Inexact Bregman Sparse Newton Method for Efficient Optimal Transport

Jianting Pan[*], **Ji'an Li**[*], **Sirong Dai**[*], **Ming Yan**[†]
School of Data Science, The Chinese University of Hong Kong, Shenzhen
jiantingpan,jianli1,sirongdai@link.cuhk.edu.cn, yanming@cuhk.edu.cn

Abstract

Computing exact optimal transport (OT) distances at large scale remains computationally challenging. Entropy-regularized OT can be solved efficiently, but its solution is biased toward the regularized problem, and small regularization parameters often lead to slow convergence and numerical instability. In this paper, we propose *Inexact Bregman Sparse Newton* (IBSN), an efficient method for high-accuracy computation of the original OT problem. Building on a verifiable inexact Bregman proximal framework, IBSN develops a sparse Newton inner solver tailored to the semi-dual formulation, which yields a lower-dimensional Newton system. To accelerate the inner subproblem solves, we introduce a Hessian sparsification strategy tailored to the semi-dual formulation, which reduces the cost of Newton direction computation while preserving key spectral properties. We establish convergence guarantees for the resulting algorithm, including convergence of the inner sparse Newton iterations and global convergence of the overall method to an optimal OT solution. Numerical experiments on synthetic and real datasets show that IBSN is effective in high-accuracy regimes and compares favorably with representative first-order and second-order OT solvers. The implementation and code are at `https://github.com/PanT12/IBSN-code`.

1 Introduction

Optimal transport (OT) has emerged as a fundamental tool for measuring the difference between probability distributions. Unlike standard metrics, OT offers a geometrically meaningful distance. This unique property has led to widespread applications across machine learning [25, 33, 31], computer vision [10, 40, 46], and statistics [24, 26].

The discrete OT problem can be characterized by the following linear programming problem:

$$\min_{X\in\Omega} \langle C, X\rangle \quad \text{where } \Omega := \{X \in \mathbb{R}^{m\times n} \mid X\mathbf{1}_n = \boldsymbol{a}, X^\top \mathbf{1}_m = \boldsymbol{b}, X \geq 0\}, \tag{1}$$

where $C \in \mathbb{R}_+^{m\times n}$ is the cost matrix, $\boldsymbol{a} \in \mathbb{R}_{++}^m$ and $\boldsymbol{b} \in \mathbb{R}_{++}^n$ are two vectors satisfying $\mathbf{1}_m^\top \boldsymbol{a} = \mathbf{1}_n^\top \boldsymbol{b} = 1$; $\mathbf{1}_m$ and $\mathbf{1}_n$ are the vectors of all ones in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. While this formulation is linear, solving large-scale problems is computationally challenging.

To address this, an entropic regularization term $\phi(X) = \sum_{i=1}^m \sum_{j=1}^n X_{ij} \log X_{ij}$ was added to the objective, which is known as entropy-regularized optimal transport (EOT) [16]:

$$\min_{X\in\Omega} \langle C, X\rangle + \eta\phi(X), \tag{2}$$

where $\eta > 0$ is a regularization parameter. This modification yields a η-strongly convex problem on Ω that can be solved efficiently via the Sinkhorn algorithm [16]. Despite its simplicity and scalability, it

[*]These authors contributed equally to this work.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

is a first-order method with sublinear convergence, meaning it often requires many iterations to reach high accuracy [49]. Recently, to accelerate convergence, researchers have turned to second-order methods such as the Newton method [48, 49, 47]. By combining Newton updates with sparse Hessian approximations and efficient linear solvers, these methods achieve fast local convergence and handle large datasets much better than standard Sinkhorn. However, the EOT solution remains only an approximation of the original OT problem. While reducing the regularization parameter improves approximation accuracy, excessively small regularization values cause the algorithm to slow down and lead to severe numerical instability, such as overflow and underflow.

Instead of settling for the approximate solution provided by EOT, recent research has shifted focus back to solving the original OT problem exactly. This is typically achieved by viewing EOT (2) as a subproblem within a larger iterative framework, such as the Bregman proximal point algorithm. A natural approach is to solve these subproblems exactly until convergence. [50] accelerated this process by applying Newton methods with sparsification techniques. However, requiring an exact solution at every step created a heavy computational burden. Inexact frameworks reduce this burden by allowing approximate subproblem solutions [52, 21, 45], but their stopping criteria can be difficult to verify or justify theoretically. The recent framework of [53] addresses this issue with an easily verifiable stopping condition. However, when implemented with first-order inner solvers, it may still become slow as the required inner accuracy tightens. This motivates our use of an efficient sparse Newton-type solver within the inexact Bregman framework.

In this paper, we propose **I**nexact **B**regman **S**parse **N**ewton (IBSN), an efficient algorithm for solving the original optimal transport problem (1). IBSN adopts the inexact Bregman proximal framework as its outer backbone, but redesigns the inner solver using the semi-dual formulation. The semi-dual formulation itself is standard, while it achieves a lower-dimensional Newton system and yields a different Hessian structure. To exploit this structure, we develop a novel sparsification strategy that preserves the key spectral properties of the semi-dual Hessian. It enables efficient Newton updates while retaining fast local convergence. We establish convergence guarantees for the resulting method and demonstrate its effectiveness on synthetic and real OT benchmarks.

Contribution. The main contributions of this work are summarized as follows:

- We develop a sparse Newton solver for the semi-dual formulation of (2). Unlike existing sparse Newton methods based on the $(m+n)$-dimensional two-dual system, our method operates on an n-dimensional Newton system. We design a sparsification strategy for the semi-dual Hessian and establish its spectral properties and approximation-error bounds.
- We integrate the proposed sparse Newton solver into the inexact Bregman framework and establish convergence guarantees for both inner and outer iterations (proof in Appendix C). Extensive experiments demonstrate its effectiveness against iterative OT solvers.

Notation. For any $\boldsymbol{x} \in \mathbb{R}^n$, we denote its i-th entry by x_i. For an index set $\mathcal{I} \subseteq [n] := \{1, 2, \ldots, n\}$, $\boldsymbol{x}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ represents the subvector of $\boldsymbol{x}$ corresponding to the indices in $\mathcal{I}$, where $|\mathcal{I}|$ denotes the cardinality of $\mathcal{I}$. The span of vector $\boldsymbol{x}$ is defined as $\text{span}\{\boldsymbol{x}\} = \{c\boldsymbol{x} \mid c \in \mathbb{R}\}$. We denote $\mathbf{1}_n$ and $\mathbf{0}_n$ as the n-dimensional vectors of all ones and all zeros, respectively. We use $\lambda^+_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ to represent the smallest positive and largest eigenvalues of real symmetric matrices, respectively. For a matrix $A \in \mathbb{R}^{m\times n}$, let $A_{i\cdot}$ be the vector of the i-th row of A. The kernel of matrix A is defined as $\ker(A) = \{\boldsymbol{x} \mid A\boldsymbol{x} = \mathbf{0}_m\}$. We denote $\|\cdot\|_p$ as the ℓ_p norm of a vector or ℓ_p induced norm of a matrix. For simplicity, we use $\|\cdot\|$ to denote the Euclidean norm of a vector or the ℓ_2 induced norm of a matrix.

2 Related Work

Applications of OT. The versatility of OT has led to its adoption in a broad range of scientific and engineering domains. In machine learning, OT distances underpin generative adversarial networks [18, 4], domain adaptation [15, 14], and distributionally robust optimization [34] by offering geometry-aware measures of distributional discrepancy. In computer vision and graphics, OT provides a principled framework for color transfer [37], texture mixing [38], and point cloud registration [43, 36]. These diverse applications highlight the growing importance of efficient OT computation as a core building block for modern data science.

Computational OT. Algorithmic advances in OT have followed two main directions. The first direction leverages entropy regularization for its computational benefits. Initial progress relied on first-order methods, most notably the Sinkhorn algorithm [16] and its variants (e.g., Greenkhorn [1] and stabilized Sinkhorn [41]). For further acceleration, Newton-based methods (as seen in [48, 47, 49, 12, 50]) exploit second-order information and sparsity. The second direction tackles the original OT problem directly, often by embedding the EOT form into an iterative framework. The strategy of employing EOT as a subproblem within a Bregman framework is a natural and common approach. Other direct solvers include Halpern-accelerated approaches (HOT) [54], primal-dual hybrid gradient method (PDHG) [3], Douglas-Rachford splitting algorithm [30], and extragradient method [28].

Bregman proximal point. Bregman divergence has been widely adopted as a proximal regularizer in large-scale optimization, particularly when the decision variables lie in the probability simplex [5, 6, 9, 7, 35]. Compared to Euclidean proximal mappings, Bregman proximal mappings naturally encode the geometry of distributions and yield algorithms with improved stability and interpretability.

Inexact optimization of inner subproblems. A common strategy in large-scale optimization, including OT, is to solve inner subproblems only approximately rather than to full precision [52, 53, 13, 21]. Such inexactness reduces per-iteration cost while still guaranteeing global convergence, provided that the approximation error is controlled relative to the outer iteration.

3 Background

In this section, we review the subproblem formulation and its corresponding semi-dual problem based on Bregman divergence.

To compute the optimal transport plan, we employ a Bregman proximal point framework. In this iterative scheme, each iteration solves a regularized subproblem of the form:

$$X^{k+1} \in \arg\min_{X \in \Omega} \left\{ \langle C, X \rangle + \eta D_\phi(X, X^k) \right\}, \tag{3}$$

where $D_\phi(X, Y) = \sum_{i,j} X_{ij} \log \frac{X_{ij}}{Y_{ij}} - \sum_{i,j} X_{ij} + \sum_{i,j} Y_{ij}$ is the Bregman divergence based on the negative entropy function ϕ. It has been shown that the subproblem (3) can be equivalently stated via its two-dual formulation [48]:

$$\min_{\boldsymbol{\gamma} \in \mathbb{R}^n, \boldsymbol{\zeta} \in \mathbb{R}^m} Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta}) := -\boldsymbol{a}^\top \boldsymbol{\zeta} - \boldsymbol{b}^\top \boldsymbol{\gamma} + \eta \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}^k \exp\{(\zeta_i + \gamma_j - C_{ij})/\eta\}. \tag{4}$$

Under this framework, the Sinkhorn algorithm can be interpreted as a simple alternating minimization algorithm that iteratively minimizes $Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta})$ with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\zeta}$.

To simplify the optimization landscape, especially for our Newton-type method, we work with the semi-dual formulation. This form is derived by eliminating one set of dual variables. Without loss of generality, we proceed to eliminate the dual variable $\boldsymbol{\zeta}$ by letting $\frac{\partial Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = \mathbf{0}_m$, i.e.,

$$\zeta(\boldsymbol{\gamma})_i = \eta \log a_i - \eta \log \Big(\sum_{j=1}^{n} X_{ij}^k \exp\{(\gamma_j - C_{ij})/\eta\} \Big). \tag{5}$$

Substituting this back into the two-dual problem (4), we obtain the semi-dual formulation that minimizes over only $\boldsymbol{\gamma}$:

$$\min_{\boldsymbol{\gamma} \in \mathbb{R}^n} \mathcal{L}^k(\boldsymbol{\gamma}) := -\sum_{j=1}^{n} \gamma_j b_j + \eta \sum_{i=1}^{m} a_i \log \Big(\sum_{j=1}^{n} X_{ij}^k \exp\left((\gamma_j - C_{ij})/\eta\right) \Big). \tag{6}$$

The following lemma guarantees the existence of an optimal solution for this formulation (6).

Lemma 3.1. *The semi-dual problem* (6) *has a solution.*

Let $\boldsymbol{\gamma}^{k\star}$ denote an optimal solution to (6), then the corresponding primal update X^{k+1} can be recovered via $X^{k+1} = \operatorname{diag}(\boldsymbol{a})\, P^k(\boldsymbol{\gamma}^{k\star})$, where $P^k(\boldsymbol{\gamma})$ is a transport probability matrix with entries:

$$\left[P^k(\boldsymbol{\gamma})\right]_{ij} = \frac{X_{ij}^k \exp\left((\gamma_j - C_{ij})/\eta\right)}{\sum_{l=1}^{n} X_{il}^k \exp\left((\gamma_l - C_{il})/\eta\right)}. \tag{7}$$

The partial gradient g^k of $\mathcal{L}^k(\boldsymbol{\gamma})$ with respect to $\boldsymbol{\gamma}$ and its corresponding Hessian H^k are given by

$$g^k(\boldsymbol{\gamma}) = (P^k(\boldsymbol{\gamma}))^\top \boldsymbol{a} - \boldsymbol{b}, \quad H^k(\boldsymbol{\gamma}) = \frac{1}{\eta}\left(\operatorname{diag}\left(P^k(\boldsymbol{\gamma})^\top \boldsymbol{a}\right) - P^k(\boldsymbol{\gamma})^\top \operatorname{diag}(\boldsymbol{a})\, P^k(\boldsymbol{\gamma})\right). \tag{8}$$

This highly smooth and structured formulation provides major advantages for using Newton methods to solve (6). First, the existence and easy computation of the Hessian enable the use of the Newton method. This guarantees a much faster rate of convergence compared to the sublinear convergence typical of first-order algorithms like Sinkhorn. Second, compared to the classical two-dual formulation, this approach reduces the number of dual variables from $(m+n)$ to n. This significantly reduces the memory requirement for the Hessian (from $(m+n)\times(m+n)$ to $n\times n$) and the computational effort required to compute the Newton direction. Third, the Hessian's inherent structure is ideally suited to the sparsification technique we detail in the next subsection. This mathematical property is key to reducing the computational cost without sacrificing accuracy.

4 Methodology

This section details the Inexact Bregman Sparse Newton (IBSN) framework, a Newton-type scheme for solving the original OT problem (1). Section 4.1 describes the Hessian sparsification strategy designed to enhance computational efficiency. In addition, Section 4.2 outlines the complete algorithmic framework, detailing the inexact Newton update mechanism and overall procedure.

4.1 Sparsifying the Hessian matrix

In this subsection, we detail our strategy for accelerating the computation of Newton directions. For brevity, in the current subproblem iteration k, we denote the matrix, gradient, and Hessian by $P = P^k(\boldsymbol{\gamma})$, $\boldsymbol{g} = g^k(\boldsymbol{\gamma})$, and $H = H^k(\boldsymbol{\gamma})$, respectively in this subsection.

The classical Newton method requires solving the linear system $H(\Delta\boldsymbol{\gamma}) = -\boldsymbol{g}$. For large-scale OT problems, this operation can be computationally expensive, particularly when Hessian-vector products involve dense matrices. Our approach exploits the structure of the transport probability matrix. When the regularization parameter η is small, the transport plan and the associated probability matrix often contain many entries that are close to zero. This motivates replacing P with a normalized sparse approximation, so that Hessian-vector products can be computed more efficiently while preserving the dominant curvature information.

Specifically, we construct P_ρ by thresholding small entries of P and then renormalizing each row to preserve the row-stochastic structure. The row indexed by $i^\star = \arg\max_{i\in[m]} a_i$ is kept unchanged, which is essential for maintaining the spectral lower bound used in the subsequent analysis. The full procedure is summarized in Algorithm 1. In practice, the 'sparsified' Hessian H_ρ does not need to be formed explicitly; storing the sparse matrix P_ρ is sufficient for efficient Hessian-vector products.

Algorithm 1 Sparsifying the Hessian matrix

1: **Input:** Outer iteration k, dual variable $\boldsymbol{\gamma}$, threshold $\rho \in [0, 1/n)$, $i^\star = \arg\max_{i\in[m]} a_i$.
2: Compute $P \leftarrow P^k(\boldsymbol{\gamma})$ using (7).
3: Construct the sparsified matrix P_ρ by setting

$$[P_\rho]_{ij} \leftarrow \frac{P_{ij}\mathbf{1}_{\{P_{ij}\ge\rho\}}}{\sum_{l=1}^n P_{il}\mathbf{1}_{\{P_{il}\ge\rho\}}}\ \forall i \neq i^\star, \qquad [P_\rho]_{i^\star\cdot} \leftarrow P_{i^\star\cdot}.$$

4: Compute $H_\rho \leftarrow \left(\operatorname{diag}\left(P_\rho^\top \boldsymbol{a}\right) - P_\rho^\top \operatorname{diag}(\boldsymbol{a})\, P_\rho\right)/\eta$.
5: **Output:** H_ρ.

This construction naturally raises two important questions. First, **is H_ρ invertible on $\mathbf{1}_n^\perp$?** This property is crucial for implementing the Newton step, as computing the direction $\Delta\boldsymbol{\gamma}$ requires solving the linear system $H_\rho(\Delta\boldsymbol{\gamma}) = -\boldsymbol{g}$. Ensuring invertibility guarantees that the system admits a unique solution and that the Newton direction is well defined. Second, **how accurate is H_ρ as an approximation of the true Hessian H?** A quantitative understanding of this discrepancy guides

the choice of the sparsification threshold ρ, balancing computational efficiency with approximation accuracy.

To address these questions, we first establish the structural properties of H_ρ. The following result shows that the sparsified Hessian constructed by Algorithm 1 remains positive definite on the feasible subspace, ensuring that the Newton step is always well defined.

Theorem 4.1. *The matrix H_ρ output by Algorithm 1 is symmetric and positive semidefinite, and its kernel satisfies ker(H_ρ) = span$\{\mathbf{1}_n\}$ for any $\rho \in [0, 1/n)$.*

Theorem 4.1 implies that, for any $\rho \in [0, 1/n)$, H_ρ is positive definite on the orthogonal complement $\mathbf{1}_n^\perp$ of the kernel. Note that the gradient $\boldsymbol{g}$ satisfies $\mathbf{1}_n^\top \boldsymbol{g} = 0$, so the Newton system can be efficiently solved using the conjugate gradient (CG) method. The next result gives spectral bounds for the Hessian on this restricted subspace.

Theorem 4.2. *Let H_ρ be the matrix output by Algorithm 1, then for any $\rho \in [0, 1/n)$,*

$$\lambda_{\min}^{+}(H_\rho) \geq n a_{i^\star} p^2/\eta, \quad \lambda_{\max}(H_\rho) \leq 1/(2\eta),$$

where $p := \min_j [P]_{i^\star j}$.

These bounds ensure that H_ρ is well-conditioned on the feasible subspace, providing stability for the Newton update and reliable convergence behavior across different sparsification levels.

To address the second question regarding accuracy, we establish a quantitative bound on the approximation error between the exact and sparsified Hessians.

Theorem 4.3. *Let H_ρ be the matrix output by Algorithm 1, then $\|H - H_\rho\| \leq \frac{6mn\|\boldsymbol{a}\|_\infty}{\eta}\rho$.*

Theorem 4.3 shows that the approximation error increases with the problem size m, n and decreases with the regularization parameter η. To maintain a small and stable error across different scales, the threshold ρ can be chosen proportional to $\eta/(mn)$. This choice balances sparsity and accuracy while ensuring consistent performance across varying values of m, n, and η.

The preceding analysis confirms that the Hessian sparsification technique constructed via Algorithm 1 yields an approximation H_ρ that is both structurally robust and quantitatively accurate. However, even with these sparsification and second-order acceleration techniques, solving the problem (6) to high precision in every iteration remains computationally prohibitive. To fully address this efficiency bottleneck, we adopt the inexact Bregman framework proposed by [53]. This framework allows the subproblems (3) to be solved approximately while still ensuring global convergence to the optimal OT solution. The details of this inexact criterion and its integration into our method are shown in the next subsection.

4.2 The proposed algorithm

Algorithm 2 summarizes the proposed Inexact Bregman Sparse Newton (IBSN) method for solving the OT problem (1). At each outer iteration, IBSN first applies a few Sinkhorn iterations to the two-dual subproblem (4), producing a warm start for the subsequent Newton refinement in the semi-dual space.

During the inner Newton phase, the gradient $\boldsymbol{g}^{k,v}$ is used to adapt the sparsification threshold $\rho = \eta\|\boldsymbol{g}^{k,v}\|/(mn)$. This adaptive choice satisfies the admissible range $\rho < 1/n$ required by Algorithm 1 under the mild condition $\eta < m/2$. Indeed, since both $(P^k(\boldsymbol{\gamma}))^\top \boldsymbol{a}$ and $\boldsymbol{b}$ are probability vectors, $\|\boldsymbol{g}^{k,v}\| \leq 2$, which gives $\rho \leq \frac{2\eta}{mn} < \frac{1}{n}$. A larger residual yields a sparser Hessian approximation and reduces the cost of early iterations, while ρ decreases as the iterate approaches the subproblem solution, leading to a more accurate curvature approximation. The shifted system $\left(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\right) \Delta\boldsymbol{\gamma}^{k,v} = -\boldsymbol{g}^{k,v}$ is then solved to obtain a stable descent direction, followed by Armijo backtracking.

The inner loop terminates once the inexact Bregman criterion $D_\phi(\mathcal{P}_\Omega(X^{k,v+1}), X^{k,v+1}) \leq \mu_k$ is satisfied. The resulting primal and dual variables are used to initialize the next outer iteration. This criterion follows the inexact framework of [53] and avoids solving every subproblem to high precision. In implementation, the rounding procedure of [1] can be used as an efficient approximation of $\mathcal{P}_\Omega$; see Appendix B. Note that our convergence analysis assumes exact solutions of the Newton systems, whereas the practical implementation uses CG with a prescribed stopping tolerance.

Algorithm 2 Inexact Bregman Sparse Newton (IBSN)

Input: $X^0 = \boldsymbol{a}\boldsymbol{b}^\top$, $\boldsymbol{\gamma}^{0,0}$, $\boldsymbol{\zeta}^{0,0}$, $\{\mu_k\}_k$, $\eta \in (0, m/2)$, Armijo parameter $\sigma = 10^{-4}, \beta = 0.8$.
for $k = 0, ..., K-1$ **do**
  Solve subproblem (4) using the Sinkhorn algorithm with $\boldsymbol{\gamma}^{k,0}, \boldsymbol{\zeta}^{k,0}$ to obtain $\boldsymbol{\gamma}^{k,1}, \boldsymbol{\zeta}^{k,1}$.
  Update $\boldsymbol{\gamma}^{k,1} \leftarrow \boldsymbol{\gamma}^{k,1} - \frac{1}{n}\mathbf{1}_n^\top \boldsymbol{\gamma}^{k,1}\mathbf{1}_n$.
  for $v = 1, 2, ...$ **do**
    Compute the gradient $\boldsymbol{g}^{k,v} = g^k(\boldsymbol{\gamma}^{k,v})$ and update $\rho \leftarrow \frac{\eta}{mn}\left\|\boldsymbol{g}^{k,v}\right\|$.
    Construct $H_\rho^{k,v}$ by Algorithm 1.
    Compute $\Delta\boldsymbol{\gamma}^{k,v} \leftarrow -\left(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\right)^{-1}\boldsymbol{g}^{k,v}$.
    Select the stepsize $t^{k,v}$ by Armijo backtracking.
    Compute $\boldsymbol{\gamma}^{k,v+1} \leftarrow \boldsymbol{\gamma}^{k,v} + t^{k,v}\Delta\boldsymbol{\gamma}^{k,v}$.
    Compute $X^{k,v+1} \leftarrow \operatorname{diag}(\boldsymbol{a})\, P^k(\boldsymbol{\gamma}^{k,v+1})$.
    if $D_\phi(\mathcal{P}_\Omega(X^{k,v+1}), X^{k,v+1}) \le \mu_k$ **then**
      Obtain $X^{k+1} \leftarrow X^{k,v+1}$, $\boldsymbol{\gamma}^{k+1,0} \leftarrow \boldsymbol{\gamma}^{k,v+1}$.
      Calculate $\boldsymbol{\zeta}^{k+1,0}$ by (5) and **break**.
    end if
  end for
end for
Output: X^K.

5 Theoretical Analysis

This section presents the convergence analysis of the proposed IBSN framework. We first establish the inner convergence of the Newton iterations with sparsified Hessians, then prove the global convergence of the outer iterations to the exact optimal transport solution.

5.1 Inner convergence analysis

This subsection establishes the crucial convergence properties of the inner iterations in the IBSN framework, demonstrating that our sparse Newton steps converge to the unique solution of each EOT subproblem. We start with the following lemma, which validates our approach by confirming that the entire optimization sequence remains confined to this desirable subspace.

Lemma 5.1. *Let $\{\boldsymbol{\gamma}^{k,v}\}$ be the sequence generated by Algorithm 2 with $\mathbf{1}_n^\top\boldsymbol{\gamma}^{0,0} = 0$, then $\mathbf{1}_n^\top\boldsymbol{\gamma}^{k,v} = 0$ for $\forall k, v$.*

Lemma 5.1 ensures that the convergence analysis can be safely restricted to the subspace $\mathbf{1}_n^\perp$, effectively eliminating the redundant degree of freedom and restoring the strong convexity of $\mathcal{L}^k$.

We then establish that the inner iterates $\boldsymbol{\gamma}^{k,v}$ generated by Algorithm 2 converge to the unique global minimizer of $\mathcal{L}^k$ within the subspace $\mathbf{1}_n^\perp$.

Theorem 5.2. *For a fixed outer k-th iteration, let $\{\boldsymbol{\gamma}^{k,v}\}_v$ be the sequence generated by Algorithm 2 with $\mathbf{1}_n^\top\boldsymbol{\gamma}^{k,0} = 0$, then we have $\|\boldsymbol{g}^{k,v}\| \to 0$ as $v \to \infty$ and $\boldsymbol{\gamma}^{k,v} \to \boldsymbol{\gamma}^{k\star}$, where $\boldsymbol{\gamma}^{k\star}$ is the unique minimizer of $\mathcal{L}^k$ in the slice $\{\boldsymbol{\gamma} \mid \mathbf{1}_n^\top\boldsymbol{\gamma} = 0\}$.*

Next, we examine the local convergence rate achieved by the sparse Newton iterations.

Theorem 5.3. *For a fixed outer k-th iteration, let $\boldsymbol{\gamma}^{k\star}$ denote the unique solution in the slice $\{\boldsymbol{\gamma} \mid \mathbf{1}_n^\top\boldsymbol{\gamma} = 0\}$ of $\mathcal{L}^k$. When the iterates $\boldsymbol{\gamma}^{k,v}$ are sufficiently close to $\boldsymbol{\gamma}^{k\star}$ such that the full Newton step $t^{k,v} = 1$ is accepted, there exists a constant $L > 0$ such that*

$$\|\boldsymbol{\gamma}^{k,v+1} - \boldsymbol{\gamma}^{k\star}\| \le L\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\|^2.$$

Together, Theorems 5.2 and 5.3 establish that the proposed sparse Newton updates converge globally to the unique solution of each subproblem and exhibit quadratic local convergence once sufficiently close to the optimum. This demonstrates that the sparsified Newton steps preserve the accuracy and convergence speed of the classical method, making this approach feasible for large-scale optimal transport problems.

Complexity Analysis of a Newton Direction. We briefly discuss the cost of computing a Newton direction for the EOT subproblem (2). Assume for simplicity that $m = n$, the matrix $P_\rho \in \mathbb{R}^{n\times n}$ has an average of l nonzero entries per row. Then P_ρ contains $O(ln)$ nonzero entries, and each Hessian-vector product with H_ρ can be computed in $O(ln)$ operations. Consequently, each CG iteration also costs $O(ln)$. When $l \ll n$, as is typically observed after effective sparsification, this is substantially cheaper than the $O(n^2)$ cost associated with a dense representation.

Let ϵ_{cg} be the CG tolerance and κ denote the condition number of $H_\rho + \|\boldsymbol{g}\|I$ restricted to $\mathbf{1}_n^\perp$. The number of CG iterations required is $O(\sqrt{\kappa}\log(1/\epsilon_{\text{cg}}))$ [44, Chapter 10]. Applying the spectral bounds from Theorem 4.2, we can bound this condition number as

$$\kappa \le \lambda_{\max}(H)/\lambda_{\min}(H) \le \frac{1/(2\eta)}{na_{i^\star}p^2/\eta} = O(p^{-2}),$$

where $p = \min_j P_{i^\star j}$ and $a_{i^\star} \ge 1/n$. For the preserved row $i^\star$, let $s_j := \eta \log X^k_{i^\star j} + \gamma_j - C_{i^\star j},$. We can bound p from below:

$$p = \min_j P_{i^\star j} = \min_j \frac{\exp(s_j/\eta)}{\sum_{l=1}^n \exp(s_l/\eta)} \ge \frac{\exp(\min_l s_l/\eta)}{n\exp(\max_l s_l/\eta)} := \frac{1}{n}\exp\left(-\frac{\Delta}{\eta}\right),$$

where $\Delta = \max_l s_l - \min_l s_l$ represents the spread of the exponentiated terms.

Therefore, ignoring the logarithmic dependence on ϵ_{cg}, the total cost of a single CG solve is $O\left(ln\,p^{-1}\right) = O\left(ln^2\exp\left(\Delta/\eta\right)\right)$. This theoretical bound shows that the linear system becomes increasingly expensive to solve as the problem dimension n increases or the regularization parameter η decreases. In particular, a smaller η sharpens the softmax distribution, which can exponentially amplify the factor $\exp(\Delta/\eta)$.

5.2 Global convergence

We now establish the global convergence of the outer IBSN iterations. Let $f(X) = \langle C, X\rangle$, $f^\star = \min_{X\in\Omega} f(X)$, and $\widetilde{X}^k := \mathcal{P}_\Omega(X^k)$. The outer iteration is a specialization of the inexact Bregman proximal point framework in [53]. In our OT formulation, the residual and subgradient-error terms in their framework vanish, so the convergence behavior is determined by the inexactness sequence $\{\mu_k\}$.

If $\sum_{k=0}^\infty \mu_k < \infty$, then $\{X^k\}$ and $\{\widetilde{X}^k\}$ converge to the same optimal solution of problem (1). For the polynomial schedule $\mu_k = O((k+1)^{-p})$, any $p > 1$ therefore guarantees sequence convergence. The last-iterate rate depends on the stronger weighted summability of $\{\mu_k\}$: $p = 2$ yields

$$f(\widetilde{X}^K) - f^\star = O\left(\frac{\eta\log K}{K}\right), \tag{9}$$

whereas any $p > 2$ gives

$$f(\widetilde{X}^K) - f^\star = O\left(\frac{\eta}{K}\right). \tag{10}$$

The detailed specialization and rate analysis are provided in Appendix C.8.

6 Numerical Experiments

In this section, we evaluate the performance of the proposed IBSN algorithm through a series of numerical experiments on both synthetic and real datasets and compare its empirical performance with existing state-of-the-art optimal transport solvers: 1. PINS algorithm [50] (based on its implementation at the time of our submission, prior to its second version); 2. HOT algorithm [54]; 3. Inexact Bregman framework with Sinkhorn (IBSink) [53]; 4. IPOT algorithm [52]; 5. Extragradient (ExtraGrad) method [28]. All numerical experiments are implemented in Python 3.9.18 on a MacBook Pro running macOS 13.0 with an Apple M2 Pro processor and 16 GB RAM.

Setup To ensure consistency, we normalize all cost matrices to have a unit infinity norm, i.e., $C \leftarrow C/\max_{i,j}|C_{ij}|$. We initialize the transport plan as $X^0 = \boldsymbol{a}\boldsymbol{b}^\top$. We use the relative KKT residual [53] (Δ_{kkt}) as the unified stopping criterion for all methods (see Appendix D.1). An algorithm is terminated once $\Delta_{\text{kkt}} < 10^{-11}$ or the maximum number of iterations is attained.

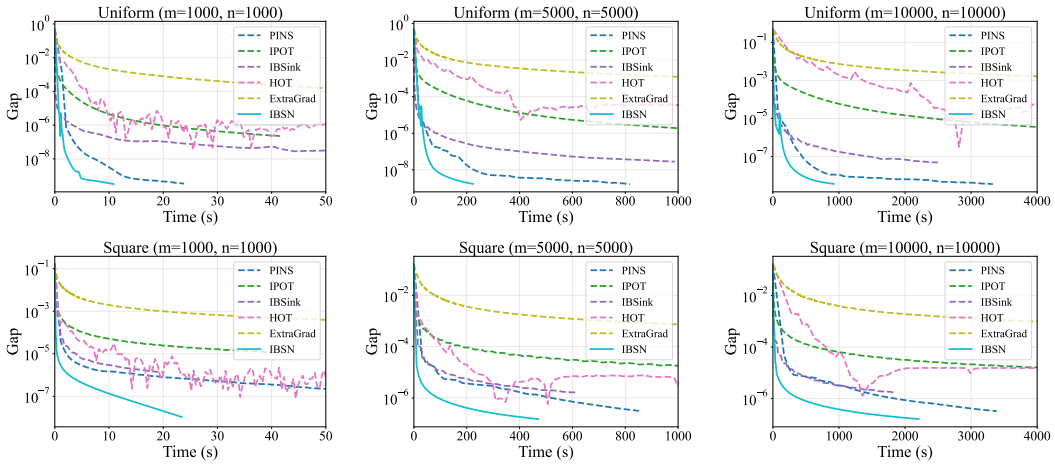

Figure 1: Performance of different algorithms on synthetic data. Top: Uniform. Bottom: Square.

Within the inner iteration of IBSN, the Sinkhorn warm-up phase stops once the gradient norm is less than 10^{-3}. In addition, we terminate the Newton phase when

$$D_\phi(\mathcal{P}_\Omega(X^{k+1}), X^{k+1}) \leq \min\{m,n\} \cdot \mu_k, \quad \text{where } \mu_k = \max\left\{10^{-4}/(k+1)^2, 10^{-11}\right\}, \quad (11)$$

where $\mathcal{P}_\Omega$ is computed using the rounding procedure in [1]. To save computational time, we only evaluate the condition (11) after a simpler gradient norm check ($\left\|\nabla \mathcal{L}^k(\boldsymbol{\gamma}^{k,v+1})\right\| < \mu_k$) passes (as suggested in [53]). To compare the solution quality, we utilize the objective gap relative to a high-precision ground truth $X^\star$ (obtained via Gurobi with tolerance 10^{-8}):

$$\text{Gap} = \left|\langle C, \mathcal{P}_\Omega(X)\rangle - \langle C, X^\star\rangle\right|.$$

6.1 Synthetic data

We generate synthetic datasets using the following procedure. The marginal distributions $\boldsymbol{a}$ and $\boldsymbol{b}$ are sampled independently from a uniform distribution on $(0,1)$ and subsequently normalized to satisfy $\mathbf{1}_m^\top \boldsymbol{a} = \mathbf{1}_n^\top \boldsymbol{b} = 1$. We evaluate performance across problem dimensions $m = n \in \{1000, 5000, 10000\}$ using two types of cost matrices:

- **Uniform**: Each entry of C is drawn independently from the uniform distribution on $(0,1)$.
- **Square**: The cost matrix C is defined by the squared ℓ_2-distance between grid coordinates.

We set $\eta = 10^{-4}$ for both IBSN and PINS, $\eta = 10^{-3}$ for IBSink, $\eta = 10^{-1}$ for IPOT. These values are selected based on empirical tuning and correspond to the best observed trade-off between convergence speed and solution accuracy for each method. A more detailed analysis of the impact of the regularization parameter η is provided in Appendix D.2.

The experimental results are presented in Figure 1. The second-order methods (IBSN and PINS) converge faster than the first-order methods in most cases. This exploits sparse matrix operations to reduce per-iteration cost and achieve fast convergence with second-order methods. In addition, IBSN surpasses PINS in overall efficiency. This advantage stems from our semi-dual formulation, which further reduces the cost of the second-order updates.

6.2 Real data

We further evaluate the proposed IBSN algorithm on real-image OT tasks to examine its effectiveness in practical scenarios. Three benchmark datasets are considered:

- **(Fashion-)MNIST**: We compute the OT map between pairs of images from the MNIST [27] and the Fashion-MNIST [51] datasets. The vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are obtained by flattening and normalizing the pixel intensity values, while the cost matrix C is generated via the squared ℓ_2-distance between pixel indices. The problem size is set to $m = n = 784$.

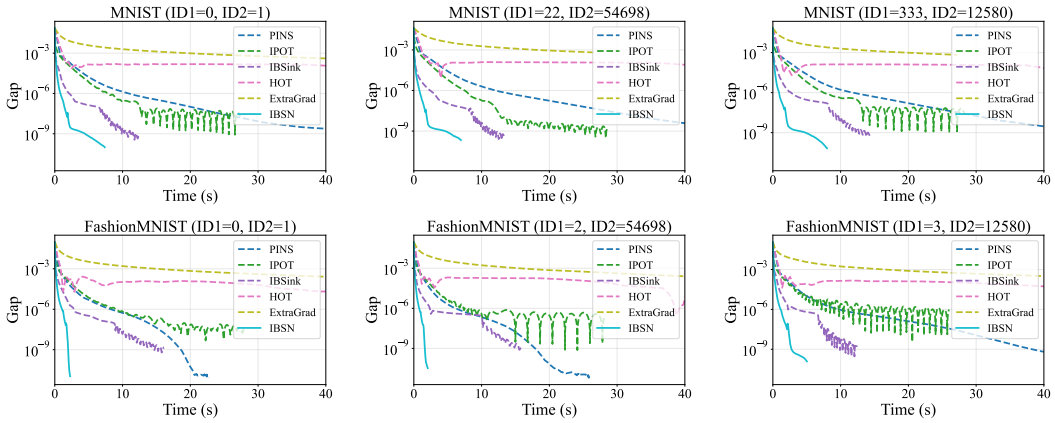

Figure 2: Performance of different algorithms on MNIST (Top) and Fashion-MNIST (Bottom).

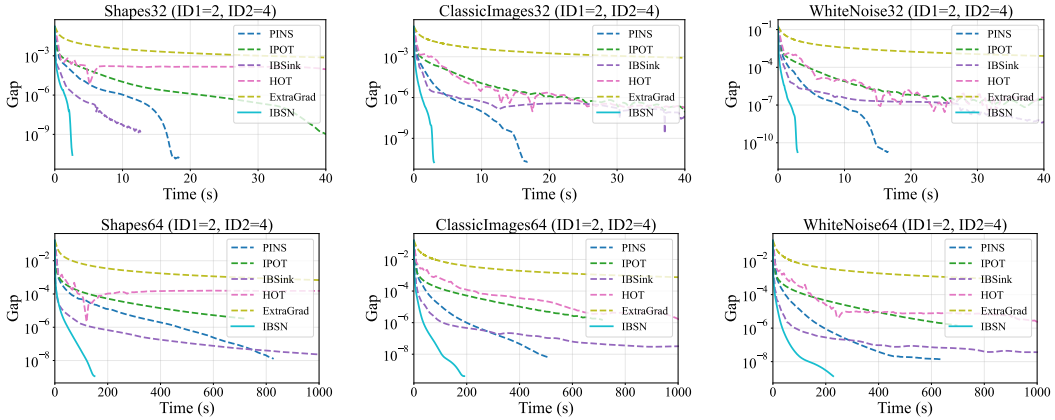

Figure 3: Performance of different algorithms on DOTmark dataset. Top: 32×32. Bottom: 64×64.

- **DOTmark**: The DOTmark dataset [42] provides a comprehensive benchmark suite for evaluating OT algorithms. In our experiments, we select two images each from three categories (`Shapes`, `ClassicImages`, `WhiteNoise`) in the dataset, with image size 32×32 ($m = n = 1024$) and 64×64 ($m = n = 4096$). The cost matrix C and marginal vectors $\boldsymbol{a}, \boldsymbol{b}$ are constructed in the same manner as in the (Fashion-)MNIST experiments.
- **ImageNet**: We further evaluate the algorithms on larger-scale image pairs from the ImageNet dataset [19]. Each image is resized to 100×100, resulting in an OT problem of size $m = n = 10000$. The marginal vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and the cost matrix C are constructed in the same manner as in the (Fashion-)MNIST experiments.

We set $\eta = 10^{-3}$ for IBSN, PINS and IBSink, $\eta = 10^{-1}$ for IPOT. These choices of η are determined in the same way as in Section 6.1.

Figure 2 shows the results on the (Fashion-)MNIST dataset. As observed, first-order methods (HOT and ExtraGrad) remain comparatively slow. Notably, the convergence trajectory of IBSink exhibits three distinct stages, a behavior attributable to the inexactness of the framework. Initially, the algorithm converges rapidly due to the loose inexactness tolerance μ_k. As the tolerance tightens, IBSink enters a second stage in which it requires significantly more iterations to satisfy the inner stopping criterion, leading to a slowdown. Finally, as the transport plan X approaches the optimal $X^\star$, convergence accelerates again. In contrast, our proposed IBSN algorithm overcomes this mid-stage bottleneck by leveraging second-order information. This allows it to maintain rapid convergence even as the tolerance μ_k becomes more stringent. Similar patterns are observed in the DOTmark dataset (Figure 3) and the ImageNet dataset results (Figure 4), where the performance gap between IBSN and competing algorithms is even more pronounced. More experiment results are given in Appendix D.

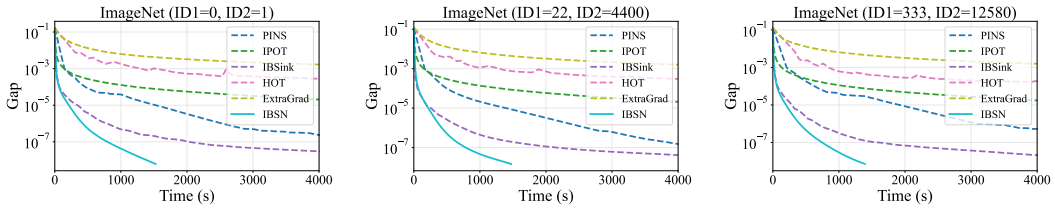

Figure 4: Performance of different algorithms on the ImageNet dataset.

7 Conclusion

In this paper, we present the Inexact Bregman Sparse Newton (IBSN) method for efficiently solving the exact optimal transport problem. Building on the verifiable inexact Bregman proximal framework, we develop a sparse Newton solver tailored to the semi-dual formulation, with a structure-preserving Hessian sparsification strategy that reduces the computational cost of second-order updates. By adaptively coupling the sparsification threshold to the optimality residual, our method improves the Hessian approximation accuracy near the solution while retaining fast local convergence. The verifiable inexact stopping criterion further avoids unnecessarily accurate subproblem solves. Our theoretical analysis establishes convergence guarantees for both the inner Newton iterations and the outer framework. Extensive experiments on synthetic and real-world datasets demonstrate the efficiency of IBSN relative to representative iterative OT solvers. Limitations of the proposed method are discussed in Appendix A.

Acknowledgements

We thank Prof. Lei Yang from Sun Yat-sen University for valuable guidance and insightful discussions. We also thank the authors of PINS for sharing their code.

This work was partially supported by the Guangdong Major Project of Basic Research [grant number 2026B0303000015], the National Natural Science Foundation of China [grant numbers 72495131, 82441027], Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence [grant number 2023B1212010001], and Shenzhen Stability Science Program.

References

[1] Jason M. Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, volume 30, pages 1964–1974, 2017.

[2] Brandon Amos, Giulia Luise, Samuel Cohen, and Ievgen Redko. Meta Optimal Transport. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 791–813. PMLR, 2023.

[3] David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. In *Advances in Neural Information Processing Systems*, volume 34, pages 20243–20257, 2021.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR, 2017.

[5] Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

[6] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[7] Aharon Ben-Tal, Tamar Margalit, and Arkadi Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal on Optimization*, 12(1):79–108, 2001.

[8] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84, pages 880–889. PMLR, 2018.

[9] Jérôme Bolte, Shoham Sabach, Marc Teboulle, and Yakov Vaisbourd. First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

[10] Nicolas Bonneel and Julie Digne. A survey of optimal transport for computer graphics and computer vision. In *Computer Graphics Forum*, volume 42, pages 439–460. Wiley Online Library, 2023.

[11] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–12, 2011.

[12] Christoph Brauer, Christian Clason, Dirk Lorenz, and Benedikt Wirth. A Sinkhorn-Newton method for entropic optimal transport. *arXiv preprint arXiv:1710.06635*, 2017.

[13] Regina S Burachik, Alfredo N Iusem, and Benar Fux Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Analysis*, 5(2):159–180, 1997.

[14] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 30, pages 3730–3739, 2017.

[15] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pages 2292–2300, 2013.

[17] Marco Cuturi and Gabriel Peyré. Semidual regularized optimal transport. *SIAM Review*, 60(4):941–965, 2018.

[18] Grady Daniels, Tyler Maunu, and Paul Hand. Score-based generative neural networks for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, volume 34, pages 12955–12965, 2021.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Erin Doxsey-Whitfield, Kytt MacManus, Susana B Adamo, Linda Pistolesi, John Squires, Olena Borkovska, and Sandra R Baptista. Taking advantage of the improved availability of census data: a first look at the gridded population of the world, version 4. *Papers in Applied Geography*, 1(3):226–234, 2015.

[21] Jonathan Eckstein. Approximate iterations in Bregman-function-based proximal algorithms. *Mathematical Programming*, 83(1):113–123, 1998.

[22] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[23] Rémi Flamary, Cédric Vincent-Cuaz, Nicolas Courty, Alexandre Gramfort, Oleksii Kachaiev, Huy Quang Tran, Laurène David, Clément Bonet, Nathan Cassereau, Théo Gnassounou, Eloi Tanguy, Julie Delon, Antoine Collas, Sonia Mazelet, Laetitia Chapel, Tanguy Kerdoncuff, Xizheng Yu, Matthew Feickert, Paul Krzakala, Tianlin Liu, and Eduardo Fernandes Montesuma. Pot python optimal transport (version 0.9.5), 2024.

[24] Ziv Goldfeld, Kengo Kato, Gabriel Rioux, and Ritwik Sadhu. Statistical inference with regularized optimal transport. *Information and Inference: A Journal of the IMA*, 13(1):iaad056, 2024.

[25] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International conference on machine learning*, pages 2357–2365. PMLR, 2019.

[26] Marcel Klatt, Carla Tameling, and Axel Munk. Empirical regularized optimal transport: Statistical theory and applications. *SIAM Journal on Mathematics of Data Science*, 2(2):419–443, 2020.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[28] Gen Li, Yanxi Chen, Yu Huang, Yuejie Chi, H Vincent Poor, and Yuxin Chen. Fast computation of optimal transport via entropy-regularized extragradient methods. *SIAM Journal on Optimization*, 35(2):1330–1363, 2025.

[29] Tianlin Liu, Joan Puigcerver, and Mathieu Blondel. Sparsity-constrained optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023.

[30] Vien V. Mai, Jacob Lindbäck, and Mikael Johansson. A fast and accurate splitting method for optimal transport: analysis and implementation. In *The International Conference on Learning Representations*, 2022.

[31] Eduardo Fernandes Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1161–1180, 2024.

[32] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[33] Luca Oneto, Michele Donini, Giulia Luise, Carlo Ciliberto, Andreas Maurer, and Massimiliano Pontil. Exploiting MMD and Sinkhorn divergences for fair and transferable representation learning. *Advances in Neural Information Processing Systems*, 33:15360–15370, 2020.

[34] Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. OT-Flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9223–9232, 2021.

[35] Jianting Pan and Ming Yan. Efficient sparse probability measures recovery via bregman gradient. *Journal of Scientific Computing*, 102(3):66, 2025.

[36] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *European conference on computer vision*, pages 527–544. Springer, 2020.

[37] Julien Rabin, Sira Ferradans, and Nicolas Papadakis. Adaptive color transfer with relaxed optimal transport. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4852–4856. IEEE, 2014.

[38] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International conference on scale space and variational methods in computer vision*, pages 435–446. Springer, 2011.

[39] Alain Rakotomamonjy, Rémi Flamary, and Nicolas Courty. Generalized conditional gradient: analysis of convergence and applications. *arXiv preprint arXiv:1510.06567*, 2015.

[40] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

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

[42] Jörn Schrieber, Dominic Schuhmacher, and Carsten Gottschlich. DOTmark–a benchmark for discrete optimal transport. *IEEE Access*, 5:271–282, 2016.

[43] Zhengyang Shen, Jean Feydy, Peirong Liu, Ariel H Curiale, Ruben San Jose Estepar, Raul San Jose Estepar, and Marc Niethammer. Accurate point cloud registration with robust optimal transport. *Advances in Neural Information Processing Systems*, 34:5373–5389, 2021.

[44] Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[45] Mikhail V Solodov and Benar Fux Svaiter. An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions. *Mathematics of Operations Research*, 25(2):214–230, 2000.

[46] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics (ToG)*, 33(4):1–12, 2014.

[47] Xun Tang, Michael Shavlovsky, Holakou Rahmanian, Elisa Tardini, Kiran Koshy Thekumparampil, Tesi Xiao, and Lexing Ying. Accelerating Sinkhorn algorithm with sparse Newton iterations. In *The Twelfth International Conference on Learning Representations.*, 2024.

[48] Zihao Tang and Yixuan Qiu. Safe and sparse Newton method for entropic-regularized optimal transport. *Advances in Neural Information Processing Systems*, 37:129914–129943, 2024.

[49] Chenrui Wang and Yixuan Qiu. The sparse-plus-low-rank quasi-Newton method for entropic-regularized optimal transport. In *Forty-second International Conference on Machine Learning*, 2025.

[50] Di Wu, Ling Liang, and Haizhao Yang. Pins: Proximal iterations with sparse newton and sinkhorn for optimal transport. *arXiv preprint arXiv:2502.03749*, 2025.

[51] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[52] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pages 433–453. PMLR, 2020.

[53] Lei Yang and Kim-Chuan Toh. Bregman proximal point algorithm revisited: A new inexact version and its inertial variant. *SIAM Journal on Optimization*, 32(3):1523–1554, 2022.

[54] Guojun Zhang, Zhexuan Gu, Yancheng Yuan, and Defeng Sun. HOT: An efficient halpern accelerating algorithm for optimal transport problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6703–6714, 2025.

A Limitations

The proposed method has several limitations. First, the efficiency of the Newton phase may be affected by the conditioning of the 'sparsified' Hessian. In particular, when the induced transport probabilities contain very small entries, the Newton system can become less well-conditioned, requiring more conjugate gradient iterations. Second, although the proposed sparsification strategy is efficient in practice, a complete worst-case complexity analysis of the overall inexact Bregman Newton framework remains open. Finally, our convergence analysis does not account for the inexact Newton directions produced by CG with finite stopping tolerances.

B Converting a matrix to a transportation plan

Given a general non-negative matrix $X \in \mathbb{R}^{m\times n}$, the authors in [1] put forward a simple algorithm that returns a transportation plan $\hat{X} \in \mathbb{R}^{m\times n}$ satisfying $\hat{X}\mathbf{1}_n = \boldsymbol{a}$ and $\hat{X}^\top \mathbf{1}_m = \boldsymbol{b}$, where $\mathbf{1}_m^\top \boldsymbol{a} = \mathbf{1}_n^\top \boldsymbol{b} = 1$. The algorithm is shown below as Algorithm 3.

Algorithm 3 Converting a matrix X to a feasible transportation plan [1]

Input: X, $\boldsymbol{a}$, $\boldsymbol{b}$.
for $i = 1, ..., m$ **do**
  $z_i \leftarrow \min\left\{\frac{a_i}{\sum_{j=1}^n X_{ij}}, 1\right\}$.
end for
Compute $F = \text{diag}\,(\boldsymbol{z})\, X$. % row sum is no greater than a_i
for $j = 1, ..., n$ **do**
  $y_j \leftarrow \min\left\{\frac{b_j}{\sum_{i=1}^m F_{ij}}, 1\right\}$.
end for
Update $F \leftarrow F\text{diag}\,(\boldsymbol{y})$. % column sum is no greater than b_j
Compute $\boldsymbol{e}_r \leftarrow \boldsymbol{a} - F\mathbf{1}_n$ and $\boldsymbol{e}_c \leftarrow \boldsymbol{b} - F^\top \mathbf{1}_m$.
Compute $\hat{X} \leftarrow F + \boldsymbol{e}_r \boldsymbol{e}_c^\top / \|\boldsymbol{e}_r\|_1$.
Output: $\hat{X}$.

C Proofs of Theoretical Results

C.1 Proof of Lemma 3.1

Proof. Let $(\boldsymbol{\gamma}', \boldsymbol{\zeta}')$ be the solution to the two-dual problem (4). Note that the objective function in the semi-dual problem (6) can be rewritten as $\mathcal{L}^k(\boldsymbol{\gamma}) = \min_{\boldsymbol{\zeta}\in\mathbb{R}^m} Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta}) + C$, where C is a constant. Therefore, for any $\boldsymbol{\gamma} \in \mathbb{R}^n$,

$$\mathcal{L}^k(\boldsymbol{\gamma}) = \min_{\boldsymbol{\zeta}\in\mathbb{R}^m} Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta}) + C \geq \min_{\boldsymbol{\gamma}\in\mathbb{R}^n, \boldsymbol{\zeta}\in\mathbb{R}^m} Z^k(\boldsymbol{\gamma}, \boldsymbol{\zeta}) + C = Z^k(\boldsymbol{\gamma}', \boldsymbol{\zeta}') + C.$$

On the other hand,

$$\mathcal{L}^k(\boldsymbol{\gamma}') = \min_{\boldsymbol{\zeta}\in\mathbb{R}^m} Z^k(\boldsymbol{\gamma}', \boldsymbol{\zeta}) + C \leq Z^k(\boldsymbol{\gamma}', \boldsymbol{\zeta}') + C.$$

Combining these two inequalities, we know

$$\mathcal{L}^k(\boldsymbol{\gamma}') \leq Z^k(\boldsymbol{\gamma}', \boldsymbol{\zeta}') + C \leq \mathcal{L}^k(\boldsymbol{\gamma}), \quad \forall \boldsymbol{\gamma} \in \mathbb{R}^n.$$

Therefore, $\boldsymbol{\gamma}'$ is a solution to the problem (6). □

C.2 Proof of Theorem 4.1

Proof. For simplicity, let $W = P_\rho^\top \text{diag}\,(\boldsymbol{a})\, P_\rho$. The matrix H_ρ is obviously symmetric. For arbitrary $\boldsymbol{z} \in \mathbb{R}^n$, we know

$$\boldsymbol{z}^\top H_\rho \boldsymbol{z} = \boldsymbol{z}^\top \left(\text{diag}\,(W\mathbf{1}_n) - W\right) \boldsymbol{z}/\eta = \frac{1}{2\eta} \sum_j \sum_k W_{jk}(z_j - z_k)^2 \geq 0, \tag{12}$$

Hence, H_ρ is positive semidefinite. Furthermore, by construction, there exists an $i^\star$ such that $(P_\rho)_{i^\star j} > 0$ for $\forall j$, then $W_{jk} > 0$ for $\forall j, k$ and the above inequality (12) holds with equality for $z_j = z_k$ for $\forall j, k$. □

C.3 Proof of Theorem 4.2

Lemma C.1. *For any $\boldsymbol{x} \in \mathbf{1}_n^\perp$, we have*

$$\sum_j \sum_k (x_j - x_k)^2 = 2n\|\boldsymbol{x}\|^2. \tag{13}$$

Proof. Expanding the square within the summation, we have

$$\begin{aligned}\sum_j \sum_k (x_j - x_k)^2 &= \sum_j \sum_k (x_j^2 + x_k^2) - 2\sum_j \sum_k x_j x_k \\ &= 2n\sum_j x_j^2 - 2\Big(\sum_j x_j\Big)^2 = 2n\|\boldsymbol{x}\|^2.\end{aligned}$$

This completes the proof. □

Now we are ready to prove Theorem 4.2.

Proof. Based on the construction of H_ρ, we know

$$W_{jk} = \sum_{i=1}^m a_i [P_\rho]_{ij}[P_\rho]_{ik} \geq a_{i^\star}[P_\rho]_{i^\star j}[P_\rho]_{i^\star k} \geq a_{i^\star} p^2.$$

Therefore, for any $\boldsymbol{z} \in \mathbf{1}_n^\perp$ with $\|\boldsymbol{z}\|^2 = 1$, we have

$$\boldsymbol{z}^\top H_\rho \boldsymbol{z} = \frac{1}{2\eta}\sum_j \sum_k W_{jk}(z_j - z_k)^2 \geq \frac{a_{i^\star}p^2}{2\eta}\sum_j \sum_k (z_j - z_k)^2 = \frac{n a_{i^\star} p^2}{\eta}, \tag{14}$$

where the first equation follows from Theorem 4.1 and the final equation follows from Lemma C.1.

We consider the off-diagonal elements of W. Since $[P_\rho]_{ij} + \sum_{k\neq j}[P_\rho]_{ik} = 1$, we have

$$\sum_{k\neq j} W_{jk} = \sum_{k\neq j}\sum_{i=1}^m a_i [P_\rho]_{ij}[P_\rho]_{ik} = \sum_{i=1}^m a_i [P_\rho]_{ij}(1 - [P_\rho]_{ij}) \leq \sum_{i=1}^m \frac{a_i}{4} = \frac{1}{4}.$$

Note that the eigenvector $\boldsymbol{v}$ corresponding to $\lambda(H_\rho) \neq 0$ satisfies $\mathbf{1}_n^\top \boldsymbol{v} = 0$, so $\boldsymbol{v} \in \mathbf{1}_n^\perp$. For any $\boldsymbol{z} \in \mathbb{R}^n$ with $\|\boldsymbol{z}\|^2 = 1$,

$$\boldsymbol{z}^\top H_\rho \boldsymbol{z} = \frac{1}{2\eta}\sum_j \sum_k W_{jk}(z_j - z_k)^2 \leq \frac{1}{\eta}\sum_j \sum_{k\neq j} W_{jk}(z_j^2 + z_k^2) = \frac{2}{\eta}\sum_j z_j^2 \sum_{k\neq j} W_{jk} \leq \frac{1}{2\eta}. \tag{15}$$

This completes the proof. □

C.4 Proof of Theorem 4.3

We first present the approximation error of $E_P := P - P_\rho$, where P_ρ is constructed via Algorithm 1.

Lemma C.2. *Let P and P_ρ constructed by Algorithm 1 for any $\rho \in [0, 1/n)$, then*

$$\|E_P\|_\infty \leq 2n\rho, \quad \|E_P\|_1 \leq 2mn\rho, \quad \textit{and} \quad \|E_P\| \leq 2\sqrt{mn}\rho.$$

Proof. To begin with, denote $K_i = \{j : P_{ij} < \rho\}$ and $J_i = \{j : P_{ij} \geq \rho\}$. Write $P_{i\cdot} = Q_{i\cdot} + S_{i\cdot}$ with

$$Q_{ij} = \begin{cases} P_{ij}, & j \in J_i \\ 0, & j \in K_i \end{cases}, \qquad S_{ij} = \begin{cases} 0, & j \in J_i \\ P_{ij}, & j \in K_i \end{cases},$$

then $\sum_{j\in J_i} Q_{ij} = 1 - \sum_{j\in K_i} P_{ij} := 1 - r_i$ and $\sum_{j\in K_i} S_{ij} = r_i \le n\rho$. Hence, for any $i \neq i^\star$,

$$\begin{aligned} \|P_{i\cdot} - [P_\rho]_{i\cdot}\|_1 &= \sum_{j\in J_i} \left| Q_{ij} - \frac{Q_{ij}}{1-r_i} \right| + \sum_{j\in K_i} |S_{ij} - 0| \\ &= \left(\frac{1}{1-r_i} - 1 \right) \sum_{j\in J_i} Q_{ij} + \sum_{j\in K_i} S_{ij} \\ &= 2r_i \le 2n\rho. \end{aligned}$$

Therefore, we obtain $\|E_P\|_\infty \le 2n\rho$ and $\|E_P\|_1 \le 2mn\rho$. In addition, $\|E_P\| \le \sqrt{\|E_P\|_1 \|E_P\|_\infty} \le 2\sqrt{m}n\rho$. □

Now we show the approximation error between H and sparsified H_ρ.

Proof. To begin with, observe

$$H - H_\rho = \frac{1}{\eta} \left(\underbrace{\operatorname{diag}\left(P^\top \boldsymbol{a} - P_\rho^\top \boldsymbol{a}\right)}_{=:D_1} - \left(\underbrace{P^\top \operatorname{diag}(\boldsymbol{a})\, E_P}_{=:D_2} + \underbrace{(E_P)^\top \operatorname{diag}(\boldsymbol{a})\, P_\rho}_{D_3} \right) \right).$$

For D_1, note that

$$\|D_1\| = \|\operatorname{diag}\left((E_P)^\top \boldsymbol{a}\right)\| = \|E_P^\top \boldsymbol{a}\|_\infty \le \|\boldsymbol{a}\|_\infty \|E_P\|_1 \overset{\text{Lemma } C.2}{\le} 2\rho mn \|\boldsymbol{a}\|_\infty.$$

For D_2 and D_3, we have

$$\|D_2\| \le \|P\|\, \|\boldsymbol{a}\|_\infty\, \|E_P\|, \qquad \|D_3\| \le \|E_P\|\, \|\boldsymbol{a}\|_\infty\, \|P_\rho\|.$$

In addition, since P_ρ satisfies $\|P_\rho\|_\infty = 1$ and $\|P_\rho\|_1 \le m$ for any $\rho \in [0, 1/n)$, we have $\|P_\rho\| \le \sqrt{m}$. Hence

$$\|D_2\| + \|D_3\| \le 2\sqrt{m}\, \|\boldsymbol{a}\|_\infty\, \|E_P\| \overset{\text{Lemma } C.2}{\le} 4\rho mn\, \|\boldsymbol{a}\|_\infty.$$

Combining the two parts,

$$\|H - H_\rho\| \le \frac{1}{\eta}\big(\|D_1\| + \|D_2\| + \|D_3\|\big) \le \frac{6mn\|\boldsymbol{a}\|_\infty}{\eta}\rho.$$

□

C.5 Proof of Lemma 5.1

Proof. In the Newton stage, recall that the gradient satisfies $\mathbf{1}_n^\top \boldsymbol{g}^{k,v} = 0$ and the regularized Newton system is given by: $\big(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\big)\Delta\boldsymbol{\gamma}^{k,v} = -\boldsymbol{g}^{k,v}$. Multiplying both sides of the Newton system by $\mathbf{1}_n^\top$ from the left yields

$$\mathbf{1}_n^\top H_\rho^{k,v} \Delta\boldsymbol{\gamma}^{k,v} + \|\boldsymbol{g}^{k,v}\| \mathbf{1}_n^\top \Delta\boldsymbol{\gamma}^{k,v} = 0.$$

From Theorem 4.1, the sparsified Hessian satisfies $H_\rho^{k,v}\mathbf{1}_n = 0$. Due to the symmetry of $H_\rho^{k,v}$, we also have $\mathbf{1}_n^\top H_\rho^{k,v} = 0$. Consequently, the first term vanishes, leaving $\mathbf{1}_n^\top \Delta\boldsymbol{\gamma}^{k,v} = 0$ if $\|\boldsymbol{g}^{k,v}\| \neq 0$. This ensures that the update maintains $\mathbf{1}_n^\top \boldsymbol{\gamma}^{k,v+1} = 0$. □

C.6 Proof of Theorem 5.2

In the following analysis, we fix the outer iteration k. We first present the following lemmas.

Lemma C.3. *The set $S^k(\boldsymbol{\gamma}) := \{\boldsymbol{\gamma} \mid \mathcal{L}^k(\boldsymbol{\gamma}) \leq \mathcal{L}^k(\boldsymbol{\gamma}^{k,1}),\ \boldsymbol{\gamma} \in \mathbf{1}_n^\perp\}$ is bounded, closed, and convex, $\mathcal{L}^k(\boldsymbol{\gamma})$ is bounded below.*

Proof. Define $\kappa_k := \min_{i,j}\{X_{ij}^k \exp(-C_{ij}/\eta)\} > 0$. For $M(\boldsymbol{\gamma}) := \max_j \gamma_j$, the log-sum-exp term yields

$$\mathcal{L}^k(\boldsymbol{\gamma}) \geq -\boldsymbol{b}^\top \boldsymbol{\gamma} + M(\boldsymbol{\gamma}) + \eta \log \kappa_k \geq n \min_j b_j\, M(\boldsymbol{\gamma}) + \eta \log \kappa_k,$$

where the last inequality uses $\mathbf{1}_n^\top \boldsymbol{\gamma} = 0$. On $\mathbf{1}_n^\perp$, $\|\boldsymbol{\gamma}\| \to \infty$ implies $M(\boldsymbol{\gamma}) \to \infty$. Thus, $\mathcal{L}^k$ is coercive on $\mathbf{1}_n^\perp$. Since $\mathcal{L}^k$ is continuous and convex, S^k is closed, bounded and convex, and it is nonempty because it contains $\boldsymbol{\gamma}^{k,1}$. Continuity and coercivity also imply that $\mathcal{L}^k$ attains its minimum on this subspace. Finally, $\mathcal{L}^k(\boldsymbol{\gamma} + c\mathbf{1}_n) = \mathcal{L}^k(\boldsymbol{\gamma})$ for all $c \in \mathbb{R}$, so this minimum is also the minimum over $\mathbb{R}^n$. □

Lemma C.4. *Let $H_\rho^k(\boldsymbol{\gamma})$ be the matrix output by Algorithm 1, then for $\rho \in [0, 1/n)$, there exist a constant $\underline{C}_k > 0$ such that $\lambda_{\min}^+(H_\rho^k(\boldsymbol{\gamma})) \geq \underline{C}_k$ for all $\boldsymbol{\gamma} \in S^k(\boldsymbol{\gamma})$.*

Proof. By the boundedness of $S^k(\boldsymbol{\gamma})$, each element of $P^k(\boldsymbol{\gamma})$ as defined in (7) is bounded below from zero over $S^k(\boldsymbol{\gamma})$. Recall the lower bound of $\lambda_{\min}^+(H_\rho^k(\boldsymbol{\gamma}))$ established in Theorem 4.2, we conclude that there exists a constant $\underline{C}_k > 0$ such that $\lambda_{\min}^+(H_\rho^k(\boldsymbol{\gamma})) \geq \underline{C}_k$ holds for any $\rho \in [0, 1/n)$ and all $\boldsymbol{\gamma} \in S^k(\boldsymbol{\gamma})$. □

As shown in Lemma 5.1, the convergence analysis can be safely restricted to $\mathbf{1}_n^\perp$ in the subsequent analysis. We then show that $\Delta\boldsymbol{\gamma}^{k,v}$ is a descent direction and provide the minimum stepsize.

Lemma C.5. *In the outer k iteration, let $\{\boldsymbol{\gamma}^{k,v}\}_v$ be the sequence generated by Algorithm 2 satisfying $\mathbf{1}_n^\top \boldsymbol{\gamma}^{k,0} = 0$, $\Delta\boldsymbol{\gamma}^{k,v}$ be a direction for $\mathcal{L}^k$ and we choose $t^{k,v}$ as stepsize by Armijo backtracking, then*

- *(i) $\Delta\boldsymbol{\gamma}^{k,v}$ is a descent direction for $\mathcal{L}^k$;*
- *(ii) the backtracking terminates finitely with a step size $t^{k,v}$ bounded below by*

$$t^{k,v} \geq t_{\min}^k := \min\left\{1,\ \beta \frac{4\eta(1-\sigma)\underline{C}_k^2}{1/(2\eta) + 2}\right\} > 0, \tag{16}$$

where $\beta \in (0,1)$ is the backtracking contraction factor and $\sigma \in (0,1)$ is the Armijo parameter.

Proof. (i) To begin with, it is easy to see that

$$(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} = -(\boldsymbol{g}^{k,v})^\top \left(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\right)^{-1} (\boldsymbol{g}^{k,v}) < 0,$$

where $\boldsymbol{g}^{k,v}$ is the gradient of $\mathcal{L}^k$ on $\boldsymbol{\gamma}^{k,v}$ and the inequality due to Theorem 4.1. Therefore, $\Delta\boldsymbol{\gamma}^{k,v}$ is a descent direction.

(ii) Since $\lambda_{\max}(H_\rho) \leq 1/(2\eta)$ for any$\rho \in [0, 1/n)$ from Theorem 4.2, $\boldsymbol{g}^{k,v}$ is $1/(2\eta)$-Lipschitz continuous, and the standard quadratic upper bound yields

$$\mathcal{L}^k(\boldsymbol{\gamma}^{k,v} + t\Delta\boldsymbol{\gamma}^{k,v}) \leq \mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) + t(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} + \frac{1}{4\eta} t^2 \|\Delta\boldsymbol{\gamma}^{k,v}\|^2.$$

Hence any t with

$$0 < t \leq \bar{t}^{k,v} := \frac{4\eta(1-\sigma)\left(-(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v}\right)}{\|\Delta\boldsymbol{\gamma}^{k,v}\|^2}$$

satisfies the Armijo inequality $\mathcal{L}^k(\boldsymbol{\gamma}^{k,v} + t\Delta\boldsymbol{\gamma}^{k,v}) \leq \mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) + \sigma t(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v}$.

We now provide a lower bound for $\bar{t}^{k,v}$. First, note that

$$-(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} = (\boldsymbol{g}^{k,v})^\top \left(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\right)^{-1} (\boldsymbol{g}^{k,v}) \geq \frac{1}{\lambda_{\max}(H_\rho^{k,v}) + \|\boldsymbol{g}^{k,v}\|} \|\boldsymbol{g}^{k,v}\|^2,$$

$$\|\Delta\boldsymbol{\gamma}^{k,v}\| = \left(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I\right)^{-1} \boldsymbol{g}^{k,v} \leq \frac{1}{\lambda_{\min}^+(H_\rho^{k,v}) + \|\boldsymbol{g}^{k,v}\|} \|\boldsymbol{g}^{k,v}\| \leq \frac{1}{\lambda_{\min}^+(H_\rho^{k,v})} \|\boldsymbol{g}^{k,v}\|.$$

Combining these,

$$\bar{t}^{k,v} \geq \frac{4\eta(1-\sigma)[\lambda_{\min}^+(H_\rho^{k,v})]^2}{\lambda_{\max}(H_\rho^{k,v}) + \|\boldsymbol{g}^{k,v}\|} \geq \frac{4\eta(1-\sigma)[\lambda_{\min}^+(H_\rho^{k,v})]^2}{\lambda_{\max}(H_\rho^{k,v}) + 2},$$

where the final inequality holds since $\|\boldsymbol{g}^{k,v}\| \leq \|(P^k(\boldsymbol{\gamma}^{k,v}))^\top \boldsymbol{a}\| + \|\boldsymbol{b}\| \leq 2$. Backtracking with ratio β accepts some $t^{k,v} \geq \beta\bar{t}^{k,v}$ and using the bound $\lambda_{\max}(H_\rho^{k,v}) \leq 1/(2\eta)$ and $\lambda_{\min}^+(H_\rho^{k,v})$ in Theorem 4.2. From the Armijo criterion, we know $\{\mathcal{L}^k(\boldsymbol{\gamma}^{k,v})\}_v$ is monotonically decreasing and $\boldsymbol{\gamma}^{k,v} \in S^k(\boldsymbol{\gamma})$ for all v. By Lemma C.4, we know $\lambda_{\min}^+(H_\rho^{k,v}) \geq \underline{C}_k > 0$ for all v and $\rho \in [0, 1/n)$ over $S^k(\boldsymbol{\gamma})$. This allows us to define a uniform lower bound for the step size:

$$t_{\min}^{k,v} \geq t_{\min}^k := \min\left\{1,\ \beta\frac{4\eta(1-\sigma)\underline{C}_k^2}{1/(2\eta)+2}\right\},$$

yields (16). □

Next, we provide the proof of Theorem 5.2.

Proof. (i) From Lemma C.5, we know

$$-(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} \geq \frac{1}{\lambda_{\max}(H^{k,v}) + \|\boldsymbol{g}^{k,v}\|} \|\boldsymbol{g}^{k,v}\|^2 \geq \frac{1}{1/(2\eta)+2} \|\boldsymbol{g}^{k,v}\|^2,$$

since $\|\boldsymbol{g}^{k,v}\| \leq 2$. It further yields

$$\mathcal{L}^k(\boldsymbol{\gamma}^{k,v+1}) - \mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) \leq -\sigma t_{\min}^k \cdot \frac{1}{1/(2\eta)+2} \|\boldsymbol{g}^{k,v}\|^2, \tag{17}$$

where $t_{\min}^k$ is defined in Lemma C.5.

In addition, since $\mathcal{L}^k$ is bounded below by Lemma C.3, the telescoping sum of (17) implies $\sum_v \|\boldsymbol{g}^{k,v}\|^2 < \infty$. Therefore, we conclude that $\|\boldsymbol{g}^{k,v}\| \to 0$ as $v \to \infty$.

(ii) By the definition of $\boldsymbol{\gamma}^{k\star}$, we know $g^k(\boldsymbol{\gamma}^{k\star}) \in \text{span}\{\mathbf{1}_n\}$, where $g^k(\boldsymbol{\gamma})$ is defined in (8). On the other hand, it is easy to check that for all $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\mathbf{1}_n^\top g^k(\boldsymbol{\gamma}) = 0$. Combining the two relations implies $g^k(\boldsymbol{\gamma}^{k\star}) = \mathbf{0}_n$.

Over $S^k(\boldsymbol{\gamma})$, since $\mathcal{L}^k$ is $\underline{C}_k$-strongly convex, the solution of $\mathcal{L}^k$ is unique and

$$\|\boldsymbol{g}^{k,v}\| = \|\boldsymbol{g}^{k,v} - g^k(\boldsymbol{\gamma}^{k\star})\| \geq \underline{C}_k \|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\|$$

by [32, Theorem 2.1.10]. Since $\|\boldsymbol{g}^{k,v}\| \to 0$, it follows that $\boldsymbol{\gamma}^{k,v} \to \boldsymbol{\gamma}^{k\star}$ as $v \to \infty$, where $\boldsymbol{\gamma}^{k\star}$ satisfies $\mathbf{1}_n^\top \boldsymbol{\gamma}^{k\star} = 0$. □

C.7 Proof of Theorem 5.3

The following lemma shows the global Lipschitz continuity of the exact Hessian $H^k(\boldsymbol{\gamma})$.

Lemma C.6. *For any outer iteration k, the exact Hessian $H^k(\boldsymbol{\gamma})$ of the subproblem objective $\mathcal{L}^k(\boldsymbol{\gamma})$ is globally Lipschitz continuous over $\mathbb{R}^n$. The global Lipschitz constant L_H is given by:*

$$L_H = \frac{6}{\eta^2}.$$

Proof. Recall that the directional derivative of a differentiable multivariate function $f(\boldsymbol{\gamma})$ along a vector $\boldsymbol{u}$ is defined as $\nabla_{\boldsymbol{u}} f(\boldsymbol{\gamma}) = \langle \nabla f(\boldsymbol{\gamma}), \boldsymbol{u} \rangle$. To establish global Lipschitz continuity, we evaluate the spectral norm of the third derivative tensor. Equivalently, we show that for any arbitrary vectors $\boldsymbol{y}, \boldsymbol{u} \in \mathbb{R}^n$, the third-order directional derivative of the objective is uniformly bounded by the constant L_H:

$$|\nabla_{\boldsymbol{u}} (\boldsymbol{y}^\top H^k(\boldsymbol{\gamma}) \boldsymbol{y})| \leq L_H \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\|.$$

For brevity, let $P = P^k(\boldsymbol{\gamma})$. We define a deterministic weighted sum operator $S_i : \mathbb{R}^n \to \mathbb{R}$, representing the inner product with the i-th row of the transport probability matrix P:

$$S_i(\boldsymbol{x}) = \sum_{j=1}^{n} P_{ij} x_j.$$

Because each row of P is a valid probability distribution (i.e., $P_{ij} \geq 0$ and $\sum_j P_{ij} = 1$), the operator S_i computes a strict convex combination. Its magnitude is universally bounded by the infinity norm, and consequently the Euclidean norm, of the input vector: $|S_i(\boldsymbol{x})| \leq \|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|$.

Using this operator, the quadratic form $Q(\boldsymbol{\gamma}) := \boldsymbol{y}^\top H^k(\boldsymbol{\gamma}) \boldsymbol{y}$ can be explicitly expanded as:

$$Q(\boldsymbol{\gamma}) = \frac{1}{\eta} \sum_{i=1}^{m} a_i \left[S_i(\boldsymbol{y}^2) - (S_i(\boldsymbol{y}))^2 \right],$$

where $\boldsymbol{y}^2$ denotes the Hadamard (element-wise) square of the vector $\boldsymbol{y}$.

To compute the directional derivative of $Q(\boldsymbol{\gamma})$ along $\boldsymbol{u}$, we first require the directional derivative of the transport probability matrix entries. From the definition of P_{ij} in (7), we can write the logarithm of P_{ij} as:

$$\ln P_{ij} = \ln X_{ij}^k + \frac{\gamma_j - C_{ij}}{\eta} - \ln Z_i,$$

where $Z_i = \sum_{l=1}^{n} X_{il}^k \exp((\gamma_l - C_{il})/\eta)$ is the row normalization factor. Taking the directional derivative of $\ln Z_i$ along $\boldsymbol{u}$ gives:

$$\nabla_{\boldsymbol{u}} \ln Z_i = \frac{1}{Z_i} \sum_{l=1}^{n} X_{il}^k \exp\left(\frac{\gamma_l - C_{il}}{\eta} \right) \frac{u_l}{\eta} = \frac{1}{\eta} \sum_{l=1}^{n} P_{il} u_l = \frac{1}{\eta} S_i(\boldsymbol{u}).$$

Applying the chain rule, $\nabla_{\boldsymbol{u}} \ln P_{ij} = \frac{1}{P_{ij}} \nabla_{\boldsymbol{u}} P_{ij}$, and differentiating the expression for $\ln P_{ij}$ yields:

$$\frac{1}{P_{ij}} \nabla_{\boldsymbol{u}} P_{ij} = \frac{u_j}{\eta} - \nabla_{\boldsymbol{u}} \ln Z_i = \frac{u_j}{\eta} - \frac{1}{\eta} S_i(\boldsymbol{u}).$$

Multiplying both sides by P_{ij}, we isolate the directional derivative of P_{ij}:

$$\nabla_{\boldsymbol{u}} P_{ij} = \frac{1}{\eta} P_{ij} (u_j - S_i(\boldsymbol{u})).$$

By linearity, the derivative of the operator $S_i(\boldsymbol{x})$ for a fixed vector $\boldsymbol{x}$ evaluates to:

$$\nabla_{\boldsymbol{u}} S_i(\boldsymbol{x}) = \sum_{j=1}^{n} (\nabla_{\boldsymbol{u}} P_{ij}) x_j = \frac{1}{\eta} \left[S_i(\boldsymbol{x} \odot \boldsymbol{u}) - S_i(\boldsymbol{x}) S_i(\boldsymbol{u}) \right].$$

Let $B_i = S_i(\boldsymbol{y}^2) - (S_i(\boldsymbol{y}))^2$. Differentiating B_i along $\boldsymbol{u}$ gives:

$$\nabla_{\boldsymbol{u}} B_i = \frac{1}{\eta} \Big[S_i(\boldsymbol{y}^2 \odot \boldsymbol{u}) - S_i(\boldsymbol{y}^2) S_i(\boldsymbol{u}) - 2 S_i(\boldsymbol{y}) \big(S_i(\boldsymbol{y} \odot \boldsymbol{u}) - S_i(\boldsymbol{y}) S_i(\boldsymbol{u}) \big) \Big].$$

Applying the absolute magnitude bound $|S_i(\boldsymbol{x})| \leq \|\boldsymbol{x}\|$, we have $|S_i(\boldsymbol{y}^2 \odot \boldsymbol{u})| \leq \|\boldsymbol{y}^2 \odot \boldsymbol{u}\|_\infty \leq \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\|$, and similarly for the other terms. Using the triangle inequality, we can strictly bound the magnitude of $\nabla_{\boldsymbol{u}} B_i$:

$$|\nabla_{\boldsymbol{u}} B_i| \leq \frac{1}{\eta} \left(\|\boldsymbol{y}\|^2 \|\boldsymbol{u}\| + \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\| + 2 \|\boldsymbol{y}\| \left(\|\boldsymbol{y}\| \|\boldsymbol{u}\| + \|\boldsymbol{y}\| \|\boldsymbol{u}\| \right) \right) = \frac{6}{\eta} \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\|.$$

Finally, substituting this bound back into the derivative of the total quadratic form:

$$|\nabla_{\boldsymbol{u}} Q(\boldsymbol{\gamma})| \leq \frac{1}{\eta} \sum_{i=1}^{m} a_i |\nabla_{\boldsymbol{u}} B_i| \leq \frac{6}{\eta^2} \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\| \sum_{i=1}^{m} a_i.$$

Since $\boldsymbol{a}$ is a marginal probability vector satisfying $\sum_{i=1}^{m} a_i = 1$, we obtain $|\nabla_{\boldsymbol{u}} Q(\boldsymbol{\gamma})| \leq \frac{6}{\eta^2} \|\boldsymbol{y}\|^2 \|\boldsymbol{u}\|$. This confirms that the third-order directional derivative is uniformly bounded, thereby establishing the global Lipschitz constant as $L_H = 6/\eta^2$. □

The following lemma guarantees that after a sufficient number of iterations, the Armijo backtracking procedure accepts the full Newton step ($t^{k,v} = 1$).

Lemma C.7. *In the outer k iteration, let $\{\boldsymbol{\gamma}^{k,v}\}_v$ be the sequence generated by Algorithm 2. Then for sufficiently large v, the Armijo line search condition is strictly satisfied with $t^{k,v} = 1$.*

Proof. From the lemma C.6, the exact Hessian $H^k(\boldsymbol{\gamma})$ of the subproblem objective $\mathcal{L}^k(\boldsymbol{\gamma})$ is globally Lipschitz continuous with the constant L_H. By [32, Lemma 1.2.4]

$$\mathcal{L}^k(\boldsymbol{\gamma}^{k,v} + \Delta\boldsymbol{\gamma}^{k,v}) \leq \mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) + (\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} + \frac{1}{2}(\Delta\boldsymbol{\gamma}^{k,v})^\top H^{k,v} \Delta\boldsymbol{\gamma}^{k,v} + \frac{L_H}{6}\|\Delta\boldsymbol{\gamma}^{k,v}\|^3. \quad (18)$$

Note that for sufficiently large v, we have

$$\begin{aligned}
D_{\text{Armijo}} &:= \mathcal{L}^k(\boldsymbol{\gamma}^{k,v} + \Delta\boldsymbol{\gamma}^{k,v}) - \mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) - \sigma(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} \\
&\leq (1-\sigma)(\boldsymbol{g}^{k,v})^\top \Delta\boldsymbol{\gamma}^{k,v} + \frac{1}{2}(\Delta\boldsymbol{\gamma}^{k,v})^\top H^{k,v} \Delta\boldsymbol{\gamma}^{k,v} + \frac{L_H}{6}\|\Delta\boldsymbol{\gamma}^{k,v}\|^3 \\
&= -(1-\sigma)(\Delta\boldsymbol{\gamma}^{k,v})^\top H_\rho^{k,v} \Delta\boldsymbol{\gamma}^{k,v} - (1-\sigma)\|\boldsymbol{g}^{k,v}\|\|\Delta\boldsymbol{\gamma}^{k,v}\|^2 \\
&\quad + \frac{1}{2}(\Delta\boldsymbol{\gamma}^{k,v})^\top H^{k,v} \Delta\boldsymbol{\gamma}^{k,v} + \frac{L_H}{6}\|\Delta\boldsymbol{\gamma}^{k,v}\|^3 \\
&= \frac{1}{2}(\Delta\boldsymbol{\gamma}^{k,v})^\top (H^{k,v} - H_\rho^{k,v}) \Delta\boldsymbol{\gamma}^{k,v} - \left(\frac{1}{2} - \sigma\right)(\Delta\boldsymbol{\gamma}^{k,v})^\top H_\rho^{k,v} \Delta\boldsymbol{\gamma}^{k,v} \\
&\quad - (1-\sigma)\|\boldsymbol{g}^{k,v}\|\|\Delta\boldsymbol{\gamma}^{k,v}\|^2 + \frac{L_H}{6}\|\Delta\boldsymbol{\gamma}^{k,v}\|^3 \\
&\leq \left((3\|\boldsymbol{a}\|_\infty - (1-\sigma))\|\boldsymbol{g}^{k,v}\| + \frac{L_H}{6}\|\Delta\boldsymbol{\gamma}^{k,v}\| - \left(\frac{1}{2} - \sigma\right)\lambda_{\min}^+(H_\rho^{k,v})\right)\|\Delta\boldsymbol{\gamma}^{k,v}\|^2,
\end{aligned} \quad (19)$$

where $\sigma \in (0, 1/2)$ is the Armijo parameter. The first inequality comes from Equation (18), the second equality comes from the fact that $(H_\rho^{k,v} + \|\boldsymbol{g}^{k,v}\| I)\Delta\boldsymbol{\gamma}^{k,v} = -\boldsymbol{g}^{k,v}$, and the final inequality holds since $\|H^{k,v} - H_\rho^{k,v}\| \leq \frac{6mn\|\boldsymbol{a}\|_\infty}{\eta}\rho = 6\|\boldsymbol{a}\|_\infty\|\boldsymbol{g}^{k,v}\|$ from Theorem 4.3 and $\mathbf{1}_n^\top \Delta\boldsymbol{\gamma}^{k,v} = 0$ from Lemma 5.1. In addition, following the proof of Lemma C.5, we know $\|\Delta\boldsymbol{\gamma}^{k,v}\| \leq \|\boldsymbol{g}^{k,v}\|/[\lambda_{\min}^+(H_\rho^{k,v})]$ and $\lambda_{\min}^+(H_\rho^{k,v}) \geq \underline{C}_k > 0$ for all v and $\rho \in [0, 1/n)$ over $S^k(\boldsymbol{\gamma})$. Therefore, we further bound the D_{Armijo} by:

$$D_{\text{Armijo}} \leq \left[\left(3\|\boldsymbol{a}\|_\infty - (1-\sigma) + \frac{L_H}{6\underline{C}_k}\right)\|\boldsymbol{g}^{k,v}\| - \left(\frac{1}{2} - \sigma\right)\underline{C}_k\right]\|\Delta\boldsymbol{\gamma}^{k,v}\|^2. \quad (20)$$

Since $\|\boldsymbol{g}^{k,v}\| \to 0$ as $v \to \infty$, for sufficiently large v, we have $D_{\text{Armijo}} \leq 0$, which means $t^{k,v} = 1$. □

Now we begin to proof Theorem 5.3.

Proof. The following proof is motivated by [48]. By Theorem 4.2 and Lipschitz continuity of $H^k(\boldsymbol{\gamma})$, we have

$$\begin{aligned}
&\left\|\boldsymbol{g}^{k,v}\right\| \leq 1/(2\eta)\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|, \\
&\left\|\boldsymbol{g}^{k,v} - g^k\left(\boldsymbol{\gamma}^{k\star}\right) - H^{k,v}\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right)\right\| \leq \frac{L_H}{2}\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|^2.
\end{aligned}$$

In addition, denote $B_\rho^{k,v} := H_\rho^{k,v} + \left\|\boldsymbol{g}^{k,v}\right\| I$, we have

$$\left\|H^{k,v} - B_\rho^{k,v}\right\| \le \left\|H^{k,v} - H_\rho^{k,v}\right\| + \left\|\boldsymbol{g}^{k,v}\right\| \le (6\|\boldsymbol{a}\|_\infty + 1)\left\|\boldsymbol{g}^{k,v}\right\| \le \frac{6\|\boldsymbol{a}\|_\infty + 1}{2\eta}\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|,$$

where the second inequality holds due to Theorem 4.3.

In addition, since $\boldsymbol{\gamma}^{k,v} \in S^k(\boldsymbol{\gamma})$, we have

$$\left\|(B_\rho^{k,v})^{-1}\right\| = \left(\lambda_{\min}^+(B_\rho^{k,v})\right)^{-1} \le \left(\lambda_{\min}^+(H_\rho^{k,v})\right)^{-1} \le \underline{C}_k^{-1},$$

where $\underline{C}_k > 0$ is defined in Lemma C.4. As a result,

$$\begin{aligned}
\left\|\boldsymbol{\gamma}^{k,v+1} - \boldsymbol{\gamma}^{k\star}\right\| &= \left\|\boldsymbol{\gamma}^{k,v} + \Delta\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\| = \left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star} - (B_\rho^{k,v})^{-1} g^{k,v}\right\| \\
&= \left\|(B_\rho^{k,v})^{-1}\left(B_\rho^{k,v}\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right) - g^{k,v}\right)\right\| \\
&\le \left\|(B_\rho^{k,v})^{-1}\right\| \cdot \left\|g^{k,v} - g^k\left(\boldsymbol{\gamma}^{k\star}\right) - H^{k,v}\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right) + \left(H^{k,v} - B_\rho^{k,v}\right)\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right)\right\| \\
&\le \underline{C}_k^{-1}\left(\left\|g^{k,v} - g^k\left(\boldsymbol{\gamma}^{k\star}\right) - H^{k,v}\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right)\right\| + \left\|\left(H^{k,v} - B_\rho^{k,v}\right)\left(\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right)\right\|\right) \\
&\le \underline{C}_k^{-1}\left(\frac{1}{2}L_H\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|^2 + \frac{6\|\boldsymbol{a}\|_\infty + 1}{2\eta}\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|^2\right) \\
&= \underline{C}_k^{-1}\left(\frac{1}{2}L_H + \frac{6\|\boldsymbol{a}\|_\infty + 1}{2\eta}\right)\left\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\right\|^2.
\end{aligned}$$

By setting $L \coloneqq \underline{C}_k^{-1}\left(\frac{1}{2}L_H + \frac{6\|\boldsymbol{a}\|_\infty + 1}{2\eta}\right)$, we have

$$\|\boldsymbol{\gamma}^{k,v+1} - \boldsymbol{\gamma}^{k\star}\| \le L\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\|^2.$$

□

The following lemma quantifies when the inner iterations transition from the globalization phase governed by Armijo line search to the pure Newton phase, and provides an explicit upper bound on the number of iterations required for this transition.

Lemma C.8. *Fix an outer iteration k, and let $\{\boldsymbol{\gamma}^{k,v}\}_v$ be the inner sequence generated by Algorithm 2. Let $\mathcal{L}^{k\star} := \mathcal{L}^k(\boldsymbol{\gamma}^{k\star})$ denote the optimal value of the subproblem restricted to $\mathbf{1}_n^\perp$. Define*

$$C_A := 3\|\boldsymbol{a}\|_\infty - (1-\sigma) + \frac{L_H}{6\underline{C}_k}.$$

Then the following two cases hold.

(i) ***Case 1:*** *$C_A \le 0$. The Armijo line search accepts the full Newton step at every inner iteration; that is, $t^{k,v} = 1$, $\forall v \ge 1$.*

(ii) ***Case 2:*** *$C_A > 0$. The inner iterations enter the pure Newton phase once the gradient norm satisfies*

$$\|\boldsymbol{g}^{k,v}\| \le \epsilon_{\text{phase}},$$

where $\epsilon_{\text{phase}} := \min\{\epsilon_{\text{Newton}}, \epsilon_{\text{contract}}\}$, $\epsilon_{\text{Newton}} := \frac{(1/2-\sigma)\underline{C}_k}{C_A}$, and $\epsilon_{\text{contract}} := \frac{2\eta\underline{C}_k^2}{L}$. From that point onward, the full Newton step is accepted at all subsequent inner iterations. Moreover, let V_{global} denote the maximum number of inner iterations required to enter this local quadratic convergence phase. Then

$$V_{\text{global}} \le \left\lfloor \frac{\left(\mathcal{L}^k(\boldsymbol{\gamma}^{k,0}) - \mathcal{L}^{k\star}\right)(1+4\eta)}{2\eta\sigma t_{\min}^k \epsilon_{\text{phase}}^2} \right\rfloor, \tag{21}$$

where $t_{\min}^k$ is the lower bound on the Armijo step size given in Lemma C.5, and L is the quadratic convergence constant in Theorem 5.3.

Proof. To guarantee that the algorithm permanently transitions into the pure Newton phase, it is sufficient to reach a state where the full Newton step ($t^{k,v} = 1$) is unconditionally accepted and the subsequent gradient norm is non-increasing ($\|\boldsymbol{g}^{k,v+1}\| \leq \|\boldsymbol{g}^{k,v}\|$). This ensures that once the iterates enter this local convergence region, they will not escape back into the global phase where the backtracking line search truncates the step.

First, we revisit the condition for accepting the full Newton step. In the proof of Lemma C.7, we established that the Armijo condition is naturally satisfied for $t^{k,v} = 1$ when the difference $D_{\text{Armijo}} \leq 0$. We demonstrated this is algebraically guaranteed when:

$$C_A \|\boldsymbol{g}^{k,v}\| - \left(\frac{1}{2} - \sigma\right) \underline{C}_k \leq 0.$$

This leads to two distinct cases. If $C_A \leq 0$, the inequality holds trivially for any valid gradient norm because the subtracted term is strictly positive ($\sigma \in (0, 1/2)$ and $\underline{C}_k > 0$). In this scenario, the Armijo condition never truncates the step, and the algorithm executes pure Newton steps globally from $v = 1$.

Conversely, if $C_A > 0$, isolating $\|\boldsymbol{g}^{k,v}\|$ yields the explicit finite threshold ϵ_{Newton}. Thus, whenever $\|\boldsymbol{g}^{k,v}\| \leq \epsilon_{\text{Newton}}$, the full step $t^{k,v} = 1$ is guaranteed to be accepted.

Second, we determine the non-increasing gradient threshold. From Theorem 5.3, a full Newton step yields the local quadratic bound $\|\boldsymbol{\gamma}^{k,v+1} - \boldsymbol{\gamma}^{k\star}\| \leq L\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\|^2$. Using the $1/(2\eta)$-Lipschitz continuity of the gradient established in Theorem 4.2 (where $g^k(\boldsymbol{\gamma}^{k\star}) = \mathbf{0}_n$), and the strong convexity parameter $\underline{C}_k$ on the restricted subspace, we relate this distance bound back to the gradient norm:

$$\|\boldsymbol{g}^{k,v+1}\| \leq \frac{1}{2\eta}\|\boldsymbol{\gamma}^{k,v+1} - \boldsymbol{\gamma}^{k\star}\| \leq \frac{L}{2\eta}\|\boldsymbol{\gamma}^{k,v} - \boldsymbol{\gamma}^{k\star}\|^2 \leq \frac{L}{2\eta\underline{C}_k^2}\|\boldsymbol{g}^{k,v}\|^2.$$

For the sequence of gradient norms to be non-increasing (i.e., $\|\boldsymbol{g}^{k,v+1}\| \leq \|\boldsymbol{g}^{k,v}\|$), it is sufficient that the coefficient satisfies $\frac{L}{2\eta\underline{C}_k^2}\|\boldsymbol{g}^{k,v}\| \leq 1$, which algebraically reduces to:

$$\|\boldsymbol{g}^{k,v}\| \leq \frac{2\eta\underline{C}_k^2}{L} := \epsilon_{\text{contract}}.$$

Therefore, assuming $C_A > 0$, if an iterate satisfies the joint condition $\|\boldsymbol{g}^{k,v}\| \leq \min(\epsilon_{\text{Newton}}, \epsilon_{\text{contract}}) := \epsilon_{\text{phase}}$, the algorithm accepts the full step $t^{k,v} = 1$ and ensures that $\|\boldsymbol{g}^{k,v+1}\| \leq \|\boldsymbol{g}^{k,v}\| \leq \epsilon_{\text{phase}}$. By mathematical induction, all subsequent inner iterations will remain bounded by this threshold, executing pure Newton steps indefinitely.

Consequently, to mathematically bound the maximum number of iterations V_{global} prior to this permanent transition, we analyze the worst-case trajectory where the iterates have not yet satisfied this joint sufficient condition, meaning $\|\boldsymbol{g}^{k,v}\| > \epsilon_{\text{phase}}$.

We recall the uniform sufficient decrease condition established in (17):

$$\mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) - \mathcal{L}^k(\boldsymbol{\gamma}^{k,v+1}) \geq \sigma t_{\min}^k \frac{1}{1/(2\eta) + 2}\|\boldsymbol{g}^{k,v}\|^2 = \sigma t_{\min}^k \frac{2\eta}{1 + 4\eta}\|\boldsymbol{g}^{k,v}\|^2.$$

Coupling this with the condition $\|\boldsymbol{g}^{k,v}\| > \epsilon_{\text{phase}}$, the objective function is guaranteed to decrease by a strictly bounded minimum amount at every step prior to the guaranteed phase transition:

$$\mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) - \mathcal{L}^k(\boldsymbol{\gamma}^{k,v+1}) > \frac{2\eta\sigma t_{\min}^k}{1 + 4\eta}\epsilon_{\text{phase}}^2 := \Delta_{\min}.$$

Because the objective function evaluates to $\mathcal{L}^k(\boldsymbol{\gamma}^{k,0})$ at initialization and is bounded globally from below by $\mathcal{L}^{k\star}$, the cumulative decrease over the first V_{global} iterations cannot exceed $\mathcal{L}^k(\boldsymbol{\gamma}^{k,0}) - \mathcal{L}^{k\star}$. Summing the functional decreases over these initial global iterations yields:

$$V_{\text{global}} \cdot \Delta_{\min} \leq \sum_{v=0}^{V_{\text{global}}-1} (\mathcal{L}^k(\boldsymbol{\gamma}^{k,v}) - \mathcal{L}^k(\boldsymbol{\gamma}^{k,v+1})) \leq \mathcal{L}^k(\boldsymbol{\gamma}^{k,0}) - \mathcal{L}^{k\star}.$$

Rearranging this inequality for V_{global} and substituting the definition of $\Delta_{\min}$ yields the final bound on the number of iterations required to guarantee reaching the permanent quadratic phase:

$$V_{\text{global}} \leq \frac{\mathcal{L}^k(\boldsymbol{\gamma}^{k,0}) - \mathcal{L}^{k\star}}{\Delta_{\min}} = \frac{(\mathcal{L}^k(\boldsymbol{\gamma}^{k,0}) - \mathcal{L}^{k\star})(1 + 4\eta)}{2\eta\sigma t_{\min}^k \epsilon_{\text{phase}}^2}.$$

Taking the floor of this bound completes the proof. □

C.8 Mapping to the inexact Bregman convergence theorem

Let $\widetilde{X}^{k+1} := P_\Omega(X^{k+1}) \in \Omega$ denote the feasible point used in the stopping criterion. The subproblem (3) is a specialization of the inexact Bregman proximal point framework of [53] with constant proximal parameter η. In the OT setting, the dual first-order relation implies that the residual and subgradient-error terms in their framework vanish. Hence, the only remaining inexactness condition is

$$D_\phi(\widetilde{X}^{k+1}, X^{k+1}) \le \mu_k. \tag{22}$$

Therefore, if $\sum_{k=0}^{\infty} \mu_k < \infty$, [53, Theorem 3.2] guarantees that $\{X^k\}$ and $\{\widetilde{X}^k\}$ converge to the same optimal solution of the original OT problem.

To characterize the last-iterate rate, define $\xi_k := f(\widetilde{X}^{k+1}) - f(\widetilde{X}^k)$. Specializing [53, Eq. (3.5)] gives

$$\xi_k \le \eta(\mu_{k-1} + \mu_k), \tag{23}$$

where $\mu_{-1} := 0$. Since the proximal parameter is constant, the coefficient in their rate estimate satisfies $\sigma_{k-1} = k/\eta$. Thus,

$$\sum_{k=0}^{K-1} \sigma_{k-1}\xi_k \le \sum_{k=1}^{K-1} k(\mu_{k-1} + \mu_k) \le 2\sum_{k=0}^{K-1} (k+1)\mu_k. \tag{24}$$

Hence, the last-iterate rate is determined by the growth of the weighted inexactness sequence.

For $\mu_k = O((k+1)^{-2})$, we have $\sum_k \mu_k < \infty$, so sequence convergence holds, while

$$\sum_{k=0}^{K-1} (k+1)\mu_k = O(\log K). \tag{25}$$

[53, Theorem 3.1] therefore yields

$$f(\widetilde{X}^K) - f^\star = O\left(\frac{\eta \log K}{K}\right). \tag{26}$$

In contrast, if $\mu_k = O((k+1)^{-p})$ with $p > 2$, then

$$\sum_{k=0}^{\infty} (k+1)\mu_k < \infty, \tag{27}$$

and the weighted error term remains uniformly bounded. Consequently,

$$f(\widetilde{X}^K) - f^\star = O\left(\frac{\eta}{K}\right). \tag{28}$$

Thus, $p = 2$ is sufficient for global sequence convergence but introduces a logarithmic factor in the general last-iterate bound, whereas $p > 2$ recovers the standard $O(\eta/K)$ rate.

D Additional Experiment Details

D.1 Derivation of relative KKT residual Δ_{kkt}

One can show that the dual problem of (1) is

$$\max_{\boldsymbol{\gamma}\in\mathbb{R}^n, \boldsymbol{\zeta}\in\mathbb{R}^m} \boldsymbol{a}^\top\boldsymbol{\zeta} + \boldsymbol{b}^\top\boldsymbol{\gamma} \quad \text{s.t.} \quad U(\boldsymbol{\gamma},\boldsymbol{\zeta}) := C - \boldsymbol{\zeta}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\gamma}^\top \ge 0 \tag{29}$$

with the KKT system

$$X\mathbf{1}_n = \boldsymbol{a}, \quad X^\top\mathbf{1}_m = \boldsymbol{b}, \quad \langle X, U(\boldsymbol{\gamma},\boldsymbol{\zeta})\rangle = 0, \quad X \ge 0, \quad U(\boldsymbol{\gamma},\boldsymbol{\zeta}) \ge 0. \tag{30}$$

Based on (30), the relative KKT residual Δ_{kkt} is defined as:

$$\Delta_{\text{kkt}} := \max\{\Delta_p, \Delta_d, \Delta_c\}, \tag{31}$$

where $\Delta_p := \max\left\{\frac{\|X\mathbf{1}_n - \boldsymbol{a}\|}{1+\|\boldsymbol{a}\|}, \frac{\|X^\top\mathbf{1}_m - \boldsymbol{b}\|}{1+\|\boldsymbol{b}\|}, \frac{\|\min\{X,0\}\|_F}{1+\|X\|_F}\right\}$, $\Delta_d := \frac{\|\min\{U(\boldsymbol{\gamma},\boldsymbol{\zeta}),0\}\|_F}{1+\|C\|_F}$, $\Delta_c := \frac{|\langle X, U(\boldsymbol{\gamma},\boldsymbol{\zeta})\rangle|}{1+\|C\|_F}$, and $\|\cdot\|_F$ is the Frobenius norm of a matrix. Clearly, $\Delta_{\text{kkt}} = 0$ if and only if $(X, \boldsymbol{\gamma}, \boldsymbol{\zeta})$ satisfies the KKT system (30), making Δ_{kkt} a natural measure of optimality. In our semi-dual formulation (6), we compute $\boldsymbol{\zeta}$ via the *c-transform*: $\zeta_i = \min_{j\in[n]}(C_{ij} - \gamma_j)\ \forall i \in [m]$.

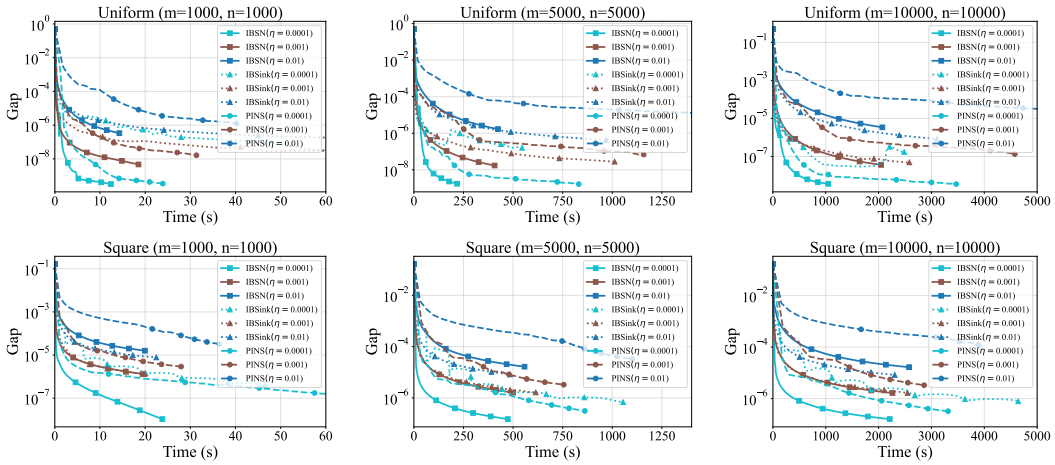

Figure 5: Impact of the regularization parameter η on IBSN, IBSink and PINS on synthetic data. Top: Uniform setting. Bottom: Square setting.

Table 1: Performance comparison between IBSN and IBN (IBSN without sparsity process during the Newton stage).

Size $m = n = 1000$									
Synthetic: Uniform					Synthetic: Square				
Alg	NIte	CGTime (s)	Time (s)	Gap	Alg	NIte	CGTime (s)	Time (s)	Gap
IBN	177	28.38	33.47	3.18e-10	IBN	301	61.55	72.03	1.07e-08
IBSN	176	**3.86**	**11.04**	3.19e-10	IBSN	301	**11.13**	**24.68**	1.07e-08
Size $m = n = 5000$									
Synthetic: Uniform					Synthetic: Square				
Alg	NIte	CGTime (s)	Time (s)	Gap	Alg	NIte	CGTime (s)	Time (s)	Gap
IBN	174	497.19	659.57	1.70e-09	IBN	301	803.53	1101.43	1.49e-07
IBSN	174	**9.69**	**219.22**	1.70e-09	IBSN	301	**74.37**	**457.10**	1.49e-07
Size $m = n = 10000$									
Synthetic: Uniform					Synthetic: Square				
Alg	NIte	CGTime (s)	Time (s)	Gap	Alg	NIte	CGTime (s)	Time (s)	Gap
IBN	167	1688.99	2522.38	3.65e-09	IBN	300	2654.52	4303.08	1.59e-07
IBSN	170	**16.82**	**921.60**	3.63e-09	IBSN	300	**267.15**	**2205.74**	1.59e-07

Bold values indicate the most favorable results. The algorithm terminates when $\Delta_{\text{kkt}} < 10^{-11}$ or after 300 outer iterations. "NIte", "CGTime" and "Time" represent the total number of Newton iterations, total runtime for finding the Newton directions, and total running time for the algorithm, respectively.

D.2 Impact of the regularization parameter η

In this section, we investigate the effect of the regularization parameter η on the performance of the IBSN, PINS, and IBSink algorithms under the synthetic settings.

The results, shown in Figure 5, demonstrate that IBSN consistently outperforms PINS across all configurations, achieving faster convergence and smaller objective gaps for the same η. In addition, as η increases, both methods exhibit slower convergence, reflecting the effect of stronger regularization.

D.3 Impact of the sparsity in the Newton phase

To assess the effectiveness of the sparsity process introduced in the Newton phase, we compare the proposed IBSN algorithm with its non-sparse counterpart, IBN (IBSN without sparsity during the

Table 2: Comparison between IBSN and Network Simplex using C++ implementations. Bold values indicate the lower runtime.

		IBSN		Network Simplex	
Instance	(m, n)	Gap	Time (s)	Gap	Time (s)
Synthetic: Uniform					
Uniform	$(1000, 1000)$	8.690e-10	**2.459**	8.092e-11	3.319
Uniform	$(2000, 1000)$	1.431e-07	**3.584**	1.435e-07	5.117
Uniform	$(4000, 1000)$	3.746e-10	**7.160**	1.884e-10	8.576
Uniform	$(5000, 1000)$	5.609e-10	**8.605**	3.532e-11	10.055
Uniform	$(5000, 500)$	9.425e-10	**4.184**	5.765e-11	6.257
Uniform	$(5000, 5000)$	5.628e-07	49.00	5.644e-11	**43.44**
Real: Fashion-MNIST					
IDs 0–1	$(784, 784)$	1.342e-09	**1.05**	1.302e-09	1.20
IDs 22–54698	$(784, 784)$	7.571e-10	1.04	5.362e-10	**0.71**
IDs 333–12580	$(784, 784)$	1.580e-09	1.33	1.180e-16	**0.62**
Real: DOTmark, balanced case ($m = n$)					
Shapes64	$(4096, 4096)$	5.890e-06	23.71	3.844e-10	**14.83**
ClassicImages64	$(4096, 4096)$	4.315e-06	**24.31**	4.016e-10	36.37
WhiteNoise64	$(4096, 4096)$	4.141e-06	19.15	3.242e-16	**11.11**
Real: DOTmark, unbalanced case ($m \neq n$)					
Shapes	$(4096, 1024)$	4.546e-09	**6.70**	1.250e-09	10.88
ClassicImages	$(4096, 1024)$	5.517e-09	**12.44**	4.945e-10	17.44
WhiteNoise	$(4096, 1024)$	5.403e-10	**13.31**	3.609e-16	14.70

Newton stage). Both methods are evaluated on the Uniform and Square synthetic cost settings with problem sizes $m = n \in \{1000, 5000, 10000\}$. The results are given in Table 1.

As shown in the table, introducing sparsity in the Hessian computation leads to a substantial reduction in the computational cost of solving the Newton systems while achieving identical final gaps. In addition, the benefit of sparsity becomes more pronounced in finding a Newton direction as the problem size increases. We also observe that the total runtime advantage decreases as problems get larger. This is primarily because other computational components, such as line search and the evaluation of the stopping criterion based on $D_\phi(\mathcal{P}_\Omega(X^{k+1}), X^{k+1})$, begin to dominate the overall cost. Nevertheless, the significant reduction in the time required for Newton direction computation clearly demonstrates the effectiveness of the proposed sparsity process in accelerating the Newton phase.

D.4 Comparison with Network Simplex

We further compare IBSN with a classical network simplex solver for exact OT. For this comparison, we implement IBSN in C++ and use the network simplex solver [11] available through the Python Optimal Transport (POT) package [22, 23].

Table 2 reports the final objective gap and running time. Neither method uniformly dominates the other. Network Simplex is faster on several balanced instances and often achieves smaller final objective gap. In contrast, IBSN demonstrates competitive performance and is faster on all three tested unbalanced DOTmark instances with $m = 4096, n = 1024$.

These results indicate that the relative performance of the two methods depends on the problem dimensions and cost structure. In particular, the semi-dual formulation of IBSN reduces the Newton system dimension from $m + n$ to n, which can be advantageous when m is substantially larger than n. Overall, the results demonstrate that Network Simplex remains highly competitive, while IBSN can offer computational advantages on certain unbalanced OT instances.

D.5 Impact of the semi-dual formulation on Newton direction computation

In this subsection, we investigate the impact of the semi-dual formulation on the efficiency of computing the Newton direction. To do this, we compare the proposed IBSN algorithm with three

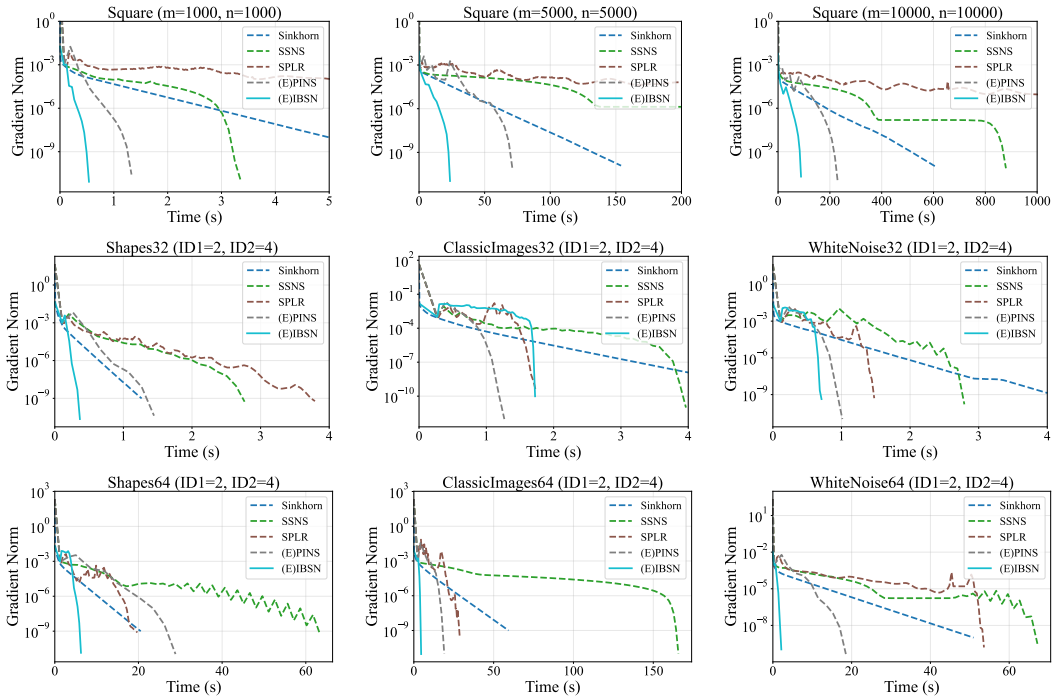

Figure 6: Performance of different algorithms on EOT problem (2). Top: Square setting. Middle: DOTmark with image size 32×32. Bottom: DOTmark with image size 64×64.

second-order methods, PINS, SSNS [48], and SPLR [49]. The baselines use the standard dual formulation with two dual variables, while IBSN works in the semi-dual space. We also use a shared warm start: run the Sinkhorn algorithm until the gradient norm is below 10^{-3}, then switch to the Newton-type phase. This reduces sensitivity to initialization and makes the comparison focus on the Newton step. In addition, we also provide the Sinkhorn algorithm as a reference.

Since PINS and IBSN are originally designed to solve the exact OT problem (which involves a sequence of subproblems), we adapted them for this specific experiment. In both frameworks, each subproblem can be regarded as an EOT problem with a modified cost matrix $C - \eta \log(X^k)$. To ensure they solve the exact same EOT problem as the other methods, we fix $X^k = \mathbf{1}_m \mathbf{1}_n^\top$ and limit the algorithms to solve the subproblem just once, rather than iteratively. This modification ensures that all algorithms address the same EOT problem with the original cost matrix C. We refer to these adapted versions as (E)PINS and (E)IBSN. For completeness, we also include their non-sparse counterparts, denoted by (E)PINS(NS) and (E)IBN, respectively.

In the following experiments, we evaluate the impact of the semi-dual formulation across two distinct scenarios: (i) the balanced setting where $m = n$, and (ii) the unbalanced setting where $m > n$. The algorithm terminates when the gradient norm is less than 10^{-9} or when the maximum number of iterations is reached.

Performance on balanced $m = n$. We first evaluate performance using the synthetic Square setting and the real DOTmark dataset, with the regularization parameter fixed at $\eta = 10^{-4}$. As shown in Figure 6, our proposed algorithm (E)IBSN performs well with both synthetic and real data. In addition, second-order methods typically exhibit a faster convergence rate than the Sinkhorn algorithm once they enter the Newton regime. Table 3 presents more details about the cost of the Newton step. Across all experimental settings, (E)IBSN and (E)IBN require the fewest CG iterations, typically around half the number required by the other second-order methods. This improvement is mainly due to the semi-dual formulation, which reduces the dimension of the dual variable and leads to a smaller and simpler linear system. In addition, the CG iteration counts of (E)IBSN and its non-sparse counterpart (E)IBN are close, suggesting that the proposed sparsification strategy does not noticeably increase the difficulty of solving the Newton system.

Table 3: Performance of different algorithms in Newton direction computation under EOT problem (2).

Synthetic: Square									
	size $m = n = 1000$			size $m = n = 5000$			size $m = n = 10000$		
Alg	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)
SSNS	305.8 (69)	34	3.36	712.4 (92)	116	373.37	749.4 (135)	61	879.52
SPLR	- (-)	-	20.44	- (-)	-	554.80	- (-)	-	2068.48
(E)PINS	292.8 (49)	14	1.33	418.0 (143)	22	71.24	425.5 (173)	17	229.77
(E)PINS(NS)	292.8 (49)	14	3.12	418.3 (143)	22	178.41	441.7 (166)	16	580.58
(E)IBN	167.2 (12)	**10**	1.42	**290.4** (88)	13	73.49	241.3 (97)	**13**	274.29
(E)IBSN	**163.6** (16)	11	**0.54**	**290.4** (88)	**13**	**23.64**	**241.2** (97)	**13**	**88.63**
Real: DOTmark (ID1 = 2, ID2 = 4) with size $m = n = 1024$									
	Shapes32			ClassicImages32			WhiteNoise32		
Alg	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)
SSNS	1071.3 (383)	23	2.77	371.9 (89)	40	3.97	346.3 (129)	29	2.79
SPLR	694.9 (528)	80	3.79	210.3 (144)	50	1.74	233.3 (110)	47	1.48
(E)PINS	763.2 (251)	9	1.45	318.6 (67)	**11**	**1.27**	317.7 (76)	**10**	1.01
(E)PINS(NS)	771.9 (263)	9	2.91	327.9 (54)	**10**	1.86	317.7 (76)	**10**	1.67
(E)IBN	**359.1** (77)	**8**	1.15	164.1 (34)	38	3.49	166.4 (26)	19	1.91
(E)IBSN	**360.3** (79)	**8**	**0.37**	**163.7** (27)	43	1.72	**164.6** (34)	20	**0.71**
Real: DOTmark (ID1 = 2, ID2 = 4) with size $m = n = 4096$									
	Shapes64			ClassicImages64			WhiteNoise64		
Alg	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)	MeanCG	Ite	Time (s)
SSNS	770.4 (190)	43	63.24	275.5 (99)	113	165.73	396.4 (87)	46	67.17
SPLR	509.4 (205)	46	19.73	151.0 (110)	71	28.75	219.8 (76)	123	53.60
(E)PINS	617.9 (206)	**12**	28.85	265.1 (57)	11	19.05	324.5 (78)	11	18.53
(E)PINS(NS)	617.9 (206)	**12**	55.95	265.1 (57)	11	27.53	324.5 (78)	11	30.70
(E)IBN	236.1 (113)	14	24.84	**162.3** (27)	**7**	10.86	**195.2** (25)	**6**	10.00
(E)IBSN	**236.0** (113)	14	**6.26**	**162.3** (27)	**7**	**4.58**	**195.2** (25)	**6**	**2.14**

Bold values indicate the most favorable results. "MeanCG" denotes the mean number (standard deviation) of CG iterations in finding a Newton direction, and "Ite" denotes the total number of iterations. "-" indicates the algorithm attains its maximum number of iterations, and its gradient norm is larger than 10^{-9}.

Performance on unbalanced $m > n$. Next, we investigate the performance of the algorithms in unbalanced settings where $m > n$. To do this, we conduct experiments on a spherical transport problem between supply and demand locations, inspired by [2]. The cost matrix $C \in \mathbb{R}^{m\times n}$ is given by the spherical distance, i.e., $C_{ij} = \arccos(\langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle)$, where $\boldsymbol{x}_i, \boldsymbol{y}_j \in \mathbb{S} := \{\boldsymbol{z} \in \mathbb{R}^3 : \|\boldsymbol{z}\| = 1\}$. For the data generation, the demand locations $\{\boldsymbol{x}_i\}_{i=1}^m$ from Earth's population density data [20], while the supply locations $\{\boldsymbol{y}_j\}_{j=1}^n$ are sampled uniformly from Earth's landmass. The marginal weights $\boldsymbol{a}$ and $\boldsymbol{b}$ are sampled independently from a uniform distribution on $(0, 1)$ and normalized such that $\mathbf{1}_m^\top \boldsymbol{a} = \mathbf{1}_n^\top \boldsymbol{b} = 1$. We set the regularization parameter to $\eta = 10^{-3}$.

Figure 7 demonstrates that (E)IBSN requires the fewest mean number of CG iterations to find a Newton direction and consistently outperforms among all algorithms as the source dimension m increases. This efficiency is attributed to our semi-dual formulation, which restricts the Hessian matrix to a low dimension regardless of the growth in the source dimension m.

D.6 Impact of the sparsification choices on EOT problem (2)

In this section, we conduct an ablation study to assess the effect of different Hessian sparsification strategies for solving the EOT problem (2). In particular, we compare the proposed threshold-based strategy with a simple magnitude-based sorting strategy.

- **Adaptive thresholding (ours):** We set the threshold as $\rho = \frac{\eta}{mn}\|\boldsymbol{g}\|$ and apply Algorithm 1 to control the sparsity level of P_ρ. The threshold is therefore adjusted dynamically according to the current optimality residual.

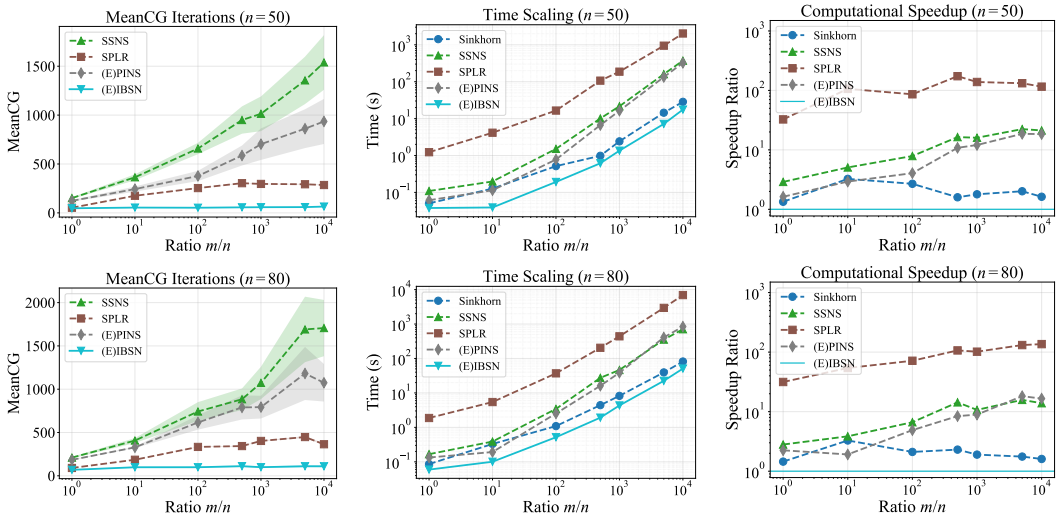

Figure 7: Scalability analysis on highly unbalanced spherical OT problems. The target dimension is fixed at $n = 50$ (top) and $n = 80$ (bottom). Left: Mean number of CG iterations required to compute a Newton direction, with shaded regions representing the standard error. Middle: Total running time in log-log scale. Right: Speedup ratio of (E)IBSN relative to other algorithms. A value of $c > 1$ indicates that (E)IBSN is c times faster than the competing algorithm.

- q/n**-sort (baseline):** This strategy uses a prescribed hyperparameter q to control the sparsity level of P_ρ. At each iteration, all entries of P are sorted by magnitude, and only the largest q entries are retained. The resulting matrix is then renormalized to obtain P_ρ, from which the sparse Hessian H_ρ is constructed.

We evaluate both strategies on synthetic EOT instances with $m = n$ and fix the regularization parameter at $\eta = 10^{-4}$. For the sorting baseline, we consider the predefined capacities $q \in \{500, 750, 1000\}$.

As shown in Figure 8, the proposed adaptive thresholding strategy maintains a stable proportion of nonzero entries across different problem dimensions. In contrast, the nonzero proportion produced by the sorting baseline decreases rapidly as n increases. Although this aggressive sparsification can reduce the worst-case cost of the CG solver, it also leads to a less accurate Hessian approximation and may introduce numerical instability. As a result, the sorting-based strategy often fails to reach the target tolerance of 10^{-9} for larger problem sizes.

Another limitation of the sorting baseline is its dependence on the capacity parameter q. In practice, the best choice of q varies with the problem size and the structure of the cost matrix, making the method sensitive to hyperparameter tuning. By contrast, the proposed adaptive thresholding rule adjusts automatically to the current residual, problem dimension, and regularization level, thereby providing a more robust and tuning-free sparsification mechanism.

D.7 More comparisons on EOT problem (2)

In this subsection, we extend our comparison to include the following methods, all implemented using the POT package [22, 23]:

- S2EOT: Solve the EOT problem using sparsity-constrained regularization [29] (sparsity is chosen as 2).
- LogSink: Solve the EOT problem using Sinkhorn-Knopp matrix scaling algorithm with the log stabilization [41].
- GCG: Solve the OT problem using ℓ_2 and entropic regularization with the generalized conditional gradient algorithm [15, 39].
- L-BFGS-B [8, 17]: Solve the EOT problem (semi-dual formulation) using L-BFGS-B.

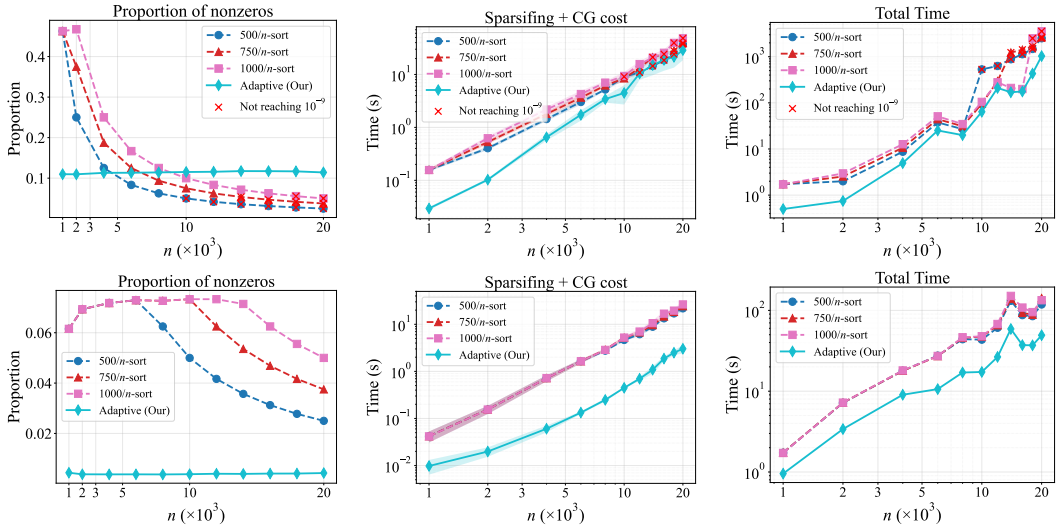

Figure 8: Performance comparison of different sparsification strategies on EOT problems (2) under the Square setting (top) and Uniform setting (bottom). From left to right, the subplots evaluate three key metrics: the proportion of non-zero elements retained in P_ρ, the combined computational cost of sparsification and the CG solver to find a Newton direction, and the total time required to reach a convergence tolerance of 10^{-9}. Instances where a strategy fails to converge are marked with a $\times$.

All regularization parameters are set to be 10^{-4}. Table 4 presents the results. Our proposed (E)IBSN algorithm consistently outperforms these baselines on both synthetic and real datasets, achieving significantly lower gradient norms in less computational time. These results demonstrate that (E)IBSN is a highly effective and robust solver for EOT problems (2).

E Color transfer

Color transfer aims to modify the color distribution of a source image to match the color palette of a target image while preserving the source image's structural content. Specifically, by interpreting image colors as probability distributions in a suitable color space, the optimal transport framework provides a method for transferring color characteristics between images.

Following the experimental protocol in [54], we conduct the color transfer experiments in the CIE-Lab color space. We apply the K-means clustering algorithm with $K = 512$ to quantize each image's color space and use the resulting cluster centers and their normalized frequencies to construct discrete color distributions. The proposed IBSN algorithm is then employed to compute the optimal transport plan between these two distributions, and the resulting mapping is used to transfer the target's color style onto the source image. The results of color transfer for selected image pairs can be found in Figure 9.

Table 4: Comparison of different algorithms in the POT package when solving the EOT problem (2).

Synthetic: Square								
size $m = n = 1000$			size $m = n = 5000$			size $m = n = 10000$		
Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)
S2EOT	2.61e-02	5.65	S2EOT	9.57e-03	209.74	S2EOT	5.88e-03	833.64
LogSink	5.14e-02	25.49	LogSink	2.31e-02	774.38	LogSink	1.63e-02	2530.27
GCG	1.37e-04	2.33	GCG	1.43e-04	64.16	GCG	3.75e-05	217.92
L-BFGS-B	2.75e-04	0.93	L-BFGS-B	3.99e-05	39.79	L-BFGS-B	3.14e-05	100.43
(E)IBSN	**8.44e-12**	**0.55**	(E)IBSN	**1.08e-11**	**22.89**	(E)IBSN	**1.92e-11**	**90.91**
Real: DOTmark (ID1 = 2, ID2 = 4) with size $m = n = 1024$								
Shapes32			ClassicImages32			WhiteNoise32		
Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)
S2EOT	8.02e-01	14.34	S2EOT	2.62e-01	10.75	S2EOT	7.74e-02	9.99
LogSink	6.20e-02	28.92	LogSink	4.75e-02	27.41	LogSink	5.09e-02	27.04
GCG	4.56e-06	2.15	GCG	2.43e-04	2.20	GCG	1.89e-04	2.17
L-BFGS-B	4.46e-04	3.10	L-BFGS-B	2.34e-06	1.88	L-BFGS-B	2.65e-06	2.73
(E)IBSN	**2.25e-11**	**0.37**	(E)IBSN	**9.72e-11**	**1.72**	(E)IBSN	**4.30e-10**	**0.71**
Real: DOTmark (ID1 = 2, ID2 = 4) with size $m = n = 4096$								
Shapes64			ClassicImages64			WhiteNoise64		
Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)	Alg	GradNorm	Time (s)
S2EOT	3.19e-01	276.25	S2EOT	8.48e-02	219.66	S2EOT	2.68e-02	199.39
LogSink	3.12e-02	439.69	LogSink	2.40e-02	440.81	LogSink	2.55e-02	435.39
GCG	4.03e-06	34.64	GCG	1.22e-04	41.49	GCG	4.09e-05	40.48
L-BFGS-B	3.75e-07	272.54	L-BFGS-B	3.28e-07	28.75	L-BFGS-B	5.69e-07	49.35
(E)IBSN	**1.39e-11**	**6.26**	(E)IBSN	**9.19e-12**	**4.58**	(E)IBSN	**1.08e-10**	**2.14**

Bold values indicate the most favorable results. The algorithm terminates when the gradient norm (GradNorm) is less than 10^{-9} or when the maximum number of iterations is reached.

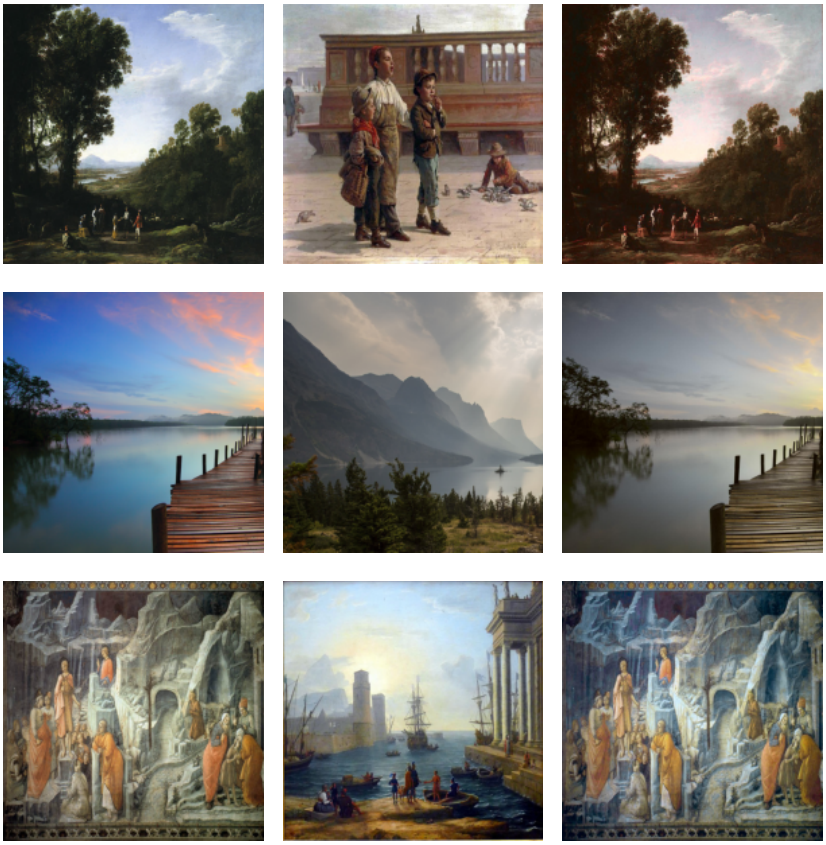

Figure 9: Color transfer results obtained using the IBSN algorithm. Each triplet shows the source image (left), the target image (middle), and the resulting image after applying color transfer (right).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have provided detailed derivations and numerical experiments to support the claims made in the abstract.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have mentioned in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We listed the assumptions of the theoretical results and provided the proof in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the code to reproduce the experiments in the article.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets used are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The detailed parameter settings are reflected in the article, Appendix, as well as the programming code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The experiments focus on deterministic optimization convergence and runtime comparisons under fixed problem instances, rather than statistical estimation or predictive performance across random trials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the details in the article.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work develops numerical optimization algorithms for OT and EOT computation and uses standard synthetic and publicly available benchmark datasets. It does not involve human subjects, sensitive personal data, or high-risk deployment settings.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This article is mostly a theoretical and computational work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The article poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have listed the original papers that provide the datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The article does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The article does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Our core methodology and method development do not involve LLMs. They were strictly used for writing and language polishing purposes only.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.