2D chrominance channel independently. Note that the 1D optimal transport plan can be found efficiently [8]. To efficiently conduct the optimal transport over the 2D chrominance channel, we first solve the equivalent reduced OT model using the HOT Algorithm 2 and then recover an optimal transport plan using Algorithm 1. To construct the supports of the two-dimensional histograms used in the reduced OT model, we apply the K-means algorithm with $K = 128$ to all chrominance values in the CIE-Lab color space that appear in the source image or the target image. For each resulting centroid, we draw a vertical and a horizontal line. The intersections of these lines form a set of (non-uniform) bins, which serve as the histogram supports. The performance of color transfer for selected image pairs can be found in Fig. 1. More results about color transfer can be found in Appendix G.

V. Conclusion

In this paper, we proposed an efficient and scalable HOT algorithm for computing the KW distance with finite supports. In particular, the HOT algorithm solves an equivalent reduced OT model where the involved linear systems are solved by a novel procedure in linear time complexity. Consequently, we can obtain an ε-approximate solution to the OT problem with M supports in $\mathbb{R}^2$ in $O(M^{1.5}/\varepsilon)$ flops, significantly enhancing the best-known computational complexity. Additionally, we have designed an efficient algorithm to recover an optimal transport plan from a solution to the reduced OT model, thereby overcoming the limitations of the reduced OT model in applications that require the transport plan. The extensive numerical results presented in this paper demonstrated the superior performance of the HOT algorithm. For future research directions, we aim to design an efficient implementation of the HOT algorithms for solving the OT problems with discrete supports in the more general $\mathbb{R}^d$ space. We also consider designing an efficient algorithm for solving the Wasserstein barycenter problem with discrete supports.

Acknowledgment

The research of Yancheng Yuan was supported by the Research Center for Intelligent Operations Research and The Hong Kong Polytechnic University under the grant P0045485. The research of Defeng Sun was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (GRF Project No. 15303720) and the Research Center for Intelligent Operations Research.

References

[1] F. Pitié and A. Kokaram, "The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer," *4th European Conference on Visual Media Production*, 2007.

[2] N. Bonneel, K. Sunkavalli, S. Paris, and H. Pfister, "Example-based video color grading," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 4, pp. 1–12, 2013.

[3] J. Solomon, F. De Goes, G. Peyré, *et al.*, "Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 4, pp. 1–11, 2015.

[4] A. Dominitz and A. Tannenbaum, "Texture mapping via optimal mass transport," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 419–433, 2009.

[5] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, "Optimal mass transport for registration and warping," *International Journal of Computer Vision*, vol. 60, pp. 225–240, 2004.

[6] S. Basu, S. Kolouri, and G. K. Rohde, "Detecting and visualizing cell phenotype differences from microscopy images using transport-based morphometry," *Proceedings of the National Academy of Sciences*, vol. 111, no. 9, pp. 3448–3453, 2014.

[7] E. Del Barrio, J. A. Cuesta-Albertos, C. Matrán, and J. M. Rodríguez-Rodríguez, "Tests of goodness of fit based on the L2-Wasserstein distance," *Annals of Statistics*, pp. 1230–1239, 1999.

[8] C. Villani, "Topics in optimal transportation," *Graduate Studies in Mathematics*, 2003.

[9] G. Peyré, M. Cuturi, *et al.*, "Computational Optimal Transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[10] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[11] J. Orlin, "A faster strongly polynomial minimum cost flow algorithm," in *Proceedings of the Twentieth Annual ACM Aymposium on Theory of Computing*, 1988, pp. 377–387.

[12] T. Lin, N. Ho, and M. Jordan, "On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms," in *International Conference on Machine Learning*, PMLR, 2019, pp. 3982–3991.

[13] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, "Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1367–1376.

[14] 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*, PMLR, 2021, pp. 3886–3898.

[15] B. Schmitzer, "Stabilized sparse scaling algorithms for entropy regularized transport problems," *SIAM Journal on Scientific Computing*, vol. 41, no. 3, A1443–A1481, 2019.

[16] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, 2009, pp. 460–467.

[17] A. V. Goldberg, É. Tardos, and R. Tarjan, "Network flow algorithm," Cornell University Operations Research and Industrial Engineering, Tech. Rep., 1989.

[18] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for general graph matching problems," *Journal of the ACM (JACM)*, vol. 38, no. 4, pp. 815–853, 1991.

[19] X. Li, D. F. Sun, and K.-C. Toh, "An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for linear programming," *SIAM Journal on Optimization*, vol. 30, no. 3, pp. 2410–2440, 2020.

[20] V. V. Mai, J. Lindbäck, and M. Johansson, "A fast and accurate splitting method for optimal transport: Analysis and implementation," *International Conference on Learning Representations*, 2022.

[21] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," *UCLA Cam Report*, vol. 34, no. 2, 2008.

[22] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.

[23] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, pp. 120–145, 2011.

[24] D. Applegate, M. Díaz, O. Hinder, *et al.*, "Practical large-scale linear programming using primal-dual hybrid gradient," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 243–20 257, 2021.

[25] A. Jambulapati, A. Sidford, and K. Tian, "A direct $\tilde{O}(1/\epsilon)$ iteration parallel algorithm for optimal transport," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] G. Zhang, Y. Yuan, and D. F. Sun, "An efficient HPR algorithm for the Wasserstein barycenter problem with $O(\mathrm{Dim(P)}/\varepsilon)$ computational complexity," *arXiv preprint arXiv:2211.14881*, 2022.

[27] N. Tupitsa, P. Dvurechensky, D. Dvinskikh, and A. Gasnikov, "Numerical methods for large-scale optimal transport," *arXiv preprint arXiv:2210.11368*, 2022.

[28] A. Khamis, R. Tsuchida, M. Tarek, V. Rolland, and L. Petersson, "Scalable optimal transport methods in machine learning: A contemporary survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[29] T. Lin, N. Ho, and M. I. Jordan, "On the efficiency of entropic regularized algorithms for optimal transport," *Journal of Machine Learning Research*, vol. 23, no. 137, pp. 1–42, 2022.

[30] 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*, vol. 4, no. 4, pp. 1369–1395, 2022.

[31] W. Wang, D. Slepčev, S. Basu, J. A. Ozolek, and G. K. Rohde, "A linear optimal transportation framework for quantifying and visualizing variations in sets of images," *International Journal of Computer Vision*, vol. 101, pp. 254–269, 2013.

[32] N. Bonneel, M. Van De Panne, S. Paris, and W. Heidrich, "Displacement interpolation using Lagrangian mass transport," in *Proceedings of the 2011 SIGGRAPH Asia Conference*, 2011, pp. 1–12.

[33] P. Indyk and N. Thaper, "Fast color image retrieval via embeddings," in *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[34] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[35] W. Leeb and R. Coifman, "Hölder–Lipschitz norms and their duals on spaces with semigroups, with applications to earth mover's distance," *Journal of Fourier Analysis and Applications*, vol. 22, pp. 910–953, 2016.

[36] H. Ling and K. Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 840–853, 2007.

[37] G. Auricchio, F. Bassetti, S. Gualandi, and M. Veneroni, "Computing Kantorovich-Wasserstein distances on d-dimensional histograms using $(d+1)$-partite graphs," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[38] J. Solomon, "Optimal transport on discrete domains," *AMS Short Course on Discrete Differential Geometry*, 2018.

[39] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014.

[40] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.

[41] D. F. Sun, Y. Yuan, G. Zhang, and X. Zhao, "Accelerating preconditioned ADMM via degenerate proximal point mappings," *arXiv preprint arXiv:2403.18618*, 2024, SIAM Journal on Optimization 35 (2025) XXX, in print.

[42] D. Kim, "Accelerated proximal point method for maximally monotone operators," *Mathematical Programming*, vol. 190, no. 1, pp. 57–87, 2021.

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

[44] F. Lieder, "On the convergence rate of the Halpern-iteration," *Optimization Letters*, vol. 15, no. 2, pp. 405–418, 2021.

[45] 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*, vol. 204, no. 1, p. 9, 2025.

[46] 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*, vol. 9, no. R2, pp. 41–76, 1975.

[47] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

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

[49] B. Halpern, "Fixed points of nonexpanding maps," *Bulletin of the American Mathematical Society*, vol. 73, no. 6, pp. 957–961, 1967.

[50] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, 2016, pp. 115–163.

[51] Y. Cui, X. Li, D. F. 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*, vol. 169, pp. 1013–1041, 2016.

[52] J. Blanchet, A. Jambulapati, C. Kent, and A. Sidford, "Towards optimal running times for optimal transport," *Operations Research Letters*, vol. 52, p. 107 054, 2024.

[53] J. Schrieber, D. Schuhmacher, and C. Gottschlich, "Dotmark–a benchmark for discrete optimal transport," *IEEE Access*, vol. 5, pp. 271–282, 2016.

[54] M. Cuturi, L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul, "Optimal transport tools (OTT): A JAX Toolbox for all things Wasserstein," *arXiv preprint arXiv:2201.12324*, 2022.

[55] R. Flamary, N. Courty, A. Gramfort, *et al.*, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.

[56] P. Kovács, "Minimum-cost flow algorithms: An experimental evaluation," *Optimization Methods and Software*, vol. 30, no. 1, pp. 94–127, 2015.

[57] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[58] Y. Xiao, L. Chen, and D. Li, "A generalized alternating direction method of multipliers with semi-proximal terms for convex composite conic programming," *Mathematical Programming Computation*, vol. 10, pp. 533–555, 2018.

Guojun Zhang is a Ph.D. candidate in the Department of Applied Mathematics at Hong Kong Polytechnic University. His research focuses on optimization theory and complexity analysis, computational optimal transport, and software development.

Zhexuan Gu is currently a Ph.D. student at The Hong Kong Polytechnic University. He received his Master's degree in the Department of Applied Mathematics at The Hong Kong Polytechnic University. His current research interests include computational optimal transport, the foundations of artificial intelligence, and its applications in healthcare and beyond.

Yancheng Yuan is an Assistant Professor at the Department of Applied Mathematics, The Hong Kong Polytechnic University. He received his PhD in Mathematics from the National University of Singapore. He was a research fellow at the NExT Research Center, National University of Singapore, mentored by Prof. Tat-Seng Chua. His research focuses on optimization theory, algorithm design and software development, the mathematical foundation of data science, and data-driven applications. His research has been published in prestigious academic journals and conferences, including Journal of Machine Learning Research, SIAM Journal on Optimization, IEEE Transactions on Neural Networks and Learning Systems, NeurIPS, ICML, WWW, SIGIR, ACL.

Defeng Sun is currently Chair Professor of Applied Optimization and Operations Research at the Hong Kong Polytechnic University. He was the President of the Hong Kong Mathematical Society. He mainly publishes in continuous optimization and machine learning. Together with Professor Kim-Chuan Toh and Dr Liuqin Yang, he was awarded the triennial 2018 Beale–Orchard-Hays Prize for Excellence in Computational Mathematical Programming by the Mathematical Optimization Society. In 2020, he was elected as a Fellow of the societies CSIAM and SIAM. In 2022, he received the RGC Senior Research Fellow Scheme award.

SUPPLEMENTARY MATERIALS FOR HOT: AN EFFICIENT HALPERN ACCELERATING ALGORITHM FOR OPTIMAL TRANSPORT PROBLEMS

In the appendix, we first establish the equivalence between the HOT algorithm and the accelerated degenerate proximal point algorithm (dPPA) [1] in Appendix A. Then, by analyzing the convergence and iteration complexity of the accelerated dPPA, we derive the global convergence result (Proposition 2) and the complexity result (Proposition 3) for the HOT algorithm in Appendices B and C, respectively. Additional numerical results, including transport plan recovery time, GPU vs. CPU comparisons, and the acceleration effects of Halpern iteration for high-accuracy solutions, are provided in Appendix D. Further comparisons between the HOT algorithm and the W2NeuralDual method from the OTT-JAX library [2] can be found in Appendix E. Moreover, we also present an application of HOT for domain adaptation in Appendix F. Finally, additional examples of color transfer are included in Appendix G.

APPENDIX A
EQUIVALENCE BETWEEN THE HOT AND THE ACCELERATED DPPA

Let $\mathcal{N}_{\mathbb{R}^N_+}(\cdot)$ denote the normal cone of $\mathbb{R}^N_+$. Note that solving problems (8) and (9) is equivalent to finding a $w^* \in \mathbb{W} = \mathbb{R}^{M_3} \times \mathbb{R}^N \times \mathbb{R}^N$ such that $0 \in \mathcal{T}w^*$, where the maximal monotone operator $\mathcal{T}$ is defined by

$$\mathcal{T}w = \begin{pmatrix} -b + Ax \\ \mathcal{N}_{\mathbb{R}^N_+}(z) + x \\ c - A^\top y - z \end{pmatrix}, \quad \forall w = (y, z, x) \in \mathbb{W}. \tag{30}$$

Consider the following self-adjoint linear operator $\mathcal{M} : \mathbb{W} \to \mathbb{W}$,

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma I_N & I_N \\ 0 & I_N & \frac{1}{\sigma} I_N \end{bmatrix}, \tag{31}$$

where I_N denotes the identity matrix in $\mathbb{R}^{N \times N}$. We can establish the equivalence between the HOT algorithm and the accelerated dPPA [1] in the following proposition.

Proposition 5. *Consider the operators $\mathcal{T}$ defined in* (30) *and $\mathcal{M}$ defined in* (31)*. Then the sequence $\{w^k\}$ generated by the HOT algorithm in Algorithm 2 coincides with the sequence $\{w^k\}$ generated by the following accelerated dPPA for any $k \geq 0$,*

$$\begin{cases} \bar{w}^k = \widehat{\mathcal{T}} w^k = (\mathcal{M} + \mathcal{T})^{-1} \mathcal{M} w^k, \\ \hat{w}^k = \widehat{\mathcal{F}} w^k = 2\bar{w}^k - w^k, \\ w^{k+1} = \dfrac{1}{k+2} w^0 + \dfrac{k+1}{k+2} \hat{w}^k, \end{cases} \tag{32}$$

with the same initial point $w^0 \in \mathbb{W}$. Additionally, $\mathcal{M}$ is an admissible preconditioner[4] such that $(\mathcal{M} + \mathcal{T})^{-1}$ is Lipschitz continuous.

Proof. We establish that $(\mathcal{M} + \mathcal{T})^{-1}$ is single-valued by contradiction. Suppose, for the sake of contradiction, that $(\mathcal{M} + \mathcal{T})^{-1}$ is not single-valued. Then, there exist distinct points $\bar{w}_1 = (\bar{y}_1, \bar{z}_1, \bar{x}_1) \in \mathbb{W}$ and $\bar{w}_2 = (\bar{y}_2, \bar{z}_2, \bar{x}_2) \in \mathbb{W}$ such that

$$\bar{w}_1, \bar{w}_2 \in (\mathcal{M} + \mathcal{T})^{-1} v$$

for some $v = (v_y, v_z, v_x) \in \mathbb{W}$. This implies that for $i = 1, 2$, the following conditions hold:

$$v_y = -b + A\bar{x}_i, \tag{33}$$
$$v_z \in \mathcal{N}_{\mathbb{R}^N_+}(\bar{z}_i) + \sigma\bar{z}_i + 2\bar{x}_i, \tag{34}$$
$$v_x = c - A^\top \bar{y}_i + \sigma^{-1}\bar{x}_i. \tag{35}$$

By direct calculations, we obtain:

$$A(\bar{x}_1 - \bar{x}_2) = 0, \tag{36}$$
$$\langle -\sigma(\bar{z}_1 - \bar{z}_2) - 2(\bar{x}_1 - \bar{x}_2), \bar{z}_1 - \bar{z}_2 \rangle \geq 0, \tag{37}$$
$$\sigma A^\top (\bar{y}_1 - \bar{y}_2) = \bar{x}_1 - \bar{x}_2, \tag{38}$$

where inequality (37) follows from the monotonicity of $\mathcal{N}_{\mathbb{R}^N_+}(\cdot)$. Substituting (38) into (36), we obtain

$$\sigma A A^\top (\bar{y}_1 - \bar{y}_2) = 0.$$

Since $AA^\top$ is invertible, it follows that

$$\bar{y}_1 - \bar{y}_2 = 0. \tag{39}$$

Substituting (39) into (38), we deduce that

$$\bar{x}_1 - \bar{x}_2 = 0. \tag{40}$$

Similarly, we conclude from (37) and (40) that

$$\bar{z}_1 - \bar{z}_2 = 0. \tag{41}$$

Thus, equations (39), (40), and (41) imply that $\bar{w}_1 = \bar{w}_2$, contradicting our assumption. Therefore, we conclude that $(\mathcal{M} + \mathcal{T})^{-1}$ is single-valued.

To establish the Lipschitz continuity of $(\mathcal{M} + \mathcal{T})^{-1}$, consider $\bar{w}_i = (\bar{y}_i, \bar{z}_i, \bar{x}_i) \in \mathbb{W}$ such that

$$\bar{w}_i = (\mathcal{M} + \mathcal{T})^{-1} v_i, \quad \text{where} \quad v_i = (v_{iy}, v_{iz}, v_{ix}), \quad i = 1, 2.$$

[4] In [3], an admissible preconditioner for the operator $\mathcal{T} : \mathbb{W} \to 2^{\mathbb{W}}$ is a linear, bounded, self-adjoint, and positive semidefinite operator $\mathcal{M} : \mathbb{W} \to \mathbb{W}$ such that $(\mathcal{M} + \mathcal{T})^{-1}\mathcal{M}$ is single-valued and has full domain.

Following the derivations in (33)–(35), we obtain for $i = 1, 2$:

$$v_{iy} = -b + A\bar{x}_i, \tag{42}$$
$$v_{iz} \in \mathcal{N}_{\mathbb{R}^N_+}(\bar{z}_i) + \sigma\bar{z}_i + 2\bar{x}_i, \tag{43}$$
$$v_{ix} = c - A^\top \bar{y}_i + \sigma^{-1}\bar{x}_i. \tag{44}$$

Substituting (44) into (42) and applying [4, Corollary 23.5.1], we obtain:

$$\bar{y}_i = (AA^\top)^{-1}\left(\frac{v_{iy}+b}{\sigma} - A(v_{ix}-c)\right),$$
$$\bar{z}_i = (\mathcal{N}_{\mathbb{R}^N_+}(\cdot) + \sigma I_N)^{-1}(v_{iz} - 2\bar{x}_i).$$

Thus, there exists a positive constant L_1 such that

$$\begin{aligned}\|\bar{y}_1 - \bar{y}_2\| &= \left\|(AA^\top)^{-1}\big(\frac{v_{1y}-v_{2y}}{\sigma} - A(v_{1x}-v_{2x})\big)\right\| \\ &\le L_1(\|v_{1y}-v_{2y}\| + \|v_{1x}-v_{2x}\|).\end{aligned} \tag{45}$$

It follows from (44) and (45) that there exists a positive constant L_2 such that

$$\begin{aligned}\|\bar{x}_1 - \bar{x}_2\| &= \|\sigma(v_{1x}-v_{2x}+A^\top(\bar{y}_1-\bar{y}_2))\| \\ &\le L_2(\|v_{1x}-v_{2x}\| + \|v_{1y}-v_{2y}\|).\end{aligned} \tag{46}$$

Similarly, since the resolvent $(\mathcal{N}_{\mathbb{R}^N_+}(\cdot) + \sigma I_N)^{-1}$ is nonexpansive [5], there also exists a positive constant L_3 such that

$$\|\bar{z}_1-\bar{z}_2\| \le L_3(\|v_{1y}-v_{2y}\|+\|v_{1z}-v_{2z}\|+\|v_{1x}-v_{2x}\|). \tag{47}$$

Hence, combining (45), (46), and (47), we conclude that there exists a positive constant L such that

$$\|\bar{w}_1 - \bar{w}_2\| \le L\|v_1 - v_2\|,$$

which establishes the Lipschitz continuity of $(\mathcal{M}+\mathcal{T})^{-1}$.

Finally, following the proof establishing the equivalence between the semi-proximal alternating direction method of multipliers (spADMM) and the (partial) PPA, as outlined in Appendix B of [6], we obtain that the sequence $\{w^k\}$ generated by Algorithm 2 coincides with the sequence $\{w^k\}$ produced by the following scheme:

$$\mathcal{M}w^k \in (\mathcal{M}+\mathcal{T})\bar{w}^k, \quad w^{k+1} = \frac{1}{k+2}w^0 + \frac{k+1}{k+2}(2\bar{w}^k - w^k).$$

Since $(\mathcal{M}+\mathcal{T})^{-1}$ is single-valued from the previous proof, it follows that

$$\bar{w}^k = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}w^k, \quad w^{k+1} = \frac{1}{k+2}w^0 + \frac{k+1}{k+2}(2\bar{w}^k - w^k).$$

For an arbitrary choice of $w^k \in \mathbb{W}$, each step in Algorithm 2 is well-defined. Thus, based on the established equivalence, we conclude that $(\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}$ has full domain. Combining this result with the Lipschitz continuity proven earlier, we deduce that $\mathcal{M}$ is an admissible preconditioner such that $(\mathcal{M}+\mathcal{T})^{-1}$ is Lipschitz continuous, completing the proof. □

Let $\mathcal{C}^\top := (0, \sqrt{\sigma}I_N, \frac{1}{\sqrt{\sigma}}I_N)$. It is straightforward to verify that $\mathcal{M} = \mathcal{C}\mathcal{C}^\top$. Define

$$\widetilde{\mathcal{T}} := \mathcal{C}^\top(\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}, \quad \widetilde{\mathcal{F}} := 2\widetilde{\mathcal{T}} - I_{\mathbb{W}}, \tag{48}$$

where $I_{\mathbb{W}}$ denotes the identity operator on $\mathbb{W}$. The following proposition summarizes some key properties of $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{F}}$.

Proposition 6. *Consider the operators $\mathcal{T}$ defined in* (30) *and $\mathcal{M}$ defined in* (31)*. Then, the operator $\widetilde{\mathcal{T}}$ in* (48) *is everywhere well-defined and firmly nonexpansive. Moreover, the operator $\widetilde{\mathcal{F}} = 2\widetilde{\mathcal{T}} - I_{\mathbb{W}}$ is nonexpansive. Furthermore, we have the equivalence*

$$\mathcal{C}^\top\mathcal{T}^{-1}(0) = \mathcal{C}^\top \operatorname{Fix}\widehat{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{F}},$$

where $\operatorname{Fix}\widehat{\mathcal{T}}$ *denotes the set of fixed points of the operator* $\widehat{\mathcal{T}}$.

Proof. The firm nonexpansiveness of $\widetilde{\mathcal{T}}$ follows from Theorem 2.13 in [3]. Furthermore, by [7, Proposition 4.4], $\widetilde{\mathcal{F}}$ is nonexpansive. Finally, from the proof of Theorem 2.14 in [3], we obtain

$$\mathcal{C}^\top \operatorname{Fix}\widehat{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{T}}.$$

Since $\mathcal{T}^{-1}(0) = \operatorname{Fix}\widehat{\mathcal{T}}$, the result follows. □

To analyze the global convergence and iteration complexity of the HOT algorithm (Algorithm 2), we introduce two shadow sequences $\{u^k\}$ and $\{\bar{u}^k\}$, defined as

$$u^k := \mathcal{C}^\top w^k, \quad \bar{u}^k := \mathcal{C}^\top \bar{w}^k, \quad \forall k \ge 0, \tag{49}$$

where $\{w^k\}$ and $\{\bar{w}^k\}$ are the sequences generated by Algorithm 2. Applying Proposition 5, we obtain the following identity:

$$u^{k+1} = \frac{1}{k+2}u^0 + \frac{k+1}{k+2}\widetilde{\mathcal{F}}u^k, \quad \forall k \ge 0. \tag{50}$$

We are now ready to prove Proposition 2.

Appendix B
Proof of Proposition 2

Proof. Note that the scheme in (50) is the Halpern iteration applied to the nonexpansive operator $\widetilde{\mathcal{F}}$. It follows from the global convergence of the Halpern iteration in [8, Theorem 2] that

$$u^k \to u^*, \tag{51}$$

where u^* is a point in $\operatorname{Fix}\widetilde{\mathcal{F}}$. By utilizing Proposition 6, we have

$$\mathcal{C}^\top\mathcal{T}^{-1}(0) = \mathcal{C}^\top \operatorname{Fix}\widehat{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{F}},$$

which implies that there exists a w^* in $\mathcal{T}^{-1}(0)$ such that $\mathcal{C}^\top w^* = u^*$. Consequently, by the relationship between $\{u^k\}$ and $\{w^k\}$ in (49), and (51), we can obtain

$$\begin{aligned}\bar{w}^k &= (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}\mathcal{C}^* w^k = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}u^k \\ &\to (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}\mathcal{C}^\top w^* = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}w^* = w^*,\end{aligned} \tag{52}$$

where the continuity of $(\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}$ is derived from the composition of a continuous function $(\mathcal{M}+\mathcal{T})^{-1}$ showed in Proposition 5 and a linear operator $\mathcal{C}$. Hence, $\{\bar{w}^k\}$ converges to w^*, which completes the proof. □

APPENDIX C
PROOF OF PROPOSITION 3

Proof. The shadow sequence $\{u^k\}$ satisfying (50) corresponds exactly to the Halpern iteration. By Proposition 6, we know that $\widetilde{\mathcal{F}}$ is nonexpansive. Applying [9, Theorem 2.1], we obtain

$$\|u^k - \widetilde{\mathcal{F}}u^k\| \leq \frac{2\|u^0 - u^*\|}{k+1}, \quad \forall k \geq 0, \quad u^* \in \operatorname{Fix}\widetilde{\mathcal{F}}. \tag{53}$$

Furthermore, by Proposition 6, we have $\operatorname{Fix}\widetilde{\mathcal{F}} = \mathcal{C}^*\mathcal{T}^{-1}(0)$. Thus, for any $u^* \in \operatorname{Fix}\widetilde{\mathcal{F}}$, there exists a point $w^* \in \mathcal{T}^{-1}(0)$ such that $\mathcal{C}^*w^* = u^*$. Substituting this into (53), we obtain, for all $k \geq 0$ and $w^* \in \mathcal{T}^{-1}(0)$,

$$\|\mathcal{C}^*w^k - \mathcal{C}^*\widehat{\mathcal{F}}w^k\| \leq \frac{2\|\mathcal{C}^*w^0 - \mathcal{C}^*w^*\|}{k+1},$$

which implies

$$\|w^k - \hat{w}^k\|_{\mathcal{M}} = \|w^k - \widehat{\mathcal{F}}w^k\|_{\mathcal{M}} \leq \frac{2\|w^0 - w^*\|_{\mathcal{M}}}{k+1} \tag{54}$$

with the seminorm defined by $\|w\|_{\mathcal{M}} := \sqrt{\langle w, w\rangle_{\mathcal{M}}} = \sqrt{\langle w, \mathcal{M}w\rangle}$.

We now estimate the convergence rate of $\mathcal{R}(\bar{w}^k)$ for any $k \geq 0$. From (54), we obtain

$$\|\hat{w}^k - w^k\|_{\mathcal{M}}^2 \leq \frac{4\|w^0 - w^*\|_{\mathcal{M}}^2}{(k+1)^2}, \quad \forall k \geq 0.$$

By the definition of $\mathcal{M}$ in (31), this can be rewritten as

$$\frac{1}{\sigma}\|\sigma(\hat{z}^k - z^k) + (\hat{x}^k - x^k)\|^2 \leq \frac{4R_0^2}{\sigma(k+1)^2}, \quad \forall k \geq 0. \tag{55}$$

From Step 2 of the accelerated dPPA scheme in (32), we obtain, for any $k \geq 0$,

$$\begin{cases} \hat{y}^k - y^k = 2(\bar{y}^k - y^k), \\ \hat{z}^k - z^k = 2(\bar{z}^k - z^k), \\ \hat{x}^k - x^k = 2(\bar{x}^k - x^k). \end{cases}$$

Thus, we can rewrite (55) as

$$\frac{1}{\sigma}\|\sigma(\bar{z}^k - z^k) + (\bar{x}^k - x^k)\|^2 \leq \frac{R_0^2}{\sigma(k+1)^2}, \quad \forall k \geq 0. \tag{56}$$

From Step 2 of Algorithm 2, we deduce that for any $k \geq 0$,

$$\begin{aligned} &\|\sigma(\bar{z}^k - z^k) + (\bar{x}^k - x^k)\| \\ &= \|\sigma(\bar{z}^k - z^k) + \sigma(A^\top\bar{y}^k + z^k - c)\| \\ &= \sigma\|A^\top\bar{y}^k + \bar{z}^k - c\|, \end{aligned}$$

which, combined with (56), yields

$$\|A^\top\bar{y}^k + \bar{z}^k - c\| \leq \frac{R_0}{\sigma(k+1)}, \quad \forall k \geq 0. \tag{57}$$

Moreover, from the optimality conditions of the subproblems in Algorithm 2, we obtain, for any $k \geq 0$,

$$\begin{cases} AA^\top\bar{y}^k = \frac{b}{\sigma} - A\left(\frac{x^k}{\sigma} + z^k - c\right), \\ \bar{z}^k = \Pi_{\mathbb{R}^N_+}\left(\bar{z}^k - \bar{x}^k - \sigma(A^\top\bar{y}^k + \bar{z}^k - c)\right). \end{cases} \tag{58}$$

Thus, from Step 2 of Algorithm 2 and (58), we have

$$\begin{aligned} &\|b - A\bar{x}^k\| \\ &= \|b - A(x^k + \sigma(A^\top\bar{y}^k + z^k - c))\| \\ &= \|b - A(x^k + \sigma(z^k - c)) - (b - A(x^k + \sigma(z^k - c)))\| \\ &= 0. \end{aligned} \tag{59}$$

Similarly, from (57) and (58), we obtain, for any $k \geq 0$,

$$\begin{aligned} \|\bar{z}^k - \Pi_{\mathbb{R}^N_+}(\bar{z}^k - \bar{x}^k)\| &\leq \sigma\|A^\top\bar{y}^k + \bar{z}^k - c\| \\ &\leq \tfrac{R_0}{(k+1)}. \end{aligned} \tag{60}$$

Therefore, by (57), (59), and (60), we conclude that for any $k \geq 0$,

$$\|\mathcal{R}(\bar{w}^k)\| \leq \frac{\sigma+1}{\sigma}\frac{R_0}{(k+1)}.$$

We now estimate the objective errors. For any $k \geq 0$, define

$$\tilde{x}^k = \bar{x}^k + \sigma(A^\top\bar{y}^k + \bar{z}^k - c). \tag{61}$$

From the second equation in (58), it follows that

$$\tilde{x}^k \in \mathbb{R}^N_+, \quad \text{and} \quad \langle\tilde{x}^k, \bar{z}^k\rangle = 0. \tag{62}$$

Thus, by the convexity of $\delta_{\mathbb{R}^N_+}(\cdot)$ and the KKT system (10), we obtain

$$\delta_{\mathbb{R}^N_+}(\tilde{x}^k) \geq \delta_{\mathbb{R}^N_+}(x^*) + \langle -z^*, \tilde{x}^k - x^*\rangle.$$

Adding $\langle c, \bar{x}^k - x^*\rangle$ to both sides and applying (57), (59), and (61), we derive

$$\begin{aligned} \langle c, \bar{x}^k - x^*\rangle &\geq \langle c, \bar{x}^k - x^*\rangle + \langle -z^*, \tilde{x}^k - x^*\rangle \\ &= \langle c - z^*, \bar{x}^k - x^*\rangle + \langle -z^*, \sigma(A^\top\bar{y}^k + \bar{z}^k - c)\rangle \\ &= \langle A^\top y^*, \bar{x}^k - x^*\rangle + \langle -z^*, \sigma(A^\top\bar{y}^k + \bar{z}^k - c)\rangle \\ &= \langle -z^*, \sigma(A^\top\bar{y}^k + \bar{z}^k - c)\rangle \\ &\geq -\|z^*\|\frac{R_0}{k+1}. \end{aligned}$$

Similarly, from the second equation in (58), we obtain

$$-\bar{z}^k \in \mathcal{N}_{\mathbb{R}^N_+}(\tilde{x}^k),$$

which implies

$$\delta_{\mathbb{R}^N_+}(x^*) \geq \delta_{\mathbb{R}^N_+}(\tilde{x}^k) + \langle -\bar{z}^k, x^* - \tilde{x}^k\rangle.$$

Adding $\langle c, x^* - \bar{x}^k\rangle$ to both sides and using (62), we obtain

$$\begin{aligned} \langle c, x^* - \bar{x}^k\rangle &\geq \langle -\bar{z}^k, x^* - \tilde{x}^k\rangle + \langle c, x^* - \bar{x}^k\rangle \\ &= \langle -\bar{z}^k, x^*\rangle + \langle c, x^* - \bar{x}^k\rangle \\ &= \langle c - \bar{z}^k, x^*\rangle - \langle c, \bar{x}^k\rangle. \end{aligned} \tag{63}$$

For convenience, let

$$\Delta^k = c - (A^\top\bar{y}^k + \bar{z}^k).$$

Using (63) and (59), we obtain

$$\begin{aligned} \langle c, \bar{x}^k - x^*\rangle &\leq -\langle\Delta^k + A^\top\bar{y}^k, x^*\rangle + \langle\Delta^k + (A^\top\bar{y}^k + \bar{z}^k), \bar{x}^k\rangle \\ &= \langle\Delta^k, \bar{x}^k - x^*\rangle + \langle\bar{z}^k, \bar{x}^k\rangle. \end{aligned}$$

TABLE 1: Numerical results of various algorithms on the DOTmark Dataset: Total time as the sum of solve time and transport plan recovery time.

Category	Resolution		**HOT**	Network Simplex	Gurobi	ADMM	Improved Sinkhorn	Sinkhorn
Classic	64×64	time(s)	**0.67**+0.17	2.73	2.16+0.17	1.77+0.17	16.18	174.82
	128×128	time(s)	**1.58**+0.67	36.18	29.15+0.67	3.53+0.67	39.40	2632.17
Shapes	64×64	time(s)	**0.64**+0.17	1.48	1.33+0.17	3.92+0.17	9.60	103.74
	128×128	time(s)	**1.68**+0.67	20.70	22.46+0.67	2.32+0.67	24.32	1616.34

By (61) and (62), we further derive

$$\begin{aligned}\langle c, \bar{x}^k - x^* \rangle &\leq \langle \Delta^k, \bar{x}^k - x^* \rangle + \sigma \langle \bar{z}^k, \Delta^k \rangle \\ &= \langle \Delta^k, \bar{x}^k - x^* + \sigma(\bar{z}^k - z^*) \rangle + \sigma \langle z^*, \Delta^k \rangle \\ &\leq \|\Delta^k\| \|\bar{x}^k - x^* + \sigma(\bar{z}^k - z^*)\| + \sigma \|z^*\| \|\Delta^k\|. \end{aligned} \tag{64}$$

By the definition of $\mathcal{M}$ in (31) and the $\mathcal{M}$-nonexpansiveness of $\widehat{\mathcal{T}}$ [3], we have

$$\begin{aligned}\|\bar{x}^k - x^* + \sigma(\bar{z}^k - z^*)\|^2 &= \sigma \|\bar{w}^k - w^*\|_{\mathcal{M}}^2 \\ &\leq \sigma \|w^0 - w^*\|_{\mathcal{M}}^2 \\ &= \|x^0 - x^* + \sigma(z^0 - z^*)\|^2 = R_0^2. \end{aligned} \tag{65}$$

Therefore, combining (57), (64), and (65), we conclude that

$$\begin{aligned}\langle c, \bar{x}^k - x^* \rangle &\leq \|\Delta^k\| R_0 + \sigma \|z^*\| \|\Delta^k\| \\ &\leq (\sigma \|z^*\| + R_0) \frac{R_0}{\sigma(k+1)}. \end{aligned}$$

This completes the proof. □

APPENDIX D
MORE NUMERICAL RESULTS ON THE DOTMARK DATASET

Note that the worst-case computational complexity of reconstructing the transport plan via Algorithm 1 is $3M^2$. In practice, we can efficiently parallelize the (k, j) loop to leverage the significant benefits of GPU acceleration, enabling an efficient reconstruction of the transport plan from a solution to the reduced OT model. We provide some additional numerical experiment results in Table 1 to better demonstrate the efficiency of the transport plan recovery with the GPU acceleration. The results shown in Table 1 demonstrate that our approach outperforms other popular algorithms, even when accounting for the plan recovery time. We have also considered the sparsity in the transport plan recovery, which facilitates handling large-scale problems.

The numerical comparison of the HOT algorithm on GPU and CPU is presented in Table 2. The results demonstrate that the GPU's acceleration is substantial, and the acceleration effect becomes more significant as the problem's dimension increases.

To better demonstrate the significance of the Halpern acceleration, we conduct additional numerical testing by changing the stopping criterion of the algorithm from relative KKT residual to absolute KKT residual and setting the maximum number of iterations to 1E6. The results are shown in Table 3 below.

TABLE 2: The comparison of HOT's performance on CPU and GPU.

Category	Resolution		**GPU**	CPU	Ratio ($t_{\text{CPU}}/t_{\text{GPU}}$)
Classic	128×128	time(s)	**1.58**	5.91	3.74
	256×256	time(s)	**12.98**	108.98	8.40
	512×512	time(s)	**81.02**	764.22	9.43
Shapes	128×128	time(s)	**1.68**	6.15	3.66
	256×256	time(s)	**14.87**	130.81	8.80
	512×512	time(s)	**87.12**	812.82	9.33

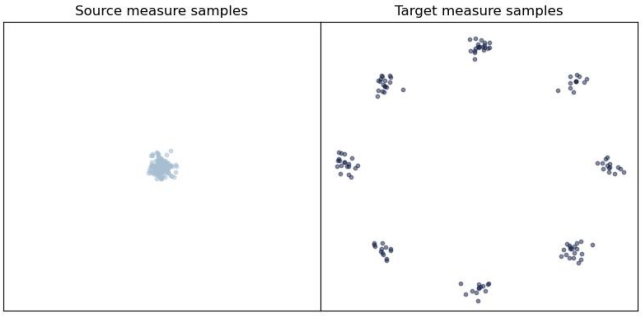
TABLE 3: The numerical results of HOT and ADMM using absolute KKT residual as the stopping criterion (1E-6).

Category	Resolution		HOT	ADMM
Classic	64×64	time(s)	6.39	406.94
		gap	**6.93E-10**	1.58E-6
		feaserr	**3.12E-13**	4.51E-10
		iter	17680	1000000
	128×128	time(s)	76.36	1093.10
		gap	**8.10E-10**	4.69E-6
		feaserr	**2.82E-14**	6.11E-10
		iter	57300	1000000
	256×256	time(s)	3774.04	9135.66
		feaserr	**3.96E-15**	9.31E-11
		iter	338240	1000000
Shapes	64×64	time(s)	6.34	404.29
		gap	**2.29E-10**	6.48E-7
		feaserr	**3.62E-13**	8.78E-10
		iter	17660	991730
	128×128	time(s)	54.86	1087.06
		gap	**2.47E-9**	2.43E-6
		feaserr	**2.33E-14**	1.01E-9
		iter	41190	1000000
	256×256	time(s)	2519.99	9101.29
		feaserr	**3.24E-15**	1.10E-10
		iter	226020	1000000

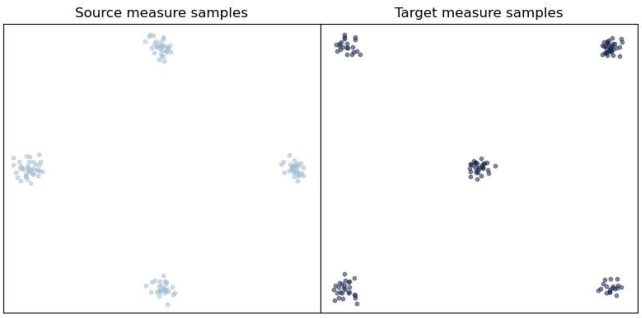
APPENDIX E
COMPARE WITH W2NEURALDUAL

In this section, we compare HOT with the W2NeuralDual method implemented in OTT-JAX [2]. We first sample some data using a built-in OTT-JAX random dataset generator in two different settings, visualized in Fig. 1. For the Circle Gaussian and the CheckerBoard dataset, we generate 128 data points for both the source and target measures, respectively. Then we

compare the performance of the HOT algorithm and the neural network dual method with default parameters in the OTT-JAX library[5]. Specifically, we train the network using a batch size of 2048 and utilize the "W2NeuralDual" API without modifying any parameters (e.g., 20000 training epochs, Adam optimizers). Note that the ICNN potential, as the default parameter, is not suitable for training the CheckerBoard dataset (see Fig. 2). We manually modify it to MLP potential as suggested by OTT-JAX documentation[6]. Here, we also use the 2D non-uniform bins constructed from the union of the supports in the source and target measure samples. We report the numerical results in Table 4. From the results, we can see that, even though the W2NeuralDual is computationally efficient in the inference stage and is memory efficient, the training phase is computationally expensive. Moreover, the quality of the solution obtained by the HOT algorithm can be much better than the ones obtained from the W2NeuralDual. We further visualize the transport plans obtained by the HOT algorithm and the W2NeuralDual in Fig. 3.



(a) Circle Gaussian.



(b) CheckerBoard.

Fig. 1: A visualization of the sample dataset.

Appendix F
Domain Adaptation

Optimal transport has a wide range of applications in domain adaptation and data alignment. Inspired by [10], we apply the HOT algorithm for domain adaptation in unsupervised classification. We first sample test data from the target domain and perform optimal transport between the source domain

[5]https://ott-jax.readthedocs.io/en/latest/neural/_autosummary/ott.neural.methods.neuraldual.W2NeuralDual.html

[6]https://ott-jax.readthedocs.io/en/latest/tutorials/neural/000_neural_dual.html

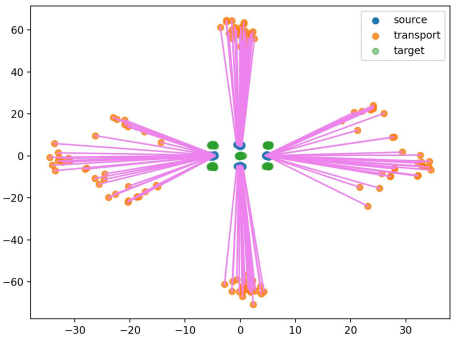
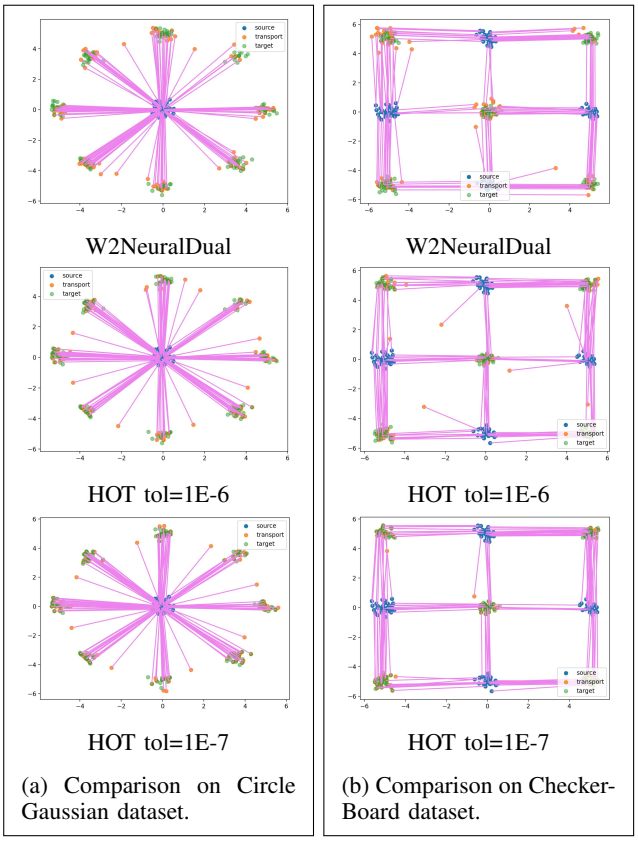


Fig. 2: W2NeuralDual on CheckerBoard Using ICNN Potential with default parameter settings.



(a) Comparison on Circle Gaussian dataset.

(b) Comparison on CheckerBoard dataset.

Fig. 3: A visualization of the transport plans on different datasets.

Dataset		HOT		W2NeuralDual	
		tol=1E-6	tol=1E-7	training	inference
Circle Gaussian	objective	22.3634	22.3601	\	22.2784
	gap	1.54E-4	1.32E-5	\	3.49E-3
	time(s)	51.73	262.78	4466.61	5.21
	memory(GB)	2.72	2.72	0.65	\
CheckerBoard	objective	21.5437	21.5348	\	23.6818
	gap	4.09E-4	1.46E-5	\	9.53E-2
	time(s)	62.51	272.29	3029.24	3.58
	memory(GB)	2.72	2.72	0.65	\

TABLE 4: Numerical results of HOT and W2NeuralDual on the sample dataset.

and the sampled test data. We then train a classifier on the transported data, which retains the source domain labels, and evaluate it on the target domain. Again, we apply the non-uniform 2D grids construction method here.

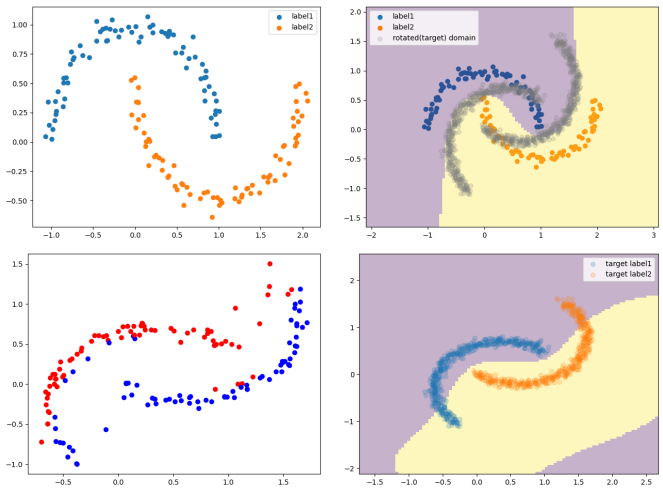


Fig. 4: Optimal transport based domain adaptation for unsupervised classification.

From Table 5, we can conclude that optimal transport-based domain adaptation significantly increases classification accuracy.

TABLE 5: Influence of optimal transport based domain adaptation on unsupervised classification accuracy.

Rotation Angle	10°	20°	40°	50°	70°	90°
SVM	100%	100%	76.5%	40.4%	26.2%	18.7%
OTDA-SVM	100%	100%	92.6%	82.8%	68.4%	59.0%

APPENDIX G
MORE EXAMPLES OF COLOR TRANSFER

We present additional examples of color transfer by the HOT algorithm, as shown in Fig. 5.

REFERENCES

[1] D. F. Sun, Y. Yuan, G. Zhang, and X. Zhao, "Accelerating preconditioned ADMM via degenerate proximal point mappings," *arXiv preprint arXiv:2403.18618*, 2024, SIAM Journal on Optimization 35 (2025) XXX, in print.

[2] M. Cuturi, L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul, "Optimal transport tools (OTT): A JAX Toolbox for all things Wasserstein," *arXiv preprint arXiv:2201.12324*, 2022.

[3] K. Bredies, E. Chenchene, D. A. Lorenz, and E. Naldi, "Degenerate preconditioned proximal point algorithms," *SIAM Journal on Optimization*, vol. 32, no. 3, pp. 2376–2401, 2022.

[4] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[5] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer, New York, 1998.

[6] 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*, vol. 204, no. 1, p. 9, 2025.

[7] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces,2nd edn.* Springer, New York, 2017.

[8] R. Wittmann, "Approximation of fixed points of nonexpansive mappings," *Archiv der Mathematik*, vol. 58, pp. 486–491, 1992.

[9] F. Lieder, "On the convergence rate of the Halpern-iteration," *Optimization Letters*, vol. 15, no. 2, pp. 405–418, 2021.

[10] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.

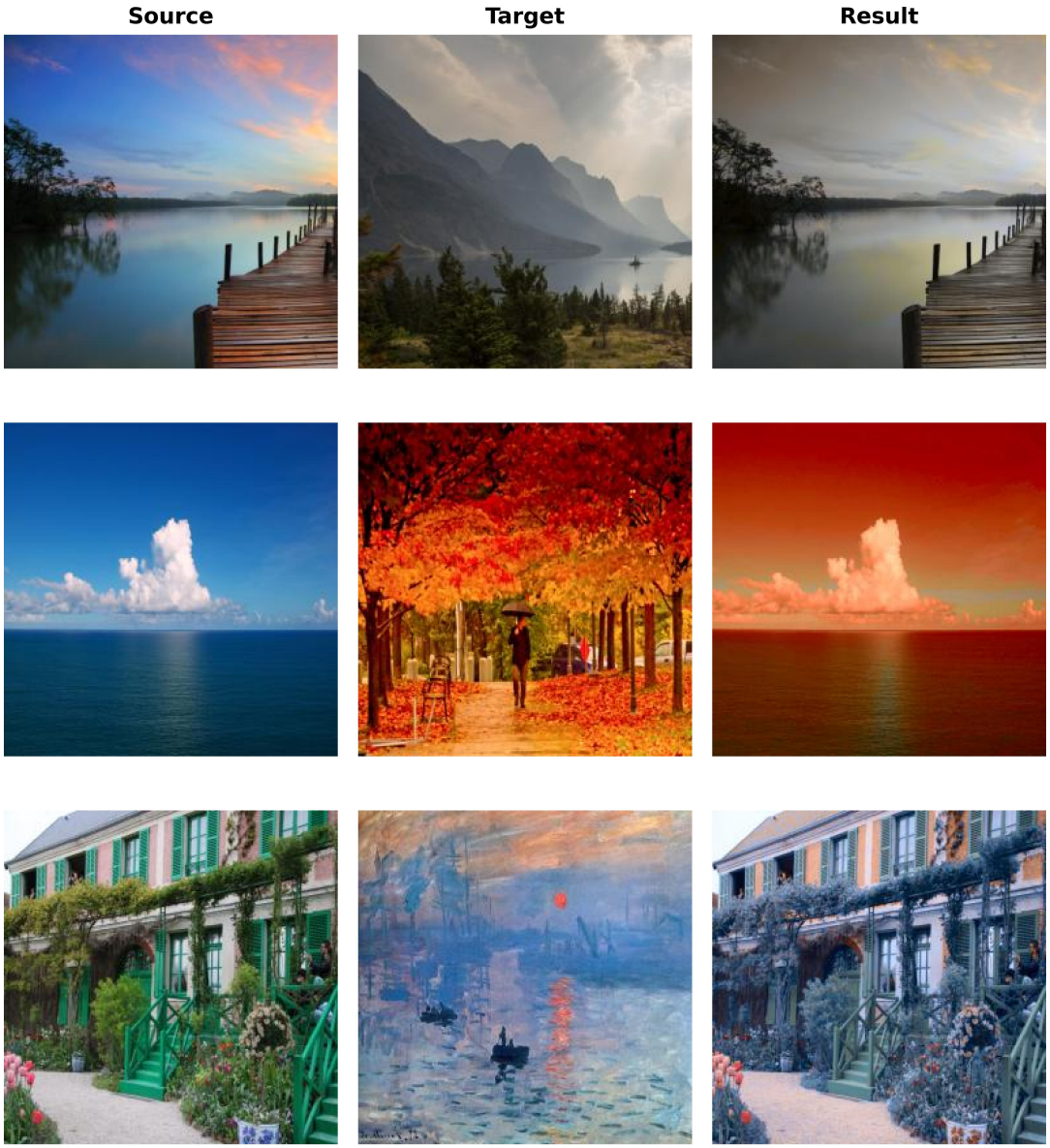


Fig. 5: More examples of color transfer.