A PROXIMAL-SINKHORN-NEWTON METHOD FOR ENTROPIC OPTIMAL TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

Entropic optimal transport (OT) enables efficient distribution alignment via the Sinkhorn method, but suffers from numerical instability and slow convergence under weak entropic regularization. To address these challenges, we first employ an inexact proximal point method to decompose entropic OT into simpler subproblems, yielding numerically stable approximate solutions. We then introduce a dynamic Newton method with global convergence and a locally quadratic convergence rate to refine these solutions. We also establish, to the best of our knowledge, the first complexity result for Newton-type methods. Under relatively heavy entropic regularization, Sinkhorn scaling can serve as an alternative to the Newton method, since numerical instability is handled in the first stage. The resulting Proximal-Sinkhorn-Newton method combines the strengths of proximal, Sinkhorn, and Newton approaches and outperforms baseline methods across a wide range of regularization strengths and accuracy requirements.

Keywords: Optimal transport, Proximal point method, Newton method, Sinkhorn scaling.

1 INTRODUCTION

Optimal transport (OT) seeks the best transportation plan from an ensemble of sources to targets (Linial et al., 1998; Peyré et al., 2017). It is an important task in machine learning, such as GAN (Arjovsky et al., 2017; Genevay et al., 2018), graph matching (Shen et al., 2024), domain adaptation (Nguyen et al., 2021; Turrisi et al., 2022), statistical learning (Oneto et al., 2020; Huynh et al., 2020), robust and efficient unsupervised learning (Lei et al., 2019; Onken et al., 2021), feature selection (Cao & Zhang, 2024; Cao et al., 2026), and so on. For an overview of further applications of optimal transport, readers are referred to (Peyré & Cuturi, 2019).

In this work, we focus on the entropic optimal transport problem of the form

$$P_\beta = \arg\min_{P\in\mathbb{U}} C\cdot P + \frac{1}{\beta}\mathcal{H}(P),\ \ \mathcal{H}(P) := P\cdot \mathring{\log} P, \tag{1}$$

where $P \in \mathbb{R}_+^{m\times n}$ represents a transport plan, $\mathbb{U} := \left\{P\in\mathbb{R}_+^{m\times n} : P\mathbf{1} = \boldsymbol{a}, P^\top\mathbf{1} = \boldsymbol{b}\right\}$ is the feasible region where $\boldsymbol{a}\in\mathbb{R}_{++}^m$ and $\boldsymbol{b}\in\mathbb{R}_{++}^n$ are the source and target distributions respectively (assume $n\geq m$), $C\in\mathbb{R}^{m\times n}$ is a cost matrix, $C\cdot P = \sum C_{ij}P_{ij}$ is the transport cost, $\beta>0$ is the regularization parameter, $\mathcal{H}(\cdot)$ is the negative entropy, and $\mathring{\log}$ denotes the element-wise logarithm operator i.e., $P\cdot\mathring{\log}P = \sum P_{ij}\log(P_{ij})$. It is well known (Cuturi, 2013) that the solution to (1) admits a closed-form characterization:

$$P_\beta = \Gamma_{\mathbb{U}}^{\mathrm{kl}}\left(\mathring{\exp}(-\beta C)\right) = D_{(\boldsymbol{u})}\ \mathring{\exp}(-\beta C)\, D_{(\boldsymbol{v})}, \tag{2}$$

where $\Gamma_{\mathbb{U}}^{\mathrm{kl}}(\cdot)$ denotes the projection onto the set $\mathbb{U}$ with respect to the Kullback–Leibler (KL) divergence (Peyré et al., 2017), $\mathring{\exp}(\cdot)$ represents the element-wise exponential operator, $D_{(\boldsymbol{u})}$ and $D_{(\boldsymbol{v})}$ are diagonal matrices constructed from scaling vectors $\boldsymbol{u}\in\mathbb{R}_{++}^m$ and $\boldsymbol{v}\in\mathbb{R}_{++}^n$. The two vectors can be computed via the Sinkhorn algorithm (Cuturi, 2013):

$$K = \exp(-\beta C),\ \boldsymbol{u}^{(l+1)} = \boldsymbol{a}\oslash(K\boldsymbol{v}^{(l)}),\quad \boldsymbol{v}^{(l+1)} = \boldsymbol{b}\oslash(K^\top\boldsymbol{u}^{(l+1)}). \tag{3}$$

where $\oslash$ is the element-wise division. (3) is called Sinkhorn scaling (Sinkhorn, 1964).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

According to (2), solving the entropic OT relies on two components: the kernel matrix K and the scaling vectors $\boldsymbol{u}, \boldsymbol{v}$. These introduce two intrinsic computational challenges for entropic OT with a large β: (a) numerical instability caused by division by zero, which occurs when K contains an entire row of zeros; and (b) the iterative computation of $\boldsymbol{u}$ and $\boldsymbol{v}$ with complexity tied to β. These challenges are inherent to the entropic OT and continue to motivate ongoing research.

Numerical instability. The log-Sinkhorn (Benamou et al., 2015) addresses the numerical instability by performing computations in log-space, but introduces considerable overhead. The *proximal point method* (POT) (Xie et al., 2020) generates a sequence $\{P^{(t)}\}_{t=1,2,\dots}$ to obtain the solution to OT by solving a series of entropic OT problems with small values of parameters $\beta^{(t)}$. To improve efficiency, the *inexact proximal point method* (IPOT) (Xie et al., 2020) computes the proximal operator approximately at each iteration, yet $\{\hat{P}^{(t)}\}_{t=1,2,\dots}$ still converges to the exact solution (shown in Fig. 5). However, the algorithm exhibits only linear convergence in OT and does not benefit from the efficiency of entropic OT, thus becoming potentially expensive for large-scale problems.

Computational complexity. The Sinkhorn algorithm enjoys an $\mathcal{O}(n^2)$ per-iteration cost, but due to its sub-linear convergence, the algorithm may require a prohibitively large number of iterations to attain a moderately high accuracy (Tang & Qiu, 2024). The Newton method has a local quadratic convergence rate, while even a single Newton step requires $\mathcal{O}(n^3)$ cost (Brauer et al., 2017). The *Sinkhorn-Newton-Sparse* (SNS) algorithm (Tang et al., 2024) sparsifies the Hessian matrix to reduce the computational cost to $\mathcal{O}(n^2)$. However, its warm start may require hundreds of Sinkhorn iterations (as shown in Fig. 5 in Appendix B). Tang and Qiu (Tang & Qiu, 2024) propose an SSNS to overcome this problem, whereas the objective value may experience a period of stagnation during the Newton iteration. Moreover, the complexity of such Newton-type methods for entropic OT remains unknown.

In light of the above challenges, a natural question arises:

> How to design an entropic optimal transport algorithm that maintains stability and efficiency under varying β and desired accuracies?

A promising approach is to weave complementary algorithms into a framework that leverages their respective strengths. Specifically, we employ the inexact proximal point method to stably construct an approximate transport plan. If this plan does not meet the desired accuracy, it can serve as a high-quality initializer for faster secondary solvers. Such an idea is supported by the following theorem.

Theorem 1 (Inexact-to-Exact Transition)**.** *If $\hat{P}^{(t)}$ denotes the transport plan updated at the t-th iteration in the inexact proximal point method with a fixed proximal parameter $\Delta\beta$, then*

$$P_\beta = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}^{(t)}), \text{ where } \beta = t\Delta\beta. \tag{4}$$

This theorem establishes a fundamental connection between the inexact proximal point method and the entropic OT. Specifically, it allows the inexact iterate $\hat{P}^{(t)}$ to be refined through the KL-projection into the exact solution of the entropic OT problem. Interpreting $\hat{P}^{(t)}$ as a processed kernel matrix reveals two distinct benefits over the naive counterpart K: improved numerical robustness and reduced distance to the solution P_β (shown in Fig. 1). These properties make it a more desirable input for subsequent refinement algorithms such as the Sinkhorn scaling or the Newton method.

The contributions of this paper are threefold:

- **Numerical instability addressed via an inexact proximal point framework** that decomposes the entropic OT problem into a sequence of simpler subproblems. This yields an inexact proximal point method for entropic OT (IP-EOT), which stably and efficiently produces approximate solutions for subsequent refinement.
- **Computational complexity reduced via a dynamic Newton method** with global and fast local convergence by adaptively switching to robust Sinkhorn iterations (3). Sparse Hessian approximation and incomplete Cholesky–preconditioned conjugate gradients reduce computational cost. We also provide, to the best of our knowledge, the first complexity analysis of Newton-type methods for entropic OT.

- **Adaptive hybrid solver**. IP-EOT serves as either a standalone solver or a warm start for Sinkhorn and Newton methods, mitigating Sinkhorn instability and costly Newton initialization. An adaptive Hessian-sparsity criterion selects Newton for sparse Hessians and Sinkhorn for denser ones.

In short, we address two fundamental challenges in entropic optimal transport—numerical instability and high computational complexity—by introducing a unified inexact-to-exact framework, termed Proximal-Sinkhorn-Newton (PSN), which systematically integrates complementary algorithms. The overall architecture of PSN is illustrated in Fig. 1.

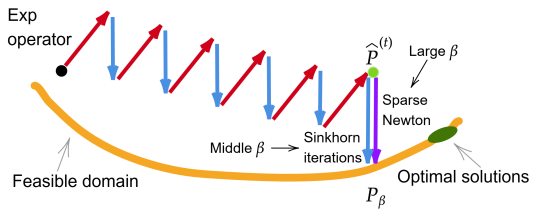


Figure 1: The trajectory of PSN consists of two stages: (a) IP-EOT for an efficient and stable approximation $\hat{P}^{(t)}$, and (b) KL-projection by either Sinkhorn scaling or the Newton method.

2 Proximal Point Methods for Entropic Optimal Transport

The proximal point method transforms complex optimization problems into a sequence of simpler subproblems by introducing a proximal operator (Rockafellar, 1976). Based on the Bregman distances, (Chen & Teboulle, 1993) formalized the Bregman proximal point algorithm and established its convergence. Recently, this method has been applied to OT, substantially reducing computational complexity (Xie et al., 2020; Chu et al., 2023; Chizat, 2024). The unclear trajectory of the regularization parameter during iteration has hindered use of the proximal point method in entropic OT, which our work seeks to address.

We propose an Inexact Proximal point method for Entropic OT (IP-EOT) that yields an approximation $\hat{P}_\beta$ by explicitly incorporating β into the proximal step. This approximation can also be derived from an approximate KL-projection $\hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}(\mathring{\exp}(-\beta C))$, a typical instance of which is early termination of Sinkhorn scaling. Furthermore, we demonstrate that the resulting $\hat{P}_\beta$ can be refined into the exact solution P_β through an exact KL-projection.

2.1 Proximal point methods

Proximal point methods are iterative algorithms for convex optimization, well-suited for nonsmooth or composite objectives. Their core principle involves solving a regularized subproblem at each step. This process optimizes the objective while ensuring the stability of the iterative sequence. To solve the OT (without the entropic term), the proximal point method reads as follows:

$$P^{(t+1)} = \arg\min_{P\in\mathbb{U}} C\cdot P + \frac{1}{\beta^{(t+1)}}\mathcal{B}(P, P^{(t)}), \tag{5}$$

$\mathcal{B}(\cdot,\cdot)$ is the Bregman divergence used to measure the distance between two probability vectors/matrices (Benamou et al., 2015). This formula can be equivalently rewritten as:

$$P^{(t+1)} = \arg\min_{P\in\mathbb{U}} C^{(t)}\cdot P + \frac{1}{\beta^{(t+1)}}\mathcal{H}(P), \text{ where } C^{(t)} = C - \frac{1}{\beta^{(t+1)}}\mathring{\log}\, P^{(t)}. \tag{6}$$

The update can be computed by applying a KL-projection to a modified kernel matrix $\mathring{\exp}\left(-C^{(t)}\beta^{(t+1)}\right)$, yielding:

$$P^{(t+1)} = \Gamma^{\mathrm{kl}}_{\mathbb{U}}\left(\mathring{\exp}(-C\beta^{(t+1)})\odot P^{(t)}\right), \tag{7}$$

where $\odot$ is the element-wise product. As $t \to \infty$, $P^{(t)}$ converges to the optimal transport plan. To improve efficiency, (Xie et al., 2020) proposed the inexact proximal point method (IPOT), which performs only one Sinkhorn iteration to approximate the projection in (7). The resulting sequence $\{\hat{P}^{(t)}\}_{t=1,2,\dots}$ can efficiently converge to the optimal transport plan.

2.2 Inexact proximal point method for entropic OT

Theorem 1 enables the inexact proximal point method to efficiently produce an approximation $\hat{P}_\beta$. This theorem relies on explicitly tracking the regularization parameter β across proximal iterations, as detailed below.

Lemma 1 (**Explicit form of** β)**.** *The iterative schemes of the exact/inexact proximal point method for entropic OT admit the following explicit representation:*

$$P^{(t)} = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})), \quad \hat{P}^{(t)} = \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})). \tag{8}$$

$\bigodot$ represents element-wise multiplication of a series of matrices. In particular, when each $\beta^{(i)}$ is fixed as $\Delta\beta$, the regularization β satisfies $\beta = t\Delta\beta$, which yields the following explicit expression:

$$\hat{P}^{(t)} = \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}(\mathring{\exp}\,(-t\Delta\beta C)) = \hat{P}_\beta, \quad \beta = t\Delta\beta. \tag{9}$$

The next Lemma establishes a theoretical bridge between $\hat{P}_\beta$ and P_β.

Lemma 2 (From Inexact to Exact)**.** *Let $\hat{P}_\beta$ be an approximation of P_β of the form*

$$\hat{P}_\beta = D_{(\hat{\boldsymbol{u}})}\ \mathring{\exp}(-\beta C)\, D_{(\hat{\boldsymbol{v}})}, \tag{10}$$

where $\hat{\boldsymbol{u}} \in \mathbb{R}^m_{++}$ and $\hat{\boldsymbol{v}} \in \mathbb{R}^n_{++}$. Then

$$P_\beta = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta). \tag{11}$$

This result enables $\hat{P}_\beta$ to serve as a warm start for computing P_β via the KL-projection.

Building on the above results, we propose an IP-EOT to provide an approximation $\hat{P}_\beta$, as outlined in Algorithm 1. The algorithm avoids the risk of numerical underflow by dividing the total regularization β into ℓ steps (line 2) and progressively constructing the kernel matrix (line 6). Our matrices ΔK and $\hat{P}$ are less susceptible to numerical underflow than the Sinkhorn kernel matrix $\mathring{\exp}(-\beta C)$. This advantage becomes more pronounced with a larger β, leading to substantially improved numerical stability. Since the cost of each iteration is $\mathcal{O}(n^2)$, the computational complexity of the algorithm is $\mathcal{O}(\ell n^2)$.

Algorithm 1 IP-EOT

Require: Probability vectors $\{\boldsymbol{a}, \boldsymbol{b}\}$ with lengths m and n, C, entropic parameter β, and proximal parameter $\Delta\hat{\beta}, v$

1: $\ell \leftarrow \left\lceil \frac{\beta}{\Delta\hat{\beta}} \right\rceil$
2: $\Delta\beta \leftarrow \beta/\ell$
3: $\Delta K \leftarrow \mathring{\exp} -(\Delta\beta C)$
4: $\hat{P} \leftarrow \mathbf{1}_m \mathbf{1}_n^\top/(mn)$
5: **for** $t = 1, 2, \dots, \ell$
6: $\quad K \leftarrow \Delta K \odot \hat{P}$
7: $\quad \boldsymbol{u} \leftarrow \boldsymbol{a} \oslash (K\boldsymbol{v}), \boldsymbol{v} \leftarrow \boldsymbol{b} \oslash (K^\top \boldsymbol{u})$
8: $\quad \hat{P} \leftarrow \mathrm{diag}(\boldsymbol{u}) K\, \mathrm{diag}(\boldsymbol{v})$
9: **end**
10: **return** $\hat{P}$, $\boldsymbol{u}$, $\boldsymbol{v}$

Algorithm 2 Dynamic Newton Method

Require: probability vectors $\{\boldsymbol{a}, \boldsymbol{b}\}$, cost matrix C, β, ρ and marginal error threshold ρ_{th}, $\boldsymbol{u}$, $\boldsymbol{v}$

1: **while** $\rho > \rho_{th}$ **do**
2: $\quad$ Compute $H, \nabla f$, and ρ using (14)
3: $\quad \hat{H} \leftarrow \mathrm{Sparsify}(H, \rho)$
4: $\quad L \leftarrow \mathrm{IC}(\hat{H})$ $\quad \triangleright$ # $\hat{H} \approx LL^\top$
5: $\quad \Delta\boldsymbol{z} \leftarrow \mathrm{PCG}(\hat{H}, -\nabla f, L)$
6: $\quad$ **if** α is searched **then** $\boldsymbol{z} \leftarrow \boldsymbol{z} + \alpha\Delta\boldsymbol{z}$
7: $\quad$ **else** update $\boldsymbol{z}$ by Sinkhorn iterations
8: $\quad (\boldsymbol{x}, \boldsymbol{y}) \leftarrow \boldsymbol{z}$
9: $\quad P \leftarrow \mathring{\exp}(\beta\left(-C + \boldsymbol{x}\mathbf{1}^\top + \mathbf{1}\boldsymbol{y}^\top\right) - 1)$
10: $\quad$ Compute marginal error ρ
11: **end while**
12: **return** P, ρ

Remark 1. *Since the iteration number of IP-EOT is tied to $\Delta\beta$, the algorithm cannot adaptively stop. Thus, it should be augmented with alternative KL-projection methods (e.g., Sinkhorn iterations or Newton methods), when higher accuracy is needed.*

3 DYNAMIC NEWTON METHOD FOR ENTROPIC OPTIMAL TRANSPORT

The Newton method is based on the Lagrange function $\mathcal{L}$ associated with the entropic OT (1):

$$\mathcal{L}(P, \boldsymbol{x}, \boldsymbol{y}) := \frac{1}{\beta} P \cdot \mathring{\log} P + C \cdot P - \boldsymbol{x} \cdot (P\mathbf{1} - \boldsymbol{a}) - \boldsymbol{y} \cdot (P^\top \mathbf{1} - \boldsymbol{b}). \tag{12}$$

Using the primal-dual and strong duality theory (Nocedal & Wright, 1999), one only needs to minimize the function f to get the solution of (1):

$$f(\boldsymbol{x}, \boldsymbol{y}) = -\min_P \mathcal{L}(P, \boldsymbol{x}, \boldsymbol{y}). \tag{13}$$

Since the unreduced Hessian matrix of f has the null direction $(\mathbf{1}, -\mathbf{1})^\top$ and is singular, we remove this redundancy by fixing the gauge $y_n = 0$ (Tang & Qiu, 2024) and write $\boldsymbol{z} = (\boldsymbol{x}, \tilde{\boldsymbol{y}})$. Therefore, the corresponding gradient and reduced Hessian used by the Newton iteration are:

$$\nabla_{\boldsymbol{z}} f(\boldsymbol{z}) = \left(P\mathbf{1} - \boldsymbol{a}, \widetilde{P}^\top \mathbf{1} - \tilde{\boldsymbol{b}}\right), \quad H(\boldsymbol{z}) = \beta \begin{bmatrix} \operatorname{diag}(P\mathbf{1}) & \widetilde{P} \\ \widetilde{P}^\top & \operatorname{diag}(\widetilde{P}^\top \mathbf{1}) \end{bmatrix}. \tag{14}$$

Here, $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{y}}$ are the vectors $\boldsymbol{b}$ and $\boldsymbol{y}$ with their last element removed, and $\widetilde{P}$ is the matrix P with its last column removed.

Since $H(\boldsymbol{z}_k)$ is invertible after removing the redundancy, the Newton method updates $\boldsymbol{z}_{k+1}$ via the formula:

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \Delta \boldsymbol{z}_k, \;\; \Delta \boldsymbol{z}_k = H(\boldsymbol{z}_k)^{-1} \left(-\nabla_{\boldsymbol{z}} f(\boldsymbol{z}_k)\right). \tag{15}$$

To avoid costly matrix inversion, $\Delta \boldsymbol{z}$ is obtained by solving a linear system $H \Delta \boldsymbol{z} = -\nabla f$, which is commonly solved by the Conjugate Gradient (CG) method (Tang et al., 2024). However, a key challenge arises from the $\mathcal{O}(n^3)$ computational cost per Newton iteration and the singularity of the unreduced Hessian matrix.

3.1 PRECONDITIONER DESIGN AND SPARSIFICATION STRATEGY FOR NEWTON METHOD

Preconditioner for Newton Method. To enhance the efficiency of each Newton iteration, we employ an IC preconditioner to concentrate the eigenvalue distribution of the Hessian matrix, thereby reducing the number of CG iterations. This design leverages the fact that CG converges more rapidly when the eigenvalues of H are tightly clustered (Hestenes et al., 1952). The effectiveness of IC is demonstrated in Fig. 6, which compares it with the diagonal loading used in previous Newton-based algorithms for entropic OT (Brauer et al., 2017; Tang & Qiu, 2024).

Fast Sparse Scheme. Prior work exploits Hessian sparsity to accelerate Newton updates. Tang et al. (2024) reduced the per-iteration cost from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ via sparse Hessian approximation, while Tang & Qiu (2024) further preserved local quadratic convergence. However, their diagonal loading may distort second-order information and hinder advanced preconditioners such as incomplete Cholesky (IC). We therefore propose a preconditioner-friendly sparse Hessian approximation by sparsifying the anti-diagonal blocks P and $P^\top$:

$$\hat{H} = \beta \begin{bmatrix} \operatorname{diag}(P\mathbf{1}) & \widehat{P} \\ \widehat{P}^\top & \operatorname{diag}(\widetilde{P}^\top \mathbf{1}) \end{bmatrix}, \text{ where } \widehat{P}_{ij} = \widetilde{P}_{ij} \text{ for } \widetilde{P}_{ij} \geq \tau, \text{ and } 0 \text{ otherwise.} \tag{16}$$

where $\tau = \|\nabla f\|_2$ is chosen to preserve quadratic convergence (detailed in the proof of Theorem 3 in the appendix).

Accordingly, we can bound the eigenvalues of the Hessian matrix $H(\boldsymbol{z})$ as follows.[1]

Theorem 2 (Truncated Hessian Eigenvalue Lower Bound)**.** *Every eigenvalue λ of the truncated Hessian matrix $\{\hat{H}_k\}$ generated by (15) and fast sparse scheme satisfies*

$$\lambda \geq \left(n - \sqrt{n(n-1)}\right) \beta p^{\mathcal{L}}, \tag{17}$$

where $p^{\mathcal{L}}$ is defined by (140) in Lemma 4, which depends on $C, \beta, \boldsymbol{a}, \boldsymbol{b}$ and the initial point.

[1] For clarity, the spectral, convergence and complexity analyses in Section 3 to Section 4 and the corresponding appendices are presented for the square case $m = n$. The algorithm itself is applicable to the general rectangular case $m \times n$.

Applying the estimation of eigenvalue lower bound, we can derive the local quadratic convergence of the fast sparse scheme, as shown in the following theorem. In the following, we use $\mathcal{N}_\delta(\boldsymbol{w}_0)$ to denote all $\boldsymbol{w}$ satisfying $\|\boldsymbol{w} - \boldsymbol{w}_0\| \leq \delta$.

Theorem 3 (Local Quadratic Convergence). *Let the sequence $\{\boldsymbol{z}_k\}$ be generated by*

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \hat{H}_k^{-1} \nabla f(\boldsymbol{z}_k), \tag{18}$$

where $\hat{H}_k$ is sparsified from the Hessian matrix $H_k = H(\boldsymbol{z}_k)$ by the fast sparse scheme. The sequence $\{\boldsymbol{z}_k\}$ converges quadratically to the minimizer $\boldsymbol{z}^*$ in $\mathcal{N}_\delta(\boldsymbol{z}^*)$, where

$$\delta = \frac{e^{-1/e}}{6\beta} \cdot \frac{n - \sqrt{n(n-1)}p^{\mathcal{L}} e^{\beta \Delta C}}{\sqrt{2}\, n \left(a_{\max} b_{\max} + e^{\beta \Delta C} - 1\right) + s G_n e^{\beta \Delta C}}. \tag{19}$$

where $\Delta_C := \max_{i,j,k,l} \left(c_{ij} + c_{kl} - c_{il} - c_{kj}\right)$, $G_n = \sqrt{n(n-1)}$ and $s = \max(a_{\max}, b_{\max})$.

3.2 Dynamic Newton Method.

Let $f(\boldsymbol{z})$ be defined in (13). A backtracking line search selects $\alpha \in \{1, \hat{\alpha}, \hat{\alpha}^2, \ldots\}$, with $\hat{\alpha} \in (0, 1)$, to ensure descent. However, excessively small step sizes may cause prolonged stagnation; diagonal loading (Tang & Qiu, 2024) alleviates but does not fully eliminate this issue. We therefore introduce a Sinkhorn fallback: if no $\alpha \in \{\hat{\alpha}^0, \hat{\alpha}^1, \ldots, \hat{\alpha}^l\}$ satisfy Armijo and Wolfe condition, we perform several Sinkhorn iterations to obtain a direction $\boldsymbol{p}_k$ satisfying

$$f(\boldsymbol{z}_k + \boldsymbol{p}_k) < f(\boldsymbol{z}_k). \tag{20}$$

Newton updates resume once a satisfactory step size is found. This dynamic strategy combines the robustness of Sinkhorn scaling with the fast local convergence of Newton's method. Moreover, the Sinkhorn direction satisfies the Armijo and Wolfe conditions (shown in Lemma 8), ensuring sufficient descent and supporting the subsequent convergence analysis.

Algorithm 2 outlines the dynamic Newton method for entropic OT. In each iteration, the Hessian is sparsified (line 3) and preconditioned using an incomplete Cholesky factorization L (line 4). The main computational cost lies in computing $\Delta \boldsymbol{z}$, which requires $\mathcal{O}(n\|\hat{H}\|_0)$ operations, where $\|\hat{H}\|_0$ denotes the number of nonzero elements (line 5). The step size α is selected via a backtracking line search from $\{1, 0.5, 0.5^2, 0.5^3, 0.5^4\}$ (line 6). If none of these values satisfies Armijo and Wolfe condition, we apply several Sinkhorn iterations (typically 20 iterations suffice) for refinement (line 7). This adjustment helps Newton's method escape flat regions caused by vanishing step sizes, and is critical for establishing global convergence.

Theorem 4 (Global Convergence). *The sequence $\{\boldsymbol{z}_k\}$ generated by Algorithm 2 converges globally to the minimizer $\boldsymbol{z}^*$.*

4 Proximal-Sinkhorn-Newton method

This section presents the main algorithm, the Proximal-Sinkhorn-Newton (PSN), in Algorithm 3. The algorithm begins with the IP-EOT to produce an approximate solution $\hat{P}_\beta$. If the $\hat{P}_\beta$ fails to meet the predefined criteria, the algorithm refines the solution: if $\hat{H}$ contains fewer than λn non-zero elements, the dynamic Newton method is employed; otherwise, the Sinkhorn scaling. This approach is motivated by the relationship between sparsity and computational efficiency: Newton's method benefits from sparser matrices, which occur as β increases (Tang et al., 2024).

We next analyze the computational complexity of PSN in the Newton-refinement regime. The other two regimes are relatively straightforward: the proximal-only case is determined by the number of IP-EOT iterations, while the proximal-Sinkhorn case has a complexity order comparable to that of standard Sinkhorn scaling. For tractable analysis, we consider a conservative Sinkhorn–Newton procedure, where Sinkhorn scaling first drives the iterate into the local quadratic-convergence region, followed by Newton refinement. In practice, PSN is typically faster than this surrogate, since IP-EOT provides a more efficient warm start than Sinkhorn scaling, as observed in our experiments, while Newton iterations may enter their fast local convergence regime earlier than required by the sufficient theoretical condition. The resulting complexity bound is given below.

Theorem 5 (Global Convergence and Newton-Refinement Complexity)**.** *For entropic OT (1), Algorithm 3 converges globally. The conservative Sinkhorn–Newton procedure achieves a marginal tolerance below ρ_{th} with total computational complexity*

$$\mathcal{O}\left(n^2\left[\frac{\beta\|C\|_\infty}{\tilde{\lambda}\delta}+(1+\tau_{avg}+k_{CG}\tau_{avg})\ln\ln\left(\frac{2\beta e^{\sqrt{2}\delta\beta}s}{\rho_{th}}\right)\right]\right), \tag{21}$$

where $\tilde{\lambda}$ is a lower bound on the eigenvalues of $\{H_k\}$ estimated in Lemma 6 (Appendix D), δ is the quadratic-convergence radius in Theorem 3, τ_{avg} is the average sparsity, k_{CG} is the number of PCG iterations per Newton step, and $s=\max(\max(\boldsymbol{a}),\max(\boldsymbol{b}))$.

Algorithm 3 Proximal-Sinkhorn-Newton (PSN)

Require: Probability vectors $\{\boldsymbol{a},\boldsymbol{b}\}$, cost matrix C, entropic parameter β, proximal parameter $\Delta\hat{\beta}$, marginal error threshold ρ_{th}, target sparsity λ.
Ensure: The transport plan P for entropic OT with β.

$\hat{P}_\beta, u, v \leftarrow$ IP-EOT$(\boldsymbol{a},\boldsymbol{b},C,\beta,\Delta\hat{\beta})$ ▷ # *IP-EOT for approximation*
$\rho=\left(\|P\mathbf{1}-\boldsymbol{a}\|_2^2+\|\tilde{P}^\top\mathbf{1}-\tilde{\boldsymbol{b}}\|_2^2\right)^{1/2}$
if $\rho\geq\rho_{th}$ **then**
  $\hat{H}\leftarrow$ Sparsify Hessian$(\hat{P}_\beta,\rho)$ according to (16)
  if $\|\hat{H}\|_0<\lambda n$ **then**
    $P,\rho\leftarrow$ Dynamic Newton$(\boldsymbol{a},\boldsymbol{b},C,\beta,\rho,\rho_{th},u,v)$
  else
    Perform Sinkhorn iterations, updating P and ρ, until $\rho\leq\rho_{th}$
  end if
end if
return P

5 Numerical Experiments

We evaluate PSN and its main components through the following three questions:

- Q1. How does the performance of IP-EOT compare to the Sinkhorn algorithm?
- Q2. How does PSN perform across different problem sizes and regularization strengths?
- Q3. How scalable is PSN to large-scale problems?

The appendix provides supplementary experiments and detailed analyses, including: (i) a comparison of PSN with baselines over a wider range of β values (I); (ii) a benchmark of PSN and Proximal-Sinkhorn against first-order methods (J); (iii) variability results (P) and a running-time analysis (K, L); (iv) sensitivity to the proximal step size (M) and the switching parameter (N); and (v) an ablation on the IC preconditioner (H).

Datasets include random linear assignments, MNIST (OT between pairs of images), and ImageNet (OT between two categories of images). Compared with the baselines (Tang et al., 2024; Tang & Qiu, 2024), our method covers an approximately $10\times$ larger range of both β and problem sizes, enabling a more comprehensive comparison. The relevant details are summarized in Table 1.

Setting. In PSN, the proximal parameter $\Delta\hat{\beta}$ is set to 25 for general OT. This ensures numerical stability and leverages the rapid early convergence of IP-EOT, as the iteration count is $\beta/\Delta\hat{\beta}$. Given the faster convergence of the Sinkhorn method in linear assignment (Altschuler et al., 2017), we set $\Delta\hat{\beta}=50$ for this special case. The sparsity parameter λ is set to 35 for linear assignment and 70 for general OT. This design is informed by the theoretical sparsity of solutions to OT (m for linear assignment and $m+n-1$ for general OT) (Peyré et al., 2017) and the fact that solutions to entropic OT are typically denser (Tang et al., 2024).

Baselines include the Sparse-Newton-Sinkhorn method (SNS, ICLR) (Tang et al., 2024), the Safe and Sparse Newton Method (SSNS, NeurIPS) (Tang & Qiu, 2024), the Newton method for entropic

Table 1: Summary of the datasets and experimental settings.

Dataset	Dims	Cost	Marginals	β
Assign.	$[500, 10^4]$	Random entry-wise costs $C_{ij} \sim \text{Unif}[0,1]$	Uniform $a = b = n^{-1}\mathbf{1}_n$	$[300, 3000]$
MNIST	$[784, 10^5]$	Pixel-coordinate distances $C_{ij} = \|r_i - r_j\|_2^2$ $C_{ij} = \|r_i - r_j\|_1$	Normalized pixel intensities $a_i = I_i^{(1)}/\sum_k I_k^{(1)}, \quad b_j = I_j^{(2)}/\sum_k I_k^{(2)}$	$[300, 10^4]$
ImageNet	1000	Image-level feature distances $C_{ij} = \|x_i - y_j\|_2^2$ $C_{ij} = \|x_i - y_j\|_1$	Uniform $a = m^{-1}\mathbf{1}_m, \quad b = n^{-1}\mathbf{1}_n$	$[500, 2000]$

OT (Brauer et al., 2017), the log-Sinkhorn (Benamou et al., 2015), and the primal Sinkhorn (Cuturi, 2013) [2]. All methods were benchmarked in terms of runtime using MATLAB R2024b on an Intel Core i7-12800HX processor[3].

5.1 IP-EOT VS. LOG-SINKHORN.

Fig. 2 shows that the error of IP-EOT decreases much faster than that of log-Sinkhorn in the early iterations. Moreover, since the per-iteration cost of IP-EOT is less than that of log-Sinkhorn, its efficiency advantage in the early iterations becomes even more pronounced.

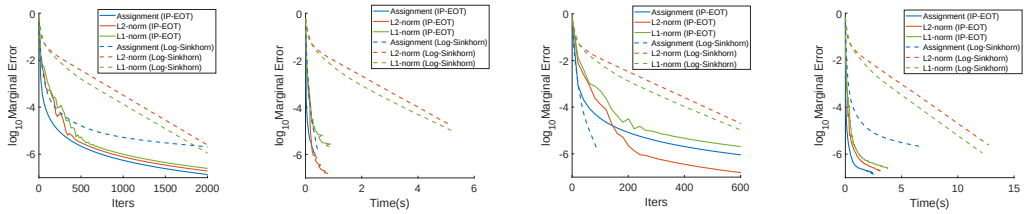


Figure 2: IP-EOT vs. log-Sinkhorn: $\beta = 500$ (first two subfigures) and $\beta = 1200$ (last two).

5.2 COMPARISON BETWEEN PSN AND BASELINES.

As illustrated in Fig. 3, PSN demonstrates strong performance across all settings. In (a)–(c), PSN mitigates the numerical overflow risk inherent in the Sinkhorn method through a few proximal steps, matching Sinkhorn's efficiency while remaining stable. Subfigures (f)–(h) show that for large β, PSN outperforms other algorithms, especially log-Sinkhorn. For problems with larger β, PSN is the only method that converges successfully across all cases (k)–(o). Further results on iterations and time-to-accuracy are included in Appendix I.

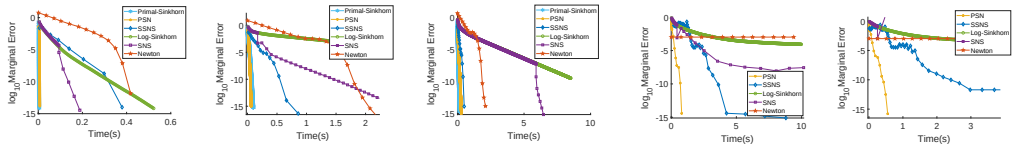
5.3 LARGER SCALE PROBLEMS.

We compare our algorithm against several baseline methods on larger-scale problems. We set $\beta = 200 \ln n$, as the distance between P_β and the optimal plan P_∞ scales with $\frac{\ln n}{\beta}$ (Altschuler et al., 2017). Fig. 4 (a) to (c) demonstrate clear and consistent performance advantages of PSN over the baselines. On the linear assignment dataset with $n = 10^4$, approaching the CPU memory limit, our method remains stable and robust.

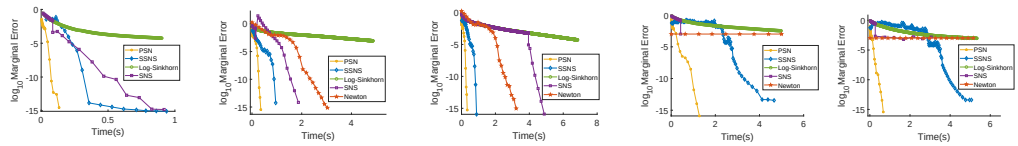
To further evaluate the scalability of the algorithm, we conducted experiments on MNIST L2-norm problems with $n = 10^5$ and $\beta = 10^4$. To address memory limitations, we implemented a scalable version of PSN with an $O(n)$ memory footprint. In this version, we avoid explicitly storing any matrices with $O(n^2)$ nonzeros, such as C, ΔK, K and $\hat{P}$ in Algorithm 1 ($\hat{H}$ with only $O(n)$ nonzeros can be stored directly). When necessary, these matrices are computed on the fly from relevant vectors and accessed in blocks. Although this approach introduces considerable additional computational

[2] when $\beta > 100$, the primal Sinkhorn may exhibit numerical instability (Appendix F of (Shen et al., 2024))

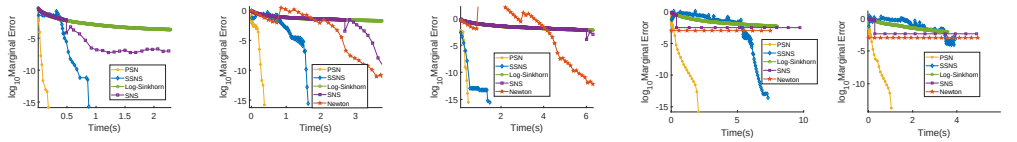
[3] For SSNS implemented in C++, its runtime was scaled to the MATLAB baseline by comparing the execution time of the Newton method in C++ and MATLAB.



(a) AS. (b) MI L2. (c) MI L1. (d) IM L2. (e) IM L1.

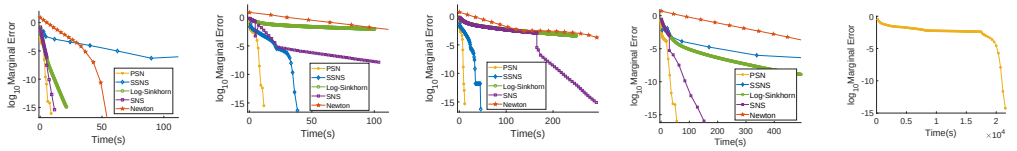


(f) AS. (g) MI L2. (h) MI L1. (i) IM L2. (j) IM L1.



(k) AS. (l) MI L2. (m) MI L1. (n) IM L2. (o) IM L1.

Figure 3: Comparison of PSN with baselines. From left to right, the columns correspond to Assignment (AS.), MNIST with L2 and L1 cost (MI L2., MI L1.), ImageNet with L2 and L1 cost (IM L2., IM L1.). The dimensions are $n = 500,\ 784,\ 784,\ 1000,\ 1000$, respectively. From top to bottom, we use $\beta = 300, 1200, 3000$ for Assignment and MNIST, and $\beta = 500, 1000, 2000$ for ImageNet.



(a) AS. (b) MI L2. (c) MI L1. (d) AS. (e) MI L2.

Figure 4: Larger-scale experiments. The first three experiments use $n = 4000$, while the last two use $n = 10^4$ and $n = 10^5$, respectively.

overhead, it enables the algorithm to be applied to larger-scale problems and demonstrates its robustness. The results are shown in Fig. 4 (e).

6 Conclusion

We elaborate the trajectory of the regularization parameter during the iterations of the proximal point method and establish its ability to approximate the entropic optimal transport solution. This theoretical insight facilitates the seamless integration of the proximal point method with complementary algorithms, leading to the proposed Proximal-Sinkhorn-Newton framework that achieves high efficiency and robustness under varying entropic regularization parameters. Furthermore, to the best of our knowledge, we present the first complexity analysis of Newton-type methods for entropic optimal transport.

Since the number of IP-EOT iterations in this work is chosen empirically, future research may investigate how the number of proximal point iterations influences overall convergence, potentially leading to more adaptive iteration scheduling strategies. Another promising direction is the integration of additional algorithmic modules into the proximal point framework to further enhance performance.

AI Use Statement. This work uses generative AI solely to improve the language and readability of the manuscript. All AI-assisted edits have been reviewed by the authors, and no AI tools were used for other research or writing tasks. The authors take full responsibility for the final content of the paper.

Ethics Statement. This work focuses on the development and analysis of algorithms for entropic optimal transport. It does not involve human subjects, sensitive data, or applications with foreseeable ethical risks. To the best of our knowledge, this research raises no ethical concerns.

Reproducibility Statement. To ensure reproducibility, we provide the source code with our submission. All algorithmic steps, parameter settings, and experimental configurations are explicitly described in the paper. With the released code and detailed specifications, independent researchers should be able to fully reproduce our results.

REFERENCES

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative Bregman Projections for Regularized Transportation Problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

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

Chunxu Cao and Qiang Zhang. Feature Selection via Maximizing Distances between Class Conditional Distributions. *arXiv preprint arXiv:2401.07488*, 2024.

Chunxu Cao, Yong Zhang, Yanke Ai, and Qiang Zhang. Unsupervised feature selection via unifying distribution alignment and structure preservation. *Information Fusion*, 126:103544, 2026. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2025.103544.

Gong Chen and Marc Teboulle. Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

Lénaïc Chizat. Annealed Sinkhorn for Optimal Transport: convergence, regularization path and debiasing. *arXiv preprint arXiv:2408.11620*, 2024.

Hong TM Chu, Ling Liang, Kim-Chuan Toh, and Lei Yang. An efficient implementable inexact entropic proximal point algorithm for a class of linear programming problems. *Computational Optimization and Applications*, 85(1):107–146, 2023.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm. In *International Conference on Machine Learning*, pp. 1367–1376. PMLR, 2018.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

Magnus R Hestenes, Eduard Stiefel, et al. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Viet Huynh, He Zhao, and Dinh Phung. OTLDA: A Geometry-Aware Optimal Transport Approach for Topic Modeling. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18573–18582, 2020.

Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng David Gu. A Geometric View of Optimal Transportation and Generative Model. *Computer Aided Geometric Design*, 68:1–21, 2019.

Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. In *Proceedings of the thirtieth annual ACM symposium on Theory of Computing*, pp. 644–652, 1998.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Jianzhou Luo, Dingchuan Yang, and Ke Wei. Improved complexity analysis of the sinkhorn and greenkhorn algorithms for optimal transport. *arXiv preprint arXiv:2305.14939*, 2023.

Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

Tuan Nguyen, Trung Le, He Zhao, Quan Hung Tran, Truyen Nguyen, and Dinh Phung. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *Uncertainty in Artificial Intelligence*, pp. 225–235. PMLR, 2021.

J Nocedal and S Wright. Numerical optimization, 2006.

Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, 1999.

Luca Oneto, Michele Donini, Giulia Luise, Carlo Ciliberto, Andreas Maurer, and Massimiliano Pontil. Exploiting MMD and Sinkhorn Divergences for Fair and Transferable Representation Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15360–15370, 2020.

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*, 2021.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, 2017.

Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends in Machine Learning, 2019.

R Tyrrell Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM J. Sci. Comput.*, 41(3):A1443–A1481, 2019.

Binrui Shen, Qiang Niu, and Shengxin Zhu. Adaptive softassign via hadamard-equipped sinkhorn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17638–17647, 2024.

Richard Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

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.

Zihao Tang and Yixuan Qiu. Safe and sparse newton method for entropic-regularized optimal transport. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, and Massimiliano Pontil. Multi-source Domain Adaptation via Weighted Joint Distributions Optimal Transport. In *Uncertainty in Artificial Intelligence*, pp. 1970–1980. PMLR, 2022.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pp. 433–453. PMLR, 22–25 Jul 2020.

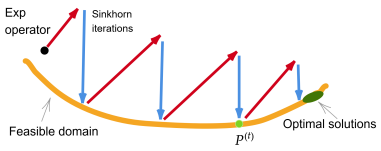
A NOTATIONS

Key notations are summarized below.

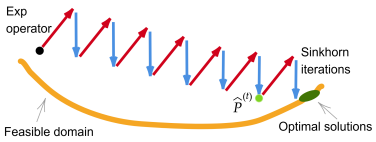
Table 2: Symbols and Notations.

Symbol	Definition
$\mathcal{H}(\cdot)$	negative entropy function
$\mathcal{B}(\cdot,\cdot)$	Bregman divergence
$\mathbb{U}$	$\{P \in \mathbb{R}_+^{m\times n} : P\mathbf{1} = \boldsymbol{a}, P^\top \mathbf{1} = \boldsymbol{b}\}, \boldsymbol{a} \in \mathbb{R}_{++}^m, \boldsymbol{b} \in \mathbb{R}_{++}^n$
$D_{(\boldsymbol{x})}$	diagonal matrix of a vector $\boldsymbol{x}$
$A \cdot B$	Frobenius inner product between matrix A and B
$\otimes$	Kronecker product
$\mathring{\exp}$	element-wise exponential
$\oslash$	element-wise division
$\odot$	element-wise product
$\bigodot$	element-wise multiplication of a series of matrices
β	the parameter in entropic OT
$\Delta\beta$	the proximal parameter in proximal point methods
$\Gamma_{\mathbb{U}}^{\mathrm{kl}}(\cdot), \hat{\Gamma}_{\mathrm{kl}}(\cdot)$	KL-projection and its approximating version
$P_\beta, \hat{P}_\beta$	the exact and an approximating solution to the entropic OT with β

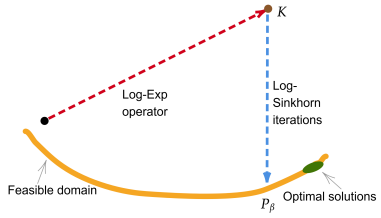
B SCHEMATIC OF THE PATH OF EXISTING METHODS



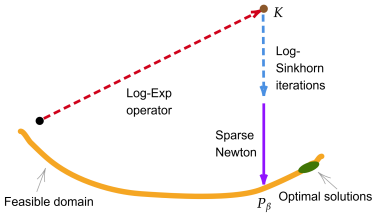
(a) POT (Xie et al., 2020)



(b) IPOT (Xie et al., 2020)



(c) Log-Sinkhorn (Benamou et al., 2015)



(d) SNS (Tang et al., 2024)

Figure 5: The distance is in the Bregman sense. POT and IPOT are designed for OT, and they require far more iterations than those illustrated in the figure to converge; log-Sinkhorn and SNS are designed for entropic OT.

C PROOFS FOR PROXIMAL POINT METHODS

Lemma 1 (Explicit form of β). *The iterative schemes of the exact/inexact proximal point method for entropic OT admit the following explicit representation:*

$$P^{(t)} = \Gamma_{\mathbb{U}}^{\mathrm{kl}}\Big(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})\Big), \quad \hat{P}^{(t)} = \hat{\Gamma}_{\mathbb{U}}^{\mathrm{kl}}\Big(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})\Big). \tag{22}$$

Proof. This proof begins with the iterative formula of the exact proximal point method:

$$P^{(t)} = \Gamma_{\mathbb{U}}^{\mathrm{kl}}\left(\mathring{\exp}(-C\beta^{(t)}) \odot P^{(t-1)}\right). \tag{23}$$

For the first step, we have

$$P^{(1)} = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\mathring{\exp}(-C\beta^{(1)})) = D_{(\boldsymbol{u}_1)}\, \mathring{\exp}(-\beta^{(1)}C)\, D_{(\boldsymbol{v}_1)}, \tag{24}$$

An equivalent form Shen et al. (2024, Proposition 3) can be derived via elementary matrix operations as

$$P^{(1)} = \mathring{\exp}\,(-C\beta^{(1)}) \odot (\boldsymbol{u}_1 \otimes \boldsymbol{v}_1^\top). \tag{25}$$

Hence, for the subsequent iteration, we have:

$$\begin{aligned}
P^{(2)} &= \Gamma^{\mathrm{kl}}_{\mathbb{U}}\left(\mathring{\exp}(-C\beta^{(2)}) \odot P^{(1)}\right) && (26)\\
&= \Gamma^{\mathrm{kl}}_{\mathbb{U}}\left(\mathring{\exp}(-C\beta^{(2)}) \odot \mathring{\exp}(-C\beta^{(1)}) \odot (\boldsymbol{u}_1 \otimes \boldsymbol{v}_1^\top)\right) && (27)\\
&= \left(\mathring{\exp}(-C\beta^{(2)}) \odot \mathring{\exp}(-C\beta^{(1)}) \odot (\boldsymbol{u}_1 \otimes \boldsymbol{v}_1^\top)\right) \odot (\boldsymbol{u}_2 \otimes \boldsymbol{v}_2^\top) && (28)\\
&= \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot (\boldsymbol{u}_1 \otimes \boldsymbol{v}_1^\top) \odot (\boldsymbol{u}_2 \otimes \boldsymbol{v}_2^\top) && (29)\\
&= \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot \left((\boldsymbol{u}_1 \odot \boldsymbol{u}_2) \otimes (\boldsymbol{v}_1^\top \odot \boldsymbol{v}_2^\top)\right). && (30)
\end{aligned}$$

By defining $\boldsymbol{u}_3 := \boldsymbol{u}_1 \odot \boldsymbol{u}_2$ and $\boldsymbol{v}_3 := \boldsymbol{v}_1 \odot \boldsymbol{v}_2$, we have

$$P^{(2)} = \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot (\boldsymbol{u}_3 \otimes \boldsymbol{v}_3^\top). \tag{31}$$

Due to $P^{(2)} \in \mathbb{U}$, $P^{(2)}$ is the KL-projection of the kernel matrix $\mathring{\exp}\left(-C(\beta^{(1)} + \beta^{(2)})\right)$. Since the KL-projection is unique for a given kernel matrix (Cuturi, 2013), it follows that

$$P^{(2)} = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{2} \mathring{\exp}(-C\beta^{(i)})). \tag{32}$$

The same structure holds for general t by induction:

$$P^{(t)} = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})). \tag{33}$$

For the inexact proximal point method, we can also obtain

$$\begin{aligned}
\hat{P}^{(2)} &= \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}\left(\mathring{\exp}(-C\beta^{(2)}) \odot \hat{P}^{(1)}\right) && (34)\\
&= \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}\left(\mathring{\exp}(-C\beta^{(2)}) \odot \mathring{\exp}(-C\beta^{(1)}) \odot (\hat{\boldsymbol{u}}_1 \otimes \hat{\boldsymbol{v}}_1^\top)\right) && (35)\\
&= \left(\mathring{\exp}(-C\beta^{(2)}) \odot \mathring{\exp}(-C\beta^{(1)}) \odot (\hat{\boldsymbol{u}}_1 \otimes \hat{\boldsymbol{v}}_1^\top)\right) \odot (\hat{\boldsymbol{u}}_2 \otimes \hat{\boldsymbol{v}}_2^\top) && (36)\\
&= \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot (\hat{\boldsymbol{u}}_1 \otimes \hat{\boldsymbol{v}}_1^\top) \odot (\hat{\boldsymbol{u}}_2 \otimes \hat{\boldsymbol{v}}_2^\top) && (37)\\
&= \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot (\underbrace{(\hat{\boldsymbol{u}}_1 \odot \hat{\boldsymbol{u}}_2)}_{\hat{\boldsymbol{u}}_3} \otimes \underbrace{(\hat{\boldsymbol{v}}_1^\top \odot \hat{\boldsymbol{v}}_2^\top)}_{\hat{\boldsymbol{v}}_3^\top}) && (38)\\
&= \mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)})) \odot (\hat{\boldsymbol{u}}_3 \otimes \hat{\boldsymbol{v}}_3^\top). && (39)
\end{aligned}$$

Since $\hat{\boldsymbol{u}}_3$ and $\hat{\boldsymbol{v}}_3$ are approximations for the scaling vectors of the corresponding kernel matrix $\mathring{\exp}(-C(\beta^{(2)} + \beta^{(1)}))$, $\hat{P}^{(2)}$ can be rewritten as

$$\hat{P}^{(2)} = \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{2} \mathring{\exp}(-C\beta^{(i)})). \tag{40}$$

The same structure holds for general t by induction:

$$\hat{P}^{(t)} = \hat{\Gamma}^{\mathrm{kl}}_{\mathbb{U}}(\bigodot_{i=1}^{t} \mathring{\exp}(-C\beta^{(i)})). \tag{41}$$

□

Lemma 2 (From Inexact to Exact). *Let $\hat{P}_\beta$ be an approximation of P_β of the form*

$$\hat{P}_\beta = D_{(\hat{\boldsymbol{u}})} \, \mathring{\exp}(-\beta C) \, D_{(\hat{\boldsymbol{v}})}, \tag{42}$$

where $\hat{\boldsymbol{u}} \in \mathbb{R}^m_{++}$ and $\hat{\boldsymbol{v}} \in \mathbb{R}^n_{++}$. Then

$$P_\beta = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta). \tag{43}$$

Proof. By assumption,

$$\hat{P}_\beta = D_{(\hat{\boldsymbol{u}})} \, \mathring{\exp}(-\beta C) \, D_{(\hat{\boldsymbol{v}})}. \tag{44}$$

Let $\tilde{u} \in \mathbb{R}^m_{++}$, $\tilde{v} \in \mathbb{R}^n_{++}$ denote the correction scaling vectors associated with the exact KL projection of $\hat{P}_\beta$ onto $\mathbb{U}$. Then

$$\Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta) = \hat{P}_\beta \odot (\tilde{u} \otimes \tilde{v}^\top) = \exp(-\beta C) \odot \left((\hat{u} \odot \tilde{u}) \otimes (\hat{v} \odot \tilde{v})^\top \right), \tag{45}$$

according to Shen et al. (2024, Proposition 3). Thus, $\Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta)$ is a feasible diagnoal scaling of the same kernel $\exp(-\beta C)$. By uniqueness of the KL projection onto $\mathbb{U}$,

$$\Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta) = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\exp(-\beta C)) = P_\beta. \tag{46}$$

This completes the proof. □

Theorem 1 (Inexact-to-Exact Transition). *If $\hat{P}^{(t)}$ denotes the transport plan updated at the t-th iteration in the inexact proximal point method with a fixed proximal parameter $\Delta\beta$, then*

$$P_\beta = \Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}^{(t)}), \ \textit{where } \beta = t\Delta\beta. \tag{47}$$

Proof. According to Lemma 1, we have

$$\hat{P}^{(t)} = \hat{\Gamma}_{\mathrm{kl}}(\mathring{\exp}\,(-t\Delta\beta C)) = \hat{P}_\beta. \tag{48}$$

Then, by Lemma 2, it follows that

$$\Gamma^{\mathrm{kl}}_{\mathbb{U}}(\hat{P}_\beta) = P_\beta, \tag{49}$$

which completes the proof. □

D Proof for truncated Hessian matrix eigenvalue lower bound

Lemma 3 (Minimum Entry Bound of Transport Plan). *Suppose P, $Q \in \mathbb{R}^{n\times n}_{++}$, $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^n_{++}$ and $\boldsymbol{x}^*$, $\boldsymbol{y}^* \in \mathbb{R}^n$ satisfying*

$$P\mathbf{1} = \boldsymbol{a}, \qquad P^\top \mathbf{1} = \boldsymbol{b}, \tag{50}$$

where

$$P = \mathrm{diag}(\exp(\beta \boldsymbol{x}^*)) Q \mathrm{diag}(\exp(\beta \boldsymbol{y}^*)), \qquad \boldsymbol{a}^\top \mathbf{1} = \boldsymbol{b}^\top \mathbf{1} = 1. \tag{51}$$

Define

$$p_{\min} := \min_{i,j} p_{ij}, \qquad a_{\min} := \min_i a_i, \qquad b_{\min} := \min_j b_j, \tag{52}$$

and

$$\vartheta(Q) := \max_{i,j,k,l} \frac{q_{ij} q_{kl}}{q_{il} q_{kj}}. \tag{53}$$

Then

$$p_{ij} \geq \frac{a_i b_j}{\vartheta(Q)}. \tag{54}$$

In particular,

$$p_{\min} \geq \frac{a_{\min} b_{\min}}{\vartheta(Q)}. \tag{55}$$

Proof. Let

$$\alpha_i^* := \exp(\beta \boldsymbol{x}_i^*), \qquad \gamma_j^* := \exp(\beta \boldsymbol{y}_j^*), \tag{56}$$

so that

$$p_{ij} = q_{ij}\alpha_i^*\gamma_j^*. \tag{57}$$

Fix an arbitrary pair (i, j). We prove the desired lower bound by contradiction. Suppose that

$$p_{ij} < \frac{a_i b_j}{\vartheta(Q)}. \tag{58}$$

From the marginal constraints, we have

$$\sum_{l\neq j} q_{il}\alpha_i^*\gamma_l^* = a_i - p_{ij}, \qquad \sum_{k\neq i} q_{kj}\alpha_k^*\gamma_j^* = b_j - p_{ij}. \tag{59}$$

Equivalently,

$$\sum_{l\neq j} q_{il}\gamma_l^* = \frac{a_i - p_{ij}}{\alpha_i^*}, \qquad \sum_{k\neq i} q_{kj}\alpha_k^* = \frac{b_j - p_{ij}}{\gamma_j^*}. \tag{60}$$

Consider the function

$$F(\boldsymbol{\alpha}, \boldsymbol{\gamma}) := \boldsymbol{\alpha}^\top Q \boldsymbol{\gamma}. \tag{61}$$

Since P has total mass one, we have

$$F(\boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*) = \mathbf{1}^\top P \mathbf{1} = 1. \tag{62}$$

Fix $\alpha_i = \mu$ and $\gamma_j = \nu$ such that

$$q_{ij}\mu\nu = p_{ij}. \tag{63}$$

We consider the feasible set

$$\Omega_{ij} := \left\{ (\boldsymbol{\alpha}, \boldsymbol{\gamma}) \in \mathbb{R}_+^{2n} : \alpha_i = \mu,\ \gamma_j = \nu,\ \sum_{l\neq j} q_{il}\gamma_l = \frac{a_i - p_{ij}}{\mu},\ \sum_{k\neq i} q_{kj}\alpha_k = \frac{b_j - p_{ij}}{\nu} \right\}. \tag{64}$$

The function F is continuous on Ω_{ij}, and its minimum is attained at a vertex of this feasible set. The relevant vertices are of the form

$$\boldsymbol{\alpha} = \mu e_i + \frac{b_j - p_{ij}}{\nu q_{kj}} e_k, \qquad k \neq i, \tag{65}$$

and

$$\boldsymbol{\gamma} = \nu e_j + \frac{a_i - p_{ij}}{\mu q_{il}} e_l, \qquad l \neq j. \tag{66}$$

Evaluating F at such a vertex gives

$$F(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = a_i + b_j - p_{ij} + \frac{q_{ij}q_{kl}}{q_{il}q_{kj}} \frac{(a_i - p_{ij})(b_j - p_{ij})}{p_{ij}}. \tag{67}$$

Define

$$g_{ijkl}(p) := a_i + b_j - p + \frac{q_{ij}q_{kl}}{q_{il}q_{kj}} \frac{(a_i - p)(b_j - p)}{p}. \tag{68}$$

By the definition of $\vartheta(Q)$,

$$\frac{q_{ij}q_{kl}}{q_{il}q_{kj}} \geq \frac{1}{\vartheta(Q)}. \tag{69}$$

Therefore,

$$g_{ijkl}(p) \geq a_i + b_j - p + \frac{1}{\vartheta(Q)} \frac{(a_i - p)(b_j - p)}{p} =: g_{ij}(p). \tag{70}$$

For

$$0 < p < \frac{a_i b_j}{\vartheta(Q)}, \tag{71}$$

we compute

$$g_{ij}'(p) = -1 + \frac{1}{\vartheta(Q)} \left(1 - \frac{a_i b_j}{p^2}\right) < 0. \tag{72}$$

Hence g_{ij} is strictly decreasing on this interval. Thus,

$$g_{ij}(p_{ij}) > g_{ij}\left(\frac{a_i b_j}{\vartheta(Q)}\right). \tag{73}$$

Since $a_i, b_j \le 1$ and $\vartheta(Q) \ge 1$, a direct calculation yields

$$g_{ij}\left(\frac{a_i b_j}{\vartheta(Q)}\right) = 1 + \left(a_i + b_j - \frac{a_i b_j}{\vartheta(Q)}\right)\left(1 - \frac{1}{\vartheta(Q)}\right) \ge 1, \tag{74}$$

Therefore,

$$F(\boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*) \ge \min_{k\neq i, l\neq j} g_{ijkl}(p_{ij}) \ge g_{ij}(p_{ij}) > 1. \tag{75}$$

This contradicts the fact that $F(\boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*) = 1$. Hence

$$p_{ij} \ge \frac{a_i b_j}{\vartheta(Q)}. \tag{76}$$

Taking the minimum over all i, j gives

$$p_{\min} \ge \frac{a_{\min} b_{\min}}{\vartheta(Q)}. \tag{77}$$

□

Lemma 4 (Uniform Minimum Entry Bound of Transport Plan)**.** *Under the same assumptions as in Lemma 3, let*

$$f(\boldsymbol{x}, \tilde{\boldsymbol{y}}) = \frac{1}{\beta}\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}\exp\big(\beta(x_i + y_j)\big) - \boldsymbol{a}^\top \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{y}, \tag{78}$$

where the gauge condition $y_n = 0$ is imposed and $\boldsymbol{z} = (\boldsymbol{x}, \tilde{\boldsymbol{y}}) \in \mathbb{R}^{2n-1}$ denotes the corresponding gauge-fixed dual variable. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \tilde{\boldsymbol{y}}^*)$ be a minimizer of f, and let $\{\boldsymbol{z}_k\}$ be the sequence generated by Algorithm 2 from an initial point $\boldsymbol{z}_0$.

Define

$$\mathcal{L}_0 := \left\{\boldsymbol{z} \in \mathbb{R}^{2n-1} : f(\boldsymbol{z}) \le f(\boldsymbol{z}_0)\right\}, \tag{79}$$

and

$$P_{ij}(\boldsymbol{z}) := q_{ij}\exp\big(\beta(x_i + y_j)\big), \qquad p_{\min}^{(k)} := \min_{i,j} P_{ij}(\boldsymbol{z}_k). \tag{80}$$

Let

$$a_{\min} := \min_i a_i, \qquad b_{\min} := \min_j b_j, \tag{81}$$

and

$$\vartheta(Q) := \max_{i,k,j,l} \frac{q_{ij} q_{kl}}{q_{il} q_{kj}}, \qquad Q = (q_{ij}). \tag{82}$$

Then $\mathcal{L}_0$ is compact, and hence

$$r_{\text{sub}} := \max_{\boldsymbol{z}\in\mathcal{L}_0} \|\boldsymbol{z} - \boldsymbol{z}^*\|_2 < \infty. \tag{83}$$

Moreover, r_{sub} admits an explicit upper bound $r_{\text{sub}} \le \bar{r}_{\text{sub}}$ defined below, and every iterate generated by Algorithm 2 satisfies

$$\boldsymbol{z}_k \in \mathcal{L}_0 \subseteq \mathcal{N}_{\bar{r}_{\text{sub}}}(\boldsymbol{z}^*), \qquad \forall k \ge 0. \tag{84}$$

Consequently, there exists a constant $p^{\mathcal{L}} > 0$, independent of the iteration index k, such that

$$p_{\min}^{(k)} \ge p^{\mathcal{L}} := \frac{a_{\min} b_{\min}}{\vartheta(Q)}\exp\big(-\sqrt{2}\beta\bar{r}_{\text{sub}}\big), \qquad \forall k \ge 0. \tag{85}$$

Proof. Let

$$c_0 := f(\boldsymbol{z}_0). \tag{86}$$

For

$$s_{ij} := x_i + y_j, \qquad w_{ij} := a_i b_j, \tag{87}$$

using $\boldsymbol{a}^\top \mathbf{1} = \boldsymbol{b}^\top \mathbf{1} = 1$, we have

$$\boldsymbol{a}^\top \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{y} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i b_j (x_i + y_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} s_{ij}. \tag{88}$$

Therefore,

$$f(\boldsymbol{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}(s_{ij}), \tag{89}$$

where

$$h_{ij}(s) := \frac{q_{ij}}{\beta} e^{\beta s} - w_{ij} s. \tag{90}$$

Since $q_{ij} > 0$ and $w_{ij} > 0$, we have

$$\lim_{s \to +\infty} h_{ij}(s) = \lim_{s \to -\infty} h_{ij}(s) = +\infty. \tag{91}$$

Moreover,

$$h'_{ij}(s) = q_{ij} e^{\beta s} - w_{ij}, \qquad h''_{ij}(s) = \beta q_{ij} e^{\beta s} > 0. \tag{92}$$

Hence h_{ij} is strictly convex and coercive and has a unique minimizer

$$s^0_{ij} = \frac{1}{\beta} \log \frac{w_{ij}}{q_{ij}}, \tag{93}$$

with minimum value

$$m_{ij} := h_{ij}(s^0_{ij}) = \frac{w_{ij}}{\beta} \left(1 - \log \frac{w_{ij}}{q_{ij}}\right). \tag{94}$$

Define

$$M := \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}. \tag{95}$$

Since $h_{ij}(s) \geq m_{ij}$ for all s, we have

$$c_0 = f(\boldsymbol{z}_0) \geq M. \tag{96}$$

Now fix an arbitrary $\boldsymbol{z} \in \mathcal{L}_0$ and an arbitrary pair (i, j). Since

$$h_{kl}(s_{kl}) \geq m_{kl}, \qquad \forall (k, l), \tag{97}$$

we obtain

$$h_{ij}(s_{ij}) = f(\boldsymbol{z}) - \sum_{(k,l) \neq (i,j)} h_{kl}(s_{kl}) \tag{98}$$

$$\leq c_0 - \sum_{(k,l) \neq (i,j)} m_{kl} \tag{99}$$

$$= c_0 - M + m_{ij}. \tag{100}$$

Define

$$c^{\mathrm{sub}}_{ij} := c_0 - M + m_{ij}. \tag{101}$$

Since $c_0 \geq M$, we have

$$c^{\mathrm{sub}}_{ij} \geq m_{ij}. \tag{102}$$

Thus the scalar sublevel set

$$\left\{ s \in \mathbb{R} : h_{ij}(s) \leq c^{\mathrm{sub}}_{ij} \right\} \tag{103}$$

is a compact interval, which we denote by

$$[L_{ij}, U_{ij}]. \tag{104}$$

We next compute these two endpoints explicitly. The equation

$$h_{ij}(s) = c^{\mathrm{sub}}_{ij} \tag{105}$$

is equivalent to

$$\frac{q_{ij}}{\beta}e^{\beta s} - w_{ij}s = c_{ij}^{\mathrm{sub}}. \tag{106}$$

Let

$$t = \beta s - \log\frac{w_{ij}}{q_{ij}}. \tag{107}$$

Then the preceding equation becomes

$$e^t - t = A_{ij}, \tag{108}$$

where

$$A_{ij} := \frac{\beta c_{ij}^{\mathrm{sub}}}{w_{ij}} + \log\frac{w_{ij}}{q_{ij}} \tag{109}$$

$$= \frac{\beta(c_0 - M + m_{ij})}{w_{ij}} + \log\frac{w_{ij}}{q_{ij}} \tag{110}$$

$$= 1 + \frac{\beta(c_0 - M)}{w_{ij}} \geq 1. \tag{111}$$

Hence the two real roots are expressed through the two real branches of the Lambert W function as

$$t_{ij}^L = -W_0(-e^{-A_{ij}}) - A_{ij}, \tag{112}$$

and

$$t_{ij}^U = -W_{-1}(-e^{-A_{ij}}) - A_{ij}. \tag{113}$$

Consequently,

$$L_{ij} = \frac{-W_0(-e^{-A_{ij}}) - A_{ij} + \log(w_{ij}/q_{ij})}{\beta}, \tag{114}$$

and

$$U_{ij} = \frac{-W_{-1}(-e^{-A_{ij}}) - A_{ij} + \log(w_{ij}/q_{ij})}{\beta}. \tag{115}$$

Therefore every $\boldsymbol{z} \in \mathcal{L}_0$ satisfies

$$L_{ij} \leq x_i + y_j \leq U_{ij}, \qquad \forall i, j. \tag{116}$$

We now use the gauge condition $y_n = 0$ to obtain coordinatewise bounds for the dual variables. Since

$$x_i = x_i + y_n, \tag{117}$$

we have

$$L_{in} \leq x_i \leq U_{in}, \qquad i = 1, \ldots, n. \tag{118}$$

For $j = 1, \ldots, n-1$,

$$y_j = (x_i + y_j) - x_i, \tag{119}$$

and therefore, for every i,

$$L_{ij} - U_{in} \leq y_j \leq U_{ij} - L_{in}. \tag{120}$$

Define

$$L_j^y := \max_{1 \leq i \leq n} \left(L_{ij} - U_{in}\right), \tag{121}$$

and

$$U_j^y := \min_{1 \leq i \leq n} \left(U_{ij} - L_{in}\right). \tag{122}$$

Then

$$L_j^y \leq y_j \leq U_j^y, \qquad j = 1, \ldots, n-1. \tag{123}$$

Hence every $\boldsymbol{z} \in \mathcal{L}_0$ belongs to the finite box

$$\prod_{i=1}^{n}[L_{in}, U_{in}] \times \prod_{j=1}^{n-1}[L_j^y, U_j^y]. \tag{124}$$

Thus $\mathcal{L}_0$ is bounded. Since f is continuous, $\mathcal{L}_0 = f^{-1}((-\infty, c_0])$ is closed. Therefore $\mathcal{L}_0$ is compact in $\mathbb{R}^{2n-1}$.

Since $\boldsymbol{z}^*$ is a minimizer of f,

$$f(\boldsymbol{z}^*) \le f(\boldsymbol{z}_0) = c_0, \tag{125}$$

and hence $\boldsymbol{z}^* \in \mathcal{L}_0$. Because $\boldsymbol{z} \mapsto \|\boldsymbol{z} - \boldsymbol{z}^*\|_2$ is continuous on the compact set $\mathcal{L}_0$, the maximum

$$r_{\mathrm{sub}} := \max_{\boldsymbol{z} \in \mathcal{L}_0} \|\boldsymbol{z} - \boldsymbol{z}^*\|_2 \tag{126}$$

exists and is finite.

Furthermore, since both $\boldsymbol{z}$ and $\boldsymbol{z}^*$ belong to the above coordinate box, we have

$$|x_i - x_i^*| \le U_{in} - L_{in}, \qquad i = 1, \dots, n, \tag{127}$$

and

$$|y_j - y_j^*| \le U_j^y - L_j^y, \qquad j = 1, \dots, n-1. \tag{128}$$

It follows that

$$r_{\mathrm{sub}} \le \bar{r}_{\mathrm{sub}}, \tag{129}$$

where

$$\bar{r}_{\mathrm{sub}} := \left[\sum_{i=1}^{n} (U_{in} - L_{in})^2 + \sum_{j=1}^{n-1} (U_j^y - L_j^y)^2 \right]^{1/2}. \tag{130}$$

We next show that all iterates generated by Algorithm 2 remain in $\mathcal{L}_0$. If the line search accepts a Newton step, the accepted step decreases the objective. If the line search fails, Algorithm 2 performs the Sinkhorn fallback, which also decreases the objective. Therefore,

$$f(\boldsymbol{z}_{k+1}) \le f(\boldsymbol{z}_k) \le \cdots \le f(\boldsymbol{z}_0), \qquad \forall k \ge 0. \tag{131}$$

Consequently,

$$\boldsymbol{z}_k \in \mathcal{L}_0 \subseteq \mathcal{N}_{r_{\mathrm{sub}}}(\boldsymbol{z}^*) \subseteq \mathcal{N}_{\bar{r}_{\mathrm{sub}}}(\boldsymbol{z}^*), \qquad \forall k \ge 0. \tag{132}$$

Finally, for every (i, j),

$$P_{ij}(\boldsymbol{z}_k) = q_{ij} \exp\left(\beta(x_i^{(k)} + y_j^{(k)})\right) \tag{133}$$

$$= P_{ij}(\boldsymbol{z}^*) \exp\left(\beta\left[(x_i^{(k)} - x_i^*) + (y_j^{(k)} - y_j^*)\right]\right). \tag{134}$$

Since both $\boldsymbol{z}_k$ and $\boldsymbol{z}^*$ satisfy the same gauge condition $y_n = 0$, we have

$$\left|(x_i^{(k)} - x_i^*) + (y_j^{(k)} - y_j^*)\right| \le \sqrt{2}\|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2 \le \sqrt{2}\bar{r}_{\mathrm{sub}}. \tag{135}$$

Hence

$$P_{ij}(\boldsymbol{z}_k) \ge P_{ij}(\boldsymbol{z}^*) \exp\bigl(-\sqrt{2}\beta\bar{r}_{\mathrm{sub}}\bigr). \tag{136}$$

Taking the minimum over all (i, j) gives

$$p_{\min}^{(k)} \ge p_{\min}^* \exp\bigl(-\sqrt{2}\beta\bar{r}_{\mathrm{sub}}\bigr), \tag{137}$$

where

$$p_{\min}^* := \min_{i,j} P_{ij}(\boldsymbol{z}^*). \tag{138}$$

By the minimum entry estimation established in Lemma 3,

$$p_{\min}^* \ge \frac{a_{\min} b_{\min}}{\vartheta(Q)}. \tag{139}$$

Therefore,

$$p_{\min}^{(k)} \ge \frac{a_{\min} b_{\min}}{\vartheta(Q)} \exp\bigl(-\sqrt{2}\beta\bar{r}_{\mathrm{sub}}\bigr) =: p^{\mathcal{L}} > 0, \qquad \forall k \ge 0. \tag{140}$$

The constant $p^{\mathcal{L}}$ depends only on the problem data and the initial sublevel set, and is independent of the iteration index k. This completes the proof. □

Lemma 5 (Unreduced Hessian Nonzero Eigenvalue Lower Bound)**.** *Suppose* P, $Q \in \mathbb{R}^{n\times n}_{++}$, $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^n_{++}$ *and* $\boldsymbol{x}^*$, $\boldsymbol{y}^* \in \mathbb{R}^n$ *satisfying*

$$P\mathbf{1} = \boldsymbol{a}, \qquad P^\top \mathbf{1} = \boldsymbol{b}, \tag{141}$$

where

$$P = \mathrm{diag}(\exp(\beta \boldsymbol{x}^*))Q\mathrm{diag}(\exp(\beta \boldsymbol{y}^*)), \qquad \boldsymbol{a}^\top \mathbf{1} = \boldsymbol{b}^\top \mathbf{1} = 1. \tag{142}$$

Define

$$p_{\min} := \min_{i,j} p_{ij}, \qquad a_{\min} := \min_i a_i, \qquad b_{\min} := \min_j b_j. \tag{143}$$

Let $\lambda > 0$ *be a non-zero eigenvalue of the unreduced Hessian*

$$\mathcal{H}_\beta(P) := \beta \begin{pmatrix} \mathrm{diag}(P\mathbf{1}) & P \\ P^\top & \mathrm{diag}(P^\top \mathbf{1}) \end{pmatrix}. \tag{144}$$

Then,

$$\bar{\lambda} \geq \frac{n\beta a_{\min} b_{\min}}{\vartheta(Q)}, \tag{145}$$

where $Q = \exp(-\beta C - \mathbf{1}\mathbf{1}^\top)$ *and* $\vartheta(Q) = \max_{i,j,k,l} \frac{q_{ij}q_{kl}}{q_{il}q_{kj}}$.

Proof. The Hessian has the null direction

$$(\mathbf{1}^\top, -\mathbf{1}^\top)^\top, \tag{146}$$

corresponding to the scaling invariance of the factorization. Therefore, the non-zero eigenvalues are characterized on the orthogonal subspace

$$\mathbf{1}^\top \boldsymbol{u} = \mathbf{1}^\top \boldsymbol{v}. \tag{147}$$

For any $(\boldsymbol{u}, \boldsymbol{v})$ in this subspace with

$$\|(\boldsymbol{u}, \boldsymbol{v})\|_2^2 = \|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2 = 1, \tag{148}$$

we have

$$\begin{aligned} (\boldsymbol{u}^\top, \boldsymbol{v}^\top)\mathcal{H}_\beta(P) \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} &= \beta \sum_{i,j} p_{ij}(u_i + v_j)^2 \\ &\geq \beta p_{\min} \sum_{i,j} (u_i + v_j)^2. \end{aligned} \tag{149}$$

Moreover,

$$\begin{aligned} \sum_{i,j}(u_i + v_j)^2 &= n\|\boldsymbol{u}\|_2^2 + n\|\boldsymbol{v}\|_2^2 + 2(\mathbf{1}^\top \boldsymbol{u})(\mathbf{1}^\top \boldsymbol{v}) \\ &= n + 2(\mathbf{1}^\top \boldsymbol{u})^2 \\ &\geq n. \end{aligned} \tag{150}$$

Thus combining Lemma 3, every non-zero eigenvalue satisfies

$$\bar{\lambda} \geq \frac{n\beta a_{\min} b_{\min}}{\vartheta(Q)}. \tag{151}$$

□

Lemma 6. *Under the same assumptions as Lemma 4 and Lemma 5, every eigenvalue* $\tilde{\lambda}$ *of the reduced Hessian matrix* $H(z)$ *from (14) satisfies*

$$\tilde{\lambda} \geq \frac{\beta}{2} p^{\mathcal{L}} \tag{152}$$

where $p^{\mathcal{L}}$ *is defined in Lemma 4.*

Proof. In Lemma 5 we proved that $\mathcal{H}_\beta(P)$ is positive definite on the subspace $\text{span}\{(\mathbf{1}, -\mathbf{1})^\top\}^\perp$, therefore, by eigenvalue decomposition,

$$\mathcal{H}_\beta(P) \geq \bar{\lambda}_{\min} \cdot \left[I - \frac{1}{2n} \cdot \begin{pmatrix} \mathbf{1}\mathbf{1}^\top & -\mathbf{1}\mathbf{1}^\top \\ -\mathbf{1}\mathbf{1}^\top & \mathbf{1}\mathbf{1}^\top \end{pmatrix} \right], \tag{153}$$

and the notation $A \geq B$ means that $A - B$ is positive semi-definite.

Now for any eigenvalue $\tilde{\lambda}$ of $H(\boldsymbol{z})$ and its eigenvector $(u, \tilde{v}) \in \mathbb{R}^n \times \mathbb{R}^{n-1}$,

$$\begin{aligned}
\tilde{\lambda} &\geq \min_{\|(u,\tilde{v})\|_2^2=1} (u^\top, \tilde{v}^\top) \cdot H(\boldsymbol{z}) \cdot \begin{pmatrix} u \\ \tilde{v} \end{pmatrix} \\
&= \min_{v_n=0, \|(u,v)\|_2^2=1} (u^\top, v^\top) \cdot \mathcal{H}_\beta(P) \cdot \begin{pmatrix} u \\ v \end{pmatrix} \\
&\geq \min_{v_n=0, \|(u,v)\|_2^2=1} (u^\top, v^\top) \cdot \bar{\lambda}_{\min} \cdot \left[I - \frac{1}{2n} \cdot \begin{pmatrix} \mathbf{1}\mathbf{1}^\top & -\mathbf{1}\mathbf{1}^\top \\ -\mathbf{1}\mathbf{1}^\top & \mathbf{1}\mathbf{1}^\top \end{pmatrix} \right] \cdot \begin{pmatrix} u \\ v \end{pmatrix} \\
&= \bar{\lambda}_{\min} \cdot \left[1 - \frac{1}{2n} \cdot \max_{v_n=0, \|(u,v)\|_2^2=1} (\mathbf{1}^\top u - \mathbf{1}^\top v)^2 \right] \\
&= \frac{1}{2n} \bar{\lambda}_{\min}.
\end{aligned} \tag{154}$$

Combining (85), (151) and (154) yields (152). □

Theorem 2 (Truncated Hessian Eigenvalue Lower Bound)**.** *Every eigenvalue λ of the truncated Hessian matrix $\{\hat{H}_k\}$ generated by (15) and fast sparse scheme satisfies*

$$\lambda \geq \left(n - \sqrt{n(n-1)} \right) \beta p^{\mathcal{L}}, \tag{155}$$

where $p^{\mathcal{L}}$ is defined by (140) in Lemma 4 (Appendix D), which depends on $C, \beta, \boldsymbol{a}, \boldsymbol{b}$ and the initial point.

Proof. For $\boldsymbol{u} \in \mathbb{R}^n$ and $\widetilde{\boldsymbol{v}} \in \mathbb{R}^{n-1}$, write $\boldsymbol{w} = (\boldsymbol{u}^\top, \widetilde{\boldsymbol{v}}^\top)^\top$ and extend $\widetilde{\boldsymbol{v}}$ to $\boldsymbol{v} \in \mathbb{R}^n$ by setting $v_n = 0$. Since the diagonal blocks retain the original row and column sums,

$$\frac{1}{\beta} \boldsymbol{w}^\top \hat{H}_k \boldsymbol{w} = \sum_{(i,j) \in S} p_{ij}^{(k)} (u_i + v_j)^2 + \sum_{\substack{1 \leq i \leq n,\, 1 \leq j < n \\ (i,j) \notin S}} p_{ij}^{(k)} (u_i^2 + v_j^2) + \sum_{i=1}^{n} p_{in}^{(k)} u_i^2, \tag{156}$$

where $p_{ij}^{(k)}$ denotes the entries of the transport plan P_k and S_k denotes the entries retained by the truncation. Every summand is nonnegative. Replacing each $p_{ij}^{(k)}$ by $p^{\mathcal{L}}$ therefore gives

$$\boldsymbol{w}^\top \hat{H}_k \boldsymbol{w} \geq \beta p^{\mathcal{L}} \boldsymbol{w}^\top B_S \boldsymbol{w}, \qquad B_S := \begin{pmatrix} nI_n & A_S \\ A_S^\top & nI_{n-1} \end{pmatrix}, \tag{157}$$

where $(A_S)_{ij} = 1$ for $(i, j) \in S$ and zero otherwise. The singular value decomposition of A_S yields

$$\lambda_{\min}(B_S) = n - \|A_S\|_2. \tag{158}$$

For any $\boldsymbol{t} \in \mathbb{R}^{n-1}$, the entrywise inequality $|A_S \boldsymbol{t}| \leq \mathbf{1}_n \mathbf{1}_{n-1}^\top |\boldsymbol{t}|$ and Cauchy–Schwarz give

$$\|A_S \boldsymbol{t}\|_2 \leq \sqrt{n}\, \|\boldsymbol{t}\|_1 \leq \sqrt{n(n-1)}\, \|\boldsymbol{t}\|_2. \tag{159}$$

Thus $\|A_S\|_2 \leq \sqrt{n(n-1)}$. Combining it with Lemma 4 implies (156). □

E PROOF FOR LOCAL QUADRATIC CONVERGENCE AND NEWTON'S CONVERGENCE RADIUS

Lemma 7 (Lipschitz Continuity of the Reduced Hessian)**.** *Rewrite P_β as*

$$P_\beta = \mathrm{diag}(\exp(\beta \boldsymbol{x}_\beta)) \exp(-\beta C - \mathbf{1}\mathbf{1}^\top) \mathrm{diag}(\exp(\beta \boldsymbol{y}_\beta)), \tag{160}$$

where $(\boldsymbol{x}_\beta, \boldsymbol{y}_\beta)$ is the minimizer of the dual potential function $f(\boldsymbol{x}, \boldsymbol{y})$. Then the reduced Hessian of f is Lipschitz continuous in $\mathcal{N}_\eta(\boldsymbol{x}_\beta, \boldsymbol{y}_\beta)$ with Lipschitz constant

$$L_\eta = 2\sqrt{2} \exp(\sqrt{2}\eta\beta) n \beta^2 p_{\max}, \tag{161}$$

where

$$p_{\max} := \max_{i,j} (P_\beta)_{ij}. \tag{162}$$

Proof. Let

$$(\boldsymbol{x}_k, \boldsymbol{y}_k) = (\boldsymbol{x}_\beta + \boldsymbol{\varphi}_k, \boldsymbol{y}_\beta + \boldsymbol{\psi}_k), \quad P_k = P(\boldsymbol{x}_k, \boldsymbol{y}_k), \qquad k = 1, 2, \tag{163}$$

with

$$\|(\boldsymbol{\varphi}_k, \boldsymbol{\psi}_k)\|_2 \le \eta. \tag{164}$$

Then for all i, j,

$$|\varphi_{ki} + \psi_{kj}| \le \sqrt{2}\eta. \tag{165}$$

Let

$$\Delta \boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_2, \qquad \Delta \boldsymbol{y} = \boldsymbol{y}_1 - \boldsymbol{y}_2. \tag{166}$$

For any $\|(\boldsymbol{u}, \boldsymbol{v})\|_2 = 1$, we have

$$\left| (\boldsymbol{u}^\top, \boldsymbol{v}^\top) \left(\mathcal{H}_\beta(P_1) - \mathcal{H}_\beta(P_2) \right) \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} \right| = \beta \left| \sum_{i,j} q_{ij} \left[\exp(\beta(x_{1i} + y_{1j})) - \exp(\beta(x_{2i} + y_{2j})) \right] (u_i + v_j)^2 \right|. \tag{167}$$

By the mean value theorem,

$$|\exp(\beta(x_{1i} + y_{1j})) - \exp(\beta(x_{2i} + y_{2j}))| \le \beta \exp(\sqrt{2}\eta\beta) \exp(\beta(x_{\beta,i} + y_{\beta,j})) |\Delta x_i + \Delta y_j|. \tag{168}$$

Since

$$q_{ij} \exp(\beta(x_{\beta,i} + y_{\beta,j})) = (P_\beta)_{ij} \le p_{\max}, \tag{169}$$

we obtain

$$\left| (\boldsymbol{u}^\top, \boldsymbol{v}^\top) \left(\mathcal{H}_\beta(P_1) - \mathcal{H}_\beta(P_2) \right) \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} \right| \le \beta^2 p_{\max} \exp(\sqrt{2}\eta\beta) \sum_{i,j} |\Delta x_i + \Delta y_j| (u_i + v_j)^2. \tag{170}$$

Moreover,

$$|\Delta x_i + \Delta y_j| \le \sqrt{2} \|(\Delta \boldsymbol{x}, \Delta \boldsymbol{y})\|_2. \tag{171}$$

Thus

$$\left| (\boldsymbol{u}^\top, \boldsymbol{v}^\top) \left(\mathcal{H}_\beta(P_1) - \mathcal{H}_\beta(P_2) \right) \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} \right| \le \sqrt{2}\beta^2 p_{\max} \exp(\sqrt{2}\eta\beta) \|(\Delta \boldsymbol{x}, \Delta \boldsymbol{y})\|_2 \sum_{i,j} (u_i + v_j)^2. \tag{172}$$

Finally,

$$\begin{aligned} \sum_{i,j} (u_i + v_j)^2 &= n\|\boldsymbol{u}\|_2^2 + n\|\boldsymbol{v}\|_2^2 + 2(\mathbf{1}^\top \boldsymbol{u})(\mathbf{1}^\top \boldsymbol{v}) \\ &\le n + 2 \cdot \frac{n}{2} = 2n. \end{aligned} \tag{173}$$

Therefore,

$$\|\mathcal{H}_\beta(P_1) - \mathcal{H}_\beta(P_2)\|_2 \le 2\sqrt{2} \exp(\sqrt{2}\eta\beta) n \beta^2 p_{\max} \|(\Delta \boldsymbol{x}, \Delta \boldsymbol{y})\|_2. \tag{174}$$

Since $H(\boldsymbol{z})$ is a principal submatrix of $\mathcal{H}_\beta(P)$, the same Lipschitz constant applies to $H(\boldsymbol{z})$, which proves the claimed Lipschitz constant. □

Theorem 3 (Local Quadratic Convergence)**.** *Let the sequence $\{\boldsymbol{z}_k\}$ be generated by*

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \hat{H}_k^{-1}\nabla f(\boldsymbol{z}_k), \tag{175}$$

where $\hat{H}_k$ is sparsified from the Hessian matrix $H_k = H(\boldsymbol{z}_k)$ by the fast sparse scheme. The sequence $\{\boldsymbol{z}_k\}$ converges quadratically to the minimizer $\boldsymbol{z}^*$ in $\mathcal{N}_\delta(\boldsymbol{z}^*)$, where

$$\delta = \frac{e^{-1/e}}{6\beta} \cdot \frac{n - \sqrt{n(n-1)}p^{\mathcal{L}}e^{\beta\Delta C}}{\sqrt{2}\,n\,(a_{\max}b_{\max} + e^{\beta\Delta C} - 1) + sG_n e^{\beta\Delta C}}. \tag{176}$$

where $\Delta_C := \max_{i,j,k,l}(c_{ij} + c_{kl} - c_{il} - c_{kj})$, $G_n = \sqrt{n(n-1)}$ and $s = \max(a_{\max}, b_{\max})$.

Proof. Let

$$\mathscr{A} := \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^n : y_n = 0\} \tag{177}$$

be the gauge-fixed subspace. Since the potential function is invariant along the direction $(\mathbf{1}, -\mathbf{1})$, we analyze Newton's method on this subspace and write the corresponding reduced coordinate as

$$\boldsymbol{z} = (\boldsymbol{x}, \tilde{\boldsymbol{y}}) \in \mathbb{R}^{2n-1}. \tag{178}$$

Let $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathscr{A}$ be the minimizer of the potential function defined in (13) restricted to $\mathscr{A}$, and write

$$\boldsymbol{z}^* = (\boldsymbol{x}^*, \tilde{\boldsymbol{y}}^*). \tag{179}$$

We first show that the radius stated in the theorem is contained in the local quadratic convergence region of the gauge-fixed Newton method. By Theorem 2, the truncated reduced Hessian satisfies

$$\lambda_{\min}(\hat{H}(\boldsymbol{z}^*)) \geq \kappa_n \beta p^{\mathcal{L}}. \tag{180}$$

Next we bound the error of the Hessian approximation, $\|H_k - \hat{H}_k\|$, on $\mathcal{N}_\eta(\boldsymbol{z}^*)$. Actually the error of Hessian approximation is gained from $D_k = \widetilde{P}_k - \widehat{P}_k$, then

$$\|H_k - \hat{H}_k\|_2 = \beta\|D_k\|_2 \leq \beta\|D_k\|_F = \beta\sqrt{\sum_{i,j} d_{ij}^{(k)2}} \leq \beta\sqrt{n(n-1)}\tau_k, \tag{181}$$

where τ_k is the threshold used in the sparsification of P_k. Since $\nabla f(\boldsymbol{z}^*) = 0$, we have

$$\tau_k \leq \|\nabla f(\boldsymbol{z}_k) - \nabla f(\boldsymbol{z}^*)\|_2 \leq \sup_{\|\boldsymbol{z}-\boldsymbol{z}^*\|_2 \leq \eta} \|H(\boldsymbol{z})\|_2 \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2. \tag{182}$$

For any $\boldsymbol{w} = (\boldsymbol{u}, \boldsymbol{v})$, using $(u_i + v_j)^2 \leq 2u_i^2 + 2v_j^2$ we obtain

$$\begin{aligned} \boldsymbol{w}^\top H(\boldsymbol{z})\boldsymbol{w} = \beta\sum_{i,j} P_{ij}(u_i + v_j)^2 &\leq 2\beta\sum_{i,j} P_{ij}(u_i^2 + v_j^2) \\ &= 2\beta\left(\sum_i (P\mathbf{1})_i u_i^2 + \sum_j (P^\top\mathbf{1})_j v_j^2\right) \\ &\leq 2\beta\max\{\|P\mathbf{1}\|_\infty, \|P^\top\mathbf{1}\|_\infty\}\|\boldsymbol{w}\|_2^2. \end{aligned} \tag{183}$$

Since we use the parametrization

$$P_{ij}(\boldsymbol{z}) = P_{ij}^* \exp\big(\beta(x_i - x_i^*) + \beta(y_j - y_j^*)\big), \tag{184}$$

and $\|\boldsymbol{z} - \boldsymbol{z}^*\|_2 \leq \eta$, we have

$$|(x_i - x_i^*) + (y_j - y_j^*)| \leq \sqrt{2}\eta. \tag{185}$$

Therefore, $P_{ij}(z) \leq e^{\sqrt{2}\eta\beta}P_{ij}^*$. Using $P(\boldsymbol{z}^*)\mathbf{1} = \boldsymbol{a}$ and $P(\boldsymbol{z}^*)^\top\mathbf{1} = \boldsymbol{b}$,

$$(P(\boldsymbol{z})\mathbf{1})_i \leq e^{\sqrt{2}\eta\beta}a_i, \qquad (P(\boldsymbol{z})^\top\mathbf{1})_j \leq e^{\sqrt{2}\eta\beta}b_j. \tag{186}$$

Hence, with $s = \max(\max(\boldsymbol{a}), \max(\boldsymbol{b}))$, we obtain

$$\sup_{\|\boldsymbol{z}-\boldsymbol{z}^*\|\leq\eta} \|H(\boldsymbol{z})\|_2 \leq 2\beta e^{\sqrt{2}\eta\beta} s. \tag{187}$$

Therefore, we have

$$\|H_k - \hat{H}_k\|_2 \leq 2\beta^2 G_n e^{\sqrt{2}\eta\beta} s \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2. \tag{188}$$

By the proof of Lemma 7 and (188), all $\boldsymbol{z}_k \in \mathcal{N}_\eta(\boldsymbol{z}^*)$ satisfy

$$\begin{aligned} \|H_k - H(\boldsymbol{z}^*)\|_2 &\leq 2\sqrt{2} e^{\sqrt{2}\eta\beta} n\beta^2 p_{\max} \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2, \\ \|H_k - \hat{H}_k\|_2 &\leq 2\beta^2 G_n e^{\sqrt{2}\eta\beta} s \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2, \end{aligned} \tag{189}$$

Using triangle inequality, we have

$$\begin{aligned} \|\hat{H}_k - \hat{H}(\boldsymbol{z}^*)\|_2 &\leq \|\hat{H}_k - H_k\|_2 + \|H_k - \hat{H}(\boldsymbol{z}^*)\|_2 \\ &\leq 2\beta^2 G_n e^{\sqrt{2}\eta\beta} s \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2 + \|H_k - H(\boldsymbol{z}^*)\|_2 \\ &\leq 2\beta^2 G_n e^{\sqrt{2}\eta\beta} s \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2 + 2\sqrt{2} e^{\sqrt{2}\eta\beta} n\beta^2 p_{\max} \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2 \\ &:= L_{\eta'} \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2, \end{aligned} \tag{190}$$

where $L_{\eta'} = 2\beta^2 G_n e^{\sqrt{2}\eta\beta} s + 2\sqrt{2} e^{\sqrt{2}\eta\beta} n\beta^2 p_{\max}$. Thus the truncated approximate Newton estimate (Nocedal & Wright, 2006) gives

$$\frac{1}{2L_{\eta'} \|\hat{H}(\boldsymbol{z}^*)^{-1}\|_2} \geq \frac{\kappa_n p^{\mathcal{L}}}{4\beta\left(\sqrt{2} n p_{\max} + s G_n\right)} e^{-\sqrt{2}\eta\beta} = \frac{1}{4\beta} e^{-\sqrt{2}\eta\beta} R, \tag{191}$$

where $R = \kappa_n p^{\mathcal{L}} / \left(\sqrt{2} n p_{\max} + s G_n\right)$. We choose η so that

$$\eta = \frac{1}{4\beta} e^{-\sqrt{2}\eta\beta} R. \tag{192}$$

Equivalently,

$$\sqrt{2}\eta\beta e^{\sqrt{2}\eta\beta} = \frac{\sqrt{2}}{4} R. \tag{193}$$

Use inequality $x \geq e^{-1/e} \ln(1 + x e^x)$ and replace x by $\sqrt{2}\eta\beta$, we can get

$$\sqrt{2}\eta\beta \geq e^{-1/e} \ln(1 + \frac{\sqrt{2}}{4} R), \tag{194}$$

which implies

$$\eta \geq \frac{e^{-1/e} \ln(1 + \frac{\sqrt{2}}{4} R)}{\sqrt{2}\beta} \geq \frac{e^{-1/e}}{\sqrt{2}\beta} \cdot \frac{\sqrt{2} R}{4 + \sqrt{2} R} \geq \frac{e^{-1/e}}{6\beta} R, \tag{195}$$

where the last inequality follows from $0 < R \leq 1$ and $4 + \sqrt{2}R \leq 6$. Moreover, for each i, j,

$$p_{ij} = 1 - \sum_{(k,l)\neq(i,j)} p_{kl} \leq 1 - \frac{1}{e^{\beta\Delta C}} \sum_{(k,l)\neq(i,j)} a_k b_l. \tag{196}$$

In particular,

$$p_{ij} \leq \frac{a_i b_j}{e^{\beta\Delta C}} + 1 - \frac{1}{e^{\beta\Delta C}}. \tag{197}$$

Therefore,

$$p_{\max} \leq \frac{a_{\max} b_{\max}}{e^{\beta\Delta C}} + 1 - \frac{1}{e^{\beta\Delta C}}. \tag{198}$$

So we can estimate R as follows

$$R = \frac{\kappa_n p^{\mathcal{L}}}{\sqrt{2} n p_{\max} + s G_n} \geq \frac{\kappa_n p^{\mathcal{L}} e^{\beta\Delta C}}{\sqrt{2}\, n \left(a_{\max} b_{\max} + e^{\beta\Delta C} - 1\right) + s G_n e^{\beta\Delta C}}. \tag{199}$$

Combine (195) and (199) we can let

$$\delta = \frac{e^{-1/e}}{6\beta} \cdot \frac{\kappa_n p^{\mathcal{L}} e^{\beta\Delta C}}{\sqrt{2}\, n \left(a_{\max} b_{\max} + e^{\beta\Delta C} - 1\right) + s G_n e^{\beta\Delta C}}. \tag{200}$$

Consequently, every initial point in $\mathcal{N}_\delta(\boldsymbol{z}^*)$ lies inside the local quadratic convergence region of the gauge-fixed Newton method.

□

F PROOF FOR GLOBAL CONVERGENCE AND COMPLEXITY

Lemma 8. *Let $f(\boldsymbol{z})$ be defined in (13), and the sequence of search directions $\{\boldsymbol{p}_k\}$ be generated by the Sinkhorn scaling. Then in each Sinkhorn block update, direction $\boldsymbol{p}_k$ with unit step size $\alpha_k = 1$ satisfies the following conditions:*

$$f(\boldsymbol{z}_k + \boldsymbol{p}_k) \leq f(\boldsymbol{z}_k) + c_1 \nabla f(\boldsymbol{z}_k)^\top \boldsymbol{p}_k, \tag{201}$$

$$\nabla f(\boldsymbol{z}_k + \boldsymbol{p}_k)^\top \boldsymbol{p}_k \geq c_2 \nabla f(\boldsymbol{z}_k)^\top \boldsymbol{p}_k, \tag{202}$$

for constants $0 < c_1 < c_2 < 1$.

Proof. Since Sinkhorn algorithm is a Block Coordinate Descent, (201) is naturally satisfied on the compact set $\mathcal{L}_0$ defined in Lemma 4. Therefore, we only need to prove that the Sinkhorn iteration satisfies (202). Let us analyze a single step of updating the $\boldsymbol{y}$ block while keeping the $\boldsymbol{x}$ block fixed, and symmetric argument applies to the update of the $\boldsymbol{x}$ block.

A Sinkhorn iteration updates the $\boldsymbol{y}$ block by solving the subproblem:

$$\tilde{\boldsymbol{y}}_{k+1} = \arg\min_{\boldsymbol{y} \in \mathbb{R}^{n-1}} f(\boldsymbol{x}_{k+1}, \boldsymbol{y}). \tag{203}$$

The first-order optimality condition for this subproblem requires the partial gradient with respect to $\boldsymbol{y}$ is zero, i.e.,

$$\nabla_{\boldsymbol{y}} f(\boldsymbol{x}_{k+1}, \tilde{\boldsymbol{y}}_{k+1}) = 0. \tag{204}$$

The search direction $\boldsymbol{p}_k$ associated with this update has a block structure $\boldsymbol{p}_k = (0, \boldsymbol{p}_k^{\boldsymbol{y}})$, as it only modifies the $\boldsymbol{y}$ components. Consequently, the projection of the new gradient $\nabla f(\boldsymbol{z}_{k+1})$ onto the search direction $\boldsymbol{p}_k$ is:

$$\nabla f(\boldsymbol{z}_{k+1})^\top \boldsymbol{p}_k = \nabla_{\boldsymbol{x}} f(\boldsymbol{z}_{k+1})^\top \cdot 0 + \nabla_{\boldsymbol{y}} f(\boldsymbol{z}_{k+1})^\top \boldsymbol{p}_k^{\boldsymbol{y}} = 0. \tag{205}$$

Since the Sinkhorn iteration employs a descent direction, we have $\nabla f(\boldsymbol{z}_k)^\top \boldsymbol{p}_k < 0$. Combining with (205) it leads to

$$c_2 \nabla f(\boldsymbol{z}_k)^\top \boldsymbol{p}_k < 0 = \nabla f(\boldsymbol{z}_{k+1})^\top \boldsymbol{p}_k, \tag{206}$$

which satisfies the (202) for any $c_2 \in (c_1, 1)$. Thus, the conditions of this Lemma are always met by every single Sinkhorn iterations, which implies Sinkhorn fallback satisfies (201) and (202). □

Lemma 9. *Let $f(\boldsymbol{z})$ be defined in (13), the sequences $\{\boldsymbol{z}_k\}$ and $\{\alpha_k\}$ be generated by Algorithm 2, and p_k be the search direction. Then, the function $f(\boldsymbol{z})$ and the generated sequences possess the following properties:*

- *$f(\boldsymbol{z})$ is bounded below, i.e., there exists a scalar $\bar{f}$ such that $f(\boldsymbol{z}) \geq \bar{f}$ for all $\boldsymbol{z}$.*
- *$f(\boldsymbol{z})$ is continuously differentiable.*
- *The gradient of f, ∇f, is Lipschitz continuous on the sublevel set $\mathcal{L}_0$ defined in Lemma 4, i.e., there exists a constant $l_{\nabla f} > 0$ such that*

$$\|\nabla f(\boldsymbol{z}_x) - \nabla f(\boldsymbol{z}_y)\|_2 \leq l_{\nabla f} \|\boldsymbol{z}_x - \boldsymbol{z}_y\|_2, \quad \forall\, \boldsymbol{z}_x, \boldsymbol{z}_y \in \mathcal{L}_0. \tag{207}$$

- *The step size α_k is determined by a line search procedure that ensures the Wolfe conditions are satisfied for all k:*

$$f(\boldsymbol{z}_k + \alpha_k p_k) \leq f(\boldsymbol{z}_k) + c_1 \alpha_k \nabla f(\boldsymbol{z}_k)^\top p_k, \tag{208}$$

$$\nabla f(\boldsymbol{z}_k + \alpha_k p_k)^\top p_k \geq c_2 \nabla f(\boldsymbol{z}_k)^\top p_k, \tag{209}$$

for constants $0 < c_1 < c_2 < 1$. Specifically, $p_k = \Delta \boldsymbol{z}_k$ when the line search accepts a Newton step satisfying the Wolfe conditions.

Consequently, the sequence of gradients $g_k = \nabla f(\boldsymbol{z}_k)$ converges to zero, that is,

$$\lim_{k \to \infty} \|g_k\|_2 = 0. \tag{210}$$

Proof. By the properties of the (1), the first three conditions are already met. Algorithm 2 chooses Sinkhorn iterations when the line search fails to satisfy the Wolfe conditions. By Lemma 8, the Sinkhorn fallback, being performed within the same gauge-fixed subspace as the Newton iteration after line-search failure, also satisfies the Wolfe conditions. Hence, all four conditions hold for Algorithm 2.

Given that $f(\boldsymbol{z})$ is bounded below, continuously differentiable with a Lipschitz continuous gradient, and the iterates satisfy the Wolfe conditions, we can apply Zoutendijk's Theorem, which states that:

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|g_k\|_2^2 < \infty, \tag{211}$$

where θ_k is the angle between the search direction $\Delta \boldsymbol{z}_k$ and the negative gradient $-g_k$, defined by $\cos\theta_k = \frac{-g_k^\top \Delta \boldsymbol{z}_k}{\|g_k\|_2 \|\Delta \boldsymbol{z}_k\|_2}$. The convergence $\lim_{k\to\infty} \|g_k\| = 0$ follows if $\cos\theta_k$ is uniformly bounded by positive constant.

When the line search finds a step size α_k satisfying the Wolfe conditions, the search direction is given by $\Delta \boldsymbol{z}_k = -\hat{H}_k^{-1} g_k$, where $\hat{H}_k$ is a positive definite matrix. From Gershgorin's Circle Theorem we know that the largest eigenvalue of $\hat{H}_k$ is bounded by the upper bound of $\|H(\boldsymbol{z}_k)\|_2$, which can be found in (187). Combine with Theorem 2, there exists a constant $c_{\text{line search}} > 0$ such that $\kappa(\hat{H}_k) \le c_{\text{line search}}$.

Then we establish a positive lower bound for $\cos\theta_k$:

$$\cos\theta_k = \frac{-g_k^\top(-\hat{H}_k^{-1} g_k)}{\|g_k\|_2 \|\hat{H}_k^{-1} g_k\|_2} = \frac{g_k^\top \hat{H}_k^{-1} g_k}{\|g_k\|_2 \|\hat{H}_k^{-1} g_k\|_2}. \tag{212}$$

Using the Rayleigh quotient and properties of matrix norms, we bound the numerator and denominator:

- Numerator: $g_k^\top \hat{H}_k^{-1} g_k \ge \lambda_{\min}(\hat{H}_k^{-1}) \|g_k\|_2^2 = \frac{1}{\lambda_{\max}(\hat{H}_k)} \|g_k\|_2^2$.
- Denominator: $\|\hat{H}_k^{-1} g_k\|_2 \le \|\hat{H}_k^{-1}\|_2 \|g_k\|_2 = \lambda_{\max}(\hat{H}_k^{-1}) \|g_k\|_2 = \frac{1}{\lambda_{\min}(\hat{H}_k)} \|g_k\|_2$.

Combining these inequalities yields:

$$\cos\theta_k \ge \frac{\frac{1}{\lambda_{\max}(\hat{H}_k)} \|g_k\|_2^2}{\|g_k\|_2 \cdot \frac{1}{\lambda_{\min}(\hat{H}_k)} \|g_k\|_2} = \frac{\lambda_{\min}(\hat{H}_k)}{\lambda_{\max}(\hat{H}_k)} = \frac{1}{\kappa(\hat{H}_k)}. \tag{213}$$

When the line search fails, Algorithm 2 employs a Sinkhorn step as a fallback. Now we demonstrate that this step can also be framed as a preconditioned gradient descent, where the corresponding preconditioner has a uniformly bounded condition number.

After each exact Sinkhorn block update, its gradient vanishes at the start of the next alternating update, except for Newton-to-Sinkhorn switch step. Even if dynamic Newton method repeatedly falls back to Sinkhorn, the global convergence of Sinkhorn scaling ensures eventual entry into the local convergence region of Theorem 3. Thus, these transition updates occur only finitely many times and do not affect the asymptotic convergence analysis. Therefore, for each such Sinkhorn block update, the direction $\boldsymbol{p}_k$ can be written as

$$\boldsymbol{p}_k = -\tilde{H}_k^{-1} g_k, \tag{214}$$

where $\tilde{H}_k$ is the matrix given by:

$$\tilde{H}_k = \beta \begin{pmatrix} \operatorname{diag}(\mathcal{D}(P_k \mathbf{1}_n, \boldsymbol{a})) & \mathbf{0} \\ \mathbf{0} & \operatorname{diag}(\mathcal{D}(\tilde{P}_k^\top \mathbf{1}_n, \tilde{\boldsymbol{b}})) \end{pmatrix} \in \mathbb{R}^{(2n-1)\times(2n-1)}, \tag{215}$$

$\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y}) \oslash (\ln \boldsymbol{x} - \ln \boldsymbol{y})$, $\mathcal{D}(\boldsymbol{x}, \boldsymbol{x}) = x$ and P_k is the transport plan at the k-th iteration.

Let us analyze the properties of $\tilde{H}_k$. Since the target marginals $\boldsymbol{a}$ and $\boldsymbol{b}$ consist of strictly positive elements, the row sums and column sums of the transport plan P_k also remain strictly positive

throughout the iterations. Consequently, $\tilde{H}_k$ is a diagonal matrix with strictly positive entries, which ensures it is positive definite.

Let $r_i(\boldsymbol{z}) = (P(\boldsymbol{z})\mathbf{1})_i$, $c_j(\boldsymbol{z}) = (P(\boldsymbol{z})^\top \mathbf{1})_j$. Since $\mathcal{L}_0$ is compact and r_i, c_i are both continuous function, we can find that $\exists\, r_{\min}, r_{\max}, c_{\min}, c_{\max} > 0$, for any $\boldsymbol{z} \in \mathcal{L}_0$ such that

$$r_{\min} \le r_i(\boldsymbol{z}) \le r_{\max} < \infty, \quad c_{\min} \le c_i(\boldsymbol{z}) \le c_{\max} < \infty. \tag{216}$$

Therefore, with the definition of $\mathcal{D}$ we have

$$0 < \beta \min\{a_{\min}, b_{\min}, r_{\min}, c_{\min}\} \le \lambda(\tilde{H}_k) \le \beta \max\{a_{\max}, b_{\max}, r_{\max}, c_{\max}\} < \infty, \tag{217}$$

which implies $\kappa(\tilde{H}_k)$, the condition number of $\tilde{H}_k$, can be uniformly bounded by a positive constant c_{Sinkhorn}. Substituting this result into the same inequality for $\cos\theta_k$ as in the former case, we obtain:

$$\cos\theta_k \ge \frac{1}{\kappa(\tilde{H}_k)} \ge \frac{1}{c_{\text{Sinkhorn}}} > 0. \tag{218}$$

Since $\kappa(\hat{H}_k) \le c_{\text{line search}}$, $\kappa(\tilde{H}_k) \le c_{\text{Sinkhorn}}$, let $c = \max(c_{\text{line search}}, c_{\text{Sinkhorn}})$, we have $\cos\theta_k \ge \frac{1}{c} > 0$, which provides a uniform positive lower bound. Applying this to Zoutendijk's condition:

$$\infty > \sum_{k=0}^{\infty} \cos^2\theta_k \|g_k\|_2^2 \ge \sum_{k=0}^{\infty} \frac{1}{c^2}\|g_k\|_2^2 = \frac{1}{c^2}\sum_{k=0}^{\infty}\|g_k\|_2^2. \tag{219}$$

This implies that the series $\sum_{k=0}^{\infty}\|g_k\|_2^2$ must converge. Therefore,

$$\lim_{k\to\infty} \|g_k\|_2^2 = 0, \tag{220}$$

which implies

$$\lim_{k\to\infty} \|g_k\|_2 = 0. \tag{221}$$

□

Theorem 4 (Global Convergence)**.** *The sequence $\{\boldsymbol{z}_k\}$ generated by Algorithm 2 converges globally to the minimizer $\boldsymbol{z}^*$.*

Proof. Now we begin to prove global convergence. Let $\boldsymbol{z}^*$ be the global optimum of (13). Notice that f and ∇f are both Lipschitz continuous on the sublevel set $\mathcal{L}_0$ defined in Lemma 4, i.e. $\exists\, l_f, l_{\nabla f} > 0$, for any $\boldsymbol{z}_k \in \mathcal{L}_0$ such that

$$|f(\boldsymbol{z}_k) - f(\boldsymbol{z}^*)| \le l_f \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2, \tag{222}$$

$$\|\nabla f(\boldsymbol{z}_k) - \nabla f(\boldsymbol{z}^*)\|_2 \le l_{\nabla f} \|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2. \tag{223}$$

Firstly, we show that the smallest eigenvalue of the Hessian matrix of the function f defined by (13) has a positive lower bound $\tilde{\lambda}$ in sublevel set $\mathcal{L}_0$ defined in Lemma 4, i.e. for all $\boldsymbol{z} \in \mathcal{L}_0$, $\lambda_{\min}(H(\boldsymbol{z})) \ge \tilde{\lambda}$, which yields

$$(\nabla f(\boldsymbol{z}_k) - \nabla f(\boldsymbol{z}^*))^\top (\boldsymbol{z}_k - \boldsymbol{z}^*) \ge \tilde{\lambda}\|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2^2, \tag{224}$$

which implies

$$\|\nabla f(\boldsymbol{z}_k)\|_2 = \|\nabla f(\boldsymbol{z}_k) - \nabla f(\boldsymbol{z}^*)\|_2 \ge \tilde{\lambda}\|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2. \tag{225}$$

On the other hand, by Taylor's theorem,

$$f(\boldsymbol{z}_k) = f(\boldsymbol{z}^*) + (\nabla f(\boldsymbol{z}^*))^\top (\boldsymbol{z}_k - \boldsymbol{z}^*) + \frac{1}{2}(\boldsymbol{z}_k - \boldsymbol{z}^*)^\top H(\tilde{\boldsymbol{z}})(\boldsymbol{z}_k - \boldsymbol{z}^*), \tag{226}$$

where $\tilde{\boldsymbol{z}}$ is a point between $\boldsymbol{z}_k$ and $\boldsymbol{z}^*$, so we have

$$|f(\boldsymbol{z}_k) - f(\boldsymbol{z}^*)| \ge \frac{\tilde{\lambda}}{2}\|\boldsymbol{z}_k - \boldsymbol{z}^*\|_2^2. \tag{227}$$

Since f is nonincreasing and bounded below, $\exists\, \epsilon \ge 0$ such that

$$\lim_{k\to\infty} |f(\boldsymbol{z}_k) - f(\boldsymbol{z}^*)| = \epsilon. \tag{228}$$

Combining (222) and (228), for sufficiently large k we have

$$\frac{1}{2}l_f^{-1}\epsilon \leq l_f^{-1}\left|f\left(\boldsymbol{z}_k\right)-f\left(\boldsymbol{z}^*\right)\right| \leq \left\|\boldsymbol{z}_k-\boldsymbol{z}^*\right\|_2 \leq \tilde{\lambda}^{-1}\left\|\nabla f\left(\boldsymbol{z}_k\right)\right\|_2. \tag{229}$$

Combining (223), (227) and (228), for sufficiently large k we have

$$\frac{\tilde{\lambda}}{2l_{\nabla f}^2}\left\|\nabla f\left(\boldsymbol{z}_k\right)\right\|_2^2 \leq \frac{\tilde{\lambda}}{2}\left\|\boldsymbol{z}_k-\boldsymbol{z}^*\right\|_2^2 \leq \left|f\left(\boldsymbol{z}_k\right)-f\left(\boldsymbol{z}^*\right)\right| \to \epsilon. \tag{230}$$

Since our line search method and Sinkhorn iteration both satisfy Wolfe condition, applying Lemma 9 we get that $\epsilon = 0$. Combine (229) and (230) with squeeze theorem, we have

$$\lim_{k\to\infty}\left\|\boldsymbol{z}_k-\boldsymbol{z}^*\right\|_2 = 0. \tag{231}$$

□

Theorem 5 (Global Convergence and Newton-Refinement Complexity)**.** *For entropic OT (1), Algorithm 3 converges globally. The conservative Sinkhorn–Newton procedure achieves a marginal tolerance below ρ_{th} with total computational complexity*

$$\mathcal{O}\left(n^2\left[\frac{\beta\|C\|_\infty}{\tilde{\lambda}\delta}+(1+\tau_{avg}+k_{CG}\tau_{avg})\ln\ln\left(\frac{2\beta \mathrm{e}^{\sqrt{2}\delta\beta}s}{\rho_{th}}\right)\right]\right), \tag{232}$$

where $\tilde{\lambda}$ is a lower bound on the eigenvalues of $\{H_k\}$, δ is the quadratic-convergence radius in Theorem 3, τ_{avg} is the average sparsity, k_{CG} is the number of PCG iterations per Newton step, and $s = \max(\max(\boldsymbol{a}), \max(\boldsymbol{b}))$.

Proof. The global convergence of the PSN algorithm follows from the global convergence of its two components: the Proximal-Sinkhorn (PS) and Proximal-Newton (PN). The convergence of PN is established by Theorem 4, while that of PS is guaranteed by the global convergence of the Sinkhorn algorithm (Cuturi, 2013). Hence, PSN is globally convergent.

The total complexity is the sum of the costs from the Sinkhorn algorithm and the dynamic Newton refinement stage:

$$C_{\mathrm{Total}} = C_{\mathrm{Stage1}} + C_{\mathrm{Stage2}}. \tag{233}$$

Let marginal error be $\rho(P) = \|\nabla f(\boldsymbol{z}_P)\|_2$, the marginal error and the dual-variable error satisfy

$$\|\boldsymbol{z}_P - \boldsymbol{z}^*\|_2 \leq \frac{1}{\tilde{\lambda}}\rho(P), \tag{234}$$

where $\boldsymbol{z}_P$ is the dual variable corresponding to P. Conversely, within the local Newton convergence region, we bound the marginal error in terms of the dual-variable error. By the Lipschitz continuity of ∇f, we have

$$\|\nabla f(\boldsymbol{z}_P) - \nabla f(\boldsymbol{z}^*)\|_2 \leq \sup_{\|\boldsymbol{z}-\boldsymbol{z}^*\|_2\leq\delta}\|H(\boldsymbol{z})\|_2\|\boldsymbol{z}_P-\boldsymbol{z}^*\|_2. \tag{235}$$

From (187) we obtain

$$\rho(P) \leq 2\beta \mathrm{e}^{\sqrt{2}\delta\beta}s\|\boldsymbol{z}_P-\boldsymbol{z}^*\|_2. \tag{236}$$

Stage 1: Sinkhorn. To guarantee that the iterate enters the local quadratic convergence zone, Sinkhorn must reduce the dual variable error to δ. Applying (234), if marginal error satisfies

$$\rho(P) \leq \tilde{\lambda}\delta, \tag{237}$$

then the corresponding dual variable lies within the δ-neighborhood. Luo et al. (2023) has proven that using Sinkhorn algorithm to reduce the marginal error to $\tilde{\lambda}\delta$ requires

$$\mathcal{O}\left(\frac{\beta\|C\|_\infty}{\tilde{\lambda}\delta}\right) \tag{238}$$

iterations. Since each iteration uses matrix-vector multiplications with cost $\mathcal{O}(n^2)$, the first-stage cost is

$$C_{\text{Stage1}} = \mathcal{O}\left(\frac{\beta n^2 \|C\|_\infty}{\tilde{\lambda}\delta}\right). \tag{239}$$

Stage 2: dynamic Newton. Once the iterate is inside the local quadratic convergence zone, the cost is determined by the number of PCG steps required per Newton iteration and by the total number of Newton iterations.

Each Newton step solves

$$\hat{H}\Delta \boldsymbol{z} = -\nabla f(\boldsymbol{z}) \tag{240}$$

with PCG. Under IC preconditioning, suppose the majority of eigenvalues are clustered in a bulk interval $[a, b]$ with p extreme outliers. The PCG error then decays proportionally to

$$\left(\frac{\sqrt{\kappa_{\text{eff}}} - 1}{\sqrt{\kappa_{\text{eff}}} + 1}\right)^{k-p}, \tag{241}$$

where $\kappa_{\text{eff}} = b_{\max}/a_{\min}$, and $a_{\min}$ and $b_{\max}$ are the global minimum left endpoint and maximum right endpoint of the bulk intervals, respectively. Thus, to meet a local PCG tolerance ϵ_d, the number of PCG iterations satisfies

$$k_{CG} = \mathcal{O}\left(p + \sqrt{\kappa_{\text{eff}}} \ln\left(\frac{1}{\epsilon_d}\right)\right). \tag{242}$$

For each Newton iteration, gradient computation, Hessian construction, and sparsification cost $\mathcal{O}(n^2)$. With average sparsity density τ_{avg}, IC adds $\mathcal{O}(\tau_{\text{avg}} n^2)$ operations and the sparse PCG operations add $\mathcal{O}(k_{CG}\tau_{\text{avg}} n^2)$. Hence the cost per Newton step is

$$C_{\text{step}} = \mathcal{O}\left((1 + \tau_{\text{avg}} + k_{CG}\tau_{\text{avg}})n^2\right). \tag{243}$$

Since the second stage starts inside the local quadratic convergence zone, the number of Newton iterations required to reach final tolerance ϵ is

$$M_{\text{Newton}} = \mathcal{O}\left(\ln\ln\left(\frac{1}{\epsilon}\right)\right). \tag{244}$$

Therefore, one can substitute ϵ by $\rho_{th}/2\beta \mathrm{e}^{\sqrt{2}\delta\beta} s$ and get

$$C_{\text{Stage2}} = M_{\text{Newton}}\, C_{\text{step}} = \mathcal{O}\left((1 + \tau_{\text{avg}} + k_{CG}\tau_{\text{avg}})n^2 \ln\ln\left(\frac{2\beta \mathrm{e}^{\sqrt{2}\delta\beta} s}{\rho_{th}}\right)\right). \tag{245}$$

Combining the two stages gives

$$\begin{aligned} C_{\text{Total}} &= C_{\text{Stage1}} + C_{\text{Stage2}} \\ &= \mathcal{O}\left(\frac{\beta n^2 \|C\|_\infty}{\tilde{\lambda}\delta}\right) + \mathcal{O}\left((1 + \tau_{\text{avg}} + k_{CG}\tau_{\text{avg}})n^2 \ln\ln\left(\frac{2\beta \mathrm{e}^{\sqrt{2}\delta\beta} s}{\rho_{th}}\right)\right) \\ &= \mathcal{O}\left(n^2\left[\frac{\beta\|C\|_\infty}{\tilde{\lambda}\delta} + (1 + \tau_{\text{avg}} + k_{CG}\tau_{\text{avg}}) \ln\ln\left(\frac{2\beta \mathrm{e}^{\sqrt{2}\delta\beta} s}{\rho_{th}}\right)\right]\right), \end{aligned} \tag{246}$$

which proves the claim. □

G THE SPECTRAL DISTRIBUTION OF THE HESSIAN MATRIX

The following Lemma and Theorem characterize the convergence behavior of CG. Lemma 10 shows that the error at each iteration of CG is bounded by a polynomial $q_k(\lambda)$, and Theorem 6 reveals the connection between the polynomial $q_k(\lambda)$ and the spectral distribution of the Hessian matrix H while solving $H\Delta \boldsymbol{z} = -\nabla f$. Furthermore, Theorem 6 also demonstrates that the error vector in each iteration of the CG method remains bounded by the polynomial $q_k(\lambda)$, even after excluding a small number of eigenvalues at one or both extremes. This reveals that the CG method's convergence rate is closely tied to the eigenvalue distribution of the matrix H, potentially leading to superlinear rates.

Lemma 10 (Saad (2003)). *The approximate solution $\boldsymbol{x}_k$ generated by the CG method of solving $H\boldsymbol{x} = \boldsymbol{b}$ at step k satisfies*

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_H \leqslant \min_{q_k \in \mathcal{P}_k^{(0)}} \max_{\lambda \in \lambda(H)} |q_k(\lambda)| \, \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_H \,, \tag{247}$$

where $\boldsymbol{x}^*$ denotes the exact solution of the linear system, $\|\cdot\|_H$ represents the H-norm, $\mathcal{P}_k^{(0)}$ represents the set of all real coefficient polynomials of degree not exceeding k that satisfy $q_k(0) = 1$.

Theorem 6 (Hestenes et al. (1952)). *Considering the error vectors $\boldsymbol{u}_k = \boldsymbol{x}_k - \boldsymbol{x}^*$ defined by Lemma 10, we establish the following bounds:*

(1) for $\lambda(H) \subset \{\lambda_1, \cdots, \lambda_p\} \cup [a_1, b_1]$, where $\lambda_i \in \lambda(H), \lambda_i < a_1$, we have

$$\|\boldsymbol{u}_k\|_H \leqslant F\left(a_1, b_1; k-p\right) \prod_{i=1}^{p} \frac{b_1}{\lambda_i} \|\boldsymbol{u}_0\|_H \,; \tag{248}$$

(2) for $\lambda(H) \subset [a_2, b_2] \cup \{\lambda_1', \cdots, \lambda_p'\}$, where $\lambda_i' \in \lambda(H)$, $\lambda_i' > b_2$, we have

$$\|\boldsymbol{u}_k\|_H \leqslant F\left(a_2, b_2; k-p\right) \|\boldsymbol{u}_0\|_H \,; \tag{249}$$

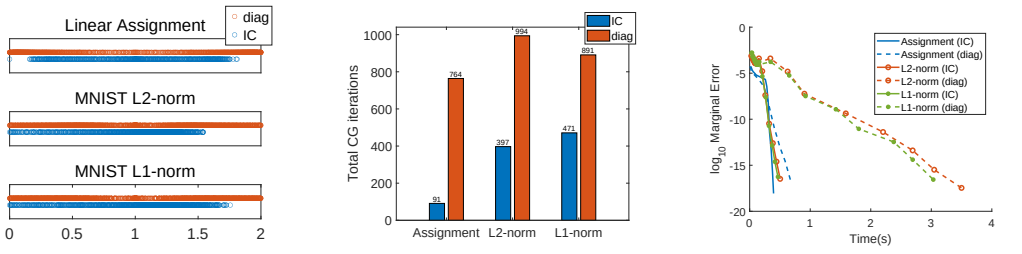
(3) for $\lambda(H) \subset \{\lambda_1, \cdots, \lambda_p\} \cup [a_3, b_3] \cup \{\lambda_1', \cdots, \lambda_p'\}$, where $\lambda_i, \lambda_i' \in \lambda(H), \lambda_i' > b_3, \lambda_i < a_3$, we have

$$\|\boldsymbol{u}_k\|_H \leqslant \frac{1}{4} \prod_{i=1}^{p} \frac{\lambda_i'}{\lambda_i} \left(1 - \frac{\lambda_i}{\lambda_i'}\right)^2 F\left(a_3, b_3; k-2\right) \|\boldsymbol{u}_0\|_H \,, \tag{250}$$

where $F(a, b\,; k) = \left(T_k\left(\frac{b+a}{b-a}\right)\right)^{-1}$, and $T_k(x)$ denotes Chebyshev polynomial of degree k.

H Comparison between incomplete Cholesky and diagonal loading

This section compares the improvements provided by the Incomplete Cholesky (IC) and diagonal loading when applied to the Newton method. Initial approximations are generated by 300 Sinkhorn iterations. Figure 6 shows that the IC preconditioner enhances spectral clustering, significantly reduces CG iterations, and consistently outperforms the diagonal loading across all datasets. Since Theorem 6 shows that CG's per-step error is mainly governed by the clustered bulk of eigenvalues, these few isolated eigenvalues do not noticeably reduce the performance gains.

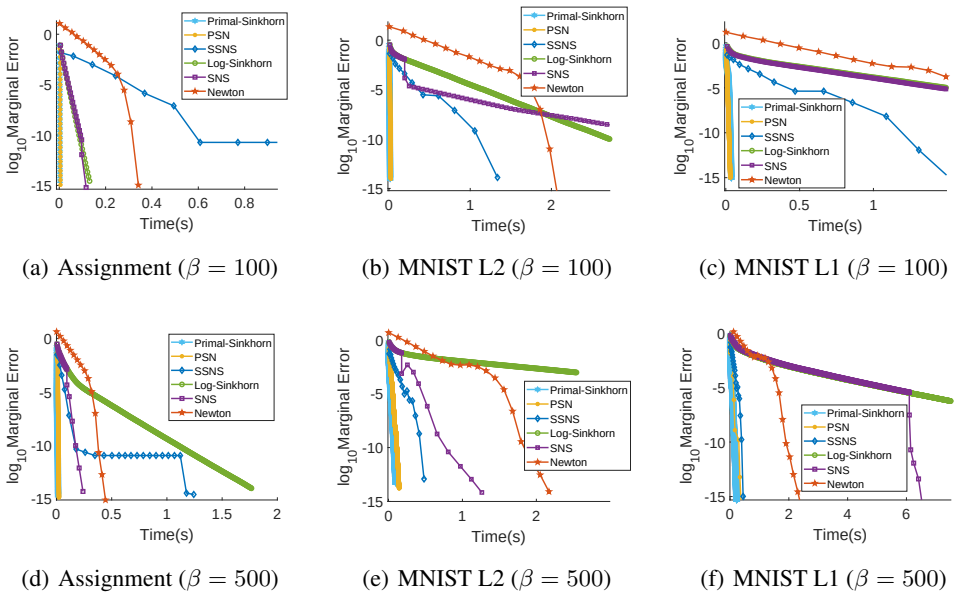


(a) Distribution of eigenvalues (b) CG iterations improved by IC (c) Convergence improved by IC

Figure 6: Comparison between IC preconditioner and diagonal loading.

I Performance on different entropic parameters

We evaluate PSN, SNS, SSNS, and the Newton method across entropic parameters $\beta \in \{100, 500\}$ covering both low and medium regularization. Figure 7 indicates that our proposed algorithm, PSN, exhibits superior robustness and effectiveness when β is large. For smaller values of β, PSN reduces to a combination of the proximal point method and Sinkhorn scaling, and outperforms other Newton-based algorithms as well as the log-Sinkhorn method.



(a) Assignment ($\beta = 100$) (b) MNIST L2 ($\beta = 100$) (c) MNIST L1 ($\beta = 100$)

(d) Assignment ($\beta = 500$) (e) MNIST L2 ($\beta = 500$) (f) MNIST L1 ($\beta = 500$)

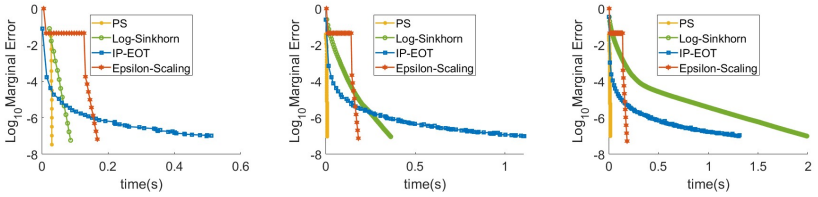
Figure 7: Performance on different β.

J Comparison with other first order algorithms

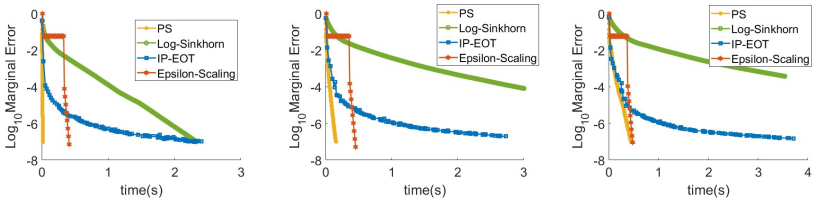
To illustrate the efficiency of our Proximal-Sinkhorn method (PS), we compare it with two classical entropic solvers: log-Sinkhorn and Epsilon-Scaling method (Schmitzer, 2019). We compare the algorithms for β in the range (100, 700), since for larger values second-order methods gain a clear advantage (as shown in Figure 11). As shown in Figure 8, the error curve of the epsilon-scaling method exhibits an initial plateau before decreasing, because it requires a warm-up phase. Our PS method does not suffer from this issue and consistently achieves superior performance.

We compare our PS and PSN against several well-established first order optimization methods. For PSN, we include Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS, (Liu & Nocedal, 1989)) for the quasi-Newton method, the adaptive primal-dual accelerated gradient descent method (ADPAGD, (Dvurechensky et al., 2018)) as an accelerated first-order approach, and the Sinkhorn method implemented as a block coordinate descent method (BCD, (Luo & Tseng, 1992)). We adopted the C++ implementation by Tang & Qiu (2024) for the baseline algorithms[4]. As depicted in Figure 9, PSN consistently delivers superior performance, significantly outperforming these baseline methods across all tested settings.

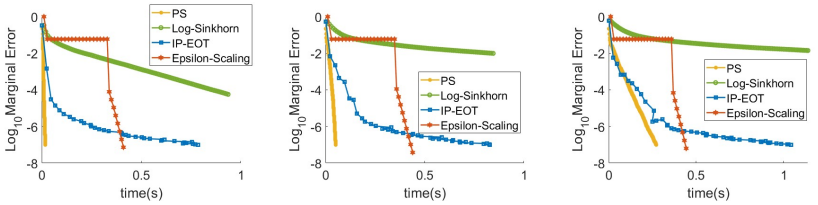
[4]https://github.com/yixuan/regot-python.



(a) Assign. ($\beta = 100$) (b) Assign. ($\beta = 400$) (c) Assign. ($\beta = 700$)

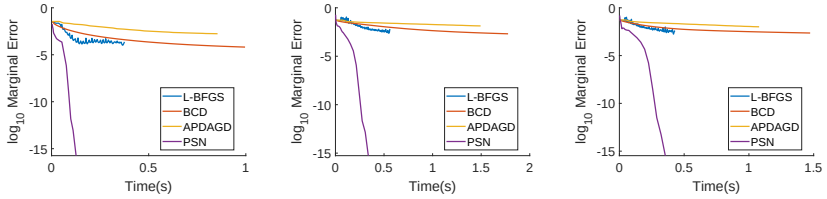


(d) L1. ($\beta = 100$) (e) L1. ($\beta = 400$) (f) L1. ($\beta = 700$)



(g) L2. ($\beta = 100$) (h) L2. ($\beta = 400$) (i) L2. ($\beta = 700$)

Figure 8: The comparison of our PS with other first-order algorithms.



(a) Assign. ($\beta = 1200$) (b) L1. ($\beta = 1300$) (c) L2. ($\beta = 1300$)

Figure 9: The comparison of our PSN with some first-order algorithms.

K Detailed results of Figure 3 (a)-(c), (f)-(h), and Figure 4 (a)-(c)

Table 3: Comparison of iterations and runtime (in seconds) at different accuracy levels with $\beta = 300$. Underlined numbers indicate the iteration count of Newton methods, while non-underlined numbers correspond to the iteration count of IP-EOT/Sinkhorn.

Case	Error	IP-EOT		Primal-Sinkhorn		Log-Sinkhorn		PSN		SSNS		SNS		Newton		SSNS/PSN
		# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	
Assignment $n = 500$	10^{-4}	16	0.01s	12	0.001s	12	0.06s	11	0.005s	6	0.14s	12	0.06s	12	0.32s	28×
	10^{-6}	148	0.08s	25	0.002s	25	0.10s	23	0.006s	7	0.20s	20+1	0.09s	13	0.35s	33×
	10^{-8}	1466	0.76s	43	0.003s	43	0.19s	42	0.007s	8	0.30s	20+2	0.11s	14	0.38s	42×
MNIST L2 $n = 784$	10^{-4}	39	0.04s	176	0.02s	176	1.48s	109	0.02s	10	0.22s	20+1	0.17s	13	1.45s	11×
	10^{-6}	111	0.12s	392	0.04s	392	3.33s	324	0.03s	13	0.38s	20+4	0.4s	14	1.57s	13×
	10^{-8}	671	0.77s	608	0.05s	608	5.16s	540	0.04s	15	0.54s	20+9	0.82s	15	1.69s	14×
MNIST L1 $n = 784$	10^{-4}	57	0.06s	148	0.02s	148	1.25s	114	0.04s	10	0.16s	148	1.26s	13	1.51s	4×
	10^{-6}	296	0.32s	409	0.05s	409	3.50s	354	0.07s	15	0.36s	409	3.45s	15	1.75s	5.1×
	10^{-8}	2948	3.42s	674	0.07s	674	5.75s	617	0.13s	16	0.42s	674	5.69s	16	1.87s	3.2×

Table 4: Comparison of iterations and runtime (in seconds) at different accuracy levels and $\beta = 1200$. Underlined numbers indicate the iteration count of Newton methods, while non-underlined numbers correspond to the iteration count of IP-EOT/Sinkhorn. The last column reports the SSNS/PSN ratio on time.

Case	Error	IP-EOT		Log-Sinkhorn		PSN		SNS		SSNS		Newton		SSNS/PSN
		# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	
Assignment $n = 500$	10^{-4}	26	0.021s	73	0.26s	24+1	0.03s	20+1	0.09s	11	0.2s	break		6.7×
	10^{-6}	236	0.15s	654	2.23s	24+2	0.04s	20+5	0.28s	13	0.26s			6.5×
	10^{-8}	2335	2.93s	9679	33.6s	24+3	0.06s	20+6	0.38s	14	0.30s			5×
MNIST L2 $n = 784$	10^{-4}	77	0.09s	770	4.68s	52+1	0.14s	20+13	1.0s	19	0.42s	14	1.64s	3.0×
	10^{-6}	357	0.41s	1713	10.0s	52+3	0.20s	20+16	1.30s	26	0.78s	16	1.92s	3.9×
	10^{-8}	2776	4.90s	3128	18.0s	52+5	0.26s	20+17	1.41s	28	0.88s	17	2.05s	3.4×
MNIST L1 $n = 784$	10^{-4}	111	0.14s	589	3.39s	52+1	0.13s	589	3.32s	24	0.44s	15	1.86s	3.4×
	10^{-6}	427	0.54s	1540	8.72s	52+3	0.18s	700+1	3.94s	29	0.68s	17	2.11s	3.8×
	10^{-8}	3152	8.94s	3052	17.3s	52+5	0.24s	700+3	4.18s	31	0.78s	18	2.23s	3.3×

Table 5: Comparison of iterations and runtime (in seconds) at different accuracy levels and $\beta = 200 \ln n$ on large scale problem. Underlined numbers indicate the iteration count of Newton methods, while non-underlined numbers correspond to the iteration count of IP-EOT/Sinkhorn.

Case	Error	IP-EOT		Log-Sinkhorn		PSN		SSNS		SNS		Newton		SSNS/PSN	SNS/PSN
		# iters	time	# iters	time	# iters	time	# iters	time	# iters	time	# iters	time		
Assignment $n = 4000$	10^{-4}	13	0.69s	8	2.10s	34	2.29s	7	40.2s	8	2.32s	10	26.4s	18×	1.0×
	10^{-6}	120	6.23s	16	4.08s	34+1	3.26s	9	89.8s	16	4.65s	13	36.6s	27×	1.4×
	10^{-8}	1189	63.1s	27	6.44s	34+2	4.24s	12	204s	20+1	5.33s	15	49.1s	48×	1.3×
MNIST L2 $n = 4096$	10^{-4}	85	4.61s	956	246s	68	5.15s	29	17.7s	20+6	15.5s	15	238s	3.4×	3.0×
	10^{-6}	400	21.6s	2133	489s	68+3	7.06s	42	29.4s	20+9	23.2s	19	310s	4.2×	3.3×
	10^{-8}	693	37.7s	3311	761s	68+4	8.01s	46	33.4s	20+31	77.6s	20	328s	4.2×	9.7×
MNIST L1 $n = 4096$	10^{-4}	124	6.94s	757	189s	68+2	5.71s	32	20.2s	700	164s	16	259s	3.5×	29×
	10^{-6}	484	26.8s	1974	444s	68+4	7.53s	42	28.0s	700+1	165s	20	328s	3.7×	22×
	10^{-8}	3189	207s	3191	718s	68+5	8.37s	48	33.4s	700+4	178s	21	345s	4.0×	21×

L TIME PER ITERATION

We further investigate the runtime performance under different regularization settings. Table 6 and Table 7 present the time per iteration for the first-order and second-order methods, with β set to 300 and 1200, respectively. For two-stage algorithms (PN and SNS), only the Newton phase is recorded in Table 7. From the two tables, we observe that PS and PN achieve among the shortest per-iteration times within the first- and second-order methods, respectively.

Table 6: Mean per-iteration computation time of first-order methods with $\beta = 300$.

Case	IP-EOT	Log-Sink	PS	Epsilon-Scal
Assign $n = 500$	5e-4s	5e-3s	6e-4s	4e-3s
MNIST L2 $n = 784$	1e-3s	8e-3s	6e-4s	1e-2s
MNIST L1 $n = 784$	1e-3s	9e-3s	6e-4s	1e-2s

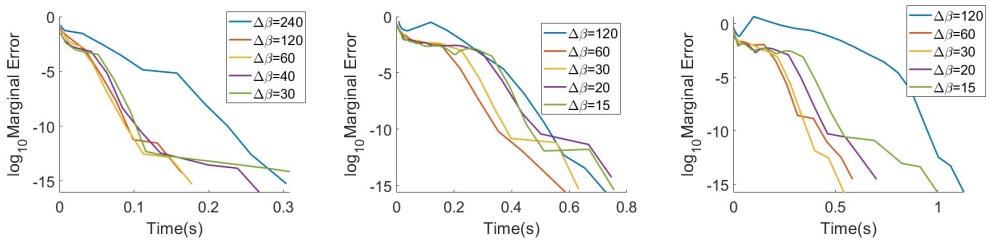
Table 7: Mean per-iteration computation time of second-order methods with $\beta = 1200$.

Case	PN	SSNS	SNS	Newton
Assign $n = 500$	0.03s	0.04s	0.1s	break
MNIST L2 $n = 784$	0.03s	0.03s	0.2s	0.1s
MNIST L1 $n = 784$	0.03s	0.03s	0.3s	0.15s

M DISCUSSION OF THE PROXIMAL STEP SIZE

This section examines the sensitivity of PSN to the proximal step size $\Delta\beta$. According to Theorem 1, the proximal step size $\Delta\beta$ influences only the initialization of the second stage. When Sinkhorn scaling is used in the second stage, $\Delta\beta$ does not affect the convergence rate. Therefore, this section focuses on the sensitivity of the Proximal–Newton (PN) method to the choice of $\Delta\beta$.

As illustrated in Figure 10, the PN algorithm exhibits considerable robustness to the choice of proximal step size, consistently achieving stable convergence across different datasets and values of β. The convergence behavior remains highly similar across a wide range of step sizes, except when $\Delta\beta$ is excessively large—that is, when the number of IP-EOT iterations ℓ is particularly small. Even in such cases, the algorithm only suffers from a slightly slower initial convergence rate, with the



(a) Assign. ($\beta = 500$) (b) L2. ($\beta = 1000$) (c) L1. ($\beta = 2000$)

Figure 10: Impact of the proximal step size $\Delta\beta$ on proximal-newton method.

total computation time remaining comparable. On the other hand, when β is large and the proximal step size is small (corresponding to a large number of IP-EOT iterations ℓ), the computational cost of IP-EOT increases somewhat. Overall, as long as the proximal step size is chosen to ensure the numerical stability of the IP-EOT phase, the algorithm PSN achieves favorable convergence performance.

N Discussion of the switching parameter

The choice of λ aims to select the more efficient solver between Proximal-Sinkhorn (PS) and Proximal-Newton (PN). Accordingly, this section presents the efficiency of PS and PN under different values of β, together with the corresponding changes in the sparsity of $\hat{H}$, providing guidance for choosing an appropriate λ.

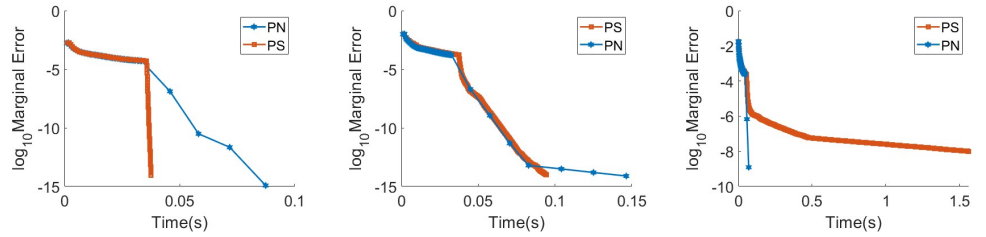
To investigate the choice of λ, we evaluate PS and PN under different values of β and examine the relationship between $\|\hat{H}\|_0$ and runtime. We select three representative settings in which PS is faster, the two methods are comparable, and PN is faster, respectively. As shown in Fig. 11, as β increases, the number of non-zero elements $|\hat{H}|_0$ gradually decreases, and PN progressively outperforms PS. Interestingly, under both L1 and L2 settings, when PS and PN achieve comparable time performance, their sparsity levels are similar even though the values of β differ, which validates the effectiveness of using λ as the switching parameter.

O Truncation Details

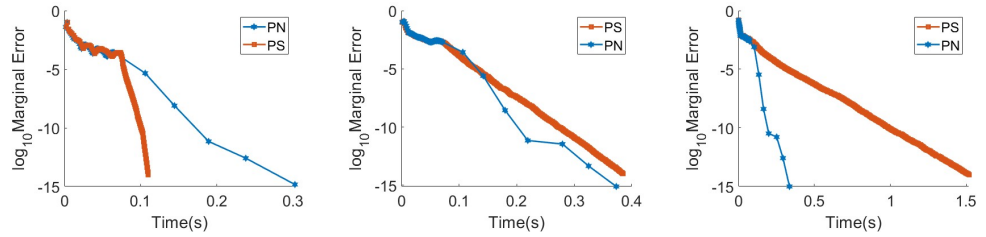
We explore the sparsity of the Hessian matrix under various truncation thresholds and visualize the results in Figure 12. Figure 12 demonstrates that the truncated matrix exhibits significant sparsity. Moreover, under the same truncation threshold, larger values of β lead to a sparser matrix. Interestingly, the resulting sparsity level is nearly linear in the logarithm of the truncation threshold.

P Variability of the methods under different random realizations

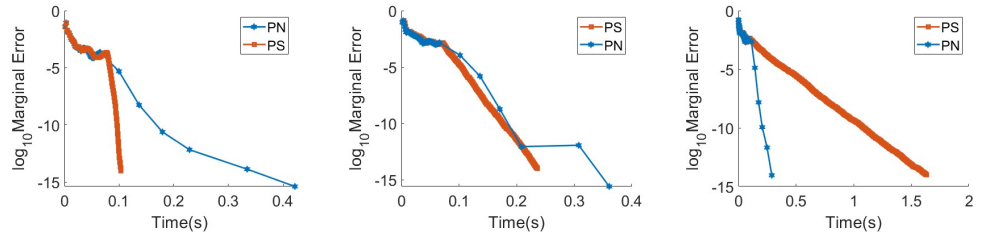
We conduct twenty independent trials and report the resulting variability. Figure 13 shows that Newton and PSN exhibit relatively small variation across random realizations, whereas SNS and SSNS show substantially larger variability.



(a) Assign.($\beta = 200$, $\|\hat{H}\|_0 = 57n$) (b) Assign.($\beta = 705$, $\|\hat{H}\|_0 = 21n$) (c) Assign.($\beta = 1200$, $\|\hat{H}\|_0 = 14n$)



(d) L2.($\beta = 200$, $\|\hat{H}\|_0 = 127n$) (e) L2.($\beta = 600$, $\|\hat{H}\|_0 = 49n$) (f) L2.($\beta = 1200$, $\|\hat{H}\|_0 = 30n$)



(g) L1.($\beta = 100$, $\|\hat{H}\|_0 = 141n$) (h) L1.($\beta = 300$, $\|\hat{H}\|_0 = 55n$) (i) L1.($\beta = 1000$, $\|\hat{H}\|_0 = 24n$)

Figure 11: Performance of the PS and PN algorithms on three datasets for different values of β.

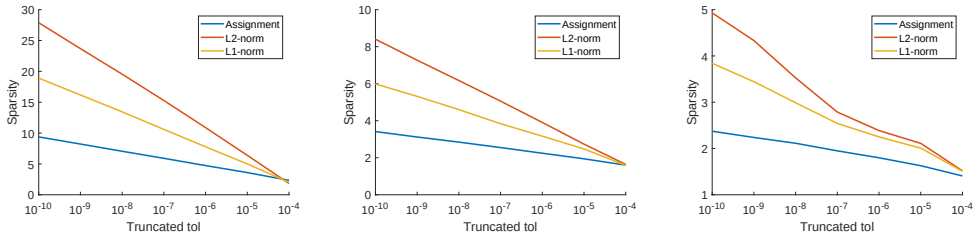
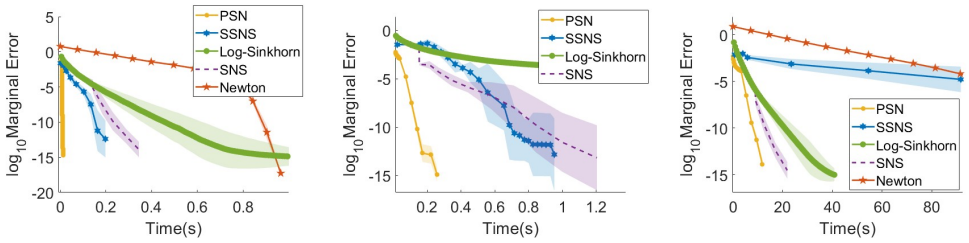


Figure 12: Relationship between truncation threshold and Sparsity. The three subfigures correspond to $\beta = \{1000, 2000, 4000\}$, respectively. The x-axis denotes the truncation threshold, while the y-axis represents the Hessian matrix sparsity as a multiple of n.



(a) Assign. ($n = 500, \beta = 300$) (b) Assign. ($n = 500, \beta = 1200$) (c) Assign. ($n = 4000, \beta = 200 \ln n$)

Figure 13: Performance under random realizations on the linear assignment tasks.