HOT: An Efficient Halpern Accelerating Algorithm for Optimal Transport Problems[1]

Yancheng Yuan

Department of Applied Mathematics
The Hong Kong Polytechnic University

May 2025, Shanghai

Based on the joint work with Guojun Zhang, Zhexuan Gu, and Defeng Sun

[1] Zhang, G., Gu, Z., Yuan, Y., & Sun, D. (2025). HOT: An efficient Halpern accelerating algorithm for optimal transport problems. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Overview

Optimal Transport

Probability distribution:

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

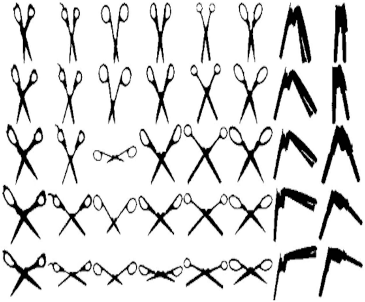
with support point $\boldsymbol{q}$ and associated probability μ satisfying $\sum_{i=1}^{M} \mu_i = 1$.

The Optimal Transport (OT) problem between $\mathcal{P}^1$ and $\mathcal{P}^2$:

$$\begin{aligned} \min_{\pi} \quad & \langle c, \pi \rangle \\ \text{s.t.} \quad & \pi^\top \mathbf{1}_M = \mu^1, \\ & \pi \mathbf{1}_M = \mu^2, \\ & \pi \geq 0. \end{aligned} \tag{1}$$

Applications of Optimal Transport

(a) Color transfer[2].



(b) Shape matching[3].

[2] F. Pitie and A. Kokaram, "The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer." 4th European Conference on Visual Media Production, London, 2007, pp. 1-9, doi: 10.1049/cp:20070055.

[3] Ling, H., & Okada, K. (2007). "An efficient earth mover's distance algorithm for robust histogram comparison." IEEE transactions on pattern analysis and machine intelligence, 29(5), 840-853.

Algorithmic challenges in solving the OT problem

1. **Entropy-Regularized approach**:

$$\text{Objective: } \langle c, \pi \rangle - \gamma H(\pi) \tag{2}$$

with the entropy $H(\pi) = -\sum_{i,j}^{M} \pi_{ij} \log \pi_{ij}$.

- **Method**:
    - Sinkhorn algorithm[4]
- **Advantages**: Low per-iteration cost, easy to implement.
- **Challenges:** Small γ for high-accuracy results in numerical instability and slow convergence.

2. **Linear Programming (LP) Approach**:
- **Methods**:
    - Interior Point Method[5]: Robust, high computational complexity.
    - Network Simplex Method[6]: Robust, not efficiently parallelizable.

[4] Cuturi, Marco. "Sinkhorn distances: Lightspeed computation of optimal transport." Advances in neural information processing systems 26 (2013).

[5] Pele, O., & Werman, M. (2009, September). "Fast and robust earth mover's distances." In 2009 IEEE 12th International Conference on Computer Vision (pp. 460-467). IEEE.

[6] Goldberg, A. V., Tardos, É., & Tarjan, R. (1989). "Network flow algorithm." Cornell University Operations Research and Industrial Engineering.

A reduced OT model I

Dimensions of variables in OT model: $O(M^2)$.
e.g., 256×256 gray-scale image, Double type, 32GB!

Challenges: high memory and computational cost!

Considering the ground distance $c_{i,j;k,l}$ for supports in R^2:

$$c_{i,j;k,l} = \|(i,j)^\top - (k,l)^\top\|_p^p = (|i-k|^p + |j-l|^p). \tag{3}$$

For both the L_1^1 distance[7] and the L_2^2 distance[8], the following property holds:

$$c_{i,j;k,l} = c_{i,j;k,j} + c_{k,j;k,l}. \tag{4}$$

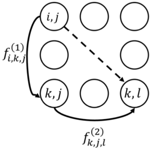


Figure: A nice property of L_1^1 and L_2^2 distance.

[7] Ling, Haibin, and Kazunori Okada. "An efficient earth mover's distance algorithm for robust histogram comparison." IEEE Transactions on Pattern Analysis and Machine Intelligence 29.5 (2007): 840-853.

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

A reduced OT model II

Auricchio et al. proposed an equivalent reduced OT model[9] under L_2^2 distance:

$$
\begin{aligned}
\min_{f^{(1)},f^{(2)}} \quad & \sum_{(i,j)\in\mathcal{I}} \left[\sum_{k=1}^{m} c^{(1)}_{i,k,j} f^{(1)}_{i,k,j} + \sum_{l=1}^{n} c^{(2)}_{k,j,l} f^{(2)}_{k,j,l} \right] \\
\text{s.t.} \quad & \sum_{i=1}^{m} f^{(1)}_{i,k,j} = \sum_{l=1}^{n} f^{(2)}_{k,j,l}, \qquad \forall (k,j)\in\mathcal{I}, \\
& \sum_{k=1}^{m} f^{(1)}_{i,k,j} = \mu^1_{i,j}, \qquad \forall (i,j)\in\mathcal{I}, \\
& \sum_{j=1}^{n} f^{(2)}_{k,j,l} = \mu^2_{k,l}, \qquad \forall (k,l)\in\mathcal{I}, \\
& f^{(1)}_{i,k,j} \geq 0,\ f^{(2)}_{k,j,l} \geq 0, \qquad \forall (i,j),(k,l)\in\mathcal{I},
\end{aligned}
\tag{5}
$$

where $\mathcal{I} = \{(i,j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$,
$c^{(1)}_{i,k,j} := (k-i)^2,\ k = 1,\ldots,m, \forall (i,j)\in\mathcal{I}$, and $c^{(2)}_{k,j,l} := (j-l)^2,\ j = 1,\ldots,n, \forall (k,l)\in\mathcal{I}$.
Dimension of variables: from $(mn)^2$ to $(mn^2 + m^2n)$.

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

A standard LP form of the reduced model

The reduced model can be further written as:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^N} & \langle c, x \rangle + \delta_{\mathbb{R}^N_+}(x) \\ \text{s.t.} & Ax = b, \end{array} \tag{6}$$

where

1. $M_3 = 3M - 1$, $N = m^2 n + mn^2$;
2. $x = [f^{(1)}; f^{(2)}] \in \mathbb{R}^{m^2 n} \times \mathbb{R}^{mn^2}$;
3. $c = [c^1; c^2] \in \mathbb{R}^{m^2 n} \times \mathbb{R}^{mn^2}$;
4. $b = [0_M; \mu^1; \bar{I}_M \mu^2] \in \mathbb{R}^{M_3}$ with $\bar{I}_m = [I_{m-1}, \mathbf{0}_{m-1}] \in \mathbb{R}^{(m-1)\times m}$;
5. $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & \mathbf{0} \\ \mathbf{0} & A_4 \end{bmatrix} \in \mathbb{R}^{M_3 \times N}$ has full row rank with

$$A_1 = I_M \otimes \mathbf{1}_m^\top \in \mathbb{R}^{M \times m^2 n},\ A_2 = -\mathbf{1}_n^\top \otimes I_M \in \mathbb{R}^{M \times mn^2}, A_3 = I_n \otimes (\mathbf{1}_m^\top \otimes I_m) \in \mathbb{R}^{M \times m^2 n},$$

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

Reconstruct the transport plan

Algorithm 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 (5).
Output: An optimal transport mapping π of problem (1).
for $(k, j) \in \mathcal{I}$ **do**
    for $i = 1, \ldots, m$ **do**
        for $l = 1, \ldots, n$ **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

A Halpern accelerating method for solving the OT problem

The dual problem of (6):

$$\min_{y\in\mathbb{R}^{M_3}, z\in\mathbb{R}^N} \left\{ -\langle b, y\rangle + \delta_{\mathbb{R}^N_+}(z) \mid A^\top y + z = c \right\}. \tag{7}$$

The augmented Lagrange function to (7):

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

A fast Halpern accelerating method[10] for solving dual problem (7):

Algorithm HOT: A Halpern accelerating method for solving the reduced OT problem.

1: Input: Choose $w^0 = (y^0, z^0, x^0) \in \mathbb{R}^{M_3} \times \mathbb{R}^N \times \mathbb{R}^N$ and $\sigma > 0$. For $k = 0, 1, \ldots,$ perform the following steps in each iteration.

2: Step 1. $\bar{y}^k = \arg\min_{y\in\mathbb{Y}} \left\{ L_\sigma\left(y, z^k; x^k\right)\right\}$.

3: Step 2. $\bar{x}^k = x^k + \sigma(A^\top \bar{y}^k + z^k - c)$.

4: Step 3. $\bar{z}^k = \arg\min_{z\in\mathbb{Z}} \left\{ L_\sigma\left(\bar{y}^k, z; \bar{x}^k\right)\right\}$.

5: Step 4. $w^{k+1} = \frac{1}{k+2}w^0 + \frac{k+1}{k+2}(2\bar{w}^k - w^k)$. [Halpern's iteration with stepsize $\frac{1}{k+2}$]

[10] Sun, D., Yuan, Y., Zhang, G., & Zhao, X. (2024). "Accelerating preconditioned ADMM via degenerate proximal point mappings." arXiv preprint arXiv:2403.18618, SIAM J. Optim. 35 (2025) XXX, in print.

The iteration complexity of HOT algorithm

Proposition 1 ([SYZZ24])

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 (7) and x^* is a solution to problem (6).

The Karush-Kuhn-Tucker (KKT) residual mapping:

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

Proposition 2 ([SYZZ24])

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:

$$\|\mathcal{R}(\bar{w}^k)\| \leq \left(\frac{\sigma + 1}{\sigma}\right) \frac{R_0}{(k+1)}. \tag{8}$$

Computational bottleneck of HOT

The major computational bottleneck of HOT is to solve the linear system:

$$AA^\top \bar{y}^k = \frac{b}{\sigma} - A\left(\frac{x^k}{\sigma} + z^k - c\right), \tag{9}$$

where $A \in \mathbb{R}^{M_3 \times N}$.

The sparse Cholesky decomposition for large-scale reduced OT problems encounters memory and computational efficiency challenges in general (for 256×256 gray-scale image, $M_3 = 196,607$).

We propose a **linear time complexity** procedure to solve the linear system (9).

The structure of $AA^\top$

The matrix $AA^\top$ has the following structure:

$$AA^\top = \begin{bmatrix} E_1 & E_2 & E_3 \\ E_2^\top & E_4 & \mathbf{0} \\ E_3^\top & \mathbf{0} & E_5 \end{bmatrix}, \tag{10}$$

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. $E_3 = -\mathbf{1}_n \otimes \left(I_m, \ldots, I_m, \bar{I}_m^\top\right) \in \mathbb{R}^{M\times(M-1)}$;
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

$$AA^\top y = \begin{bmatrix} E_1 & E_2 & E_3 \\ E_2^\top & E_4 & \mathbf{0} \\ E_3^\top & \mathbf{0} & E_5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}, \tag{11}$$

where $y := (y_1; y_2; y_3) \in \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R}^{M-1}$ and $R := (R_1; R_2; R_3) \in \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R}^{M-1}$.

A linear time complexity procedure for solving $AA^\top y = R$

Proposition 3

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

$$y_2^j = \frac{1}{m}(R_2^j - \mathbf{1}_m^\top y_1^j), \quad j = 1, \ldots, n, \tag{12}$$

$$y_3^j = \frac{1}{n}(R_3^j + \sum_{j=1}^n y_1^j), \quad j = 1, \ldots, n-1, \tag{13}$$

$$y_3^n = \frac{1}{n}(R_3^n + \bar{I}_m \sum_{j=1}^n y_1^j), \tag{14}$$

$$y_1^j = \hat{y}_1^j - \hat{y}_1^a, \quad j = 1, \ldots, n, \tag{15}$$

where

1. $\hat{y}_1^j = \frac{1}{m+n}\left(\tilde{R}_1^j + \tilde{R}_2^j + \tilde{R}_3\right), \quad j = 1, \ldots, n,$ *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* $w = \frac{1}{m} - \frac{1}{(m+1)}\left(1 - \frac{1}{n}\right)$.

The complexity of HOT for solving the OT problem

Corollary 1

The linear system (9) *can be solved in* $O(M_3)$ *flops.*

Theorem 2

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 (6) *such that the KKT residual* $\left\|\mathcal{R}\left(\bar{w}^k\right)\right\| \leq \varepsilon$*, 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*

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

The explicit solution of the linear system for the original OT problem

The structure of $AA^\top$ in the original OT problem:

$$AA^\top = \begin{bmatrix} MI_M & \mathbf{1}_M \mathbf{1}_{M-1}^\top \\ \mathbf{1}_{M-1}\mathbf{1}_M^\top & MI_{M-1} \end{bmatrix}. \tag{16}$$

The solution of the linear system for the original OT problem[11, 12]

$$AA^\top \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

can be obtained by

$$\begin{aligned} y_1 &= \frac{R_1}{M} + \frac{1}{M}\left(\frac{M-1}{M}\mathbf{1}_M^\top R_1 - \mathbf{1}_{M-1}^\top R_2\right)\mathbf{1}_M, \\ y_2 &= \frac{R_2}{M} + \frac{1}{M}\left(\mathbf{1}_{M-1}^\top R_2 - \mathbf{1}_M^\top R_1\right)\mathbf{1}_{M-1}. \end{aligned} \tag{17}$$

[11] Zhang, G., Yuan, Y., & Sun, D. (2022). An Efficient HPR Algorithm for the Wasserstein Barycenter Problem with $O(Dim(P)/\varepsilon)$ Computational Complexity. arXiv preprint arXiv:2211.14881.

[12] An explicit projection onto the affine constraint Ax = b can be found in Romero, D. (1990). Easy transportation-like problems on K-dimensional arrays. Journal of Optimization Theory and Applications, 66(1), 137-147.

Complexity bounds of different algorithms for the OT problem

Table: 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	Complexity result
Sinkhorn [DGK18]	$\widetilde{O}(M^2C^2/\varepsilon^2)$
APDAGD [DGK18, LHJ22]	$\widetilde{O}(M^{2.5}C/\varepsilon)$
Greenkhorn [LHJ22]	$\widetilde{O}(M^2C^2/\varepsilon^2)$
Accelerated Sinkhorn [LHJ22]	$\widetilde{O}(M^{7/3}C^{4/3}/\varepsilon^{4/3})$
AAM [GDTG21]	$\widetilde{O}(M^{2.5}C/\varepsilon)$
Dual extrapolation [JST19]	$\widetilde{O}(M^2C/\varepsilon)$
HPD [CC22]	$\widetilde{O}(M^{2.5}C/\varepsilon)$
HPR [ZYS22]	$O(M^2R/\varepsilon)$
HOT (Ours)	$\mathbf{O(M^{1.5}R/\varepsilon)}$

Data and baselines

Dataset: DOTmark[13].

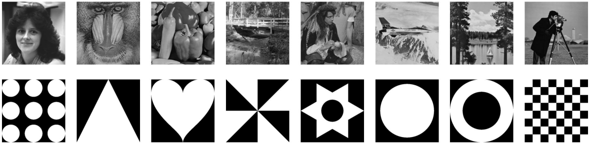


Figure: Upper row: Classic Images category, bottom row: Shapes category.

Baselines:

1. Interior Point Method in Gurobi [Gur24];
2. Network Simplex in Lemon C++ library[14];
3. ADMM;
4. Sinkhorn in POT library[15];
5. Improved Sinkhorn (also explores the property of L_2^2 ground distance).

[13] Schrieber, J., Schuhmacher, D., & Gottschlich, C. (2016). "Dotmark–a benchmark for discrete optimal transport." IEEE Access, 5, 271-282.

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

[15] Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., ... & Vayer, T. (2021). "Pot: Python optimal transport." Journal of Machine Learning Research, 22(78), 1-8.

Implementation details

Environment:

1. Ubuntu server equipped with Intel(R) Xeon(R) Platinum 8480C processor;
2. Nvidia GeForce RTX 4090 GPU (24GB).

Stopping criterion:

1. HOT & ADMM:

$$\mathrm{KKT}_{\mathrm{res}} = \max\left\{ \frac{\|A^\top y + z - c\|}{1 + \|c\|}, \frac{\|\min(x, z)\|}{1 + \|x\| + \|z\|}, \frac{\|Ax - b\|}{1 + \|b\|} \right\} \leq 10^{-6}; \tag{18}$$

2. Other baselines: Default stopping criterion.

Evaluations of solution quality:

1. relative primal feasibility error: $\mathrm{feaserr} = \max\left\{ \frac{\|\min(x,\mathbf{0})\|}{1+\|x\|}, \frac{\|Ax-b\|}{1+\|b\|} \right\}$;
2. relative objective gap: $\mathrm{gap} = \frac{|\langle c,x\rangle - \langle c,x_b\rangle|}{|\langle c,x_b\rangle|+1}$, where x_b is the solution obtained using Gurobi with the tolerance set to 10^{-8}.

Numerical results

Table: Numerical results on Shapes category.

Category	Resolution		**HOT (Reduced)**	HOT (Original)	Sinkhorn (0.01%)
Shapes	64×64	time (s)	**0.64**	28.43	103.74
		gap	3.78E-04	8.14E-04	6.07E-05
		feaserr	5.77E-07	9.73E-07	9.68E-07
		iter	1610	3080	37077
	128×128	time (s)	**1.68**	286.20	1616.34
		gap	2.51E-03	2.60E-03	3.09E-04
		feaserr	1.01E-06	9.58E-07	9.83E-07
		iter	1240	3220	35009

Numerical results

Table: Numerical results on Classic Images category.

Category	Resolution		**HOT (reduced)**	Network Simplex	Gurobi	ADMM	Improved Sinkhorn (0.01%)	Sinkhorn (0.01%)
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		

Table: Numerical results on Shapes category.

Category	Resolution		**HOT (reduced)**	Network Simplex	Gurobi	ADMM	Improved Sinkhorn (0.01%)	Sinkhorn (0.01%)
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		

128×128 case:

$$\text{HOT VS} \begin{cases} \text{Gurobi, } 15.83x \text{ faster,} \\ \text{Network Simplex, } 17.44x \text{ faster,} \\ \text{Improved Sinkhorn, } 19.54x \text{ faster.} \end{cases}$$

Numerical results

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

Category	Resolution		**GPU**	CPU	Ratio ($t_{\text{CPU}}/t_{\text{GPU}}$)	Best Baseline
Classic	128×128	time (s)	**1.58**	5.91	3.74	29.15
	256×256	time (s)	**12.98**	108.98	8.40	2562.92
	512×512	time (s)	**81.02**	764.22	9.43	\
Shapes	128×128	time (s)	**1.68**	6.15	3.66	20.70
	256×256	time (s)	**14.87**	130.81	8.80	959.77
	512×512	time (s)	**87.12**	812.82	9.33	\

Findings:

1. Acceleration ratio gets larger as the dimension of the problem increases.
2. HOT can outperform the baseline methods (excluding ADMM) even without GPU acceleration.

A comparison of sparse Cholesky decomposition and Proposition 3

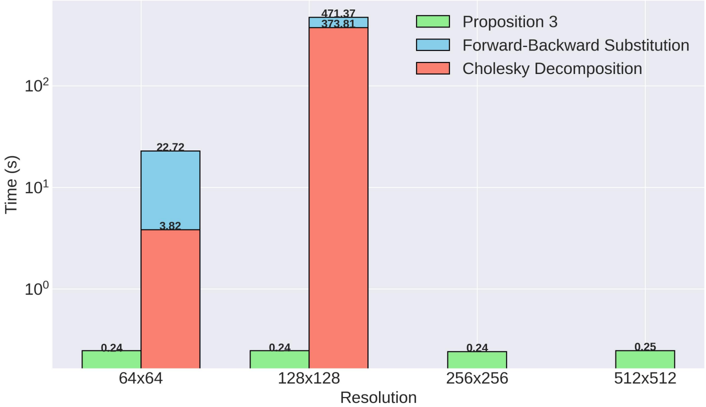


Figure: Comparison of solving the linear system (9) using Proposition 3 and the sparse Cholesky decomposition[16].

[16] https://github.com/rgl-epfl/cholespy

Color transfer

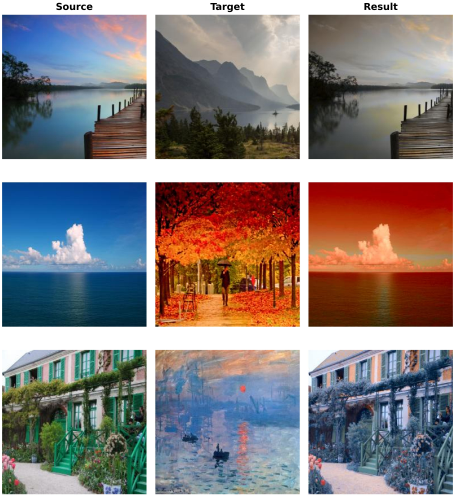


Figure: More examples on color transfer.

Conclusion

1. We proposed an efficient HOT algorithm for solving the OT problem.
2. We designed a linear time complexity procedure to solve the linear system involved in the HOT algorithm.
3. We designed an efficient algorithm to recover an optimal transport map from a solution to the reduced OT model.
4. Extensive numerical results demonstrated the superiority of the HOT algorithm.

An implementation of HPR for solving general large-scale LP problems can be found in

Chen, Kaihuang, Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. "HPR-LP: An implementation of an HPR method for solving linear programming." arXiv preprint arXiv:2408.12179 (2024).

Thanks for listening!

References I

[CC22] Antonin Chambolle and Juan Pablo Contreras.
Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein barycenter problems.
arXiv preprint arXiv:2203.00802, 2022.

[DGK18] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin.
Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm.
In *International conference on machine learning*, pages 1367–1376. PMLR, 2018.

[GDTG21] Sergey Guminov, Pavel Dvurechensky, Nazarii Tupitsa, and Alexander Gasnikov.
On a combination of alternating minimization and Nesterov's momentum.
In *International Conference on Machine Learning*, pages 3886–3898. PMLR, 2021.

[Gur24] Gurobi Optimization, LLC.
Gurobi Optimizer Reference Manual, 2024.

References II

[JST19] Arun Jambulapati, Aaron Sidford, and Kevin Tian.
A direct $\tilde{O}(1/\epsilon)$ iteration parallel algorithm for optimal transport.
Advances in Neural Information Processing Systems, 32, 2019.

[LHJ22] Tianyi Lin, Nhat Ho, and Michael I Jordan.
On the efficiency of entropic regularized algorithms for optimal transport.
Journal of Machine Learning Research, 23(137):1–42, 2022.

[SYZZ24] Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao.
Accelerating preconditioned ADMM via degenerate proximal point mappings.
arXiv preprint arXiv:2403.18618, 2024.
SIAM J. Optim. 35 (2025) XXX, in print.

[ZYS22] Guojun Zhang, Yancheng Yuan, and Defeng Sun.
An efficient HPR algorithm for the Wasserstein barycenter problem with $O(Dim(P)/\varepsilon)$ computational complexity.
arXiv preprint arXiv:2211.14881, 2022.