Fast Computation of Optimal Transport via Entropy-Regularized Extragradient Methods

Gen Li* Yanxi Chen† Yuejie Chi‡ H. Vincent Poor† Yuxin Chen*

January 2023; Revised: June 2023

Abstract

Efficient computation of the optimal transport distance between two distributions serves as an algorithm subroutine that empowers various applications. This paper develops a scalable first-order optimization-based method that computes optimal transport to within ε additive accuracy with runtime $\widetilde{O}(n^2/\varepsilon)$, where n denotes the dimension of the probability distributions of interest. Our algorithm achieves the state-of-the-art computational guarantees among all first-order methods, while exhibiting favorable numerical performance compared to classical algorithms like Sinkhorn and Greenkhorn. Underlying our algorithm designs are two key elements: (a) converting the original problem into a bilinear minimax problem over probability distributions; (b) exploiting the extragradient idea — in conjunction with entropy regularization and adaptive learning rates — to accelerate convergence.

Keywords: optimal transport, extragradient methods, entropy regularization, first-order methods, adaptive learning rates

Contents

*Department of Statistics and Data Science, Wharton School, University of Pennsylvania, Philadelphia, PA 19104, U.S.A.
†Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, U.S.A.
‡Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

1 Introduction

Quantifying the distance between two probability distributions is an algorithm subroutine that permeates and empowers a wealth of modern data science applications. For instance, how to measure the difference between the model distribution and the real distribution in generative adversarial networks (Arjovsky et al., 2017), how to evaluate the intrinsic dissimilarity between two point clouds in computer graphics (Kim et al., 2013; Solomon et al., 2015), and how to assess the distribution shift in transfer learning (Gayraud et al., 2017), are all representative examples built upon probability distances.

This paper focuses attention on computing an elementary instance within this arena that gains increasing popularity, that is, the optimal transport distance between two distributions (Peyré et al., 2019). It also sometimes goes by the name of the earth mover's distance (Pele and Werman, 2009; Rubner et al., 2000; Werman et al., 1985) or the Wasserstein distance (Villani, 2009). In light of the celebrated Kantorovich relaxation (Kantorovich, 1942; Peyré et al., 2019), computing the optimal transport between two n-dimensional probability distributions can be cast as solving a linear program over probability matrices with fixed marginals:

$$\begin{array}{ll} \underset{\boldsymbol{P}\in\mathbb{R}^{n\times n}}{\text{minimize}} & \langle \boldsymbol{W}, \boldsymbol{P}\rangle \\ \text{subject to} & \boldsymbol{P}\geq \boldsymbol{0},\ \boldsymbol{P}\boldsymbol{1}=\boldsymbol{r},\ \boldsymbol{P}^\top \boldsymbol{1}=\boldsymbol{c}. \end{array} \tag{1}$$

Here, $\boldsymbol{r}=[r_i]_{1\leq i\leq n}$ and $\boldsymbol{c}=[c_i]_{1\leq i\leq n}$ are n-dimensional probability vectors representing the prescribed row and column marginals, respectively, and $\boldsymbol{W}=[w_{i,j}]_{1\leq i,j\leq n}\in\mathbb{R}_+^{n\times n}$ stands for a given non-negative cost matrix (so that the objective function $\langle \boldsymbol{W},\boldsymbol{P}\rangle$ measures the total transportation cost). In a nutshell, the optimal transport problem amounts to finding the most cost-efficient reshaping of one distribution into another, or equivalently, the most economical coupling of the two distributions.

At first glance, the optimal transport problem (1) seems to be readily solvable via relatively mature toolboxes in linear programming. Nevertheless, the unprecedentedly large problem dimensionality in contemporary applications calls for a thorough (re)-examination of existing algorithms, so as to ensure feasibility of computing optimal transport at scale. For example, the linear-programming-based algorithm (Lee and Sidford, 2014) requires a runtime $\widetilde{O}(n^{2.5})$, which takes much longer than the time needed to read the cost matrix $\boldsymbol{W}$. In comparison, another alternative tailored to this problem, called Sinkhorn iteration (Cuturi, 2013; Sinkhorn, 1967), exploits the special structure underlying the solution to an entropy-regularized variant of (1). This classical approach and its variants have been shown to be near linear-time (Altschuler et al., 2017; Dvurechensky et al., 2018; Lin et al., 2022), attaining ε additive accuracy[1] with a computational complexity of $\widetilde{O}(n^2/\varepsilon^2)$. Despite their favorable scaling in n, however, the Sinkhorn-type algorithms fall short of achieving optimal scaling in ε, thereby stimulating further pursuit for theoretical improvement. Blanchet et al. (2018); Quanrud (2018) led this line of studies by developing the first algorithms with theoretical runtime $\widetilde{O}(n^2/\varepsilon)$, although practical implementation of these algorithms remain unavailable.[2] Jambulapati et al. (2019) went on to propose an implementable *first-order method* — i.e., dual extrapolation — that enjoys matching complexity $\widetilde{O}(n^2/\varepsilon)$; however, this method is numerically outperformed by Sinkhorn iteration as reported in their experiments. Another recent breakthrough in theoretical computer science van den Brand et al. (2020) solved a more general class of problems called maximum cardinality bipartite matching and showed that even logarithmic ε-dependency is feasible; the lack of practical implementation, once again, inhibits real-world adoption.

In sum, despite an exciting line of theoretical advances towards solving optimal transport, there is significant mismatch between the state-of-the-art theoretical results and the practical runtime. This motivates one to pursue other algorithmic alternatives that could be appealing in both theory and practice.

[1] A *feasible* point $\widetilde{\boldsymbol{P}}$ is said to achieve ε additive accuracy if $\langle \boldsymbol{W},\widetilde{\boldsymbol{P}}\rangle\leq\langle \boldsymbol{W},\boldsymbol{P}^\star\rangle+\varepsilon$, where $\boldsymbol{P}^\star$ is an optimal solution.

[2] These methods invoke black-box subroutines (e.g., positive linear programs, matrix scaling) that remain impractical so far.

reference	runtime	algorithm	first-order method?	implementable?
Altschuler et al. (2017)	n^2/ε^3	Sinkhorn	yes	yes
Dvurechensky et al. (2018)	n^2/ε^2	Sinkhorn	yes	yes
Lin et al. (2022)	n^2/ε^2	Greenkhorn	yes	yes
Dvurechensky et al. (2018) Lin et al. (2022)	$n^{2.5}/\varepsilon$	APDAGD	yes	yes
Guminov et al. (2021)	$n^{2.5}/\varepsilon$	AAM	yes	yes
Lin et al. (2022)	$n^{2.5}/\varepsilon$	APDAMD	yes	yes
Guo et al. (2020)	$n^{2.5}/\varepsilon$	APDRCD	yes	yes
An et al. (2022)	$n^{2.5}/\varepsilon$	Nesterov's smoothing	yes	yes
Chambolle and Contreras (2022)	$n^{2.5}/\varepsilon$	HPD	yes	yes
Xie et al. (2022a)	$n^{2.5}/\varepsilon$	PDASGD	yes	yes
Blanchet et al. (2018)	n^2/ε	packing linear program	yes	—
Quanrud (2018)	n^2/ε	packing linear program	yes	—
Blanchet et al. (2018)	n^2/ε	matrix scaling	no	—
Lahn et al. (2019)	$n^2/\varepsilon + n/\varepsilon^2$	combinatorial	no	yes
van den Brand et al. (2020)	$n^2 \log^2(1/\varepsilon)$	max-cardinality bipartite matching	no	—
Jambulapati et al. (2019)	n^2/ε	dual extrapolation	yes	yes
This work	n^2/ε	extragradient	yes	yes

Table 1: Comparisons with prior works. Here, we assume $\|\boldsymbol{W}\|_\infty = 1$ without loss of generality, and we omit all logarithmic factors except for van den Brand et al. (2020) (as its ε-dependency is only logarithmic). In the last column, "—" means that no practical implementation has been available so far.

Main contributions. In this paper, we contribute to the abovementioned growing literature by proposing a scalable algorithm tailored to the optimal transport problem. Our focal point is first-order optimization-based methods — a family of practically appealing algorithms for large-scale optimization. Our algorithm is built around the following key ideas.

1) We start with an ℓ_1-penalized variant of the original problem, and reformulate it into a bilinear minimax problem over two sets of probability distributions.

2) In an attempt to solve this minimax problem, we design a variant of entropy-regularized extragradient methods. On a high level, the algorithm performs two mirror-descent-type updates per iteration, with learning rates chosen adaptively in accordance with the corresponding row or column marginals.

Encouragingly, the proposed entropy-regularized extragradient method is capable of achieving ε additive accuracy with[3]

$$\widetilde{O}\left(\frac{1}{\varepsilon}\right) \text{ iterations} \qquad \text{or} \qquad \widetilde{O}\left(\frac{n^2}{\varepsilon}\right) \text{ runtime}, \tag{2}$$

thus constituting a nearly linear-time algorithm with desired iteration complexity; see Theorem 1. Table 1 provides more detailed comparisons with previous results. In short, our algorithm enjoys computational guarantees that match the best-known theory (i.e., Jambulapati et al. (2019)) among all first-order methods for computing optimal transport, while at the same time compare favorably to the classical Sinkhorn and Greenkhorn algorithms in numerical experiments.

Notation. Let $\Delta_n \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{1}^\top \boldsymbol{x} = 1\}$ and $\Delta_{n\times n} \coloneqq \{\boldsymbol{X} \in \mathbb{R}^{n\times n} \mid \boldsymbol{1}^\top \boldsymbol{X} \boldsymbol{1} = 1, X_{i,j} \geq 0, \forall i,j\}$ denote the n-dimensional and $n \times n$-dimensional probability simplices, respectively. For any probability vector $\boldsymbol{p} \in \Delta_d$, its entropy is defined and denoted by $\mathcal{H}(\boldsymbol{p}) \coloneqq -\sum_{i=1}^d p_i \log p_i$. For any probability vectors

[3]Here and throughout, $f(n, 1/\varepsilon) = O(g(n, 1/\varepsilon))$ means there exists a universal constant C such that $|f(n, 1/\varepsilon)| \leq C \cdot g(n, 1/\varepsilon)$ for all n and $1/\varepsilon$. The notation $\widetilde{O}(\cdot)$ is defined similarly except that it hides all logarithmic factors.

$\boldsymbol{p}, \boldsymbol{q} \in \Delta_d$, the Kullback-Leibler (KL) divergence of between $\boldsymbol{p}$ and $\boldsymbol{q}$ is defined by $\mathsf{KL}(\boldsymbol{p} \,\|\, \boldsymbol{q}) \coloneqq \sum_i p_i \log \frac{p_i}{q_i}$. For any matrix $\boldsymbol{W} = [W_{i,j}]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$, we denote by $\|\boldsymbol{W}\|_\infty \coloneqq \max_{1\le i,j\le n} |W_{i,j}|$ its entrywise infinity norm, and $\|\boldsymbol{W}\|_1 \coloneqq \sum_{1\le i,j\le n} |W_{i,j}|$ its entrywise ℓ_1 norm. Let $\mathbb{R}_+^{n\times n}$ denote the set of all $n \times n$ matrices with non-negative entries. For any matrix $\boldsymbol{F} \in \mathbb{R}_+^{n\times n}$, let $\mathsf{row}(\boldsymbol{F}) \coloneqq \boldsymbol{F}\boldsymbol{1}$ (resp. $\mathsf{col}(\boldsymbol{F}) \coloneqq \boldsymbol{F}^\top \boldsymbol{1}$) represent an n-dimensional vector consisting of all row sums (resp. column sums) of $\boldsymbol{F}$, and let $\mathsf{row}_i(\boldsymbol{F})$ (resp. $\mathsf{col}_i(\boldsymbol{F})$) denote the sum of the i-th row (resp. column) of $\boldsymbol{F}$. For any vector $\boldsymbol{x} = [x_i]_{1\le i\le n} \in \mathbb{R}^n$, we denote by $\mathsf{diag}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ a diagonal matrix whose diagonals consist of the entries of $\boldsymbol{x}$.

2 Algorithm and main results

In this section, we present our algorithm design, followed by its convergence guarantees. Before continuing, let us introduce several more notation that facilitates our discussion. Let $\boldsymbol{w}_i \in \mathbb{R}^n$ represent the i-th row of $\boldsymbol{W}$; for any $\boldsymbol{P} \in \mathbb{R}_+^{n\times n}$ obeying $\boldsymbol{P}\boldsymbol{1} = \boldsymbol{r}$, introduce a collection of probability vectors $\{\boldsymbol{p}_i \in \Delta_n\}$ such that $r_i\boldsymbol{p}_i$ represents the i-th row of $\boldsymbol{P}$; similarly, we denote by $\boldsymbol{P}^\star$ a solution to the problem (1), and employ $r_i\boldsymbol{p}_i^\star$ to represent the i-th row of $\boldsymbol{P}^\star$ (so that $\boldsymbol{p}_i^\star$ is a probability vector). In other words, we can write

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \boldsymbol{w}_2^\top \\ \vdots \\ \boldsymbol{w}_n^\top \end{bmatrix}, \qquad \boldsymbol{P} = \begin{bmatrix} r_1\boldsymbol{p}_1^\top \\ r_2\boldsymbol{p}_2^\top \\ \vdots \\ r_n\boldsymbol{p}_n^\top \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{P}^\star = \begin{bmatrix} r_1\boldsymbol{p}_1^{\star\top} \\ r_2\boldsymbol{p}_2^{\star\top} \\ \vdots \\ r_n\boldsymbol{p}_n^{\star\top} \end{bmatrix}. \tag{3}$$

Armed with the above notation, we can readily reformulate (1) as follows

$$\begin{array}{ll} \underset{\{\boldsymbol{p}_i\}_{i=1}^n}{\text{minimize}} & \sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \boldsymbol{p}_i\rangle \\ \text{subject to} & \boldsymbol{p}_i \in \Delta_n \ (1\le i\le n), \ \ \sum_{i=1}^n r_i\boldsymbol{p}_i = \boldsymbol{c}, \end{array} \tag{4}$$

for which $\{\boldsymbol{p}_i^\star\}_{i=1}^n$ stands for an optimal solution.

2.1 Approximate transportation solution of a penalized variant

Naturally, the equivalent formulation (4) motivates one to look at a related ℓ_1-penalized problem as follows

$$\underset{\{\boldsymbol{p}_i\in\Delta_n\}_{i=1}^n}{\text{minimize}} \qquad \frac{1}{2}\sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \boldsymbol{p}_i\rangle + \|\boldsymbol{W}\|_\infty \left\| \sum_{i=1}^n r_i\boldsymbol{p}_i - \boldsymbol{c}\right\|_1, \tag{5}$$

a trick that has been adopted in prior optimization literature; see, e.g., Jambulapati et al. (2019). Evidently, if we are able to compute an ε-optimal solution $\{\widehat{\boldsymbol{p}}_i\}_{i=1}^n$ to (5), then one necessarily has

$$\begin{aligned} \frac{1}{2}\sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \widehat{\boldsymbol{p}}_i\rangle &\le \frac{1}{2}\sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \widehat{\boldsymbol{p}}_i\rangle + \|\boldsymbol{W}\|_\infty \left\| r_i\sum_{i=1}^n \widehat{\boldsymbol{p}}_i - \boldsymbol{c}\right\|_1 \le \frac{1}{2}\sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \boldsymbol{p}_i^\star\rangle + \|\boldsymbol{W}\|_\infty \left\| r_i\sum_{i=1}^n \boldsymbol{p}_i^\star - \boldsymbol{c}\right\|_1 + \varepsilon \\ &= \frac{1}{2}\sum_{i=1}^n r_i\langle \boldsymbol{w}_i, \boldsymbol{p}_i^\star\rangle + \varepsilon. \end{aligned} \tag{6}$$

In general, however, the solution to (5) does not satisfy the feasibility constraints of the optimal transport problem, and one still needs to convert it into a feasible transportation plan. This can be accomplished by resorting to the following result derived in Altschuler et al. (2017, Lemma 7).

Lemma 1. *For any non-negative matrix $\widehat{\boldsymbol{P}} \in \mathbb{R}_+^{n\times n}$, Algorithm 3 (see Appendix A) is able to find a probability matrix $\widetilde{\boldsymbol{P}} \in \Delta_{n\times n}$ with $O(n^2)$ computation complexity such that*

$$\widetilde{\boldsymbol{P}}\boldsymbol{1} = \boldsymbol{r}, \qquad \widetilde{\boldsymbol{P}}^\top\boldsymbol{1} = \boldsymbol{c}, \qquad \textit{and} \qquad \|\widehat{\boldsymbol{P}} - \widetilde{\boldsymbol{P}}\|_1 \le 2\big(\|\widehat{\boldsymbol{P}}\boldsymbol{1} - \boldsymbol{r}\|_1 + \|\widehat{\boldsymbol{P}}^\top\boldsymbol{1} - \boldsymbol{c}\|_1\big). \tag{7}$$

As a consequence, it boils down to finding a near-optimal solution $\{\widehat{\boldsymbol{p}}_i\}_{1\le i\le n}$ to (5) in a computationally efficient manner, while ensuring sufficiently small $\|\sum_{i=1}^n r_i\widehat{\boldsymbol{p}}_i - \boldsymbol{c}\|_1$ (a quantity that is equivalent to $\|\sum_{i=1}^n r_i\widehat{\boldsymbol{p}}_i - \boldsymbol{c}\|_1 + \sum_{i=1}^n \big|r_i\widehat{\boldsymbol{p}}_i^\top\boldsymbol{1} - r_i\big| = \|\widehat{\boldsymbol{P}}^\top\boldsymbol{1} - \boldsymbol{c}\|_1 + \|\widehat{\boldsymbol{P}}\boldsymbol{1} - \boldsymbol{r}\|_1$ in this case given that $\widehat{\boldsymbol{p}}_i^\top\boldsymbol{1} = 1$).

2.2 Entropy-regularized extragradient methods

In this subsection, we propose a method for solving the ℓ_1-penalized problem (5). To streamline the presentation, we assume without loss of generality that $\|\boldsymbol{W}\|_\infty = 1$; this can be implemented by running $\boldsymbol{W} \leftarrow \boldsymbol{W}/\|\boldsymbol{W}\|_\infty$ at the very beginning of the algorithm.

An equivalent minimax problem and entropy regularization. The first step of the proposed algorithm lies in converting the objective function of (5) into a bilinear function, for which the key lies in handling the ℓ_1 penalty term. Towards this end, we introduce a set of auxiliary 2-dimensional probability vectors $\boldsymbol{\mu}_j = [\mu_{j,+}, \mu_{j,-}] \in \Delta_2$ $(1 \le j \le n)$. As can be easily verified, this allows us to recast the objective of (5) as follows:

$$\frac{1}{2}\sum_{i=1}^n r_i \langle \boldsymbol{w}_i, \boldsymbol{p}_i \rangle + \left\| \sum_{i=1}^n r_i \boldsymbol{p}_i - \boldsymbol{c} \right\|_1 = \max_{\boldsymbol{\mu}_j \in \Delta_2, \forall j} f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big), \tag{8}$$

where we define

$$\begin{aligned} f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big) &\coloneqq \frac{1}{2}\sum_{i=1}^n r_i \langle \boldsymbol{w}_i, \boldsymbol{p}_i \rangle + \sum_{j=1}^n (\mu_{j,+} - \mu_{j,-}) \left(\sum_{i=1}^n r_i p_{i,j} - c_j \right) \\ &= \frac{1}{2}\sum_{i=1}^n r_i \langle \boldsymbol{w}_i, \boldsymbol{p}_i \rangle - \sum_{j=1}^n (\mu_{j,+} - \mu_{j,-}) c_j + \sum_{i=1}^n \sum_{j=1}^n r_i (\mu_{j,+} - \mu_{j,-}) p_{i,j}. \end{aligned} \tag{9}$$

Armed with this function, one can readily recast (5) as the following minimax problem:

$$\operatorname*{minimize}_{\boldsymbol{p}_i \in \Delta_n, \forall i} \ \max_{\boldsymbol{\mu}_j \in \Delta_2, \forall j} f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big), \tag{10}$$

or equivalently (by virtue of von Neumann's minimax theorem (von Neumann, 1928)),

$$\operatorname*{maximize}_{\boldsymbol{\mu}_j \in \Delta_2, \forall j} \ \min_{\boldsymbol{p}_i \in \Delta_n, \forall i} f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big). \tag{11}$$

Given that the bilinear objective function is convex-concave but not strongly-convex-strongly-concave, one strategy for accelerating the optimization procedure is to augment the objective function with entropy regularization terms. This leads to the following entropy-regularized minimax problem:

$$\operatorname*{maximize}_{\boldsymbol{\mu}_j \in \Delta_2, \forall j} \ \min_{\boldsymbol{p}_i \in \Delta_n, \forall i} F\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big) \coloneqq f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{j=1}^n\big) + \sum_{j=1}^n \tau_{\mu,j} \mathcal{H}(\boldsymbol{\mu}_j) - \sum_{i=1}^n \tau_{p,i} \mathcal{H}(\boldsymbol{p}_i), \tag{12}$$

where $\mathcal{H}(\cdot)$ denotes the entropy (which is a strongly concave and non-negative function), and $\{\tau_{\mu,j}\}_{1\le j\le n}$ and $\{\tau_{p,i}\}_{1\le i\le n}$ are a set of *positive* regularization parameters that we shall specify momentarily. The remainder of this subsection is dedicated to solving (12) in an efficient fashion.

An extragradient method for solving (12). The family of extragradient methods has proven effective for solving convex-concave minimax problems (Harker and Pang, 1990; Korpelevich, 1976; Mokhtari et al., 2020a; Tseng, 1995). Inspired by a recent development Cen et al. (2021), we propose to solve (12) by means of a variant of extragradient methods.

Let us begin by introducing a basic operation. Suppose the current iterate is $\big(\{\boldsymbol{p}_i^{\mathsf{current}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\mathsf{current}}\}_{j=1}^n\big)$. One step of mirror descent (with the KL divergence chosen to monitor the displacement) takes the following form:

$$\boldsymbol{\mu}_j^{\mathsf{next}} = \arg\max_{\boldsymbol{\mu}_j \in \Delta_2} \left\{ \left\langle \nabla_{\boldsymbol{\mu}_j} F\big(\{\boldsymbol{p}_i^{\mathsf{grad}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\mathsf{grad}}\}_{j=1}^n\big), \boldsymbol{\mu}_j \right\rangle - \frac{1}{\eta_{\mu,j}} \mathsf{KL}\big(\boldsymbol{\mu}_j \,\|\, \boldsymbol{\mu}_j^{\mathsf{current}}\big) \right\}, \qquad 1 \le j \le n \tag{13a}$$

$$\boldsymbol{p}_i^{\mathsf{next}} = \arg\min_{\boldsymbol{p}_i \in \Delta_n} \left\{ \left\langle \nabla_{\boldsymbol{p}_i} F\big(\{\boldsymbol{p}_i^{\mathsf{grad}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\mathsf{grad}}\}_{j=1}^n\big), \boldsymbol{p}_i \right\rangle + \frac{1}{\eta_{p,i}} \mathsf{KL}\big(\boldsymbol{p}_i \,\|\, \boldsymbol{p}_i^{\mathsf{current}}\big) \right\}, \qquad 1 \le i \le n \tag{13b}$$

or equivalently,

$$\mu_{j,s}^{\mathsf{next}} \propto \big(\mu_{j,s}^{\mathsf{current}}\big)^{1-\eta} \exp\bigg(\eta_{\mu,j} s\Big(\sum_{i=1}^{n} r_i p_{i,j}^{\mathsf{grad}} - c_j\Big)\bigg), \qquad s \in \{+,-\} \tag{14a}$$

$$p_{i,l}^{\mathsf{next}} \propto \big(p_{i,l}^{\mathsf{current}}\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i\big(0.5w_{i,l} + \mu_{l,+}^{\mathsf{grad}} - \mu_{l,-}^{\mathsf{grad}}\big)\Big), \qquad l = 1, \cdots, n \tag{14b}$$

for all $1 \le i, j \le n$, where $\{\eta_{\mu,j}\}$ and $\{\eta_{p,i}\}$ are two collections of positive learning rates. Here, we allow the gradient ∇F to be evaluated at a point $\big(\{\boldsymbol{p}_i^{\mathsf{grad}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\mathsf{grad}}\}_{j=1}^n\big)$ deviating from the current iterate $\big(\{\boldsymbol{p}_i^{\mathsf{current}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\mathsf{current}}\}_{j=1}^n\big)$, which plays a crucial role in describing the extragradient update rule.

We are now ready to present the proposed method, which maintains several sequences of the iterates; for each iteration t, we maintain and update the following iterates:

- *Updates w.r.t. the variables* $\{\boldsymbol{p}_i\}_{i=1}^n$*:* main sequence $\{\boldsymbol{p}_i^t \in \Delta_n\}_{i=1}^n$; midpoints $\{\overline{\boldsymbol{p}}_i^t \in \Delta_n\}_{i=1}^n$.
- *Updates w.r.t. the variables* $\{\boldsymbol{\mu}_j\}_{j=1}^n$*:* main sequence $\{\boldsymbol{\mu}_j^t = [\mu_{j,+}^t, \mu_{j,-}^t] \in \Delta_2\}_{j=1}^n$; midpoints $\{\overline{\boldsymbol{\mu}}_j^t = [\overline{\mu}_{j,+}^t, \overline{\mu}_{j,-}^t] \in \Delta_2\}_{j=1}^n$; adjusted main sequence $\{\boldsymbol{\mu}_j^{t,\mathsf{adjust}} = [\mu_{j,+}^{t,\mathsf{adjust}}, \mu_{j,-}^{t,\mathsf{adjust}}] \in \Delta_2\}_{j=1}^n$.

In the t-th iteration, the proposed algorithm performs the following three sets of updates, with the first two embodying the extragradient idea.

1) **Computing the midpoints:** for each $1 \le j \le n$,

$$\overline{\mu}_{j,s}^{t+1} \propto \big(\mu_{j,s}^{t,\mathsf{adjust}}\big)^{1-\eta} \exp\bigg(\eta_{\mu,j} s\Big(\sum_{i=1}^{n} r_i p_{i,j}^t - c_j\Big)\bigg), \qquad s \in \{+,-\}, \tag{15a}$$

and for each $1 \le i \le n$,

$$\overline{p}_{i,j}^{t+1} \propto \big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i\big(0.5w_{i,j} + \mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}\big)\Big), \qquad j = 1, \cdots, n, \tag{15b}$$

with the learning rates $\{\eta_{\mu,j}\}$ and $\{\eta_{p,i}\}$ to be specified shortly. In words, this constitutes one step of mirror descent (cf. (14)) from the point $\big(\{\boldsymbol{p}_i^t\}_{i=1}^n, \{\boldsymbol{\mu}_j^{t,\mathsf{adjust}}\}_{j=1}^n\big)$, with the gradient evaluated at the same point.

2) **Updating the main sequence:** for each $1 \le j \le n$,

$$\mu_{j,s}^{t+1} \propto \big(\mu_{j,s}^{t,\mathsf{adjust}}\big)^{1-\eta} \exp\bigg(\eta_{\mu,j} s\Big(\sum_{i=1}^{n} r_i \overline{p}_{i,j}^{t+1} - c_j\Big)\bigg), \qquad s \in \{+,-\}; \tag{16a}$$

and for each $1 \le i \le n$,

$$p_{i,j}^{t+1} \propto \big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i\big(0.5w_{i,j} + \overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1}\big)\Big), \qquad j = 1, \cdots, n. \tag{16b}$$

This implements another step of mirror descent (cf. (14)) from the same point $\big(\{\boldsymbol{p}_i^t\}_{i=1}^n, \{\boldsymbol{\mu}_j^{t,\mathsf{adjust}}\}_{j=1}^n\big)$ as above, albeit using a gradient evaluated at the midpoint $\big(\{\overline{\boldsymbol{p}}_i^{t+1}\}_{i=1}^n, \{\overline{\boldsymbol{\mu}}_j^{t+1}\}_{j=1}^n\big)$. In a nutshell, the midpoint computed in the previous step assists in predicting a better search direction.

3) **Adjusting the current iterates:** for each $1 \le j \le n$,

$$\mu_{j,s}^{t+1,\mathsf{adjust}} \propto \max\Big\{\mu_{j,s}^{t+1},\, e^{-B} \max\big\{\mu_{j,+}^{t+1}, \mu_{j,-}^{t+1}\big\}\Big\}, \qquad s \in \{+,-\}, \tag{16c}$$

where $B > 0$ is some parameter to be specified momentarily. This operation prevents the ratio $\frac{\max\{\mu_{j,+}^{t+1}, \mu_{j,-}^{t+1}\}}{\min\{\mu_{j,+}^{t+1}, \mu_{j,-}^{t+1}\}}$ from being exponentially large (i.e., it is no larger than e^B), a condition that helps facilitate analysis.

After running the above updates for $t_{\max}$ iterations, we reach a probability matrix taking the following form:

$$\widehat{\boldsymbol{P}} = \big[r_1 \boldsymbol{p}_1^{t_{\max}}, \ldots, r_n \boldsymbol{p}_n^{t_{\max}}\big]^\top, \tag{17}$$

which can be converted into a feasible transportation plan $\widetilde{\boldsymbol{P}}$ by invoking Algorithm 3. The whole procedure is summarized in Algorithm 1.

Algorithm 1: The proposed entropy-regularized extragradient method for optimal transport.

Input: cost matrix $\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$, probability vectors $\boldsymbol{r} = [r_i]_{1\le i\le n}, \boldsymbol{c} = [c_i]_{1\le i\le n} \in \Delta_n$, target accuracy level ε, number of iterations $t_{\max}$.

// Initialization

$\mu_{j,+}^0 = \mu_{j,-}^0 = \mu_{j,+}^{0,\mathsf{adjust}} = \mu_{j,-}^{0,\mathsf{adjust}} = 1/2$ for all $1 \le j \le n$; $\boldsymbol{p}_i^0 = [1/n, \ldots, 1/n]$ for all $1 \le i \le n$.

$\boldsymbol{W} \leftarrow \boldsymbol{W}/\|\boldsymbol{W}\|_\infty$; $\varepsilon \leftarrow \varepsilon/\|\boldsymbol{W}\|_\infty$. /* normalization */

// Main loop

for $t = 0$ **to** $t_{\max} - 1$ **do**

  // Compute the midpoints

  for $j = 1$ **to** n **do**

    $\overline{\mu}_{j,s}^{t+1} \leftarrow \big(\mu_{j,s}^{t,\mathsf{adjust}}\big)^{1-\eta} \exp\Big(\eta_{\mu,j} s\big(\sum_{i=1}^n r_i p_{i,j}^t - c_j\big)\Big)$, $\quad s \in \{+,-\}$;

    $\overline{\boldsymbol{\mu}}_j^{t+1} \leftarrow \mathtt{Normalize}(\overline{\boldsymbol{\mu}}_j^{t+1})$. /* Call Algorithm 2 */

  for $i = 1$ **to** n **do**

    $\overline{p}_{i,j}^{t+1} \leftarrow \big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i\big(0.5 w_{i,j} + \mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}\big)\Big)$, $\quad j = 1, \cdots, n$;

    $\overline{\boldsymbol{p}}_i^{t+1} \leftarrow \mathtt{Normalize}(\overline{\boldsymbol{p}}_i^{t+1})$. /* Call Algorithm 2 */

  // Update the main sequence

  for $j = 1$ **to** n **do**

    $\mu_{j,s}^{t+1} \leftarrow \big(\mu_{j,s}^{t,\mathsf{adjust}}\big)^{1-\eta} \exp\Big(\eta_{\mu,j} s\big(\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} - c_j\big)\Big)$, $\quad s \in \{+,-\}$;

    $\boldsymbol{\mu}_j^{t+1} \leftarrow \mathtt{Normalize}(\boldsymbol{\mu}_j^{t+1})$. /* Call Algorithm 2 */

  for $i = 1$ **to** n **do**

    $p_{i,j}^{t+1} \leftarrow \big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i\big(0.5 w_{i,j} + \overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1}\big)\Big)$, $\quad j = 1, \cdots, n$;

    $\boldsymbol{p}_i^{t+1} \leftarrow \mathtt{Normalize}(\boldsymbol{p}_i^{t+1})$. /* Call Algorithm 2 */

  // Adjust the main sequence

  for $j = 1$ **to** n **do**

    $\mu_{j,s}^{t+1,\mathsf{adjust}} \leftarrow \max\big\{\mu_{j,s}^{t+1}, e^{-B} \max\big\{\mu_{j,+}^{t+1}, \mu_{j,-}^{t+1}\big\}\big\}$, $\quad s \in \{+,-\}$;

    $\boldsymbol{\mu}_j^{t+1,\mathsf{adjust}} \leftarrow \mathtt{Normalize}(\boldsymbol{\mu}_j^{t+1,\mathsf{adjust}})$. /* Call Algorithm 2 */

Set $\widehat{\boldsymbol{P}} = [r_1 \boldsymbol{p}_1^{t_{\max}}, \ldots, r_n \boldsymbol{p}_n^{t_{\max}}]^\top$. /* Solution of regularized problem (5) */

// Convert an almost-transportation plan $\widehat{\boldsymbol{P}}$ to a feasible transportation plan $\widetilde{\boldsymbol{P}}$

Output $\widetilde{\boldsymbol{P}}$, obtained by invoking Algorithm 3 with input $\widehat{\boldsymbol{P}}$.

Choice of algorithmic parameters. Thus far, we have not yet discussed the choices of multiple parameters required to run Algorithm 1. Let us begin by looking at the regularization parameters $\{\tau_{\mu,j}\}$ and $\{\tau_{p,j}\}$, which cannot be taken to be too large. Evidently, if the regularization parameters are chosen such that

$$\sum_{j=1}^n \tau_{\mu,j} \log 2 + \sum_{i=1}^n \tau_{p,i} \log n \le \varepsilon, \tag{18}$$

then it follows from elementary properties of the entropy that: for any $\{\boldsymbol{p}_i \in \Delta_n\}_{i=1}^n$ and $\{\boldsymbol{\mu}_j \in \Delta_2\}_{i=1}^n$,

$$\Big|F\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{i=1}^n\big) - f\big(\{\boldsymbol{p}_i\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{i=1}^n\big)\Big| \le \sum_{j=1}^n \tau_{\mu,j} \log 2 + \sum_{i=1}^n \tau_{p,i} \log n \le \varepsilon. \tag{19}$$

Consequently, any ε-optimal solution to (12) is an 2ε-optimal solution to (5). Moverover, the theory developed in Cen et al. (2021) for matrix games suggests that a feasible learning rate can be chosen to be inversely proportional to the regularization parameter. As a result, we take

$$\eta_{\mu,j} = \frac{\eta}{\tau_{\mu,j}} \qquad \text{and} \qquad \eta_{p,i} = \frac{\eta}{\tau_{p,i}}, \qquad \forall 1 \le i,j \le n \tag{20}$$

Algorithm 2: `Normalize`($\boldsymbol{x}$).

Input: $\boldsymbol{x} = [x_i]_{1\le i\le d}$

Output $\boldsymbol{y} = [y_i]_{1\le i\le d}$, where $y_i = \frac{x_i}{\sum_j x_j}$.

for some quantity $\eta > 0$.

With the above considerations in mind, we recommend the following choices of parameters:

$$B = C_1 \log\frac{n}{\varepsilon}, \qquad \eta = \frac{C_2^2\varepsilon}{\sqrt{B}\log n}, \qquad \eta_{\mu,j} = \frac{15C_2\sqrt{B}}{c_j + C_3/n}, \qquad \eta_{p,i} = \frac{C_2}{\sqrt{B}r_i}, \qquad \forall 1\le i,j\le n \tag{21}$$

with $C_1 > 0, C_2 > 0, 0 < C_3 \le 1$ some suitable universal constants, which correspond to

$$\tau_{\mu,j} = \frac{\eta}{\eta_{\mu,j}} = \frac{C_2(c_j + C_3/n)\varepsilon}{15C_1(\log n)\big(\log\frac{n}{\varepsilon}\big)} \qquad \text{and} \qquad \tau_{p,i} = \frac{\eta}{\eta_{p,i}} = \frac{C_2 r_i\varepsilon}{\log n}, \qquad \forall 1\le i,j\le n. \tag{22}$$

Three remarks are in order. Firstly, the learning rate $\eta_{p,i}$ (resp. $\eta_{\mu,j}$) is chosen adaptively to be inversely proportional to the row sum r_i (resp. column sum c_j), which is crucial in achieving our advertised convergence rate; in contrast, fixing the learing rates across all i (resp. j) as in prior works results in slow convergence particularly when the r_i's (resp. c_j's) are far from uniform. Secondly, $\eta_{\mu,j}(c_j + O(1/n))$ is chosen to be larger than $\eta_{p,i}r_i$, in the hope that $\{\boldsymbol{\mu}_j^t\}$ converges more rapidly than $\{\boldsymbol{p}_i^t\}$. Furthermore, if C_1 is large enough and C_2 small enough, the above regularization parameters obey

$$\sum_{j=1}^n \tau_{\mu,j}\log 2 + \sum_{i=1}^n \tau_{p,i}\log n = \frac{C_2(1+C_3)\varepsilon\log 2}{15C_1\log n\log\frac{n}{\varepsilon}} + C_2\varepsilon \le \varepsilon \tag{23}$$

given that $\sum_j c_j = \sum_i r_i = 1$, thus satisfying (19).

2.3 Theoretical guarantees

Our theoretical analysis delivers intriguing news about the convergence properties of the proposed algorithm, as asserted by the following theorem.

Theorem 1. *Consider any $0 < \varepsilon < \|\boldsymbol{W}\|_\infty$. Algorithm 1 with the parameters (21) returns a probability matrix $\widetilde{\boldsymbol{P}} \in \Delta_{n\times n}$ obeying*

$$\widetilde{\boldsymbol{P}}\boldsymbol{1} = \boldsymbol{r}, \qquad \widetilde{\boldsymbol{P}}^\top\boldsymbol{1} = \boldsymbol{c}, \qquad \textit{and} \qquad \langle \boldsymbol{W}, \widetilde{\boldsymbol{P}}\rangle \le \langle \boldsymbol{W}, \boldsymbol{P}^\star\rangle + \varepsilon, \tag{24}$$

provided that $C_1 > 0$ is large enough, $C_2 > 0$ is small enough, $C_2\sqrt{C_1}$ is large enough, $0 < C_3 \le 1$, C_2^2/C_3 is small enough, and

$$t_{\max} \ge C_4\frac{\|\boldsymbol{W}\|_\infty}{\eta}\log\Big(\frac{n\|\boldsymbol{W}\|_\infty}{\varepsilon}\Big) \tag{25}$$

for some large enough constant $C_4 > 0$, with η defined in (21).

Remark 1 (Explanations of constants $C_1, \ldots, C_4$.). The requirements of these universal constants are equivalent to: 1) $0 < C_3 \le 1$; 2) $0 < C_2 < \sqrt{\omega_1 C_3}$ for some universal constant $0 < \omega_1 < 1$; 3) $C_1 > \omega_2/C_2^2$ for some universal constant $\omega_2 > 1$; 4) $C_4 > \omega_3$ for some universal constant $\omega_3 > 0$. In other words, there exist some universal constants $0 < \omega_1 < 1, \omega_2 > 1, \omega_3 > 0$ such that for arbitrary choices of $C_1, \ldots, C_4$ obeying the above conditions, Theorem 1 holds.

Assuming without loss of generality that $\|\boldsymbol{W}\|_\infty = 1$, Theorem 1 in conjunction with the choice of η in (21) asserts that the iteration complexity of our algorithm is

$$\text{(iteration complexity)} \qquad O\bigg(\frac{1}{\varepsilon}\log^{2.5}\frac{n}{\varepsilon}\bigg). \tag{26}$$

Given that each iteration can be implemented in $O(n^2)$ time, the total computational complexity of Algorithm 1 is no larger than

$$\text{(computation complexity)} \qquad O\bigg(\frac{n^2}{\varepsilon}\log^{2.5}\frac{n}{\varepsilon}\bigg). \tag{27}$$

This matches the state-of-the-art theory Jambulapati et al. (2019) among all first-order methods tailored to the optimal transport problem; we shall demonstrate the practical efficacy of our algorithm momentarily. With regards to the memory complexity, all computation in our algorithm only involves matrices of dimension no larger than $n \times n$. In comparison to Blanchet et al. (2018); Quanrud (2018); van den Brand et al. (2020), our algorithm is easy-to-implement and amenable to parallelism, without the need of calling any unimplementable blackbox subroutine that is still only of theoretical interest.

Finally, we note that while our analysis is inspired by the prior work Cen et al. (2021), a direct application of their analysis framework can only lead to highly suboptimal iteration complexity. Particularly, they focus on the ℓ_∞ type bound, which can ensure $\|\widehat{\boldsymbol{P}}^\top\mathbf{1} - \boldsymbol{c}\|_\infty \le \varepsilon$ within $\widetilde{O}\big(\frac{1}{\varepsilon}\big)$ iterations. However, we need $\|\widehat{\boldsymbol{P}}^\top\mathbf{1} - \boldsymbol{c}\|_1 \le \varepsilon$ when paired with Lemma 1 to get our desired result, which in turn requires $\widetilde{O}\big(\frac{n^3}{\varepsilon}\big)$ computation complexity if naively adopting the ℓ_∞ type bound in Cen et al. (2021). Novel algorithmic and analysis ideas (e.g., how to exploit the use of adaptive learning rates) tailored to the optimal transport problem play a central role in establishing the desired performance guarantees.

3 Related works

In this section, we discuss a broader set of past works that are related to this paper.

Entropy regularization. The advantages of entropy regularization have been exploited in a diverse array of optimization problems over probability distributions, with prominent examples including equilibrium computation in game theory (Ao et al., 2023; Cen et al., 2022a, 2021; McKelvey and Palfrey, 1995; Mertikopoulos and Sandholm, 2016; Savas et al., 2019) and policy optimization in reinforcement learning (Cen et al., 2022b, 2023; Geist et al., 2019; Lan, 2022; Mei et al., 2020; Neu et al., 2017; Zhan et al., 2023). The idea of employing entropy regularization to speed up convergence in optimal transport has been studied for multiple decades (e.g., Altschuler (2022); Chakrabarty and Khanna (2021); Kalantari et al. (2008); Knight (2008)) and recently popularized by Cuturi (2013). By adding a reasonably small entropy penalty term (so that it does not bias the objective function by much), the optimal solution to the entropy-regularized problem exhibits a special form $\boldsymbol{D}_r \exp(-\eta\boldsymbol{W})\boldsymbol{D}_c$, where $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$ are certain diagonal matrices and the $\exp(\cdot)$ operator is applied in an entrywise manner (Sinkhorn, 1967). This special structure motivates one to alternate between row and column rescaling until convergence, the key idea behind the Sinkhorn algorithm.

Extragradient methods. Dating back to Korpelevich (1976); Tseng (1995), extensive research efforts have been put forth towards understanding extragradient methods for saddle-point optimization, where a clever step of extrapolation is leveraged to accelerate convergence; partial examples include the optimistic gradient descent ascent (OGDA) method (Mertikopoulos et al., 2018a,b; Rakhlin and Sridharan, 2013; Wei et al., 2021), the implicit update method (Liang and Stokes, 2019), and their stochastic variants (Hsieh et al., 2019). Mokhtari et al. (2020b) analyzed the convergence of extragradient methods for unconstrained smooth convex-concave saddle-point problems under the Euclidean metric, with Wei et al. (2021) focusing on constrained saddle-point problems. While earlier works analyzed primarily average-iterate or ergodic convergence (Nemirovski, 2004), significant emphasis was put on achieving last-iterate convergence motivated by machine learning applications (Mertikopoulos et al., 2018b; Wei et al., 2021). By using entropy regularization, Cen et al. (2021) demonstrated fast last-iterate convergence of extragradient methods for matrix games, under weaker assumptions than those needed for solving the unregularized games directly (Daskalakis and Panageas, 2018).

Prior algorithms for the optimal transport problem. Earlier effort towards computing the optimal transport include the development of the Hungarian algorithm, which is not a linear-time algorithm due to

its complexity $\widetilde{O}(n^3)$ (Kuhn, 1956; Munkres, 1957); this algorithm has recently been revisited by Xie et al. (2022b), which further came up with a variant that runs faster in a special class of problem instances. In comparison, Sinkhorn iteration and its variants have achieved widespread adoption in practice since Cuturi (2013). Altschuler et al. (2017) developed the first theory uncovering the linear-time feature of Sinkhorn iteration with computational complexity $\widetilde{O}(n^2/\varepsilon^3)$, and inspired a recent strand of works (e.g., Dvurechensky et al. (2018); Feydy et al. (2019)) that strengthened the runtime for Sinkhorn-type algorithms to $\widetilde{O}(n^2/\varepsilon^2)$ (including a fast greedy variant called the Greenkhorn algorithm (Altschuler et al., 2017; Lin et al., 2022)). First-order methods and their stochastic variants have received much recent attention, including but not limited to accelerated gradient descent (Dvurechensky et al., 2018), stochastic gradient descent (SGD) (Genevay et al., 2016), Nesterov's smoothing (An et al., 2022), and accelerated primal-dual methods (Chambolle and Contreras, 2022; Lin et al., 2022; Xie et al., 2022a). The convergence guarantees of these algorithms remain suboptimal in terms of the dependency on either n or ε.

As mentioned previously, the algorithms designed by Blanchet et al. (2018); Quanrud (2018), while achieving an appealing $\widetilde{O}(n^2/\varepsilon)$ runtime, rely heavily on reduction to blackbox methods developed in theoretical computer science (e.g., positive linear programming (Allen-Zhu and Orecchia, 2015)), which hinder practical realization and do not yet admit fast parallelization. Inspired by Nesterov (2007); Sherman (2017), Jambulapati et al. (2019) leveraged the concept of area convexity when designing and analyzing the dual extrapolation algorithm, which has become the best-performing (in theory) first-order method in the previous literature. It is also worth noting that Jambulapati et al. (2019) also reformulated the problem into a minimax form, albeit using different constraint sets (for instance, the decision variables therein are not all probability vectors). Another recent work Mai et al. (2022) proposed an algorithm based on the Douglas-Rachford splitting, which enjoys competitive numerical performance compared with Sinkhorn and is amenable to GPU acceleration. While this algorithm has been shown to enjoy an iteration complexity of $O\big(1/(\rho\varepsilon)\big)$ (with ρ some suitable penalty parameter) and per-iteration cost $O(n^2)$, the penalty parameter ρ is taken to be on the order of $1/n$ (see Mai et al. (2022, Section 4)), resulting in a total computational complexity far exceeding n^2/ε. Finally, another recent work Lahn et al. (2019) tackled this problem via combinatorial algorithms, yielding a runtime as fast as $\widetilde{O}(n^2/\varepsilon + n/\varepsilon^2)$.

4 Numerical experiments

In this section, we illustrate empirical results that validate our theoretical studies and confirm the efficacy of the proposed extragradient method. In particular, we compare the numerical performance of Algorithm 1 with the classical Sinkhorn method, its greedy variant called Greenkhorn Altschuler et al. (2017), as well as the recently proposed dual extrapolation method Jambulapati et al. (2019, Algorithm 3) and the DROT method Mai et al. (2022, Algorithm 1); in addition, we validate the theoretical runtime $\widetilde{O}(n^2/\varepsilon)$ for achieving an ε-accurate solution. All algorithms are implemented and tested in MATLAB R2020a on an iMac with a 3 GHz 6-Core Intel Core i5 processor and 24 GB memory.[4]

In our experiments, each optimal transport problem instance is generated in one of the following three ways.

(i) *"Synthetic"*. We first produce two $m \times m$ images as follows: each image has a randomly placed square foreground that accounts for 50% of the pixels; the foreground and background have pixel values uniformly sampled from $[0, 10]$ and $[0, 1]$, respectively. With two images in place, we flatten and normalize them to obtain $\boldsymbol{r}, \boldsymbol{c} \in \mathbb{R}^n$, where $n = m^2$. In addition, we let each entry of the cost matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ be the ℓ_1 distance between each pair of pixels in an $m \times m$ image.

(ii) *"MNIST"*. First, a pair of images are randomly selected from the MNIST dataset,[5] and downsampled to size $m \times m$; then, a value of 0.01 is added to all pixels. The remaining steps for obtaining $\boldsymbol{r}, \boldsymbol{c}$ and $\boldsymbol{W}$ are the same as in the "Synthetic" setting.

[4]Implementation with hardware acceleration (e.g. using GPUs), like the one in Mai et al. (2022), is beyond the scope of our current paper.

[5]http://yann.lecun.com/exdb/mnist/

(iii) *"Point clouds".* We first generate a pair of point clouds, each with n points randomly sampled from a 2-dimensional Gaussian distribution. Then, the marginal distributions $\boldsymbol{r}$ (resp. $\boldsymbol{c}$) stands for the uniform distribution supported on the first (resp. second) point cloud, i.e. an average of Dirac distributions; moreover, each entry of the cost matrix $\boldsymbol{W}$ is defined by the Euclidean distance between two points.

For each optimal transport instance, we also assign a target accuracy level ε, which will be employed to set parameters of the algorithms.

4.1 Comparisons with Sinkhorn and Greenkhorn

To demonstrate the practical applicability of the proposed algorithm, we first compare its empirical performance with that of the Sinkhorn algorithm and its greedy variant called Greenkhorn, which are still among the most widely used baselines for solving optimal transport.

Setup. Each algorithm is implemented with varying parameters. For the Sinkhorn and Greenkhorn algorithms, recall from Altschuler et al. (2017) that η^{-1} is the strength of entropic regularization; in our experiments, we consider the theoretical choice $\eta = 4\varepsilon^{-1}\log n$, as well as less conservative options $\eta \in \{10, 100, 500\}$. Regarding the proposed extragradient method, recall that Algorithm 1 requires parameters $B, \eta, \{\eta_{p,i}\}, \{\eta_{\mu,j}\}$. We let $\eta_{p,i} = C/(\sqrt{B}r_i), \eta_{\mu,j} = C\sqrt{B}/(c_j + C_3/n)$, and consider two options: (1) the theoretical choice according to (21), with $B = \log(n/\varepsilon), \eta = \varepsilon/(\sqrt{B}\log n)$ and $C = 1, C_3 = 1$; (2) a fine-tuned option, with $B = 1, \eta = 0, C = 1, C_3 = 10^{-2}$.

Convergence of algorithms. The numerical results for various settings are illustrated in Figure 1. Each subfigure represents one specific setting of optimal transport, and each curve is an average over multiple independent trials under that setting. The X-axis reflects the computation cost, measured either by the total number of matrix-vector products[6] (akin to Jambulapati et al. (2019)), or the actual runtime; the Y-axis reports the gap between the cost of the current iterate (rounded to the probability simplex with marginals $\boldsymbol{r}$ and $\boldsymbol{c}$) and the true optimal transport value (computed via standard linear programming). The curves w.r.t. Sinkhorn, Greenkhorn and the extragradient method are plotted in blue, green and red, respectively. The dashed lines stand for the theoretical choices of parameters, while solid lines correspond to the practical/fine-tuned choices.

As illustrated in Figure 1 for multiple settings, the proposed extragradient method (with fine-tuned parameters) compares favorably to both Sinkhorn and Greenkhorn, especially when the target accuracy level ε is small. The numerical results also hint at practical choices of algorithmic parameters.

- For example, in the presence of a small ε, the theoretical choice of $\eta = 4\varepsilon^{-1}\log n$ for Sinkhorn and Greenkhorn barely works in practice, since it causes numerical instability in computing $\exp(-\eta\boldsymbol{W})$ (which is a commonly known issue, cf. Peyré et al. (2019, Chapter 4)); in addition, the solid lines show that within a reasonable range, reducing regularization tends to result in slower convergence but also a smaller error floor after convergence, just as expected.

- With regards to our extragradient method, the theoretical choices of parameters also tend to be too conservative, while the fine-tuned parameters work substantially better, with the aggressive choices of $\eta = 0$ (i.e., no entropic regularization), large C and small C_3 for the stepsizes $\{\eta_{p,i}, \eta_{\mu,j}\}$, and $B = 1$ for strong adjustment of the $\boldsymbol{\mu}$ sequences. Although these fine-tuned parameters were chosen based on early experiments in one or two settings, it turns out that they work uniformly well under various settings; therefore, we will mostly focus on these fine-tuned parameters in the remaining numerical results. The superior performance under such choices might merit further theoretical studies.

Moreover, the careful reader might remark that the numerical curves of the proposed extragradient methods exhibit non-monotonicity; such an oscillation behavior is common in extragradient-type methods, as they are, in general, not descent methods.

[6]The number of matrix-vector products for each iteration of Sinkhorn and extragradient method is 1 and 2, respectively; for Greenkhorn, we set this value to $1/n$, based on the ratio between the numbers of row/column updates per iteration for Sinkhorn and Greenkhorn Altschuler et al. (2017).

We make a few more comments. (1) Our extragradient method converges fast (almost linearly in early iterations) in the "Synthetic" and "MNIST" settings, while at a sublinear rate in the "Point clouds" setting. This is likely because, in the latter case, the optimal transport plan typically lies on or close to the boundary of the polytope, where the optimization landscape is less well-conditioned and thus results in slower convergence. (2) Perhaps unsurprisingly, the actual runtime of one "matrix-vector product" for our extragradient method is longer than that for Sinkhorn, by a constant factor. Note that this constant depends heavily on detailed implementation and hardware resources (CPU/GPU/parallel computation), and our codes are not specifically optimized in these aspects. (3) Another interesting observation is that the adjustment step (cf. (16c)) in Algorithm 1 turns out to have a significant impact on the practical performance under the "Synthetic" and "MNIST" settings. This can be seen from Figure 2, where the solid lines stand for the extragradient method with fine-tuned parameters (as in Figure 1), and the dashed lines correspond to the same except that the adjustment step in Algorithm 1 is skipped. These results showcase that the adjustment step can help avoid getting stuck at undesirable points, thus accelerating convergence.

Comparisons with log-domain Sinkhorn. As we have mentioned, one disadvantage of the classical Sinkhorn method is that it faces a dilemma when the target accuracy ε is small: with a smaller parameter η, the sub-optimality gap saturates after a certain iteration (which can be seen from the "Point clouds" settings in Figure 1), while with the theoretical choice η (on the order of $\log(n)/\varepsilon$, it encounters numerical issues in its first step, i.e., calculating $\exp(-\eta \boldsymbol{W})$. Researchers have found that this issue can be partially addressed by implementing Sinkhorn in the log-domain, though at the cost of heavier computation in practice. Therefore, for a fair comparison, we implement log-domain Sinkhorn according to Peyré et al. (2019, Remark 4.23), and (optionally) add a burn-in phase, i.e. running a few iterations with a larger ε at the beginning, in order to obtain a good initialization.

An empirical comparison between our extragradient method and log-domain Sinkhorn can be found in Figure 3. We observe that, with a small $\varepsilon < 10^{-4}$, log-domain Sinkhorn still struggles with converging. The burn-in trick seems to successfully alleviate this in the "Point clouds" setting and leads to fast convergence to a high-precision solution; however, despite our best efforts, log-domain Sinkhorn still fails to converge properly in the "Synthetic" and "MNIST" settings.

4.2 Comparisons with more recent approaches

We further implement, and compare our algorithm numerically with, two recently proposed methods: (a) the dual extrapolation method from Jambulapati et al. (2019, Algorithm 3)[7], and (b) the DROT method from Mai et al. (2022, Algorithm 1). For the dual extrapolation method, we try our best to fine-tune its parameters (e.g., the step sizes and the numbers of iterations for the inner loops), while for the DROT method, we follow the choices of initialization and parameters described in Mai et al. (2022, Section 4).

The empirical results are illustrated in Figure 4. The dual extrapolation method is outperformed by the other approaches in all three settings. Our extragradient method achieves the best numerical performance in both "Synthetic" and "MNIST" settings, but is outperformed by the DROT algorithm in the "Point Clouds" setting. These numerical findings (together with the previous numerical results) suggest that there might not be a single algorithm that empirically dominates others in every setting, and each algorithm has its own pros and cons.

[7] As mentioned by the authors of Jambulapati et al. (2019), the algorithm that they actually implemented and tested numerically in their work is different from the one with theoretical guarantees (i.e. the one that we implement and compare with); its (pseudo-)codes are not yet publicly available.

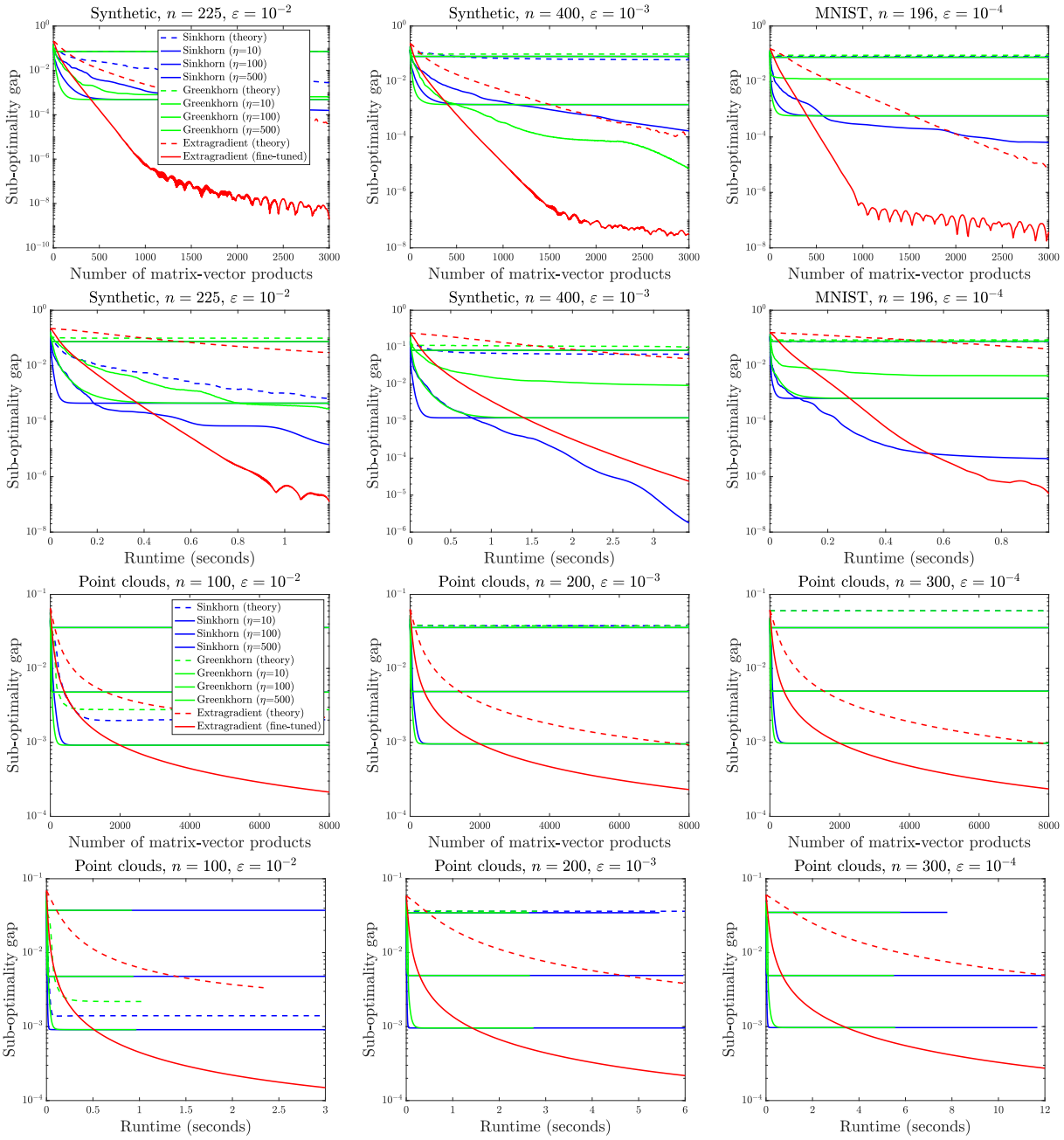

Figure 1: An empirical comparison of the convergence of various algorithms under different settings. Each curve is an average over 10 independent trials. The first and third rows use the number of matrix-vector products as a metric of computational complexities, while the second and fourth use the actual runtime.

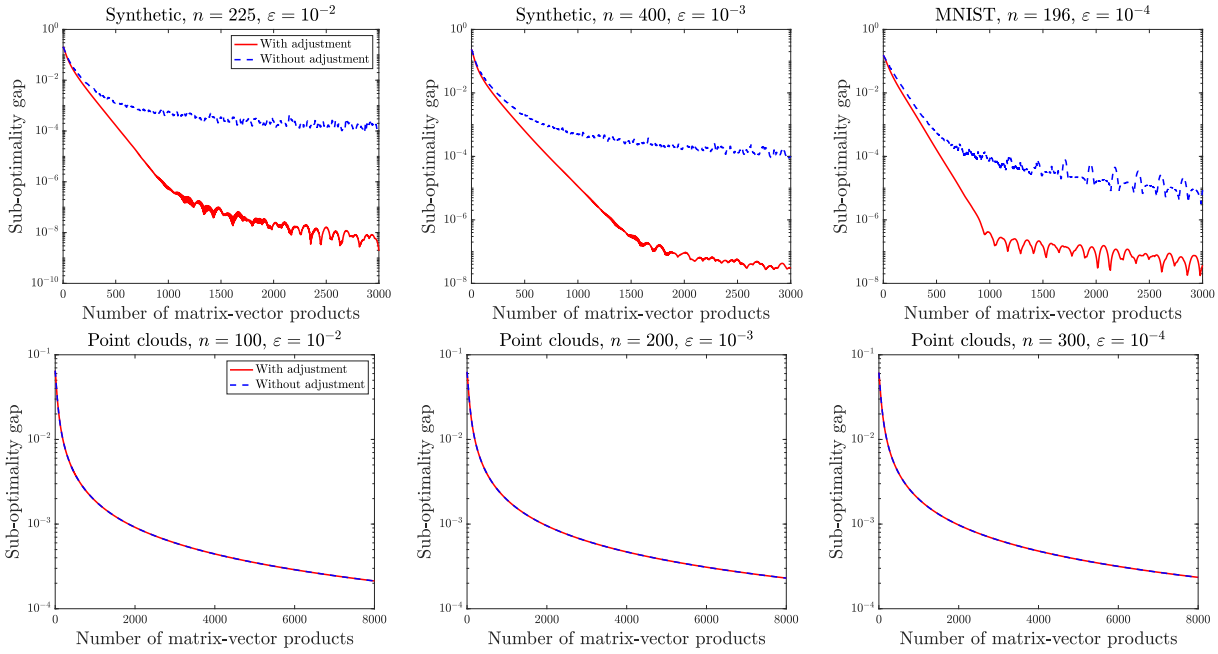

Figure 2: The proposed extragradient method (Algorithm 1) with adjustment step vs. the version without adjustment. The problem settings are the same as those in Figure 1.

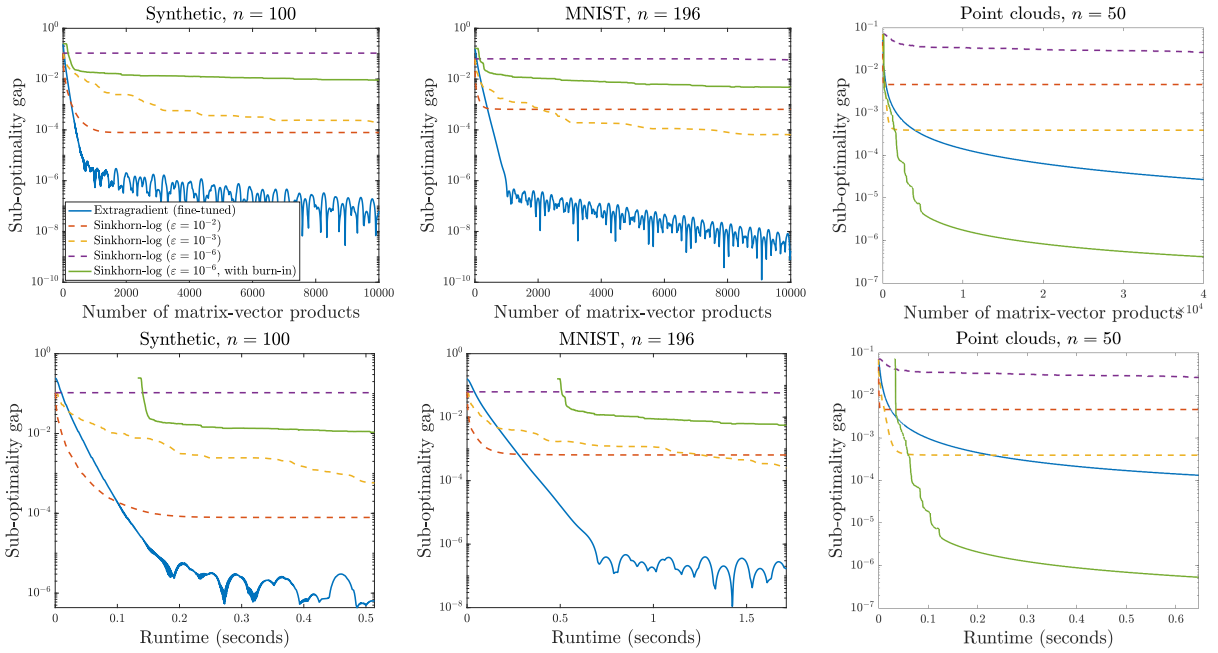

Figure 3: Empirical comparisons between our extragradient method and log-domain Sinkhorn Peyré et al. (2019, Remark 4.23). Each curve is an average over 10 independent trials.

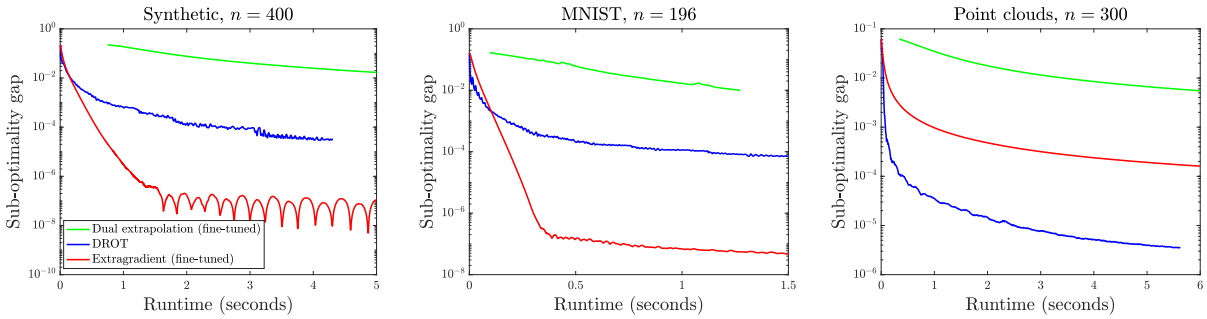

Figure 4: Empirical comparisons between our extragradient method and two recently proposed algorithms, namely the dual extrapolation method Jambulapati et al. (2019, Algorithm 3) and the DROT method Mai et al. (2022, Algorithm 1). Each curve is an average over 10 independent trials.

4.3 Validation of the theoretical convergence guarantees

Finally, we design experiments to validate our theoretical convergence rate, confirming that the runtime of our extragradient method for finding an ε-accurate solution is indeed $\widetilde{O}(n^2/\varepsilon)$. Numerical results under the "Point clouds" setting can be found in Figure 5. The X-axis represents either n^2 (with n up to 10000) or $1/\varepsilon$, and the Y-axis stands for the actual runtime to reach an accuracy ε or a fixed number of iterations; the linear relationship seems evident from the figures.

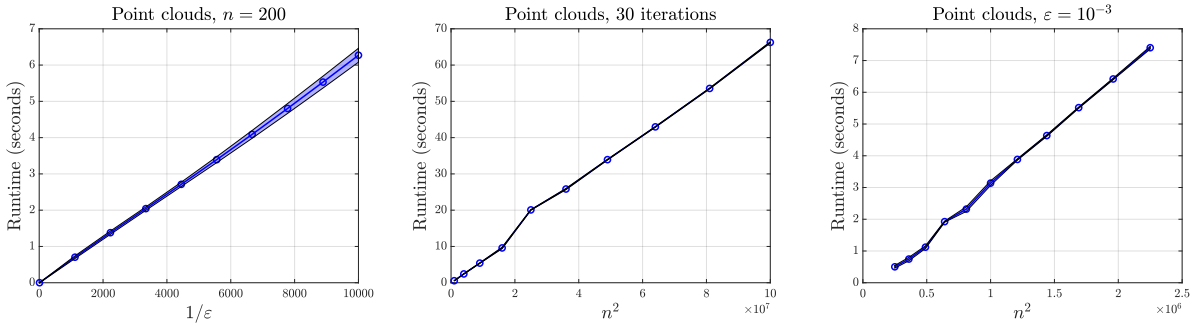

Figure 5: Eempirical validation of our theoretical rate $\widetilde{O}(n^2/\varepsilon)$, where n is the dimension, and ε is the target accuracy. Each curve plots the mean and standard deviation for 10 independent trials. Left: runtime vs. $1/\varepsilon$ for a fixed n. Middle: runtime vs. n^2 for a fixed number of iterations. Right: runtime vs. n^2 for a fixed ε; for this setting, we use the solution output by Sinkhorn as (an approximation of) the optimal solution, since exact calculation (by direct linear programming) of the optimal transport for large n is computationally infeasible.

5 Analysis

In this section, we present the proof for our main result: Theorem 1.

5.1 Preliminary facts and additional notation

Before proceeding to the main proof, let us collect several elementary facts concerning the "adjusting" step in our algorithm. Consider any probability vector $\boldsymbol{q} = [q_i]_{1\le i\le d} \in \Delta_d$, and transform it into another probability vector $\boldsymbol{q}^{\mathsf{adjust}} = [q_i^{\mathsf{adjust}}]_{1\le i\le d} \in \Delta_d$ via the following two steps:

$$q_i^{\mathsf{adjust}} = \frac{\max\left\{q_i,\, e^{-B}\|\boldsymbol{q}\|_\infty\right\}}{\sum_{j=1}^d \max\left\{q_j,\, e^{-B}\|\boldsymbol{q}\|_\infty\right\}}, \qquad i = 1, \cdots, d. \tag{28}$$

We first make note of the following basic property of this transformation:

$$\max_{1\le i\le d} \frac{q_i}{q_i^{\mathsf{adjust}}} \le 1 + de^{-B}. \tag{29}$$

Proof. For each $1 \le i \le d$, it is seen that

$$\begin{aligned}\frac{q_i}{q_i^{\mathsf{adjust}}} &= \frac{q_i}{\max\{q_i,\, e^{-B}\|\boldsymbol{q}\|_\infty\}} \cdot \frac{\sum_j \max\{q_j,\, e^{-B}\|\boldsymbol{q}\|_\infty\}}{\sum_j q_j} \le \frac{\sum_j \max\{q_j,\, e^{-B}\|\boldsymbol{q}\|_\infty\}}{\sum_j q_j} \\ &\le \frac{\sum_j q_j}{\sum_j q_j} + \frac{\sum_j e^{-B}\|\boldsymbol{q}\|_\infty}{\sum_j q_j} \le 1 + de^{-B}\|\boldsymbol{q}\|_\infty\end{aligned}$$

as claimed. □

In addition, consider the special two-dimensional case where $\boldsymbol{q} = [q_+, q_-]^\top \in \Delta_2$ and let $\boldsymbol{q}^{\mathsf{adjust}} = [q_+^{\mathsf{adjust}}, q_-^{\mathsf{adjust}}]^\top \in \Delta_2$. If $B > 0$, then it holds that

$$\frac{q_+^{\mathsf{adjust}}}{q_-^{\mathsf{adjust}}} = \begin{cases} \min\left\{\frac{q_+}{q_-}, e^B\right\} \le e^B, & \text{if } q_+ \ge q_-, \\ \max\left\{\frac{q_+}{q_-}, e^{-B}\right\} \ge e^{-B}, & \text{if } q_+ < q_-. \end{cases} \tag{30}$$

Proof. If $q_+ \ge q_-$, then it is readily seen that

$$\frac{q_+^{\mathsf{adjust}}}{q_-^{\mathsf{adjust}}} = \frac{\max\left\{q_+, e^{-B}q_+\right\}}{\max\left\{q_-, e^{-B}q_+\right\}} = \frac{q_+}{\max\left\{q_-, e^{-B}q_+\right\}} = \min\left\{\frac{q_+}{q_-}, e^B\right\} \le e^B.$$

Similarly, if $q_+ < q_-$, then one can also derive

$$\frac{q_+^{\mathsf{adjust}}}{q_-^{\mathsf{adjust}}} = \frac{\max\left\{q_+, e^{-B}q_-\right\}}{\max\left\{q_-, e^{-B}q_-\right\}} = \frac{\max\left\{q_+, e^{-B}q_-\right\}}{q_-} = \max\left\{\frac{q_+}{q_-}, e^{-B}\right\} \ge e^{-B}.$$

□

Moreover, we would also like to introduce several additional convenient notation that is useful for presenting the proof. Define

$$\boldsymbol{\zeta}^t \coloneqq \left(\left\{\frac{1}{\eta_{p,i}}\boldsymbol{p}_i^t\right\}_{i=1}^n, \left\{\frac{1}{\eta_{\mu,j}}\boldsymbol{\mu}_j^t\right\}_{j=1}^n\right), \tag{31a}$$

$$\overline{\boldsymbol{\zeta}}^t \coloneqq \left(\left\{\frac{1}{\eta_{p,i}}\overline{\boldsymbol{p}}_i^t\right\}_{i=1}^n, \left\{\frac{1}{\eta_{\mu,j}}\overline{\boldsymbol{\mu}}_j^t\right\}_{j=1}^n\right), \tag{31b}$$

$$\boldsymbol{\zeta}^{t,\mathsf{adjust}} \coloneqq \left(\left\{\frac{1}{\eta_{p,i}}\boldsymbol{p}_i^t\right\}_{i=1}^n, \left\{\frac{1}{\eta_{\mu,j}}\boldsymbol{\mu}_j^{t,\mathsf{adjust}}\right\}_{j=1}^n\right), \tag{31c}$$

$$\boldsymbol{\zeta}^\star \coloneqq \left(\left\{\frac{1}{\eta_{p,i}}\boldsymbol{p}_i^{\star,\mathsf{reg}}\right\}_{i=1}^n, \left\{\frac{1}{\eta_{\mu,j}}\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\right\}_{j=1}^n\right), \tag{31d}$$

where $\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\}_{j=1}^n$ denotes the optimizer of the entropy-regularized minimax problem (12). Additionally, for any $\boldsymbol{\zeta}^{(1)} = \big(\{\frac{1}{\eta_{p,i}}\boldsymbol{p}_i^{(1)}\}_{i=1}^n, \{\frac{1}{\eta_{\mu,j}}\boldsymbol{\mu}_j^{(1)}\}_{j=1}^n\big)$ and $\boldsymbol{\zeta}^{(2)} = \big(\{\frac{1}{\eta_{p,i}}\boldsymbol{p}_i^{(2)}\}_{i=1}^n, \{\frac{1}{\eta_{\mu,j}}\boldsymbol{\mu}_j^{(2)}\}_{j=1}^n\big)$ (with the $\boldsymbol{p}_i^{(1)}$'s, $\boldsymbol{p}_i^{(2)}$'s, $\boldsymbol{\mu}_i^{(1)}$'s and $\boldsymbol{\mu}_i^{(2)}$'s being probability vectors), we introduce a weighted KL divergence metric:

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^{(1)} \,\|\, \boldsymbol{\zeta}^{(2)}\big) \coloneqq \sum_{i=1}^n \frac{1}{\eta_{p,i}}\mathsf{KL}\big(\boldsymbol{p}_i^{(1)} \,\|\, \boldsymbol{p}_i^{(2)}\big) + \sum_{j=1}^n \frac{1}{\eta_{\mu,j}}\mathsf{KL}\big(\boldsymbol{\mu}_j^{(1)} \,\|\, \boldsymbol{\mu}_j^{(2)}\big). \tag{32}$$

5.2 Proof of Theorem 1

We are now positioned to prove Theorem 1. To begin with, we make the observation that

$$\begin{aligned}\mathsf{KL}(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^{t,\mathsf{adjust}}) &= \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log \frac{\mu_{j,s}^{\star,\mathsf{reg}}}{\mu_{j,s}^{t,\mathsf{adjust}}} = \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log \frac{\mu_{j,s}^{\star,\mathsf{reg}}}{\mu_{j,s}^{t}} + \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log \frac{\mu_{j,s}^{t}}{\mu_{j,s}^{t,\mathsf{adjust}}} \\ &\le \mathsf{KL}(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^{t}) + \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log \max\left\{ \frac{\mu_{j,+}^{t}}{\mu_{j,+}^{t,\mathsf{adjust}}}, \frac{\mu_{j,-}^{t}}{\mu_{j,-}^{t,\mathsf{adjust}}} \right\} \\ &\le \mathsf{KL}(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^{t}) + \log\big(1+2e^{-B}\big),\end{aligned}$$

where the last inequality arises from the relation (29) and the fact that $\sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} = 1$. As a result, combine the above inequality with the definitions (31c) and (31a) to arrive at

$$\begin{aligned}\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) &= \sum_{i=1}^n \frac{1}{\eta_{p,i}} \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^t\big) + \sum_{j=1}^n \frac{1}{\eta_{\mu,j}} \mathsf{KL}\big(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^{t,\mathsf{adjust}}\big) \\ &\le \sum_{i=1}^n \frac{1}{\eta_{p,i}} \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^t\big) + \sum_{j=1}^n \frac{1}{\eta_{\mu,j}} \mathsf{KL}\big(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^{t}\big) + \log\big(1+2e^{-B}\big) \sum_{j=1}^n \frac{1}{\eta_{\mu,j}} \\ &= \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t}\big) + \log\big(1+2e^{-B}\big) \cdot \sum_{j=1}^n \frac{c_j + C_3/n}{15C_2\sqrt{B}} \\ &= \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t}\big) + \frac{1+C_3}{15C_2\sqrt{C_1 \log\frac{n}{\varepsilon}}} \log\big(1+2e^{-B}\big) \\ &\le \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t}\big) + e^{-B}, \end{aligned} \tag{33}$$

where the third line results from the choice (21), the penultimate line relies on $\sum_j c_j = 1$, and the last line makes use of $\log(1+x) \le x$ for all $x \ge 0$ and holds if $C_2\sqrt{C_1}$ is sufficiently large (recall that $C_3 \le 1$). In a nutshell, (33) indicates that the KL divergence between the optimal point and the t-th iterate is not increased by much when $\boldsymbol{\zeta}^t$ is replaced with $\boldsymbol{\zeta}^{t,\mathsf{adjust}}$.

The next step consists of establishing the following result that monitors the change of KL divergence when $\boldsymbol{\zeta}^{t,\mathsf{adjust}}$ is further replaced with $\boldsymbol{\zeta}^{t+1}$:

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t+1}\big) \le (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) + 8ne^{-B}, \qquad \forall t \ge 0. \tag{34}$$

Crucially, a contraction factor of $1-\eta$ appears in the above claim, revealing the progress made per iteration. Suppose for the moment that this claim (34) is valid. Then taking it together with (33) gives

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t+1}\big) \le (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t}\big) + (8n+1)e^{-B}, \qquad \forall t \ge 0. \tag{35}$$

Applying this relation recursively further implies that

$$\begin{aligned}\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t_{\max}}\big) &\le (1-\eta)^{t_{\max}} \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{0}\big) + (8n+1)e^{-B} \sum_{t=0}^{t_{\max}-1} (1-\eta)^t \\ &\le (1-\eta)^{t_{\max}} \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{0}\big) + \frac{(8n+1)e^{-B}}{\eta}. \end{aligned} \tag{36}$$

Regarding the first term in (36), it is observed from our initialization (cf. line 2 of Algorithm 1) that

$$\begin{cases} \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^0\big) &= -\mathcal{H}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}}\big) + \sum_{j=1}^n p_{i,j}^{\star,\mathsf{reg}} \log\frac{1}{p_{i,j}^0} \le \sum_{j=1}^n p_{i,j}^{\star,\mathsf{reg}} \log n = \log n, \\ \mathsf{KL}\big(\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \,\|\, \boldsymbol{\mu}_j^0\big) &= -\mathcal{H}\big(\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\big) + \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log\frac{1}{\mu_{j,s}^0} \le \sum_{s\in\{+,-\}} \mu_{j,s}^{\star,\mathsf{reg}} \log 2 = \log 2, \end{cases}$$

and consequently,

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{0}\big) \le \sum_{i=1}^n \frac{\log n}{\eta_{p,i}} + \sum_{j=1}^n \frac{\log 2}{\eta_{\mu,j}} = \frac{\sqrt{B}\log n}{C_2} \sum_{i=1}^n r_i + \frac{\log 2}{15C_2\sqrt{B}} \sum_{j=1}^n \Big(c_j + \frac{C_3}{n}\Big) \le \frac{2\sqrt{B}\log n}{C_2},$$

where we recall that $0 < C_3 \leq 1$. This in turn leads to $(1-\eta)^{t_{\max}} \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^0\big) \leq \varepsilon^2$, provided that $t_{\max} \geq C_4 \frac{1}{\eta} \log \frac{n}{\varepsilon}$ for some large enough constant $C_4 > 0$. Turning to the second term in (36), one utilizes $B = C_1 \log \frac{n}{\varepsilon}$ to obtain

$$\frac{(8n+1)e^{-B}}{\eta} = \frac{(8n+1)e^{-B}\sqrt{B}\log n}{C_2^2 \varepsilon} \leq \frac{(8n+1)e^{-\frac{1}{2}B}\log n}{C_2^2\varepsilon} = \frac{(8n+1)e^{-\frac{1}{2}C_1 \log \frac{n}{\varepsilon}}\log n}{C_2^2\varepsilon} \leq \varepsilon^2,$$

with the proviso that $C_1 > 0$ is sufficiently large. Substitution into (36) thus yields

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t_{\max}}\big) \leq \varepsilon^2 + \varepsilon^2 = 2\varepsilon^2. \tag{37}$$

As it turns out, it is more convenient to work with the ℓ_1-based error. To convert the above bound on KL divergence into ℓ_1-based distance, one invokes Pinsker's inequality (Tsybakov, 2009, Lemma 2.5) to reach

$$\begin{aligned}
\big\|\widehat{\boldsymbol{P}} - \boldsymbol{P}^{\star,\mathsf{reg}}\big\|_1 &= \sum_{i=1}^n r_i \big\|\boldsymbol{p}_i^{\star,\mathsf{reg}} - \boldsymbol{p}_i^{\star,t_{\max}}\big\|_1 \leq \sum_{i=1}^n r_i \sqrt{2\mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^{t_{\max}}\big)} \\
&\leq \left\{\sum_{i=1}^n r_i\right\}^{\frac{1}{2}} \left\{2\sum_{i=1}^n r_i \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^{t_{\max}}\big)\right\}^{\frac{1}{2}} \\
&= \left\{2\sum_{i=1}^n r_i \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^{t_{\max}}\big)\right\}^{\frac{1}{2}} \\
&= \left\{\frac{2C_2}{\sqrt{B}} \sum_{i=1}^n \frac{1}{\eta_{p,i}} \mathsf{KL}\big(\boldsymbol{p}_i^{\star,\mathsf{reg}} \,\|\, \boldsymbol{p}_i^{t_{\max}}\big)\right\}^{\frac{1}{2}} \\
&\leq \left\{\frac{2C_2}{\sqrt{C_1 \log \frac{n}{\varepsilon}}} \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t_{\max}}\big)\right\}^{\frac{1}{2}} \leq \frac{\varepsilon}{6}.
\end{aligned} \tag{38}$$

Here, the first inequality invokes Pinsker's inequality, the second inequality results from Cauchy-Schwarz, the third line holds since $\sum_i r_i = 1$, the fourth line uses the choice (20) of $\eta_{p,i}$, whereas the last line results from the definition (32) and the bound (37) and is valid as long as C_2 (resp. C_1) is sufficiently small (resp. large).

Thus far, we have demonstrated fast convergence of our algorithm to the solution to the entropy-regularized problem (12). To finish up, we still need to show the proximity of the objective values under the regularized solution $\boldsymbol{P}^{\star,\mathsf{reg}}$ and under the true optimizer $\boldsymbol{P}^\star$. For notational convenience, define

$$\overline{\boldsymbol{\mu}}_j^\star := \arg\max_{\boldsymbol{\mu}_j \in \Delta_2} f\big(\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{i=1}^n\big), \qquad 1 \leq j \leq n. \tag{39}$$

Given that $\sum_{j=1}^n \tau_{\mu,j} \log 2 + \sum_{i=1}^n \tau_{p,i} \log n \leq \varepsilon/4$ under our choice (22) (see (23) as long as C_2/C_1 is sufficiently small), we obtain

$$\begin{aligned}
\frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{P}^{\star,\mathsf{reg}}\rangle + \big\|\boldsymbol{P}^{\star,\mathsf{reg}}\mathbf{1} - \boldsymbol{c}\big\|_1 &= f\big(\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\overline{\boldsymbol{\mu}}_j^\star\}_{i=1}^n\big) \\
&\leq F\big(\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\overline{\boldsymbol{\mu}}_j^\star\}_{i=1}^n\big) + \sum_{i=1}^n \tau_{p,i} \log n \\
&\leq F\big(\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\}_{i=1}^n\big) + \sum_{i=1}^n \tau_{p,i} \log n \\
&\leq F\big(\{\boldsymbol{p}_i^{\star}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\}_{i=1}^n\big) + \sum_{i=1}^n \tau_{p,i} \log n \\
&\leq f\big(\{\boldsymbol{p}_i^{\star}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\}_{i=1}^n\big) + \sum_{i=1}^n \tau_{p,i} \log n + \sum_{j=1}^n \tau_{\mu,j} \log 2
\end{aligned}$$

$$
\begin{aligned}
&\le \max_{\boldsymbol{\mu}_j \in \Delta_2, \forall j} f\big(\{\boldsymbol{p}_i^\star\}_{i=1}^n, \{\boldsymbol{\mu}_j\}_{i=1}^n\big) + \sum_{i=1}^n \tau_{p,i} \log n + \sum_{j=1}^n \tau_{\mu,j} \log 2 \\
&= \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{P}^\star \rangle + \Big\| \sum_{i=1}^n r_i \boldsymbol{p}_i^\star - \boldsymbol{c} \Big\|_1 + \sum_{i=1}^n \tau_{p,i} \log n + \sum_{j=1}^n \tau_{\mu,j} \log 2 \\
&\le \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{P}^\star \rangle + \frac{\varepsilon}{4}. \qquad (40)
\end{aligned}
$$

Here, the first identity comes from (8) and the definition of $\overline{\boldsymbol{\mu}}_j^\star$, the second line is valid since $\mathcal{H}(\boldsymbol{p}_i^{\star,\mathsf{adjust}}) \le \log n$ for all $1 \le i \le n$, the third and the fourth lines hold since $\big(\{\boldsymbol{p}_i^{\star,\mathsf{reg}}\}_{i=1}^n, \{\boldsymbol{\mu}_j^{\star,\mathsf{reg}}\}_{i=1}^n\big)$ corresponds to the minimax solution of $F(\cdot,\cdot)$, the fifth line relies on $\mathcal{H}(\boldsymbol{\mu}_j^{\star,\mathsf{adjust}}) \le \log 2$ for all $1 \le j \le n$, the penultimate line arises from (8), while the last line results from the fact $\sum_i r_i \boldsymbol{p}_i^\star = \boldsymbol{c}$ and the assumption $\sum_{j=1}^n \tau_{\mu,j} \log 2 + \sum_{i=1}^n \tau_{p,i} \log n \le \varepsilon/4$. As a consequence, we are ready to conclude that

$$
\begin{aligned}
\langle \boldsymbol{W}, \widetilde{\boldsymbol{P}} \rangle &\le \langle \boldsymbol{W}, \widehat{\boldsymbol{P}} \rangle + \|\boldsymbol{W}\|_\infty \|\widehat{\boldsymbol{P}} - \widetilde{\boldsymbol{P}}\|_1 \\
&\le \langle \boldsymbol{W}, \widehat{\boldsymbol{P}} \rangle + 2\big\|\widehat{\boldsymbol{P}}\boldsymbol{1} - \boldsymbol{c}\big\|_1 \\
&\le \langle \boldsymbol{W}, \boldsymbol{P}^{\star,\mathsf{reg}} \rangle + \|\boldsymbol{W}\|_\infty \big\|\widehat{\boldsymbol{P}} - \boldsymbol{P}^{\star,\mathsf{reg}}\big\|_1 + 2\big\|\boldsymbol{P}^{\star,\mathsf{reg}}\boldsymbol{1} - \boldsymbol{c}\big\|_1 + 2\big\|\widehat{\boldsymbol{P}} - \boldsymbol{P}^{\star,\mathsf{reg}}\big\|_1 \|\boldsymbol{1}\|_\infty \\
&= \langle \boldsymbol{W}, \boldsymbol{P}^{\star,\mathsf{reg}} \rangle + 2\big\|\boldsymbol{P}^{\star,\mathsf{reg}}\boldsymbol{1} - \boldsymbol{c}\big\|_1 + 3\big\|\widehat{\boldsymbol{P}} - \boldsymbol{P}^{\star,\mathsf{reg}}\big\|_1 \\
&\le \langle \boldsymbol{W}, \boldsymbol{P}^\star \rangle + \varepsilon/2 + \varepsilon/2 = \langle \boldsymbol{W}, \boldsymbol{P}^\star \rangle + \varepsilon,
\end{aligned}
$$

where the second inequality invokes Lemma 1, the assumption $\|\boldsymbol{W}\|_\infty = 1$, and the fact $\widehat{\boldsymbol{P}}\boldsymbol{1} = \boldsymbol{r}$, and the last inequality arises from (38) and (40). This establishes the advertised result in Theorem 1.

The remainder of the proof is thus dedicated to establishing the claim (34), detailed in the next subsection.

5.3 Proof of Claim (34)

5.3.1 Step 1: decomposing the KL divergence of interest

Elementary calculation together with the definition (32) of $\mathsf{KL}_{\mathsf{gen}}(\cdot \,\|\, \cdot)$ reveals that

$$
\begin{aligned}
&(1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) - (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) - \eta\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^\star\big) - \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^{t+1} \,\|\, \overline{\boldsymbol{\zeta}}^{t+1}\big) \\
&\quad + \big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1} \big\rangle + \big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^{t+1} - (1-\eta)\log \boldsymbol{\zeta}^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\zeta}^\star \big\rangle \\
&= (1-\eta)\big\langle \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^\star - \log \boldsymbol{\zeta}^{t,\mathsf{adjust}} \big\rangle - (1-\eta)\big\langle \overline{\boldsymbol{\zeta}}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t,\mathsf{adjust}} \big\rangle \\
&\quad - \eta\big\langle \overline{\boldsymbol{\zeta}}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^\star \big\rangle - \big\langle \boldsymbol{\zeta}^{t+1}, \log \boldsymbol{\zeta}^{t+1} - \log \overline{\boldsymbol{\zeta}}^{t+1} \big\rangle \\
&\quad + \big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1} \big\rangle + \big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^{t+1} - (1-\eta)\log \boldsymbol{\zeta}^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\zeta}^\star \big\rangle \\
&= \big\langle \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^\star \big\rangle - \big\langle \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^{t+1} \big\rangle = \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t+1}\big); \qquad (41)
\end{aligned}
$$

here and throughout, the logarithmic operator in $\log \boldsymbol{\zeta}$ is applied in an entrywise manner. In addition, inspired by Cen et al. (2021, Lemma 1), we observe that

$$
\big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^\star, \log \boldsymbol{\zeta}^{t+1} - (1-\eta)\log \boldsymbol{\zeta}^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\zeta}^\star \big\rangle = 0; \qquad (42)
$$

see Appendix B for the proof of this relation. Substitution into (41) leads to

$$
\begin{aligned}
\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t+1}\big) = (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) - (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) - \eta\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^\star\big) \\
- \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^{t+1} \,\|\, \overline{\boldsymbol{\zeta}}^{t+1}\big) + \big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1} \big\rangle. \qquad (43)
\end{aligned}
$$

Everything then boils down to controlling the inner product term $\big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1} \big\rangle$.

Towards this end, we first invoke the update rules (15) and (16) to yield

$$\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1}, \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1} \rangle = \sum_{j=1}^{n} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big) \Big\{ \sum_{i=1}^{n} r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \Big\}$$
$$+ \sum_{j=1}^{n} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big) \Big\{ \sum_{i=1}^{n} r_i \big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big) \Big\}, \quad (44)$$

whose proof is also deferred to Appendix B. In order to bound the two sums on the right-hand side of (44), we find it convenient to first introduce the following index subset:

$$\mathcal{J}_t \coloneqq \Big\{ j : \sum_{i=1}^{n} r_i p_{i,j}^{t} < 2e\Big(c_j + \frac{1}{n}\Big) \Big\}. \quad (45)$$

We then divide each sum into two parts — $\{j : j \in \mathcal{J}_t\}$ and $\{j : j \notin \mathcal{J}_t\}$ — and look at them separately.

5.3.2 Step 2: controlling terms with $j \in \mathcal{J}_t$

We start by looking at those terms with $j \in \mathcal{J}_t$. With regards to the first sum on the right-hand side of (44), we have

$$\sum_{j \in \mathcal{J}_t} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big) \Big\{ \sum_{i=1}^{n} r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \Big\}$$
$$\le \sum_{j \in \mathcal{J}_t} \frac{1}{2\eta_{\mu,j}} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)^2 + \sum_{j \in \mathcal{J}_t} \frac{\eta_{\mu,j}}{2} \Big\{ \sum_{i=1}^{n} r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \Big\}^2. \quad (46)$$

Recognizing that $\overline{\boldsymbol{\mu}}_j^{t+1}, \boldsymbol{\mu}_j^{t+1} \in \Delta_2$, one can invoke Pinsker's inequality (Tsybakov, 2009, Lemma 2.5) to bound the first term on the right-hand side of (46) as follows:

$$\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)^2 = \big\|\overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\big\|_1^2 \le 2\mathsf{KL}\big(\boldsymbol{\mu}_j^{t+1} \,\|\, \overline{\boldsymbol{\mu}}_j^{t+1}\big). \quad (47)$$

Regarding the second term on the right-hand side of (46), we see that, for $\eta < 1/2$,

$$\sum_i (1-\eta) \frac{1}{\eta_{p,i}} \mathsf{KL}\big(\overline{\boldsymbol{p}}_i^{t+1} \,\|\, \boldsymbol{p}_i^{t}\big) \ge \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\overline{p}_{i,j}^{t+1}}{\eta_{p,i}} \log \frac{\overline{p}_{i,j}^{t+1}}{p_{i,j}^{t}} \quad (48)$$
$$\ge \frac{1}{2\max_i \eta_{p,i} r_i} \sum_{i=1}^{n} \sum_{j=1}^{n} r_i p_{i,j}^{t} \big((1+x_{ij}) \log(1+x_{i,j})\big)$$
$$\overset{\text{(i)}}{=} \frac{1}{2\max_i \eta_{p,i} r_i} \sum_{j=1}^{n} \sum_{i=1}^{n} r_i p_{i,j}^{t} \big((1+x_{ij}) \log(1+x_{i,j}) - x_{i,j}\big)$$
$$\overset{\text{(ii)}}{\ge} \frac{3}{4\max_i \eta_{p,i} r_i} \sum_{j=1}^{n} \frac{1}{2\sum_{i=1}^{n} r_i p_{i,j}^{t} + \sum_{i=1}^{n} r_i \overline{p}_{i,j}^{t+1}} \Big\{ \sum_{i=1}^{n} r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \Big\}^2$$
$$\overset{\text{(iii)}}{\ge} \frac{1}{20e\big(\max_i \eta_{p,i} r_i\big)\big(\max_j \eta_{\mu,j}(c_j + 1/n)\big)} \sum_{j \in \mathcal{J}_t} \eta_{\mu,j} \Big\{ \sum_{i=1}^{n} r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \Big\}^2, \quad (49)$$

where we define $x_{i,j} \coloneqq \frac{\overline{p}_{i,j}^{t+1}}{p_{i,j}^{t}} - 1$. Here, (i) is valid due to $\sum_j p_{i,j}^{t} x_{i,j} = \sum_j (\overline{p}_{i,j}^{t} - p_{i,j}^{t}) = 0$; (ii) arises since

$$\sum_{i=1}^{n} r_i p_{i,j}^{t} \big((1+x_{i,j}) \log(1+x_{i,j}) - x_{i,j}\big) \ge \frac{1}{2} \sum_{i=1}^{n} r_i p_{i,j}^{t} \frac{x_{i,j}^2}{1 + x_{i,j}/3} = \frac{1}{2} \sum_{i=1}^{n} \frac{(r_i p_{i,j}^{t})^2 x_{i,j}^2}{r_i p_{i,j}^{t} (1 + x_{i,j}/3)}$$

$$\geq \frac{1}{2}\frac{1}{\sum_{i=1}^n r_i p_{i,j}^t (1+x_{i,j}/3)}\Big(\sum_{i=1}^n r_i p_{i,j}^t x_{i,j}\Big)^2 = \frac{3}{4\sum_{i=1}^n r_i p_{i,j}^t + 2\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1}}\Big(\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^t\big)\Big)^2,$$

where the first inequality holds since $(1+z)\log(1+z) - z \geq \frac{z^2}{2(1+z/3)}$ for all $z \geq -1$, and the second line follows from Sedrakyan's inequality (Sedrakyan and Sedrakyan, 2018, Chapter 8, Lemma 1); (iii) comes from the definition (45) of $\mathcal{J}_t$, as well as the following claim (which will be proved momentarily):

$$\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} < 5e\Big(c_j + \frac{1}{n}\Big) \qquad \text{and} \qquad \sum_{i=1}^n r_i p_{i,j}^{t+1} < 5e\Big(c_j + \frac{1}{n}\Big), \qquad \forall j \in \mathcal{J}_t. \tag{50}$$

Substituting (47) and (49) into (46) yields

$$\begin{aligned}
&\sum_{j\in\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)\Big\{\sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big)\Big\} \\
&\leq \sum_{j\in\mathcal{J}_t}\frac{1}{\eta_{\mu,j}}\mathsf{KL}\big(\boldsymbol{\mu}_j^{t+1} \,\|\, \overline{\boldsymbol{\mu}}_j^{t+1}\big) + (1-\eta)40e\Big(\max_i \eta_{p,i} r_i\Big)\Big(\max_j \eta_{\mu,j}(c_j + 1/n)\Big)\sum_i \frac{1}{\eta_{p,i}}\mathsf{KL}\big(\overline{\boldsymbol{p}}_i^{t+1} \,\|\, \boldsymbol{p}_i^t\big) \\
&\leq \sum_{j\in\mathcal{J}_t}\frac{1}{\eta_{\mu,j}}\mathsf{KL}\big(\boldsymbol{\mu}_j^{t+1} \,\|\, \overline{\boldsymbol{\mu}}_j^{t+1}\big) + (1-\eta)\sum_i \frac{1}{\eta_{p,i}}\mathsf{KL}\big(\overline{\boldsymbol{p}}_i^{t+1} \,\|\, \boldsymbol{p}_i^t\big),
\end{aligned} \tag{51}$$

provided that (using the choice (21) and the assumption $0 < C_3 \leq 1$)

$$\Big(\max_i \eta_{p,i} r_i\Big)\Big(\max_j \eta_{\mu,j}(c_j + 1/n)\Big) \leq \frac{1}{C_3}\Big(\max_i \eta_{p,i} r_i\Big)\Big(\max_j \eta_{\mu,j}(c_j + C_3/n)\Big) = \frac{15C_2^2}{C_3} \leq \frac{1}{40e}. \tag{52}$$

We then move on to the second sum on the right-hand side of (44) when restricted to $\mathcal{J}_t$. Repeating the arguments as above, we can guarantee that

$$\begin{aligned}
&\sum_{j\in\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big)\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big) \\
&\quad\leq \sum_{j\in\mathcal{J}_t}\frac{1-\eta}{2\eta_{\mu,j}}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big)^2 + \frac{1}{2(1-\eta)}\sum_{j\in\mathcal{J}_t}\eta_{\mu,j}\Big\{\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big)\Big\}^2 \\
&\quad\leq \sum_j \frac{1-\eta}{\eta_{\mu,j}}\mathsf{KL}\big(\overline{\boldsymbol{\mu}}_j^{t+1} \,\|\, \boldsymbol{\mu}_j^{t,\mathsf{adjust}}\big) + \sum_i \frac{1}{\eta_{p,i}}\mathsf{KL}\big(\boldsymbol{p}_i^{t+1} \,\|\, \overline{\boldsymbol{p}}_i^{t+1}\big),
\end{aligned}$$

as long as $0 < \eta < 1/2$ (so that $\frac{1}{1-\eta} \leq 2$) and Condition (52) is met; the details are omitted here for brevity. Combining this result with (51), we reach

$$\begin{aligned}
&\sum_{j\in\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)\sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big) \\
&\qquad + \sum_{j\in\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big)\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big) \\
&\quad\leq (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) + \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^{t+1} \,\|\, \overline{\boldsymbol{\zeta}}^{t+1}\big).
\end{aligned} \tag{53}$$

Proof of Claim (50). The analyses for $\overline{p}_{i,j}^{t+1}$ and for $p_{i,j}^{t+1}$ are essentially the same; we shall thus only present how to establish the first inequality in (50) for the sake of brevity. According to the update rule (15),

$$\overline{p}_{i,j}^{t+1} = \frac{\big(p_{i,j}^t\big)^{1-\eta}\exp\Big(-\eta_{p,i} r_i\big(0.5w_{i,j} + \mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}\big)\Big)}{\sum_{k=1}^n \big(p_{i,k}^t\big)^{1-\eta}\exp\Big(-\eta_{p,i} r_i\big(0.5w_{i,k} + \mu_{k,+}^{t,\mathsf{adjust}} - \mu_{k,-}^{t,\mathsf{adjust}}\big)\Big)}. \tag{54}$$

Given that $\|\boldsymbol{W}\|_\infty = 1$, $|\mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}| \le 1$ and $0 < \eta < 1$, we can bound

$$\big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i \big(0.5 w_{i,j} + \mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}\big)\Big) \ge p_{i,j}^t \exp\big(-1.5\eta_{p,i} r_i\big) \tag{55a}$$

and

$$\begin{aligned}
\big(p_{i,j}^t\big)^{1-\eta} \exp\Big(-\eta_{p,i} r_i \big(0.5 w_{i,j} + \mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}}\big)\Big) &\le \big(p_{i,j}^t\big)^{1-\eta} \exp\big(1.5\eta_{p,i} r_i\big) \\
&\le \Big\{ p_{i,j}^t e^{\eta B} \mathbb{1}\{p_{i,j}^t > e^{-B}\} + e^{-(1-\eta)B} \mathbb{1}\{p_{i,j}^t \le e^{-B}\}\Big\} \exp\big(1.5\eta_{p,i} r_i\big) \\
&\le p_{i,j}^t \exp\big(1.5\eta_{p,i} r_i + \eta B\big) + \exp\big(1.5\eta_{p,i} r_i - (1-\eta)B\big)
\end{aligned} \tag{55b}$$

for every $1 \le i, j \le n$. Substituting these two bounds into (54) leads to

$$\begin{aligned}
\overline{p}_{i,j}^{t+1} &\le \frac{p_{i,j}^t \exp\big(1.5\eta_{p,i} r_i + \eta B\big) + \exp\big(-(1-\eta)B + 1.5\eta_{p,i} r_i\big)}{\sum_{k=1}^n p_{i,k}^t \exp\big(-1.5\eta_{p,i} r_i\big)} \\
&= \big(p_{i,j}^t + e^{-B}\big) \exp\big(3\eta_{p,i} r_i + \eta B\big) \le 2\big(p_{i,j}^t + e^{-B}\big)
\end{aligned} \tag{56}$$

for every $1 \le i, j \le n$, with the proviso that $3\eta_{p,i} r_i + \eta B \le \log 2$ (which is satisfied under the choice (21) if C_2 is small enough). Therefore, we conclude that for any $j \in \mathcal{J}_t$,

$$\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} \le 2\sum_{i=1}^n r_i p_{i,j}^t + 2\sum_{i=1}^n r_i e^{-B} < 4e\Big(c_j + \frac{1}{n}\Big) + 2e^{-B} < 5e\Big(c_j + \frac{1}{n}\Big), \tag{57}$$

where we make use of the definition (45) of $\mathcal{J}_t$ as well as the fact $\sum_i r_i = 1$, provided that C_1 is large enough.

5.3.3 Step 3: controlling terms with $j \notin \mathcal{J}_t$

We now move on to the terms with $j \notin \mathcal{J}_t$. Employing similar analysis as for (56), we derive

$$\begin{aligned}
p_{i,j}^{t+1} &= \frac{\big(p_{i,j}^t\big)^{1-\eta} \exp\big(-\eta_{p,i} r_i (0.5 w_{i,j} + \overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1})\big)}{\sum_{k=1}^n \big(p_{i,k}^t\big)^{1-\eta} \exp\big(-\eta_{p,i} r_i (0.5 w_{i,k} + \overline{\mu}_{k,+}^{t+1} - \overline{\mu}_{k,-}^{t+1})\big)} \\
&\le \big(p_{i,j}^t + e^{-B}\big) \exp\big(3\eta_{p,i} r_i + \eta B\big)
\end{aligned} \tag{58}$$

for any $1 \le i, j \le n$, which in turn implies that

$$\begin{aligned}
\sum_{i=1}^n r_i p_{i,j}^t &\ge \sum_{i=1}^n r_i \Big\{ \big[\exp\big(3\eta_{p,i} r_i + \eta B\big)\big]^{-1} p_{i,j}^{t+1} - e^{-B} \Big\} \\
&\ge \sum_{i=1}^n r_i \Big[\exp\Big(\max_i 3\eta_{p,i} r_i + \eta B\Big)\Big]^{-1} p_{i,j}^{t+1} - \sum_{i=1}^n r_i e^{-B} \\
&= \Big[\exp\Big(\max_i 3\eta_{p,i} r_i + \eta B\Big)\Big]^{-1} \sum_{i=1}^n r_i p_{i,j}^{t+1} - e^{-B}.
\end{aligned}$$

Hence, for any $j \notin \mathcal{J}_t$ and any integer $0 < \tau < \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}$, applying the above relation recursively yields

$$\begin{aligned}
\sum_{i=1}^n r_i p_{i,j}^{t-\tau} &\ge \Big[\exp\Big(\max_i 3\eta_{p,i} r_i + \eta B\Big)\Big]^{-\tau} \sum_{i=1}^n r_i p_{i,j}^t - e^{-B} \sum_{k=0}^{\tau-1} \Big[\exp\Big(\max_i 3\eta_{p,i} r_i + \eta B\Big)\Big]^{-k} \\
&> \Big[\exp\Big(\max_i 3\eta_{p,i} r_i + \eta B\Big)\Big]^{-\tau} \sum_{i=1}^n r_i p_{i,j}^t - \frac{e^{-B}}{1 - \big[\exp\big(\max_i 3\eta_{p,i} r_i + \eta B\big)\big]^{-1}} \\
&> e^{-1} \sum_{i=1}^n r_i p_{i,j}^t - \frac{e^{-B}}{1 - e^{-\eta B}} = e^{-1} \sum_{i=1}^n r_i p_{i,j}^t - \frac{e^{-C_1 \log \frac{n}{\varepsilon}}}{1 - e^{-\frac{C_2^2 \varepsilon \sqrt{C_1 \log \frac{n}{\varepsilon}}}{\log n}}}
\end{aligned} \tag{59}$$

$$> e^{-1}\sum_{i=1}^{n} r_i p_{i,j}^{t} - \frac{0.5}{n} \geq 2c_j + \frac{1.5}{n}, \tag{60}$$

where the penultimate line relies on the condition $\tau < \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}$, and the last line uses the definition of $\mathcal{J}_t$ in (45) and holds as long as C_1 is large enough.

This lower bound (60) already leads to one important observation. For any $0 \leq t < \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}$, taking $\tau = t$ leads to

$$\sum_{i=1}^{n} r_i p_{i,j}^{0} = \sum_{i=1}^{n} r_i p_{i,j}^{t-\tau} > 2c_j + \frac{1.5}{n}, \tag{61}$$

which contracdicts our initialization $\sum_{i=1}^{n} r_i p_{i,j}^{0} = \frac{1}{n}\sum_{i=1}^{n} r_i = \frac{1}{n} < 2c_j + \frac{1.5}{n}$. This essentially implies that

$$\big\{j : j \notin \mathcal{J}_t\big\} = \emptyset, \qquad \text{for all } 0 \leq t < \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}. \tag{62}$$

Therefore, it suffices to focus on all $t \geq \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}$ when analyzing the sum over $j \notin \mathcal{J}_t$.

Next, recalling the bounds (55), one can derive, for every $1 \leq i, j \leq n$,

$$\begin{aligned}
\overline{p}_{i,j}^{t+1-\tau} &\geq \frac{p_{i,j}^{t-\tau} \exp\big(-1.5\eta_{p,i} r_i\big)}{\sum_{k=1}^{n} p_{i,k}^{t-\tau} \exp\big(1.5\eta_{p,i} r_i + \eta B\big) + \exp\big(-(1-\eta)B + 1.5\eta_{p,i} r_i\big)} \\
&= \frac{p_{i,j}^{t-\tau} \exp\big(-1.5\eta_{p,i} r_i\big)}{\exp\big(1.5\eta_{p,i} r_i + \eta B\big) + \exp\big(-(1-\eta)B + 1.5\eta_{p,i} r_i\big)} \\
&\geq \frac{\big[\exp\big(3\max_i \eta_{p,i} r_i + \eta B\big)\big]^{-1}}{1 + e^{-B}} p_{i,j}^{t-\tau}.
\end{aligned}$$

This allows us to further derive that: for any $j \notin \mathcal{J}_t$ and any integer $0 < \tau < \frac{1}{6\max_i \eta_{p,i} r_i + 2\eta B}$,

$$\begin{aligned}
\sum_{i=1}^{n} r_i \overline{p}_{i,j}^{t+1-\tau} &\geq \frac{\big[\exp\big(3\max_i \eta_{p,i} r_i + \eta B\big)\big]^{-1}}{1 + e^{-B}} \sum_{i=1}^{n} r_i p_{i,j}^{t-\tau} \\
&> \frac{\big[\exp\big(3\max_i \eta_{p,i} r_i + \eta B\big)\big]^{-\tau-1}}{1 + e^{-B}} \sum_{i=1}^{n} r_i p_{i,j}^{t} \\
&\qquad - \frac{\big[\exp\big(3\max_i \eta_{p,i} r_i + \eta B\big)\big]^{-1}}{1 + e^{-B}} \cdot \frac{e^{-B}}{1 - \big[\exp\big(3\max_i \eta_{p,i} r_i + \eta B\big)\big]^{-1}} \\
&> e^{-1}\sum_{i=1}^{n} r_i p_{i,j}^{t} - \frac{1}{1 + e^{-B}} \cdot \frac{e^{-B}}{\exp\big(\frac{3C_2}{\sqrt{B}} + \frac{C_2^2\sqrt{B}\varepsilon}{\log n}\big) - 1} \\
&> e^{-1}\sum_{i=1}^{n} r_i p_{i,j}^{t} - \frac{0.5}{n} > 2c_j + \frac{1.5}{n},
\end{aligned} \tag{63}$$

where the second inequality invokes (59), the third inequality follows as long as $\tau < \frac{1}{6\max_i \eta_{p,i} r_i + 2\eta B}$ and C_1 is sufficiently large, and the last line holds as long as C_1 is sufficiently large (since $B = C_1 \log\frac{n}{\varepsilon}$) and makes use of the assumption that $j \notin \mathcal{J}_t$.

The above two bounds (60) and (63) play a useful role for bounding the changes of $\boldsymbol{\mu}_j^{t,\mathsf{adjust}}$ and $\overline{\boldsymbol{\mu}}_j^{t}$. Consider any $t \geq \frac{1}{3\max_i \eta_{p,i} r_i + \eta B}$ and an integer $\tau_0 = \frac{1}{12\max_i \eta_{p,i} r_i + 4\eta B}$.

- Suppose for the moment that $\mu_{j,+}^{t+1-\tau_0} \geq \mu_{j,-}^{t+1-\tau_0}$. Then in view of the elementary fact (30), we have

$$\frac{\mu_{j,+}^{t+1-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t+1-\tau_0,\mathsf{adjust}}} = \min\left\{\frac{\mu_{j,+}^{t+1-\tau_0}}{\mu_{j,-}^{t+1-\tau_0}}, e^{B}\right\}.$$

Taking this together with the update rule (16) reveals that: for every $j \in \mathcal{J}_t$,

$$\begin{aligned}
\frac{\mu_{j,+}^{t+1-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t+1-\tau_0,\mathsf{adjust}}} &= \min\left\{\left(\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\right)^{1-\eta}\exp\left(2\eta_{\mu,j}\Big(\sum_{i=1}^{n} r_i\overline{p}_{i,j}^{t+1-\tau_0} - c_j\Big)\right),\ e^B\right\} \\
&\geq \min\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\cdot\left(\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\right)^{-\eta}\exp\left(2\eta_{\mu,j}\Big(c_j+\frac{1.5}{n}\Big)\right),\ e^B\right\} \\
&\geq \min\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\exp\left(2\eta_{\mu,j}\Big(c_j+\frac{C_3}{n}\Big)-\eta B\right),\ e^B\right\} \qquad (64) \\
&= \min\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\exp\left(16C_2\sqrt{B}-\frac{C_2^2\varepsilon}{\log n}\sqrt{B}\right),\ e^B\right\} \\
&\geq \min\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}},\ e^B\right\} = \frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}, \qquad (65)
\end{aligned}$$

where the third line relies on (63), the assumption $C_3 \leq 1$, and the elementary fact $\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}} \leq e^B$ (see (30)), the fourth line results from the choice (21), and the last identity holds since $\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}} \leq e^B$ (see (30)). Clearly, repeating the above argument recursively yields

$$\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}} \geq \frac{\mu_{j,+}^{t-1,\mathsf{adjust}}}{\mu_{j,-}^{t-1,\mathsf{adjust}}} \geq \cdots \geq \frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}. \qquad (66)$$

Furthermore, if $\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}} < \exp(B)$, then we have also seen from (64) that

$$\frac{\mu_{j,+}^{t+1-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t+1-\tau_0,\mathsf{adjust}}} \geq \frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\exp\left(2\eta_{\mu,j}\Big(c_j+\frac{C_3}{n}\Big)-\eta B\right),$$

and similarly,

$$\begin{aligned}
\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}} &\geq \frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\exp\left\{\tau_0\Big(2\eta_{\mu,j}\Big(c_j+\frac{C_3}{n}\Big)-\eta B\Big)\right\} \\
&\geq \exp\left\{\tau_0\Big(2\eta_{\mu,j}\Big(c_j+\frac{C_3}{n}\Big)-\eta B\Big)\right\} \geq e^B
\end{aligned}$$

as long as $\frac{2\eta_{\mu,j}(c_j+C_3/n)-\eta B}{12\max_i \eta_{p,i}r_i+4\eta B} > 2B$ (a condition that is satisfied under our choice of parameters). This, however, leads to contradiction. Thus, we necessarily have

$$\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}} = e^B, \qquad \text{if } \frac{\mu_{j,+}^{t+1-\tau_0}}{\mu_{j,-}^{t+1-\tau_0}} \geq 1. \qquad (67)$$

- On the other hand, consider the case where $\mu_{j,+}^{t+1-\tau_0} < \mu_{j,-}^{t+1-\tau_0}$. Then the basic fact (30) gives

$$\frac{\mu_{j,+}^{t+1-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t+1-\tau_0,\mathsf{adjust}}} = \max\left\{\frac{\mu_{j,+}^{t+1-\tau_0}}{\mu_{j,-}^{t+1-\tau_0}}, e^{-B}\right\}.$$

Repeating the analysis for (65) tells us that: for every $j \in \mathcal{J}_t$,

$$\frac{\mu_{j,+}^{t+1-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t+1-\tau_0,\mathsf{adjust}}} \geq \max\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}\exp\left(2\eta_{\mu,j}\Big(c_j+\frac{C_3}{n}\Big)-\eta B\right),\ e^{-B}\right\}$$

$$\geq \max\left\{\frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}},\ e^{-B}\right\} = \frac{\mu_{j,+}^{t-\tau_0,\mathsf{adjust}}}{\mu_{j,-}^{t-\tau_0,\mathsf{adjust}}}. \tag{68}$$

With such monotonicity in place, repeat the argument for (67) to show that (which we omit for brevity)

$$\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}} = e^B, \qquad \text{if } \frac{\mu_{j,+}^{t+1-\tau_0}}{\mu_{j,-}^{t+1-\tau_0}} < 1, \tag{69}$$

provided that $\frac{2\eta_{\mu,j}(c_j+C_3/n)-\eta B}{12\max_i \eta_{p,i} r_i + 4\eta B} > 2B$ (again, this is satisfied under our choice of parameters).

Combining (67) and (69) and reusing the argument in (65) further show that

$$\begin{aligned}\frac{\overline{\mu}_{j,+}^{t+1}}{\overline{\mu}_{j,-}^{t+1}} &= \min\left\{\left(\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}}\right)^{1-\eta}\exp\left(2\eta_{\mu,j}\Big(\sum_{i=1}^n r_i p_{i,j}^t - c_j\Big)\right),\ e^B\right\} \\ &\geq \min\left\{\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}}\exp\left(2\eta_{\mu,j}\Big(c_j + \frac{C_3}{n}\Big) - \eta B\right),\ e^B\right\} \geq e^B\end{aligned}$$

and

$$\begin{aligned}\frac{\mu_{j,+}^{t+1}}{\mu_{j,-}^{t+1}} &= \min\left\{\left(\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}}\right)^{1-\eta}\exp\left(2\eta_{\mu,j}\Big(\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} - c_j\Big)\right),\ e^B\right\} \\ &\geq \min\left\{\frac{\mu_{j,+}^{t,\mathsf{adjust}}}{\mu_{j,-}^{t,\mathsf{adjust}}}\exp\left(2\eta_{\mu,j}\Big(c_j + \frac{C_3}{n}\Big) - \eta B\right),\ e^B\right\} \geq e^B,\end{aligned}$$

provided that $\frac{2\eta_{\mu,j}(c_j+C_3/n)-\eta B}{12\max_i \eta_{p,i} r_i + 4\eta B} > 2B$. In turn, these two bounds tell us that

$$\overline{\mu}_{j,-}^{t+1} \leq \frac{1}{1+e^B} \leq e^{-B} \qquad \text{and} \qquad \mu_{j,-}^{t+1} \leq \frac{1}{1+e^B} \leq e^{-B}.$$

Armed with these results as well as (62), we can conclude that for all $t \geq 0$,

$$\begin{aligned}&\sum_{j\notin\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)\sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big) \\ &\qquad + \sum_{j\notin\mathcal{J}_t}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big)\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big) \\ &\leq \sum_{j\notin\mathcal{J}_t} 2\big|\overline{\mu}_{j,-}^{t+1} - \mu_{j,-}^{t+1}\big|\left|\sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big)\right| + \sum_{j\notin\mathcal{J}_t} 2\big|\overline{\mu}_{j,-}^{t+1} - \mu_{j,-}^{t,\mathsf{adjust}}\big|\left|\sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big)\right| \\ &\leq 4\sum_{j\notin\mathcal{J}_t}\Big(\big|\overline{\mu}_{j,-}^{t+1} - \mu_{j,-}^{t+1}\big| + \big|\overline{\mu}_{j,-}^{t+1} - \mu_{j,-}^{t,\mathsf{adjust}}\big|\Big) \leq 8ne^{-B}.\end{aligned} \tag{70}$$

5.3.4 Step 4: putting all this together

Putting (53) and (70) together with (44), we arrive at

$$\big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1},\, \log\overline{\boldsymbol{\zeta}}^{t+1} - \log\boldsymbol{\zeta}^{t+1}\big\rangle \leq (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\overline{\boldsymbol{\zeta}}^{t+1} \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) + \mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^{t+1} \,\|\, \overline{\boldsymbol{\zeta}}^{t+1}\big) + 8ne^{-B}.$$

Substituting it back into (43), we reach

$$\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t+1}\big) \leq (1-\eta)\mathsf{KL}_{\mathsf{gen}}\big(\boldsymbol{\zeta}^\star \,\|\, \boldsymbol{\zeta}^{t,\mathsf{adjust}}\big) + 8ne^{-B},$$

thereby concluding the proof of Claim (34).

6 Discussion

In this paper, we have put forward a first-order method for computing the optimal transport at scale, which has been shown to enjoy both intriguing convergence guarantees and favorable numerical performance. This is a step we have taken towards closing the theory-practice gap for solving this problem. Moving forward, there are several natural research directions to explore. To begin with, while the state-of-the-art theory van den Brand et al. (2020) demonstrated the feasibility of a runtime $O(n^2 \log^2(1/\varepsilon))$, the practical value of the algorithm proposed therein remains unrealized; it would be of great importance to design algorithms that are optimal in theory and practice at once. Next, the current algorithm still involves several hyper-parameters to tune, and it would be of interest to develop improved versions that are nearly parameter-free. Also, there is no shortage of applications where the problems exhibit certain low-dimensional structure (e.g., Altschuler et al. (2019)), which could be potentially leveraged to achieve further computational savings. Another natural problem to explore is whether we can solve a more general family of linear programs — e.g., the ones taking the form $\text{minimize}_{\boldsymbol{x}\in\Delta_n} f(\boldsymbol{x}) + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1$ — using the entropy-regularized extragradient method developed herein. We leave these for future investigation.

Acknowledgements

Y. Chen is supported in part by the Alfred P. Sloan Research Fellowship, the Google Research Scholar Award, the AFOSR grant FA9550-22-1-0198, the ONR grant N00014-22-1-2354, and the NSF grants CCF-2221009, CCF-1907661, IIS-2218713 and IIS-2218773. Y. Chi is supported in part by the ONR grant N00014-19-1-2404.

A Converting a non-negative matrix to a transportation plan

Given a general non-negative matrix $\widehat{\boldsymbol{P}} \in \mathbb{R}_+^{n\times n}$, Altschuler et al. (2017) put forward a simple algorithm that returns a probability matrix $\widetilde{\boldsymbol{P}} \in \Delta_{n\times n}$ satisfying $\widetilde{\boldsymbol{P}}\boldsymbol{1} = \boldsymbol{r}$ and $\widetilde{\boldsymbol{P}}^\top \boldsymbol{1} = \boldsymbol{c}$ while simultaneously obeying the condition in Lemma 1. We include this algorithm here in order to be self-contained; here, we recall that $\mathsf{row}_i(\boldsymbol{F})$ (resp. $\mathsf{col}_i(\boldsymbol{F})$) denotes the sum of the i-th row (resp. column) of a matrix $\boldsymbol{F}$.

Algorithm 3: Converting a matrix $\widehat{\boldsymbol{P}}$ to a feasible transportation plan (Altschuler et al., 2017).

Input: $\widehat{\boldsymbol{P}} \in \mathbb{R}_+^{n\times n}$, and two probability vectors $\boldsymbol{r} = [r_i]_{1\le i\le n}, \boldsymbol{c} = [c_i]_{1\le i\le n} \in \Delta_n$.
for $i = 1$ **to** n **do**
  $x_i = \min\big\{\frac{r_i}{\mathsf{row}_i(\widehat{\boldsymbol{P}})}, 1\big\}$.
$\boldsymbol{F}' = \mathsf{diag}([x_i]_{1\le i\le n})\widehat{\boldsymbol{P}}$.
for $i = 1$ **to** n **do**
  $y_i = \min\big\{\frac{c_i}{\mathsf{col}_i(\boldsymbol{F}')}, 1\big\}$.
$\boldsymbol{F}'' = \boldsymbol{F}'\mathsf{diag}([y_i]_{1\le i\le n})$.
$\boldsymbol{e}_{\mathrm{r}} = \boldsymbol{r} - \mathsf{row}(\boldsymbol{F}'')$; $\boldsymbol{e}_{\mathrm{c}} = \boldsymbol{c} - \mathsf{col}(\boldsymbol{F}'')$.
Output: $\widetilde{\boldsymbol{P}} = \boldsymbol{F}'' + \boldsymbol{e}_{\mathrm{r}}\boldsymbol{e}_{\mathrm{c}}^\top / \|\boldsymbol{e}_{\mathrm{r}}\|_1$.

B Proof of Equations (42) and (44)

Proof of the identity (42). The optimizer of the regularized minimax problem (12) necessarily satisfies the following optimality condition: for each $1 \le j \le n$,

$$\log \mu_{j,s}^{\star,\mathsf{reg}} = \alpha_{\mu,j} + \frac{\eta_{\mu,j}}{\eta} s\bigg(\sum_{i=1}^n r_i p_{i,j}^{\star,\mathsf{reg}} - c_j\bigg) \qquad s \in \{+,-\} \tag{71}$$

for some normalization factor $\alpha_{\mu,j}$; this can be easily seen by setting the gradient of the objective function to zero and utilizing the constraint $\boldsymbol{\mu}_j^{\star,\mathsf{reg}} \in \Delta_2$ as well as $\tau_{\mu,j} = \frac{\eta}{\eta_{\mu,j}}$. Additionally, the update rule (16) implies that

$$\log \mu_{j,s}^{t+1} = \beta_{\mu,j} + (1-\eta)\log \mu_{j,s}^{t,\mathsf{adjust}} + \eta \cdot \frac{\eta_{\mu,j}}{\eta} s\bigg(\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} - c_j\bigg), \qquad s \in \{+,-\}, \tag{72}$$

where $\beta_{\mu,j}$ is some normalization constant. Taking the preceding two identities together and using the basic facts $\big\langle \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{\star,\mathsf{reg}}, \mathbf{1}\big\rangle = 0$ give

$$\begin{aligned}
&\big\langle \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{\star,\mathsf{reg}},\, \log \boldsymbol{\mu}_j^{t+1} - (1-\eta)\log \boldsymbol{\mu}_j^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\mu}_j^{\star,\mathsf{reg}}\big\rangle \\
&\quad = \eta_{\mu,j}\big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{\star,\mathsf{reg}} + \mu_{j,-}^{\star,\mathsf{reg}}\big) \cdot \sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{\star,\mathsf{reg}}\big)
\end{aligned}$$

for any $1 \le j \le n$. A similar argument also leads to

$$\begin{aligned}
&\big\langle \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{\star,\mathsf{reg}},\, \log \boldsymbol{p}_i^{t+1} - (1-\eta)\log \boldsymbol{p}_i^{t} - \eta \log \boldsymbol{p}_i^{\star,\mathsf{reg}}\big\rangle \\
&\quad = \eta_{p,i} \sum_{j=1}^n \big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{\star,\mathsf{reg}}\big) \cdot r_i\big(\mu_{j,+}^{\star,\mathsf{reg}} - \mu_{j,-}^{\star,\mathsf{reg}} - \overline{\mu}_{j,+}^{t+1} + \overline{\mu}_{j,-}^{t+1}\big)
\end{aligned}$$

for any $1 \le i \le n$. Putting the above two identities together allows us to conclude that

$$\begin{aligned}
&\big\langle \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{\star},\, \log \boldsymbol{\zeta}^{t+1} - (1-\eta)\log \boldsymbol{\zeta}^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\zeta}^{\star}\big\rangle \\
&\quad = \sum_{j=1}^n \frac{1}{\eta_{\mu,j}} \big\langle \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{\star,\mathsf{reg}},\, \log \boldsymbol{\mu}_j^{t+1} - (1-\eta)\log \boldsymbol{\mu}_j^{t,\mathsf{adjust}} - \eta \log \boldsymbol{\mu}_j^{\star,\mathsf{reg}}\big\rangle \\
&\qquad + \sum_{i=1}^n \frac{1}{\eta_{p,i}} \big\langle \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{\star,\mathsf{reg}},\, \log \boldsymbol{p}_i^{t+1} - (1-\eta)\log \boldsymbol{p}_i^{t} - \eta \log \boldsymbol{p}_i^{\star,\mathsf{reg}}\big\rangle \\
&\quad = \sum_{j=1}^n \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{\star,\mathsf{reg}} + \mu_{j,-}^{\star,\mathsf{reg}}\big) \cdot \sum_{i=1}^n r_i\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{\star,\mathsf{reg}}\big) \\
&\qquad + \sum_{i=1}^n \sum_{j=1}^n \big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{\star,\mathsf{reg}}\big) \cdot r_i\big(\mu_{j,+}^{\star,\mathsf{reg}} - \mu_{j,-}^{\star,\mathsf{reg}} - \overline{\mu}_{j,+}^{t+1} + \overline{\mu}_{j,-}^{t+1}\big) \;\; = 0.
\end{aligned} \tag{73}$$

Proof of the identity (44). Repeating similar arguments as in the proof of (42) and using the update rules (15) and (16), we can also deduce that: for any $1 \le j \le n$,

$$\begin{aligned}
\big\langle \log \overline{\boldsymbol{\mu}}_j^{t+1}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\big\rangle &= \bigg\langle (1-\eta)\log \boldsymbol{\mu}_j^{t,\mathsf{adjust}} + \eta_{\mu,j}\Big(\sum_{i=1}^n r_i p_{i,j}^t - c_j\Big)\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\bigg\rangle, \\
\big\langle \log \boldsymbol{\mu}_j^{t+1}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\big\rangle &= \bigg\langle (1-\eta)\log \boldsymbol{\mu}_j^{t,\mathsf{adjust}} + \eta_{\mu,j}\Big(\sum_{i=1}^n r_i \overline{p}_{i,j}^{t+1} - c_j\Big)\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\bigg\rangle,
\end{aligned}$$

and as a result,

$$\begin{aligned}
\frac{1}{\eta_{\mu,j}}\big\langle \log \overline{\boldsymbol{\mu}}_j^{t+1} - \log \boldsymbol{\mu}_j^{t+1}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\big\rangle &= \bigg\langle \sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big)\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1}\bigg\rangle \\
&= \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big)\sum_{i=1}^n r_i\big(p_{i,j}^t - \overline{p}_{i,j}^{t+1}\big).
\end{aligned}$$

Similarly, the update rules (15) and (16) also indicate that

$$\big\langle \log \overline{\boldsymbol{p}}_i^{t+1}, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1}\big\rangle = \Big\langle (1-\eta)\log \boldsymbol{p}_i^t - \eta_{p,i} r_i\big(0.5\boldsymbol{w}_i + \boldsymbol{\mu}_+^{t,\mathsf{adjust}} - \boldsymbol{\mu}_-^{t,\mathsf{adjust}}\big), \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1}\Big\rangle$$

$$\big\langle \log \boldsymbol{p}_i^{t+1},\, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1} \big\rangle = \Big\langle (1-\eta)\log \boldsymbol{p}_i^{t} - \eta_{p,i} r_i \big(0.5\boldsymbol{w}_i + \overline{\boldsymbol{\mu}}_+^{t+1} - \overline{\boldsymbol{\mu}}_-^{t+1}\big),\, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1} \Big\rangle$$

with $\boldsymbol{\mu}_s^{t,\mathsf{adjust}} \coloneqq [\mu_{j,s}^{t,\mathsf{adjust}}]_{1\le j\le n}$ and $\boldsymbol{\mu}_s^{t} \coloneqq [\mu_{j,s}^{t}]_{1\le j\le n}$ for all $s \in \{+,-\}$, which in turn yield

$$\begin{aligned}
\frac{1}{\eta_{p,i}} \big\langle \log \overline{\boldsymbol{p}}_i^{t+1} - \log \boldsymbol{p}_i^{t+1}, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1} \big\rangle &= -r_i \Big\langle \boldsymbol{\mu}_+^{t,\mathsf{adjust}} - \boldsymbol{\mu}_-^{t,\mathsf{adjust}} - \overline{\boldsymbol{\mu}}_+^{t+1} + \overline{\boldsymbol{\mu}}_-^{t+1},\, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1} \Big\rangle \\
&= -\sum_{j=1}^{n} r_i \big(\mu_{j,+}^{t,\mathsf{adjust}} - \mu_{j,-}^{t,\mathsf{adjust}} - \overline{\mu}_{j,+}^{t+1} + \overline{\mu}_{j,-}^{t+1}\big)\big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big).
\end{aligned}$$

Taking the above results together, we arrive at the advertised relation:

$$\begin{aligned}
\big\langle \log \overline{\boldsymbol{\zeta}}^{t+1} - \log \boldsymbol{\zeta}^{t+1}, \overline{\boldsymbol{\zeta}}^{t+1} - \boldsymbol{\zeta}^{t+1} \big\rangle &= \sum_{j=1}^{n} \frac{1}{\eta_{\mu,j}} \big\langle \log \overline{\boldsymbol{\mu}}_j^{t+1} - \log \boldsymbol{\mu}_j^{t+1},\, \overline{\boldsymbol{\mu}}_j^{t+1} - \boldsymbol{\mu}_j^{t+1} \big\rangle \\
&\quad + \sum_{i=1}^{n} \frac{1}{\eta_{p,i}} \big\langle \log \overline{\boldsymbol{p}}_i^{t+1} - \log \boldsymbol{p}_i^{t+1},\, \overline{\boldsymbol{p}}_i^{t+1} - \boldsymbol{p}_i^{t+1} \big\rangle \\
&= \sum_{j=1}^{n} \sum_{i=1}^{n} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t+1} + \mu_{j,-}^{t+1}\big) r_i \big(p_{i,j}^{t} - \overline{p}_{i,j}^{t+1}\big) \\
&\quad + \sum_{j=1}^{n} \sum_{i=1}^{n} \big(\overline{\mu}_{j,+}^{t+1} - \overline{\mu}_{j,-}^{t+1} - \mu_{j,+}^{t,\mathsf{adjust}} + \mu_{j,-}^{t,\mathsf{adjust}}\big) r_i \big(\overline{p}_{i,j}^{t+1} - p_{i,j}^{t+1}\big).
\end{aligned}$$

References

Allen-Zhu, Z. and Orecchia, L. (2015). Nearly-linear time positive LP solver with faster convergence rate. In *Annual ACM symposium on Theory of Computing*, pages 229–236.

Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. (2019). Massively scalable Sinkhorn distances via the Nyström method. *Advances in neural information processing systems*, 32.

Altschuler, J., Niles-Weed, J., and Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30.

Altschuler, J. M. (2022). Flows, scaling, and entropy revisited: a unified perspective via optimizing joint distributions. *arXiv preprint arXiv:2210.16456*.

An, D., Lei, N., Xu, X., and Gu, X. (2022). Efficient optimal transport algorithm by accelerated gradient descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10119–10128.

Ao, R., Cen, S., and Chi, Y. (2023). Asynchronous gradient play in zero-sum multi-agent games. In *International Conference on Learning Representations (ICLR)*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Blanchet, J., Jambulapati, A., Kent, C., and Sidford, A. (2018). Towards optimal running times for optimal transport. *arXiv preprint arXiv:1810.07717*.

Cen, S., Chen, F., and Chi, Y. (2022a). Independent natural policy gradient methods for potential games: Finite-time global convergence with entropy regularization. In *IEEE Conference on Decision and Control (CDC)*.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2022b). Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578.

Cen, S., Chi, Y., Du, S. S., and Xiao, L. (2023). Faster last-iterate convergence of policy optimization in zero-sum Markov games. In *International Conference on Learning Representations (ICLR)*.

Cen, S., Wei, Y., and Chi, Y. (2021). Fast policy extragradient methods for competitive games with entropy regularization. *Advances in Neural Information Processing Systems*, 34:27952–27964.

Chakrabarty, D. and Khanna, S. (2021). Better and simpler error analysis of the Sinkhorn–Knopp algorithm for matrix scaling. *Mathematical Programming*, 188(1):395–407.

Chambolle, A. and Contreras, J. P. (2022). Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein Barycenter problems. *arXiv preprint arXiv:2203.00802*.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Daskalakis, C. and Panageas, I. (2018). Last-iterate convergence: Zero-sum games and constrained min-max optimization. *arXiv preprint arXiv:1807.04252*.

Dvurechensky, P., Gasnikov, A., and Kroshnin, A. (2018). Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *International conference on machine learning*, pages 1367–1376.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouvé, A., and Peyré, G. (2019). Interpolating between optimal transport and mmd using Sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR.

Gayraud, N. T., Rakotomamonjy, A., and Clerc, M. (2017). Optimal transport applied to transfer learning for P300 detection. In *BCI 2017-7th Graz Brain-Computer Interface Conference*, page 6.

Geist, M., Scherrer, B., and Pietquin, O. (2019). A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29.

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

Guo, W., Ho, N., and Jordan, M. (2020). Fast algorithms for computational optimal transport and wasserstein barycenter. In *International Conference on Artificial Intelligence and Statistics*, pages 2088–2097. PMLR.

Harker, P. T. and Pang, J.-S. (1990). Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical programming*, 48(1):161–220.

Hsieh, Y.-G., Iutzeler, F., Malick, J., and Mertikopoulos, P. (2019). On the convergence of single-call stochastic extra-gradient methods. *Advances in Neural Information Processing Systems*, 32.

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

Kalantari, B., Lari, I., Ricca, F., and Simeone, B. (2008). On the complexity of general matrix scaling and entropy minimization via the RAS algorithm. *Mathematical Programming*, 112(2):371–401.

Kantorovich, L. V. (1942). On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201.

Kim, Y. M., Mitra, N. J., Huang, Q., and Guibas, L. (2013). Guided real-time scanning of indoor objects. In *Computer Graphics Forum*, volume 32, pages 177–186. Wiley Online Library.

Knight, P. A. (2008). The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275.

Korpelevich, G. M. (1976). The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756.

Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. *Naval research logistics quarterly*, 3(4):253–258.

Lahn, N., Mulchandani, D., and Raghvendra, S. (2019). A graph theoretic additive approximation of optimal transport. *Advances in Neural Information Processing Systems*, 32.

Lan, G. (2022). Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, pages 1–48.

Lee, Y. T. and Sidford, A. (2014). Path finding methods for linear programming: Solving linear programs in $\widetilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 424–433. IEEE.

Liang, T. and Stokes, J. (2019). Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915. PMLR.

Lin, T., Ho, N., and Jordan, M. I. (2022). On the efficiency of entropic regularized algorithms for optimal transport. *Journal of Machine Learning Research*, 23(137):1–42.

Mai, V. V., Lindbäck, J., and Johansson, M. (2022). A fast and accurate splitting method for optimal transport: analysis and implementation. In *International Conference on Learning Representations*.

McKelvey, R. D. and Palfrey, T. R. (1995). Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38.

Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. (2020). On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C.-S., Chandrasekhar, V., and Piliouras, G. (2018a). Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*.

Mertikopoulos, P., Papadimitriou, C., and Piliouras, G. (2018b). Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2703–2717. SIAM.

Mertikopoulos, P. and Sandholm, W. H. (2016). Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324.

Mokhtari, A., Ozdaglar, A., and Pattathil, S. (2020a). A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507.

Mokhtari, A., Ozdaglar, A. E., and Pattathil, S. (2020b). Convergence rate of $O(1/k)$ for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Nemirovski, A. (2004). Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251.

Nesterov, Y. (2007). Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344.

Neu, G., Jonsson, A., and Gómez, V. (2017). A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*.

Pele, O. and Werman, M. (2009). Fast and robust earth mover's distances. In *2009 IEEE 12th international conference on computer vision*, pages 460–467. IEEE.

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

Quanrud, K. (2018). Approximating optimal transport with linear programs. *arXiv preprint arXiv:1810.05957*.

Rakhlin, A. and Sridharan, K. (2013). Optimization, learning, and games with predictable sequences. *arXiv preprint arXiv:1311.1869*.

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

Savas, Y., Ahmadi, M., Tanaka, T., and Topcu, U. (2019). Entropy-regularized stochastic games. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5955–5962. IEEE.

Sedrakyan, H. and Sedrakyan, N. (2018). *Algebraic inequalities*. Springer.

Sherman, J. (2017). Area-convexity, ℓ_∞ regularization, and undirected multicommodity flow. In *Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460.

Sinkhorn, R. (1967). Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015). Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):1–11.

Tseng, P. (1995). On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer.

van den Brand, J., Lee, Y.-T., Nanongkai, D., Peng, R., Saranurak, T., Sidford, A., Song, Z., and Wang, D. (2020). Bipartite matching in nearly-linear time on moderately dense graphs. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 919–930.

Villani, C. (2009). *Optimal transport: old and new*, volume 338. Springer.

von Neumann, J. (1928). Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320.

Wei, C.-Y., Lee, C.-W., Zhang, M., and Luo, H. (2021). Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations (ICLR)*.

Werman, M., Peleg, S., and Rosenfeld, A. (1985). A distance metric for multidimensional histograms. *Computer Vision, Graphics, and Image Processing*, 32(3):328–336.

Xie, Y., Luo, Y., and Huo, X. (2022a). An accelerated stochastic algorithm for solving the optimal transport problem. *arXiv preprint arXiv:2203.00813*.

Xie, Y., Luo, Y., and Huo, X. (2022b). Solving a special type of optimal transport problem by a modified Hungarian algorithm. *arXiv preprint arXiv:2210.16645*.

Zhan, W., Cen, S., Huang, B., Chen, Y., Lee, J. D., and Chi, Y. (2023). Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091.