Log-Averaged Mirror Prox for Fast, Large-Scale Optimal Transport in Linear Space

Matthew X. Burns* Jiaming Liang†

May 11, 2026

Abstract

We propose Log-Averaged Mirror Prox (LAMP), a linear-space primal-dual method for large-scale optimal transport. LAMP implements primal mirror prox updates by tracking an averaged dual sequence, reducing storage complexity from $\mathcal{O}(nm)$ to $\mathcal{O}(n+m)$ while preserving dense, GPU-friendly reductions. Consequently, LAMP preserves the last-iterate $\widetilde{\mathcal{O}}(nm\varepsilon^{-1})$ arithmetic complexity of conservatively parameterized primal-dual mirror prox. We further analyze LAMP as a direct optimal transport solver in a more performant parameter regime, providing a last-iterate sub-optimality certificate dependent on infeasibility and an explicit $\mathcal{O}(1/t)$ term. Moreover, we give a computable sufficient condition for best-iterate convergence to a saddle-point. Numerical experiments with an optimized CUDA implementation show that LAMP outperforms first-order baselines in several high-accuracy (entropic) optimal transport problems. LAMP is further shown to scale up to problems with $n = m = 2^{18}$ marginal supports, which were previously beyond the reach of primal-dual first-order methods.

Key words. optimal transport, entropic optimal transport, mirror prox, primal-dual methods, GPU acceleration

AMS subject classifications. 49Q22, 49M37, 65K05, 68Q25, 90C25

1 Introduction

Given probability mass functions $r \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ and a cost matrix $C \in \mathbb{R}^{n\times m}$, the (discrete) optimal transport (OT) problem is the linear program

$$\min_{X\in\Pi(r,c)} \langle C, X\rangle\,, \tag{1}$$

where $\Pi(r,c)$ is the set of couplings between r and c. Variations of OT have found applications in generative modeling [2], computational science [24, 18], robotics [43], domain adaptation [11], economics [15]; underscoring the need for computationally efficient, large-scale OT. Accordingly, OT has become an increasingly popular area of research at the intersection of first-order optimization and high-performance parallel computing. Following the seminal work of [12], numerous (entropic) OT-focused algorithms have been proposed that target highly parallel, GPU-amenable subroutines. The standard Sinkhorn algorithm [39, 12] computes a solution with ε-additive error in $\widetilde{\mathcal{O}}(nm\varepsilon^{-2})$ operations [13]. Primal-dual methods have been proposed to improve the dependence on ε to $\widetilde{\mathcal{O}}(nm(n+m)^{1/2}\varepsilon^{-1})$ [13, 17, 26] or $\widetilde{\mathcal{O}}(nm\varepsilon^{-1})$ [19, 30, 29] (see Appendix A and Table 2 for further literature review). However, existing primal-dual methods require primal averaging, which either necessitates $\mathcal{O}(nm)$ storage or GPU-unfriendly recovery subroutines dependent on sparse access patterns and pointer-based data structures [3]. In a recent development, entropy-regularized Primal-Dual Mirror Prox (PDMP) methods by [7, 25] achieve $\mathcal{O}(nm\varepsilon^{-1})$ last-iterate complexity for computing an ε-additive solution:

*Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 (email: mburns13@ur.rochester.edu).

†Goergen Institute for Data Science and Artificial Intelligence (GIDS-AI) and Department of Computer Science, University of Rochester, Rochester, NY 14620 (email: jiaming.liang@rochester.edu). This work was partially supported by GIDS-AI seed funding and AFOSR grant FA9550-25-1-0182.

requiring no postprocessing or ergodic averaging. However, as formulated, these methods still require $\mathcal{O}(nm)$ storage, which is prohibitive for large-scale OT applications. In many cases, entries of the cost matrix C can be computed on-the-fly using a distance kernel with $\mathcal{O}(n+m)$ storage, making explicit representation of X the dominant storage cost. In this work, we provide the following contributions:

1. We develop Log-Averaged Mirror Prox (LAMP), a dual-only variant of PDMP that works entirely in the dual space. LAMP preserves the theoretical guarantees of PDMP while reducing storage complexity from $\mathcal{O}(nm)$ to $\mathcal{O}(n+m)$, making the method scalable to large-scale OT instances. Unlike the semi-streaming dual-extrapolation approach of [3], LAMP avoids sparse, pointer-based recovery and maps directly to dense GPU reductions.
2. We further analyze LAMP as a direct OT method in the practical parameter regime, where the entropic regularization parameter is set to zero. In particular, we prove a last-iterate objective-error bound controlled by marginal infeasibility and an explicit $\mathcal{O}(1/t)$ term, as well as a computable sufficient condition for best-iterate convergence to a saddle-point.
3. Supported by numerical experiments, we demonstrate that LAMP outperforms comparable GPU-accelerated first-order solvers in several benchmark and real-world OT problems. Our LAMP implementation is publicly available and shown to scale to problems with $n = m = 2^{18}$ marginal supports, and our CUDA-accelerated codebase is shown to outperform the baseline PyKeOps library in dense reduction operations.

2 Preliminaries

Throughout, $\mathbb{R}$ denotes the real numbers and $\mathbb{R}_+$ are the non-negative reals. By Δ^n, we mean the n-dimensional unit simplex. Similarly, $\{\Delta^m\}^n$ denotes the set of $n \times m$ row-stochastic matrices (non-negative matrices whose rows sum to one). For a vector $x \in \mathbb{R}^n$, $\mathcal{D}_x \in \mathbb{R}^{n\times n}$ is the matrix with x along the diagonal. By $x \in [-1,1]^m$, we mean for all $1 \le i \le m$, $-1 \le x_i \le 1$.

Given a closed, proper, convex and differentiable function $\omega : \mathbb{R}^n \to (-\infty, \infty]$, we define its Bregman divergence as

$$\mathrm{D}_\omega(x\|y) = \omega(x) - \omega(y) - \langle \nabla\omega(y), x - y\rangle .$$

Unless otherwise indicated, $\|\cdot\|$ is the standard Euclidean vector norm, and $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are the ℓ_1 and ℓ_∞ norms. A norm applied to a matrix is meant in an entry-wise manner, e.g., $\|C\|_\infty = \max_{ij} |C_{ij}|$. We denote the Euclidean inner product by $\langle x, y\rangle = x^\top y$. When applied to matrices or vectors, functions log, exp, and tanh are to be interpreted in an entry-wise fashion. We define the entropy of a distribution $\gamma \in \Delta^n$ as $H(\gamma) = -\langle \gamma, \log\gamma\rangle$ with the convention that $0\log 0 = 0$ and the LogSumExp function $\mathrm{LSE}(x) = \log\left[\sum_{j=1}^m \exp(x_j)\right]$ where $x \in \mathbb{R}^m$. For convenience, we denote the Bregman divergence of the negative entropy as the KL-divergence $\mathrm{D}_{\mathrm{KL}}(X\|Y) = \mathrm{D}_{-H}(X\|Y)$. For a matrix $A \in \mathbb{R}^{n\times m}$, $\mathbf{r}(A) \in \mathbb{R}^n$ is the row-wise sum and $\mathbf{c}(A) \in \mathbb{R}^m$ is the column-wise sum. We denote $\mathbf{1}_m$ as the all-ones vector in $\mathbb{R}^m$ and $\mathbf{0}_m$ as the all-zeros vector in $\mathbb{R}^m$.

2.1 Optimal Transport

For $\varepsilon > 0$, we say that $X \in \Pi(r,c)$ is an ε-solution to the primal OT problem (1) if

$$\langle C, X\rangle - \min_{X\in\Pi(r,c)} \langle C, X\rangle \le \varepsilon.$$

As formulated, the OT problem is an $n \cdot m$ dimensional linear program with $m + n$ constraints. Interior point methods exploiting the low-dimensional constraints can solve (1) to high-precision in $\widetilde{\mathcal{O}}((n+m)^{5/2})$ arithmetic operations [23], however the underlying subroutines are not amenable to parallelization, and therefore cannot leverage the explosion in concurrent computing driven by widespread GPU adoption.

The computational challenges of OT have led to the wide adoption of entropic optimal transport (EOT) [12, 32]. EOT augments the OT objective function with a negative entropy term, making the objective η-strongly convex with respect to the ℓ_1 and ℓ_2 norms,

$$\min_{X\in\Pi(r,c)} \{\langle C, X\rangle - \eta H(X)\}. \tag{2}$$

The predominant method for solving (2) is the Sinkhorn-Knopp matrix-scaling algorithm (or just "Sinkhorn" for brevity). Sinkhorn is simple to analyze, performs well for $\eta \geq 10^{-3}$, and is highly parallelizable. Ever since the popularization of Sinkhorn in [12], numerous works have analyzed [1, 13, 16] or extended [5, 26] the basic approach. As proven in [13], Sinkhorn has a computational complexity of $\widetilde{\mathcal{O}}(nm\varepsilon^{-2})$ for finding an ε-solution to (1). This is unacceptably slow for high-accuracy (small ε) OT, motivating further approaches for EOT based on accelerated gradient descent [13], mirror descent [26], and saddle-point methods [19, 8].

2.2 Penalty and Saddle-Point Formulations

An alternative penalty formulation of OT/EOT investigated by [19] and [25] is the problem

$$\min_{p\in\{\Delta^m\}^n} \{P^\eta(\mathcal{D}_r p) := \langle C, \mathcal{D}_r p\rangle + 2\|C\|_\infty \|\mathbf{c}_r(p) - c\|_1 - \eta H_r(p)\}, \tag{3}$$

where $\mathbf{c}_r(p) := \mathbf{c}(\mathcal{D}_r p)$, $H_r(p) := H(\mathcal{D}_r p)$, and $\eta \geq 0$ is the entropic regularization coefficient. There are two primary differences between (1)/(2) and (3). First, we reparameterize the primal variables $X = \mathcal{D}_r p$ from the matrix simplex $X \in \Delta^{n\times m}$ to the row-stochastic matrix $p \in \{\Delta^m\}^n$. One can easily show that $\mathcal{D}_r p \in \Delta^{n\times m}$ with row marginal $\mathbf{r}(\mathcal{D}_r p) = r$, therefore the row parameterization directly enforces the row marginal constraint. The reparameterization is not unique to this formulation, and has been used in other (E)OT algorithms [8]. Second, we replace the constraint $\mathbf{c}_r(p) = c$ with an ℓ_1 penalty term, making the problem unconstrained but nonsmooth.

From [19, Lemma 2.3], an optimizer to problem (3) with $\eta = 0$ is reducible to an optimizer to (1) in $\mathcal{O}(nm)$ operations. For completeness, we extend the equivalence result to the case where $\eta > 0$ in Appendix C (see Lemma C.5), however our primary target is the unregularized problem (1).

To practically solve (3), we dualize the ℓ_1 penalty with the identity $\|x\|_1 = \max_{y\in[-1,1]^m} \langle y, x\rangle$ for $x \in \mathbb{R}^m$, obtaining the primal-dual formulation

$$\min_{p\in\{\Delta^m\}^n} \max_{\theta\in[-1,1]^m} \{K^\eta(\mathcal{D}_r p, \theta) := \langle C, \mathcal{D}_r p\rangle + 2\|C\|_\infty \langle \theta, \mathbf{c}_r(p) - c\rangle - \eta H_r(p)\}. \tag{4}$$

Since (4) is convex-concave over compact domains, Sion's minimax theorem [40] permits us to interchange the min and max, and solve (4) as a saddle-point problem.

3 Log-Averaged Mirror Prox

A popular method for solving saddle-point problems over simple, compact sets is mirror prox [31], which extends classical extragradient methods to non-Euclidean domains. In this section, we show that recently proposed PDMP [7, 25] with last-iterate guarantees can be implemented in $\mathcal{O}(n+m)$ storage by connecting primal mirror descent to dual log-averaging.

3.1 Mirror Descent as Dual Log-Averaging

First, we define the Bregman divergence $\mathrm{D}_{H_c^\alpha}(\theta^a \| \theta^b)$ where $H_c^\alpha(\cdot)$ is the negative dual entropy

$$H_c^\alpha(\nu) = \sum_{j=1}^m c_j^\alpha \left(\frac{1+\nu_j}{2}\ln\left[\frac{1+\nu_j}{2}\right] + \frac{1-\nu_j}{2}\ln\left[\frac{1-\nu_j}{2}\right]\right),$$

where $c^\alpha := c + (\alpha/m)\mathbf{1}_m$. The $\mathrm{D}_{H_c^\alpha}$ and D_{KL} divergences will serve as the movement limiting potentials for the primal and dual steps, respectively, and therefore play a key role in our mirror prox definitions.

Let $F_\theta^\eta(\cdot) = K^\eta(\mathcal{D}_r \cdot, \theta)$ be the primal function defined by the saddle objective in (4) with the dual variables fixed at θ, and similarly define $G_p^\eta(\cdot) = K^\eta(\mathcal{D}_r p, \cdot)$. We then define the primal and dual mirror maps as

$$\begin{aligned} \mathcal{M}_\tau^\eta(p^0; \theta) &= \operatorname*{argmin}_{p\in\{\Delta^m\}^n} \{\tau \left\langle \nabla F_\theta^\eta(p^0), p\right\rangle + \mathrm{D}_{\mathrm{KL}}(\mathcal{D}_r p \| \mathcal{D}_r p^0)\}, \\ \mathcal{M}_\tau^\eta(\theta^0; p) &= \operatorname*{argmax}_{\theta\in\mathbb{R}^m} \{\tau \left\langle \nabla G_p^\eta(\theta^0), \theta\right\rangle - \mathrm{D}_{H_c^\alpha}(\theta \| \theta^0)\}, \end{aligned} \tag{5}$$

which we can show have closed-form solutions (see Appendix C)

$$\mathcal{M}_\tau^\eta(p^0;\theta)_{ij} = \frac{(p^0)_{ij}^{1-\tau\eta}}{Z_i}\exp[-\tau(C_{ij}+2\|C\|_\infty\theta_j)],$$
$$\mathcal{M}_\tau^\eta(\theta^0;p)_j = \tanh\left[\frac{2\tau\|C\|_\infty}{c_j^\alpha}(\mathbf{c}_r(p)_j - c_j) + \frac{1}{2}\log\frac{1+\theta_j^0}{1-\theta_j^0}\right]. \tag{6}$$

Here Z_i is the normalization constant to ensure $\mathcal{M}_\tau^\eta(p^0;\theta) \in \{\Delta^m\}^n$. Additionally, we define the tanh clipping function $\mathrm{tclip}(\theta,\beta)$, which performs coordinate-wise clipping to the subset $[-\tanh(\beta/2),\tanh(\beta/2)]^m \subseteq [-1,1]^m$ for some $\beta > 0$. As shown in [25], dual clipping is theoretically useful and substantially improves empirical performance.

Finally, for $\theta \in [-1,1]^m$ and $\eta > 0$, we define the dual-to-primal map $p^\eta(\theta)$ as

$$p^\eta(\theta) = \operatorname*{argmin}_{p\in\{\Delta^m\}^n}\{\langle C + 2\|C\|_\infty\theta, \mathcal{D}_r p\rangle - \eta H_r(p)\} = \mathcal{D}_Z^{-1}\exp[-\eta^{-1}(C + 2\|C\|_\infty \mathbf{1}_n\theta^\top)], \tag{7}$$

where $\mathcal{D}_Z^{-1}$ enforces row normalization.

With the main ingredients in place, we prove a simple connection between the primal mirror map and the dual variable θ. The following lemma is the key observation that enables us to reduce PDMP from $\mathcal{O}(nm)$ space to $\mathcal{O}(n+m)$ space in the following subsection by replacing the primal iterates with a weighted average in the dual. The proof is deferred to Appendix C.

Lemma 3.1. *Let $\eta \geq 0$, $\gamma > 0$, and $\tau > 0$ satisfy $\tau\eta \leq 1$. Then let θ^a, $\theta^b \in [-1,1]^m$. Define $\eta' = \gamma/(1+\tau(\gamma-\eta))$ and $\theta' = \theta^a + \tau\eta'(\theta^b - \theta^a)$. Then, we have the equivalence*

$$p^{\eta'}(\theta') = \mathcal{M}_\tau^\eta(p^\gamma(\theta^a);\theta^b). \tag{8}$$

3.2 From PDMP to LAMP

Using the $\mathcal{M}_\tau^\eta(\cdot;p)$ and $\mathcal{M}_\tau^\eta(\cdot;\theta)$ primitives, [25] proposed a PDMP method with last-iterate guarantees. Algorithm 1 summarizes PDMP, where the Round function [1] is a standard OT subroutine, which returns a feasible transport plan and is given in Appendix C. Since PDMP is composed entirely of mirror map operations, Lemma 3.1 implies that we can recover the primal sequences $\{p^t\}$ and $\{\bar{p}^t\}$ using auxiliary dual sequences $\{\nu^t\}$ and $\{\bar{\nu}^t\}$. The resulting LAMP method is given in Algorithm 2.

Algorithm 1 Primal-Dual Mirror Prox

Require: $C \in \mathbb{R}_+^{n\times m}$, $r \in \Delta^n$, $c \in \Delta^m$, $\alpha, \beta > 0$, $\tau_1, \tau_2 > 0$, $\eta \geq 0$, $T \in \mathbb{N} > 0$, set $p^0 = (1/m)^{n\times m}$, $\theta^0 = \mathbf{0}_m$, $c^\alpha = c + \alpha m^{-1}\mathbf{1}_m$.
for $t = 0$ to $T-1$ **do**
Step 1) Compute

$$\bar{p}^{t+1} = \mathcal{M}_{\tau_1}^\eta(p^t;\theta^t) \tag{9}$$
$$\bar{\theta}^{t+1} = \mathcal{M}_{\tau_2}^\eta(\theta^t;p^t)$$
$$p^{t+1} = \mathcal{M}_{\tau_1}^\eta(p^t;\bar{\theta}^{t+1}) \tag{10}$$
$$\hat{\theta}^{t+1} = \mathcal{M}_{\tau_2}^\eta(\theta^t;\bar{p}^{t+1})$$
$$\theta^{t+1} = \mathrm{tclip}(\hat{\theta}^{t+1},\beta) \tag{11}$$

end for
Step 2) return $\mathrm{Round}(\mathcal{D}_r p^T, r, c)$.

Algorithm 2 Log-Averaged Mirror Prox

Require: $C \in \mathbb{R}_+^{n\times m}$, $r \in \Delta^n$, $c \in \Delta^m$, $\alpha, \beta > 0$, $\tau_1, \tau_2 > 0$, $\eta \geq 0$, $T \in \mathbb{N}$, set $\theta^0 = \nu^0 = \mathbf{0}_m$, $c^\alpha = c + \alpha m^{-1}\mathbf{1}_m$, $\eta_0 = \infty$.
for $t = 0$ to $T-1$ **do**
Step 1) Compute

$$\eta_{t+1} = \eta_t/(1+\tau_1(\eta_t - \eta)) \tag{12}$$
$$\bar{\nu}^{t+1} = \nu^t + \tau_1\eta_{t+1}(\theta^t - \nu^t) \tag{13}$$
$$\bar{\theta}^{t+1} = \mathcal{M}_{\tau_2}^\eta(\theta^t; p^{\eta_t}(\nu^t)) \tag{14}$$
$$\nu^{t+1} = \nu^t + \tau_1\eta_{t+1}(\bar{\theta}^{t+1} - \nu^t) \tag{15}$$
$$\hat{\theta}^{t+1} = \mathcal{M}_{\tau_2}^\eta(\theta^t; p^{\eta_{t+1}}(\bar{\nu}^{t+1})) \tag{16}$$
$$\theta^{t+1} = \mathrm{tclip}(\hat{\theta}^{t+1},\beta)$$

end for
Step 2) return $\mathrm{Round}(\mathcal{D}_r p^{\eta_T}(\nu^T), r, c)$.

The following proposition formalizes the equivalence between PDMP and LAMP. See Appendix C for the proof.

Proposition 3.2. *Consider the sequences $\{p^t\}$ and $\{\bar{p}^t\}$ from Algorithm 1 and $\{\nu^t\}$ and $\{\bar{\nu}^t\}$ from Algorithm 2. Assume that $\tau_1\eta \leq 1$ and set $\eta_0 = \infty$. Then, we have the equivalence*

$$p^t = p^{\eta_t}(\nu^t), \quad \bar{p}^{t+1} = p^{\eta_{t+1}}(\bar{\nu}^{t+1}).$$

Therefore, LAMP is a dual implementation of PDMP, tracking the weighted dual averages $\{\nu^t\}$ and $\{\bar{\nu}^t\}$ and the scalar sequence $\{\eta_t\}$ to implicitly store primal information. By the closed-form solution in (6), we only need the primal marginal $\mathbf{c}_r(p^\eta(\nu))$ to compute $\mathcal{M}^\eta_\tau(\theta; p^\eta(\nu))$ in (14) and (16), which can be computed using $\mathcal{O}(n+m)$ space as discussed in Subsection 4.1. Note that Round only involves low-rank corrections, therefore the last step of Algorithm 2 can be performed implicitly with $\mathcal{O}(n+m)$ memory and $\mathcal{O}(nm)$ operations.
Remark: Our contribution extends beyond PDMP, as any other mirror prox variant with non-ergodic convergence guarantees can be similarly implemented by tracking the low-dimensional dual variables.

The $\{\eta_t\}$ sequence in (12) is particularly noteworthy. Observe that η_{t+1}^{-1} satisfies the recursion $\eta_{t+1}^{-1} = \tau_1 + \eta_t^{-1}(1-\tau_1\eta)$. Note that $\eta_1 = 1/\tau_1$ by taking the limit $\eta_0 \to \infty$. Solving the recursion gives the generic form $\eta_{t+1}^{-1} = \eta^{-1}(1-(1-\tau_1\eta)^{t+1})$, which becomes $\eta_t^{-1} = \tau_1 t$ in the limit $\eta \to 0$. Therefore, PDMP and LAMP are actually *temperature annealing* methods. Annealing has been highly effective in prior EOT methods [37, 21], and provides a novel perspective on mirror prox methods for OT/EOT.

PDMP has been proposed and analyzed in recent works for entropy-regularized games [7] and OT [25]. In particular, [25] proved that, with certain parameters, Algorithm 1 achieves $\widetilde{\mathcal{O}}(nm\varepsilon^{-1})$ complexity. Since LAMP is equivalent to PDMP by Proposition 3.2, the complexity result recalled below applies to LAMP as well.

Theorem 3.3 ([25, Theorem 2.2], Informal)**.** *Given $\varepsilon > 0$, under the parameter choices of [25] with $\eta = \mathcal{O}(\varepsilon/(\|C\|_\infty \log m))$, PDMP/LAMP computes an ε-solution to (1) in $\widetilde{\mathcal{O}}(nm\|C\|_\infty\varepsilon^{-1})$ arithmetic operations.*

The full statement along with specific parameter choices can be found in Appendix F (see Theorem F.1). We note that the guarantees of Theorem F.1 hold *only* for a conservative set of parameters. Crucially, the guarantees require $\eta > 0$, therefore following the traditional Sinkhorn-type model of solving weakly regularized EOT to solve OT. As observed in [25] and our testing, a more performant and empirically stable set of parameters is $\eta = 0$, $\tau_1 = \tau_2 = 1/(2\|C\|_\infty)$, and $\alpha = 0.01$. With $\eta = 0$, we obtain a *direct* OT method: targeting (1) without the intermediate EOT step.

The following proposition provides computable optimality guarantees for the sequences generated by Algorithm 2 as well as a sufficient condition for finding approximate saddle-points of (4). Unlike Theorem 3.3, Proposition 3.4 provides guarantees for PDMP/LAMP in the fully unregularized regime.

Proposition 3.4. *Suppose $\tau_1 = \tau_2 = 1/(2\|C\|_\infty)$, $\eta = 0$, and $\alpha > 0$. Let $(X^*, \theta^*) \in \Pi(r,c) \times [-1/2, 1/2]^m$ be a saddle-point of $K^0(\cdot,\cdot)$ in (4) where X^* is also a minimizer of (1). For each iteration of Algorithm 2, define $X^t := \mathcal{D}_r p^{\eta_t}(\nu^t)$. Then, for all iterations $t \geq 1$ of Algorithm 2 (or, equivalently, Algorithm 1), the following statements hold*

a)

$$\left\langle C, \widetilde{X}^t - X^* \right\rangle \leq 4\|C\|_\infty \|\mathbf{c}(X^t) - c\|_1 + \frac{2\|C\|_\infty \log m}{t}, \tag{17}$$

where $\widetilde{X}^t = \mathit{Round}(X^t, r, c)$;

b) defining

$$D_t := \sum_{s=0}^{t-1} \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \hat{\theta}^{s+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \theta^s) + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^{s+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^s), \tag{18}$$

where $\bar{X}^t := \mathcal{D}_r p^t(\bar{\nu}^t)$, then

$$\min_{1 \leq s \leq t} \{K^0(\bar{X}^s, \theta^*) - K^0(X^*, \bar{\theta}^s)\} \leq \frac{2\|C\|_\infty[\log m + (1+\alpha)\log 2 + D_t]}{t}. \tag{19}$$

Round (see Algorithm 3) from [1] returns a feasible point $\widetilde{X}^t \in \Pi(r,c)$ in $\mathcal{O}(nm)$ operations satisfying $\|\widetilde{X}^t - X^t\|_1 \leq 2\|\mathbf{c}(X^t) - c\|_1$ (see Lemma C.1). The right-hand side of (17) decomposes the rounded objective

error into two terms: the column infeasibility and an $\mathcal{O}(1/t)$ initialization bias. Therefore, the column infeasibility and iteration count provide a *computable, last-iterate optimality certificate for LAMP*. As shown in our numerical results, LAMP finds feasible points quite rapidly. The remaining open theoretical question is to prove infeasibility convergence, which would provide a full convergence analysis.

The second part of Proposition 3.4 gives a complementary midpoint guarantee. If the partial sums D_t are $o(t)$, then the saddle-point convergence follows. Indeed, additional numerical experiments in Appendix D show that (18) is benign across a variety of OT problems, with D_t rapidly becoming nonpositive and appearing to converge to zero. A uniform bound $D_t = \mathcal{O}(\log mn)$ would yield an $\mathcal{O}(\varepsilon^{-1}\|C\|_\infty \log mn)$ iteration complexity, while a bound $D_t = \mathcal{O}(\log(nmt))$ would imply an $\mathcal{O}(\varepsilon^{-1}\|C\|_\infty \log(nm\varepsilon^{-1}))$ iteration complexity for finding an ε-approximate saddle-point of (4).

4 Implementation Details

4.1 Log-Domain Operation

We compute the column marginal $\mathbf{c}_r(p^{\eta_t}(\nu))$ using a two-step process. First, we compute the row-wise log-normalization constants $\log Z_i$ using the LogSumExp trick: $\mathrm{LSE}(x) = \mathrm{LSE}(x - \max_i x_i) + \max_i x_i$. Denoting $m_i = \max_j\{-\eta_t^{-1}(C_{ij} + 2\|C\|_\infty \nu_j)\}$, we have

$$\log Z_i = \mathrm{LSE}(-\eta_t^{-1}(C_{i:} + 2\|C\|_\infty \nu) - m_i) + m_i,$$
$$\mathbf{c}_r(p^{\eta_t}(\nu))_j = \sum_{i=1}^{n} r_i \exp\left[-\frac{1}{\eta_t}(C_{ij} + 2\|C\|_\infty \nu_j) - \log Z_i\right].$$

Each term $-\eta_t^{-1}(C_{ij} + 2\|C\|_\infty \nu_j)$ is computed on-the-fly and accumulated into either an $\mathcal{O}(n)$ buffer (for $\log Z_i$ terms) or an $\mathcal{O}(m)$ buffer (for $\mathbf{c}_r(p^{\eta_t}(\nu))_j$ terms), leading to the claimed $\mathcal{O}(n+m)$ space complexity of LAMP.

4.2 Optimized Reductions

Log-domain computations are commonly used to stabilize EOT algorithms [35, 37], and are common in widely-distributed OT/EOT packages [14]. Our primary challenge was to perform the operations in a computationally efficient manner. Log-domain computation requires three reduction operations: row-wise max, row-wise LSE, and finally a column-wise sum.

To reduce reduction overhead, we utilize custom kernels leveraging warp-tiling and kernel/loop fusion. Warp-tiling uses a single warp to perform each entry of the reduced vector, with threads within a warp coalescing global memory accesses and using warp-level reductions. Fusion performs the max and the LSE reductions in the same loop, reducing the number of memory accesses at the cost of additional exp and branching operations. Further details can be found in Appendix E.

After applying warp-tiling ("WT") and loop fusion ("Fused"), our measured reduction kernel latency outperforms the optimized PyKeOps library [9], as shown in Table 1. At $n = 1024$, the naive kernel does not achieve full GPU occupancy (32768 threads), hence warp-tiling improves performance by $32\times$. However, as the problem size increases, the naive kernel comes closer to achieving full SM occupancy, ultimately removing concurrency benefits from warp-tiling. Even when the GPU is at full occupancy, warp-tiling results in improved memory access patterns and a $1.4\times$ improvement over the naive kernel. The "fused" kernel further improves memory access overhead as n increases, gaining an additional $1.25\times$ speedup.

5 Numerical Experiments

In this section we compare Algorithm 2 with alternative first-order OT/EOT algorithms. Further details, including problem generation and hyperparameter choices, can be found in Appendix E. All experimental/solver code and plotting data is implemented in Julia and is publicly available[1].

[1] https://github.com/mxburns2022/CuLAMP.jl

Table 1: Ablation study of kernel optimizations with comparison to the LogSumExp reduction from the PyKeOps library on an RTX 4090 GPU. Across problem sizes, the optimized LogSumExp kernels outperform the widely-used PyKeOps reduction kernel baseline.

	Wall-clock time (ms)			
Kernel	$n = 1024$	$n = 4096$	$n = 16384$	$n = 65536$
Baseline	3.77 ± 0.092	14.8 ± 0.24	58.9 ± 0.86	474.6 ± 7.04
WT	$\mathbf{0.1 \pm 0.008}$	1.37 ± 0.044	21.3 ± 0.34	337.7 ± 5.36
Fused+WT	0.12 ± 0.0078	$\mathbf{1.30 \pm 0.0426}$	$\mathbf{17.13 \pm 0.26}$	$\mathbf{252.5 \pm 2.90}$
PyKeOps	1.79 ± 0.016	6.96 ± 0.015	27.8 ± 0.026	332 ± 0.16

Competitor Solvers: We compare to dense-reduction first-order baselines, matching the computational model targeted by LAMP. Baseline solvers include Sinkhorn algorithm [1], accelerated primal-dual adaptive mirror descent (APDAMD) [26], accelerated Sinkhorn [26], the Bregman hybrid primal-dual (HPD) algorithm [8], annealed Sinkhorn using warm starts as described in [21], and Dual Extrapolation as described in Algorithm 3 of [19]. Other methods were tested, such as APDAGD [13] and Greenkhorn [1], however we focused on the top performing solvers for the purposes of this work.[2] As LAMP and PDMP are equivalent up to numerical error, we do not compare them directly, as the plots would be uninformative. Solvers are set to terminate upon reaching a primal-dual gap $\leq 10^{-10}$, where the dual functions for each formulation are given in Appendix B.

Benchmarks: For head-to-head testing, we use instances from the DOTmark set [38] with $\|\cdot\|_p^p$ ground costs (except $p = \infty$, where ground costs are $\|\cdot\|_\infty$).

Fig. 1 compares fixed-η EOT solver performance in $n = m = 1024$ DOTmark problems with ℓ_2^2 ground costs. The y-axis on each plot shows the EOT primal gap, hence measuring convergence in the entropy-regularized problem alone. For $\eta \geq 10^{-4}$, Sinkhorn and Accelerated Sinkhorn significantly outperform the other methods tested, though LAMP outperforms several primal-dual methods (HPD and APDAMD) in this regime. However, LAMP's advantage is clear for the weakly regularized case ($\eta = 10^{-6}$), where LAMP converges extremely rapidly while the other solvers appear to converge sublinearly.

We now focus on comparing unregularized LAMP ($\eta = 0$) to Sinkhorn-based methods ($\eta > 0$) as the primary $\mathcal{O}(n + m)$ space baseline. Furthermore, we add temperature-annealed Sinkhorn with warm starts as described in [21] as an annealing baseline, with $\eta_t = \max\{\eta_i q^t, \eta_f\}$ for $q \in (0, 1]$. All solvers use on-the-fly, kernel-based distances with the warp-tiling+fusion optimizations discussed in Subsection 4.2. For our comparisons, we utilize the combined objective gap + infeasibility

$$\langle C, X^t - X^* \rangle + \|C\|_\infty (\|\mathbf{c}(X^t) - c\|_1 + \|\mathbf{r}(X^t) - r\|_1), \qquad (20)$$

which acts as an upper bound on the cost of the rounded iterate $\text{Round}(X^t, r, c)$ when X^t is either row or column feasible (true for both Sinkhorn and LAMP). Exact OT costs are computed using `emd2` from PythonOT [14]. As illustrated by Proposition 3.4(a), LAMP convergence is dependent on $\|C\|_\infty$, which in turn depends on the underlying metric. Fig. 2 compares each kernel OT solver on $n = m = 4096$ DOTmark problems with varying ground cost. For ℓ_∞ and ℓ_1 ground costs, LAMP converges extremely rapidly, outperforming both the Sinkhorn baselines. Since $\|C\|_\infty$ scales $\mathcal{O}(\sqrt{n})$ for both costs, the initial bias term in (17) is relatively benign.

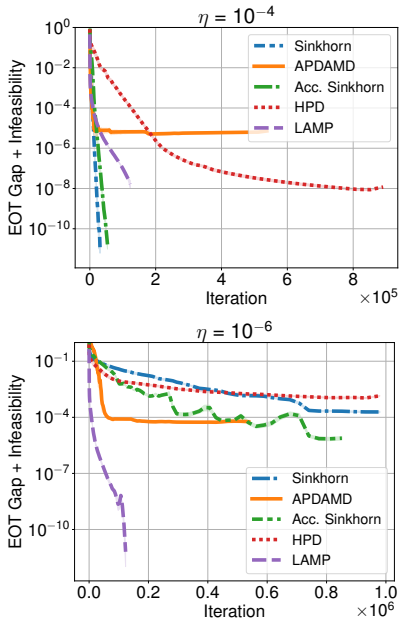


Figure 1: Comparison of various OT solvers on $n = 1024$ DOTmark problems with Euclidean ground costs. LAMP is particularly effective in the weakly-regularized $\eta = 10^{-6}$ regime.

[2] APDAGD showed significant instabilities in the small-η regime as observed in [26], while Greenkhorn is unable to efficiently utilize a GPU due to its update scheme.

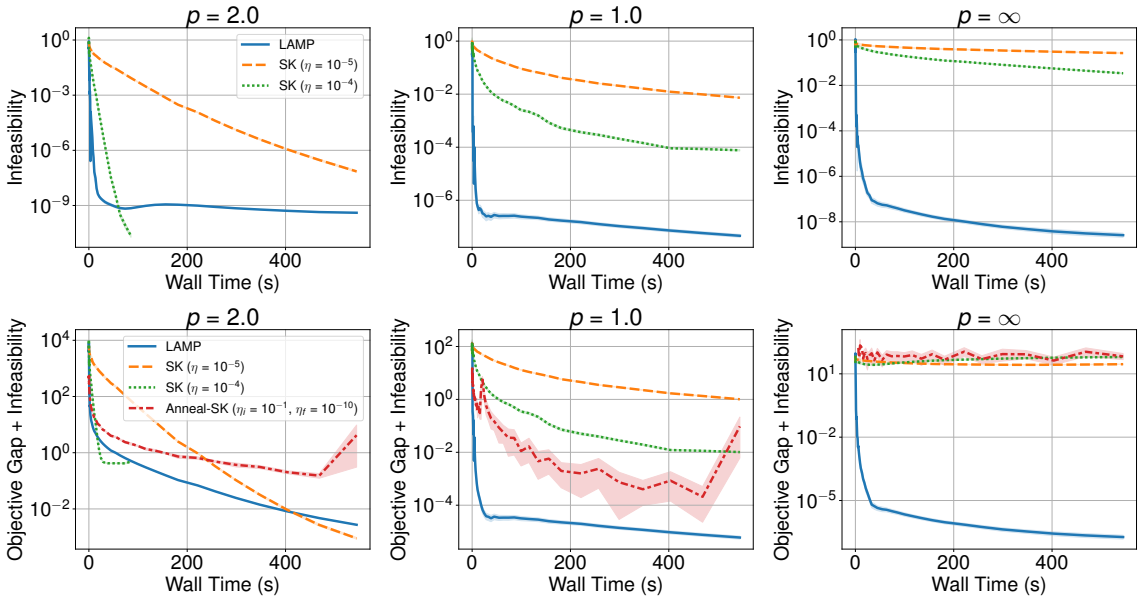


Figure 2: Solution trajectories on $n = m = 4096$ DOTmark instances with varying ground metrics comparing [top] infeasibility and [bottom] objective gap + infeasibility as described in (20). LAMP's performance is metric dependent, with particularly high performance for costs with $\|C\|_\infty = o(n)$. We omit Anneal-SK (using $(\eta_i, \eta_f, q) = (10^{-1}, 10^{-10}, 0.8)$) from the infeasibility plot, as at each outer iteration the iterate is feasible (except for the last timed-out iteration).

In contrast, LAMP exhibits slower convergence for ℓ_2^2, where $\|C\|_\infty = \mathcal{O}(n)$. LAMP is competitive for much of the ℓ_2^2 trajectory, though Sinkhorn eventually overtakes it with well-chosen regularization. This, however, highlights another benefit of LAMP. In contrast to Sinkhorn, where the value of η may require problem-dependent tuning, unregularized LAMP ($\eta = 0$) required no problem-specific tuning in our tests to obtain competitive performance.

LAMP further outperforms Sinkhorn in dataset similarity computation, shown in Fig. 3, where the dataset consists of cell omics data from [27] and preprocessed by [18]. We define the costs C_{ij} using four different similarity kernels: ℓ_1 and ℓ_2^2 costs, cosine similarities, and Pearson correlations. In each case, $\|C\|_\infty \approx 1$, leading to LAMP converging significantly faster than the Sinkhorn solvers. Appendix E provides additional description on the problem setup and specific runtime breakdowns.

We therefore broadly conclude that LAMP is particularly effective in high-accuracy settings and when costs have low-to-moderate values of $\|C\|_\infty$, while Sinkhorn may be preferable in low-accuracy/strongly-regularized problems or for large $\|C\|_\infty$ values after sufficient tuning.

Finally, we provide a proof-of-concept demonstration of LAMP on large-scale OT problems. Using the kernel-based LAMP code, we computed 512 x 512 color transfer maps with a 4 hour time limit and ℓ_2^2 ground costs, shown in Fig. 4. Explicit primal-dual methods (such as PDMP or accelerated mirror descent methods [13, 26]) would require ≈ 512 GB (the equivalent of eleven L40s GPUs), while LAMP is able to run on a single L40s GPU using approximately 38 MB.

6 Discussion

An independent line of research from [3] proved a result similar to Proposition 3.2, achieving $\mathcal{O}(n+m)$ space complexity using an auxiliary dual sequence similar to ν_t as well as a scalar λ_t. One of the main subroutines in [3] is dual extrapolation [19], which includes the update

$$X^{t+1} = \operatorname*{argmin}_{X \in \Delta^{n \times m}} \{\langle v^t, X \rangle + \tau \mathrm{D}_{\mathrm{KL}}(X \| X^t)\}, \tag{21}$$

where $v^t \in \mathbb{R}^{n \times m}$ can be computed in $\mathcal{O}(nm)$ time using additional $\mathcal{O}(n+m)$ dual variables.

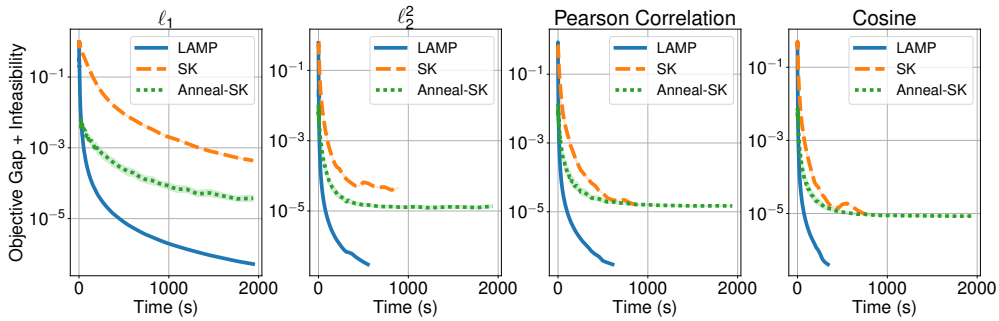


Figure 3: Comparison between Sinkhorn with $\eta = 10^{-4}$ (SK), Annealed Sinkhorn (Anneal-SK) with $(\eta_i, \eta_f, q) = (10^{-2}, 10^{-4}, 0.95)$, and LAMP ($\eta = 0$) in cell similarity tasks using the cell omics datasets from [18]. Costs are computed on the fly by comparing cell features using various similarity metrics. For these plots, we use 20 cells and 5000 features ($n = m = 5000$) and average over 10 problem instances. In each case, LAMP significantly outperforms the Sinkhorn-based methods.

Figure 4: Color transfer on 512 x 512 images with three color channels. The original images [top] were generated by Google Gemini. The color transfers [bottom] were computed with a 4-hour timeout running on an NVIDIA L40s.

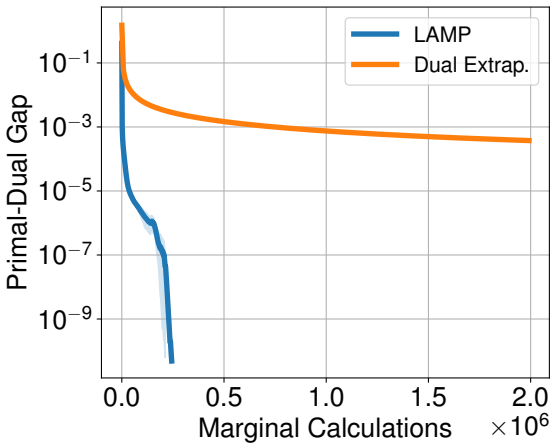


Figure 5: Comparison between the dual extrapolation method proposed in [19, Algorithm 3] and LAMP. The x-axis counts the number of marginal computations performed by each algorithm, which each requires $\mathcal{O}(nm)$ time.

The multiplicative form of (6) is the key mechanism used in the proof of Lemma 3.1. Since (21) also leads to a multiplicative update, both LAMP and dual extrapolation are able to implicitly store primal information using extra dual sequences. However, the dual extrapolation method underlying the proposed algorithm in [3] has two shortcomings. First, its convergence guarantees require an average primal iterate. To maintain convergence guarantees with only dual iterates, [3] propose a recovery procedure dependent on pointer-based data structures, making the subroutine difficult to efficiently map to GPUs. Second, the dual extrapolation method from [19] is generally slow in practice. Fig. 5 compares dual extrapolation to LAMP on $n = 1024$ DOTmark problems with ℓ_2^2 costs, showing that LAMP significantly outperforms the competing method. Since the dual extrapolation subroutine dominates the runtime complexity in [3] (see Theorem 4.1 therein), we reasonably conjecture that the conclusions of Fig. 5 hold for the linear-space framework (see Theorem 5.2 of [3]).

Theoretical Gaps As noted in the main text, proving non-ergodic optimality guarantees for $\eta = 0$ and $\tau_1, \tau_2 = 1/(2\|C\|_\infty)$ is a necessary next step to justify the "empirically performant" parameter choices.

Proposition 3.4 provides practically useful bounds for the performant regime, however the lack of a full convergence proof is our primary theoretical limitation.

Extension to Second-Order Methods Recent second-order OT solvers utilizing Krylov methods [20, 21, 42] and/or sparse Newton iterations [42, 33] have shown significant speedups over first-order baselines. These methods utilize the traditional dual of (2) and are often built on a temperature-annealing framework [21]. In this work, we have shown that mirror prox methods built on the alternative problem (4) (the dual can be found in Appendix B) can substantially outperform comparable first-order methods built on the traditional (E)OT formulation. Our claim is limited: LAMP is simple to implement, linear space, admits computable last/best-iterate certificates (Proposition 3.4), and is highly competitive in its algorithmic class (highly parallel first-order methods), not that it is a universally dominant algorithm. Our results therefore suggest the development of second-order methods targeting (4) as a promising direction for our future work.

References

[1] J. Altschuler, J. Niles-Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

[3] S. Assadi, A. Jambulapati, Y. Jin, A. Sidford, and K. Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 627–669. Society for Industrial and Applied Mathematics, 2022.

[4] A. Beck. *First-Order Methods in Optimization*. SIAM-Society for Industrial and Applied Mathematics, Philadelphia : Philadelphia, 2017.

[5] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[6] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] S. Cen, Y. Wei, and Y. Chi. Fast policy extragradient methods for competitive games with entropy regularization. *Journal of Machine Learning Research*, 25(4):1–48, 2024.

[8] A. Chambolle and J. P. Contreras. Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein barycenter problems. *SIAM Journal on Mathematics of Data Science*, 4(4):1369–1395, 2022.

[9] B. Charlier, J. Feydy, J. A. Glaunes, F.-D. Collin, and G. Durif. Kernel operations on the GPU, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021.

[10] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 2006.

[11] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[12] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[13] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1367–1376. PMLR, 2018.

[14] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. H. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[15] A. Galichon. *Optimal Transport Methods in Economics*. Princeton University Press, Princeton, 2016.

[16] P. Ghosal and M. Nutz. On the convergence rate of Sinkhorn's algorithm. *Mathematics of Operations Research*, 0(0), 2025.

[17] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov. On a combination of alternating minimization and Nesterov's momentum. In *Proceedings of the 38th International Conference on Machine Learning*, pages 3886–3898. PMLR, 2021.

[18] G.-J. Huizing, G. Peyré, and L. Cantini. Optimal transport improves cell–cell similarity inference in single-cell omics data. *Bioinformatics*, 38(8):2169–2177, 2022.

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

[20] M. Kemertas, A.-m. Farahmand, and A. D. Jepson. A truncated Newton method for optimal transport. In *The Thirteenth International Conference on Learning Representations*, 2024.

[21] M. Kemertas, A. D. Jepson, and A. massoud Farahmand. Efficient and accurate optimal transport with mirror descent and conjugate gradients. *Transactions on Machine Learning Research*, 2025.

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

[23] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{\mathcal{O}}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433, Philadelphia, PA, USA, 2014. IEEE.

[24] B. Levy, R. Mohayaee, and S. von Hausegger. A fast semidiscrete optimal transport algorithm for a unique reconstruction of the early universe. *Monthly Notices of the Royal Astronomical Society*, 506(1):1165–1185, 2021.

[25] G. Li, Y. Chen, Y. Huang, Y. Chi, H. V. Poor, and Y. Chen. Fast computation of optimal transport via entropy-regularized extragradient methods. *SIAM Journal on Optimization*, 35(2):1330–1363, 2025.

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

[27] L. Liu, C. Liu, A. Quintero, L. Wu, Y. Yuan, M. Wang, M. Cheng, L. Leng, L. Xu, G. Dong, et al. Deconvolution of single-cell multi-omics layers reveals regulatory heterogeneity. *Nature communications*, 10(1):470, 2019.

[28] H. Lu and J. Yang. PDOT: A practical primal-dual algorithm and a GPU-based solver for optimal transport. `https://arxiv.org/abs/2407.19689`, 2024.

[29] Y. Luo, Y. Xie, and X. Huo. Improved rate of first order algorithms for entropic optimal transport. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 2723–2750. PMLR, 2023.

[30] V. V. Mai, J. Lindbäck, and M. Johansson. A fast and accurate splitting method for optimal transport: analysis and implementation. In *International Conference on Learning Representations*. Curran Associates, Inc., 2022.

[31] A. Nemirovski. 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*, 2006.

[32] M. Nutz. Introduction to entropic optimal transport. *Lecture notes, Columbia University*, 2022.

[33] J. Pan, J. Li, and M. Yan. Inexact Bregman sparse Newton method for efficient optimal transport. *arXiv preprint arXiv:2603.07156*, 2026.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

[36] F. Pitié, A. C. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007.

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

[38] J. Schrieber, D. Schuhmacher, and C. Gottschlich. DOTmark – a benchmark for discrete optimal transport. *IEEE Access*, 5:271–282, 2017.

[39] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[40] M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

[41] C. Villani. *Topics in Optimal Transportation*. American Mathematical Society, Providence, RI, uk edition, 2003.

[42] D. Wu, L. Liang, and H. Yang. PINS: Proximal iterations with sparse Newton and Sinkhorn for optimal transport. *arXiv preprint arXiv:2502.03749*, 2025.

[43] G. Yao and A. Dani. Visual tracking using sparse coding and earth mover's distance. *Frontiers in Robotics and AI*, 5, 2018.

A Literature Review

Here we provide a brief review of relevant first-order methods for OT/EOT. Since our work focuses on first-order methods leveraging parallel operations, we omit discussion of combinatorial OT algorithms, such as [22].

The majority of first-order methods research targets EOT rather than directly solving OT. In the EOT family, the Sinkhorn-Knopp (a.k.a. Sinkhorn) algorithm [39] serves as the standard method: conceptually simple, highly parallel, and rapidly convergent to low-accuracy solutions. Despite its advantages, the iteration complexity of Sinkhorn scales $\widetilde{\mathcal{O}}(\varepsilon^{-2})$ [13], which results in slow convergence to high-accuracy solutions. In response, a number of alternative methods relying on classical ideas from first-order methods in optimization have been proposed, including accelerated primal-dual methods [13, 17, 26, 29], saddle-point methods [8], and variations of Sinkhorn [1, 26]. Additionally, several methods have been proposed to improve the empirical performance of EOT algorithms, including temperature annealing [37, 20, 21], and second-order subroutines, such as truncated Newton [20, 21], and sparse Newton [42, 33] iterations.

Other methods bypass EOT to target OT directly. Douglas-Rachford splitting [30], dual extrapolation [19, 3], and PDHG [28] are among the "direct" OT methods that have been proposed.

Table 2: Non-exhaustive summary of related works on first-order methods for OT. "Complexity" reflects the computational complexity required to find an ε-solution to the primal OT problem. Alternative references are provided for the complexity bounds if they differ from the results of the original work. "Space complexity" reflects the complexity required to store the algorithm iterates, excluding the cost to store the cost matrix C, which can often be computed on-the-fly for kernel-based costs. For PDHG, δ is the data precision (see Definition 2 in [28]). Acronyms are as follows: "APDA(G,M)D" are Adaptive Primal-Dual Accelerated (Gradient, Mirror) Descent, AAM is Accelerated Alternating Minimization, DROT is Douglas-Rachford OT, HPD is Hybrid Primal-Dual, PDASMD is Primal-Dual Accelerated Stochastic Mirror Descent, and PDHG is Primal-Dual Hybrid Gradient.

Algorithm(s)	Complexity	Space Complexity[3]
Sinkhorn [12]	$\widetilde{\mathcal{O}}(nm\varepsilon^{-2})$ [13]	$\boldsymbol{\mathcal{O}(n+m)}$
Greenkhorn [1]	$\widetilde{\mathcal{O}}(nm\varepsilon^{-2})$ [26]	$\boldsymbol{\mathcal{O}(n+m)}$
APDAGD [13]	$\widetilde{\mathcal{O}}(nm(n+m)^{1/2}\varepsilon^{-1})$ [26]	$\mathcal{O}(nm)$
Dual Extrapolation [19]	$\boldsymbol{\widetilde{\mathcal{O}}(nm\varepsilon^{-1})}^4$	$\boldsymbol{\mathcal{O}(n+m)}$[3]
AAM [17]	$\widetilde{\mathcal{O}}(nm(n+m)^{1/2}\varepsilon^{-1})$	$\mathcal{O}(nm)$
DROT [30]	$\boldsymbol{\mathcal{O}(nm\varepsilon^{-1})}$	$\mathcal{O}(nm)$
APDAMD [26]	$\widetilde{\mathcal{O}}(nm(n+m)^{1/2}\varepsilon^{-1})$	$\mathcal{O}(nm)$
Acc. Sinkhorn [26]	$\widetilde{\mathcal{O}}(nm(n+m)^{1/3}\varepsilon^{-4/3})$	$\boldsymbol{\mathcal{O}(n+m)}$
HPD [8]	$\widetilde{\mathcal{O}}(nm(n+m)^{1/2}\varepsilon^{-1})$	$\mathcal{O}(nm)$
PDASMD [29]	$\boldsymbol{\widetilde{\mathcal{O}}(nm\varepsilon^{-1})}$	$\mathcal{O}(nm)$
PDHG [28]	$\widetilde{\mathcal{O}}(nm(n+m)^{7/2}\delta + nm(n+m)^{1/2})$	$\mathcal{O}(nm)$
PDMP [25]	$\boldsymbol{\widetilde{\mathcal{O}}(nm\varepsilon^{-1})}$	$\mathcal{O}(nm)$
LAMP (**This Work**)	$\boldsymbol{\widetilde{\mathcal{O}}(nm\varepsilon^{-1})}$	$\boldsymbol{\mathcal{O}(n+m)}$

Table 2 catalogues a sample of the proposed first-order OT methods according to their computational complexity (relative to problem size n, m and accuracy ε) and their storage requirements. While non-exhaustive, the table captures the essence of current first-order OT solvers. Dual-only methods such as Sinkhorn, Greenkhorn, and Accelerated Sinkhorn have favorable scaling in the problem dimension n, m and attain linear-space complexity, however they have worse dependence on the accuracy ε. The dual-only Dual Extrapolation implementation of [3] manages to achieve state-of-the-art theoretical guarantees in linear space, however the underlying Dual Extrapolation method is empirically slow and the linear space implementation is difficult to implement in HPC environments, as discussed in Section 6.

B Dual Problems and Equivalence

In this section we discuss the dual functions corresponding to the OT (1), EOT (2), and saddle-point (4) problems. These functions then define the dual problems, which are useful theoretically (as we show in the next section), as well as providing a computable primal-dual gap for numerical implementation.

The dual problems for OT/EOT are obtained by first dualizing the constraints by standard Lagrangian machinery

$$\max_{\varphi\in\mathbb{R}^n,\psi\in\mathbb{R}^m}\min_{X\in\Delta^{n\times m}}\langle C,X\rangle+\langle\mathbf{r}(X)-r,\varphi\rangle+\langle\mathbf{c}(X)-c,\psi\rangle-\eta H(X), \tag{22}$$

where φ and ψ are the dual multipliers corresponding to the row and column constraints, respectively. If $\eta=0$, then minimizing with respect to X over the simplex gives the nonsmooth problem

$$\max_{\varphi\in\mathbb{R}^n,\psi\in\mathbb{R}^m}\left\{d(\varphi,\psi):=\min_{ij}\{C_{ij}+\varphi_i+\psi_j\}-\langle r,\varphi\rangle-\langle c,\psi\rangle\right\}. \tag{23}$$

If $\eta>0$, then minimizing with respect to X over the simplex gives

$$\max_{\varphi\in\mathbb{R}^n,\psi\in\mathbb{R}^m}\left\{d^\eta(\varphi,\psi):=\operatorname{smin}^\eta_{ij}\{C_{ij}+\varphi_i+\psi_j\}-\langle r,\varphi\rangle-\langle c,\psi\rangle\right\}, \tag{24}$$

where

$$\operatorname{smin}^\eta_{ij}\{C_{ij}+\varphi_i+\psi_j\}:=-\eta\log\left[\sum_{ij}\exp(-\eta^{-1}(C_{ij}+\varphi_i+\psi_j))\right]$$

is the "softmin" function and is η^{-1}-smooth with respect to the ℓ_2 and ℓ_∞ norms [4, Example 5.15].

Similarly, optimizing the primal variables p in (4) over the set of row-stochastic matrices gives the dual problem:

$$\max_{\theta\in[-1,1]^m}\left\{D(\theta)=\sum_{i=1}^{n} r_i \min_j\{C_{ij}+2\|C\|_\infty\theta_j\}-2\|C\|_\infty\langle c,\theta\rangle\}\right\},\quad \text{if } \eta=0, \tag{25}$$

and

$$\max_{\theta\in[-1,1]^m}\left\{D^\eta(\theta):=\sum_{i=1}^{n} r_i \operatorname{smin}^\eta_j\{C_{ij}+2\|C\|_\infty\theta_j\}-2\|C\|_\infty\langle c,\theta\rangle\}\right\},\quad \text{if } \eta>0. \tag{26}$$

C Deferred Proofs

First, we give a statement of the "rounding" from [1] in Algorithm 3. "Round" takes as input a matrix $X\in\mathbb{R}_+^{n\times m}$ and marginals $r\in\Delta^n$, $c\in\Delta^m$ and returns a matrix $\tilde{X}$ satisfying $\mathbf{r}(\tilde{X})=r$, $\mathbf{c}(\tilde{X})=c$.

Algorithm 3 Round [1, Algorithm 2]

Require: $X\in\mathbb{R}_+^{n\times m}$, $r\in\Delta^n$, $c\in\Delta^m$.
Step 1) Set $X'=\mathcal{D}_x X$ where $x_i=\min\left\{\frac{r_i}{\mathbf{r}(X)_i},1\right\}$.
Step 2) Set $X''=X'\mathcal{D}_y$ where $y_j=\min\left\{\frac{c_j}{\mathbf{c}(X')_j},1\right\}$.
Step 3) Compute $\delta_r=r-\mathbf{r}(X'')$, $\quad\delta_c=c-\mathbf{c}(X'')$.
Step 4) Set $\tilde{X}=X''+\|\delta_r\|_1^{-1}\delta_r\delta_c^\top$.
return $\tilde{X}$.

The following lemma, which is standard in OT literature, provides a useful property of Algorithm 3.

Lemma C.1 ([1, Lemma 7])**.** *If $r\in\Delta^n$, $c\in\Delta^m$, and $X\in\mathbb{R}_+^{n\times m}$, then Algorithm 3 takes $\mathcal{O}(nm)$ time to output a matrix $\tilde{X}\in\Pi(r,c)$ satisfying*

$$\|X-\tilde{X}\|_1\le 2\big[\|\mathbf{r}(X)-r\|_1+\|\mathbf{c}(X)-c\|_1\big]. \tag{27}$$

Next, we state several technical lemmas which will be used in the proof of Lemma C.5.

The following lemma states a general property of the dual problem (24), which allows for constant-offset transformations in each dual variable.

Lemma C.2. *Let $\varphi\in\mathbb{R}^n$, $\psi\in\mathbb{R}^m$ be two dual potentials. Then, for any $a\in\mathbb{R}$, $b\in\mathbb{R}$,*

$$d^\eta(\varphi+a\mathbf{1}_n,\psi+b\mathbf{1}_m)=d^\eta(\varphi,\psi),$$

where d^η is the EOT dual defined in (24)*.*

Proof. We observe that the LSE function is directionally affine along the $\mathbf{1}_n$-axis, i.e.,

$$\mathrm{LSE}(x+\alpha\mathbf{1}_n)=\log\left[\sum_{i=1}^{n}\exp\left(x_i+\alpha\right)\right]=\mathrm{LSE}(x)+\alpha,$$

for any $x\in\mathbb{R}^n$ and $\alpha\in\mathbb{R}$. Then, we have

$$\begin{aligned}d^\eta(\varphi+a\mathbf{1}_n,\psi+b\mathbf{1}_m)=&-\langle\varphi,r\rangle-a-\langle\psi,c\rangle-b\\&-\eta\mathrm{LSE}\left(-\eta^{-1}(C+\varphi\mathbf{1}_m^\top+\mathbf{1}_m\psi^\top)\right)+a+b=d^\eta(\varphi,\psi),\end{aligned}$$

which proves the claim. □

The next result is a technical lemma regarding exact penalty formulations.

Lemma C.3. *Consider the linearly constrained problem*

$$\min_{x\in Q} f(x) \quad s.t. \quad Ax = b, \tag{28}$$

where $f : \mathbb{R}^n \to (-\infty, \infty]$ is a closed proper and convex function, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and Q is a closed and convex set. Further assume that there exists a point $x \in \operatorname{relint}(\operatorname{dom} f) \cap \operatorname{relint}(Q)$ satisfying $Ax = b$, where $\operatorname{relint}(\operatorname{dom} f)$ is the relative interior of the domain of f (i.e., Slater's condition is satisfied).

Define the exact ℓ_1 penalization as

$$\min_{x\in Q}\{\phi_\mu(x) = f(x) + \mu\|Ax - b\|_1\}. \tag{29}$$

Suppose x^* is a solution of (28) with Lagrange multiplier $\lambda^* \in \mathbb{R}^m$. Then, for any $\mu \geq \|\lambda^*\|_\infty$, we have $x^* \in \operatorname{Argmin}_{x\in Q} \phi_\mu(x)$.

Proof. Consider the saddle-point form of (28) formed from the Lagrangian

$$\min_{x\in Q}\max_{\lambda\in\mathbb{R}^m}\{\mathcal{L}(x,\lambda) = f(x) + \langle\lambda, Ax - b\rangle\}. \tag{30}$$

By standard arguments, Slater's condition implies that the pair (x^*, λ^*) is a saddle-point of (30). By Hölder's inequality and the saddle-point property, for any $x \in Q$, we have

$$\begin{aligned} f(x^*) &= \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \\ &= f(x) + \langle\lambda^*, Ax - b\rangle \\ &\leq f(x) + \|\lambda^*\|_\infty\|Ax - b\|_1 \leq \phi_\mu(x). \end{aligned}$$

Since $f(x^*) = \phi_\mu(x^*)$, we have $x^* \in \operatorname{Argmin}_{x\in Q} \phi_\mu(x)$ as claimed. □

Therefore, if we can guarantee that $\|\psi^*\|_\infty \leq 2\|C\|_\infty$ for some optimal multiplier $\psi^* \in \mathbb{R}^m$, we can guarantee that (2) and (3) have the same minimizer (unique minimizer, for $\eta > 0$). Fortunately, we have the following bound from [26] characterizing the infinity norm of the dual multipliers. The proof follows [26, Lemma 3], however we show the argument here since our statement differs slightly from theirs, as they use a common constant rather than providing separate bounds for $\|\varphi^*\|_\infty$ and $\|\psi^*\|_\infty$.

Lemma C.4. *For the dual EOT problem defined in (24), there exists an optimal solution (φ^*, ψ^*) such that*

$$\begin{aligned} \|\varphi^*\|_\infty &\leq \|C\|_\infty - \eta \log \min_{1\leq i\leq n}\{r_i\}, \\ \|\psi^*\|_\infty &\leq \|C\|_\infty - \eta \log \min_{1\leq j\leq m}\{c_j\}. \end{aligned}$$

Proof. We first claim that there exist dual variables φ^* and ψ^* which are optimal solutions to (24) satisfying

$$\begin{aligned} \min_i \varphi_i^* \leq 0 \leq \max_i \varphi_i^*, \\ \min_j \psi_j^* \leq 0 \leq \max_j \psi_j^*. \end{aligned} \tag{31}$$

Let $\hat{\varphi}^*$, $\hat{\psi}^*$ be a pair of optimal dual potentials. Define

$$\Delta_\varphi = \frac{1}{2}(\max_i \hat{\varphi}_i^* + \min_i \hat{\varphi}_i^*), \quad \Delta_\psi = \frac{1}{2}(\max_j \hat{\psi}_j^* + \min_j \hat{\psi}_j^*),$$

and set $\varphi^* = \hat{\varphi}^* - \Delta_\varphi \mathbf{1}_n$ and $\psi^* = \hat{\psi}^* - \Delta_\psi \mathbf{1}_m$. By Lemma C.2, φ^* and ψ^* are optimal dual potentials. Furthermore, by construction, they satisfy (31). Now, taking the gradient of the dual objective in (24), we have

$$0 = -r_i + \exp[-\eta^{-1}\varphi_i^*]\sum_{j=1}^m \frac{\exp[-\eta^{-1}(C_{ij} + \psi_j^*)]}{Z}, \tag{32}$$

where

$$Z = \sum_{k,\ell} \exp[-\eta^{-1}(C_{k\ell} + \varphi_k^* + \psi_\ell^*)].$$

Rearranging (32), we obtain

$$\varphi_i^* = -\eta \log r_i + \eta \log \left(\sum_{j=1}^m \exp[-\eta^{-1}(C_{ij} + \psi_j^*)] \right) - \eta \log Z.$$

Since each $C_{ij} \in [0, \|C\|_\infty]$ we have for all i, j

$$\exp[-\eta^{-1}(C_{ij} + \psi_j^*)] \geq \exp[-\eta^{-1}(\psi_j^*)] \exp[-\eta^{-1}\|C\|_\infty].$$

We also have $-\log r_i \geq 0$, hence

$$\varphi_i^* \geq \eta \log \left(\sum_{j=1}^m \exp[-\eta^{-1}\psi_j^*] \right) - \|C\|_\infty - \eta \log Z.$$

Again using $C_{ij} \geq 0$, we have

$$\varphi_i^* \leq -\eta \log r_i + \eta \log \left(\sum_{j=1}^m \exp[-\eta^{-1}\psi_j^*] \right) - \eta \log Z.$$

Then we have

$$\max_i \varphi_i^* \leq -\eta \min_i \log r_i + \eta \log \left(\sum_{j=1}^m \exp[-\eta^{-1}\psi_j^*] \right) - \eta \log Z$$

$$\min_i \varphi_i^* \geq \eta \log \left(\sum_{j=1}^m \exp[-\eta^{-1}\psi_j^*] \right) - \|C\|_\infty - \eta \log Z.$$

So,

$$\max_i \varphi_i^* - \min_i \varphi_i^* \leq \|C\|_\infty - \eta \log \min_i r_i.$$

Since $\min_i \varphi_i^* \leq 0$ and $\max_i \varphi_i^* \geq 0$, we have that

$$0 \geq \min_i \varphi_i^* \geq -(\|C\|_\infty - \eta \log \min_i r_i),$$
$$0 \leq \max_i \varphi_i^* \leq \|C\|_\infty - \eta \log \min_i r_i,$$

which implies that

$$\|\varphi^*\|_\infty \leq \|C\|_\infty - \eta \log \min_i r_i$$

as claimed. A similar argument holds for the dual variable ψ^* with the c marginal in place of the r marginal. □

A similar result can be shown for unregularized OT with bounded costs, see [41, Remark 1.13].

We now state the full equivalence result extending [19, Lemma 2.3] referenced in Subsection 2.2.

Lemma C.5. *Given $\varepsilon > 0$ and $\eta \geq 0$, let $r \in \Delta^n$, $c \in \Delta^m$, and $C \in \mathbb{R}_+^{n \times m}$ be the input for the (E)OT problem, and assume that r and c have no zero entries. Then, we have:*

a) if $\eta = 0$ and p is an ε-solution to (3), then we can map p to an ε-solution of (1) in $\mathcal{O}(nm)$ arithmetic operations;

b) if $\eta \in (0, \|C\|_\infty / \log(\max_j c_j^{-1})]$*, then the unique minimizer* p^* *of* (3) *is equivalent to the unique minimizer* X^* *of* (2) *under the mapping* $X^* = \mathcal{D}_r p^*$*.*

Proof. a) The claim directly follows from Lemma 2.3 of [19], though we reproduce the proof here for completeness. Let $P(\mathcal{D}_r p)$ be the objective function of (3) with $\eta = 0$. First, we show that there exists an optimizer to (3) which is column feasible. Let $p^* \in \{\Delta^m\}^n$ be an arbitrary optimizer to (3) and denote $\mathcal{D}_r \tilde{p}^* = \text{Round}(\mathcal{D}_r p^*, r, c)$. By Lemma C.1, we have $\mathcal{D}_r \tilde{p}^* \in \Pi(r, c)$ and the bound

$$\|\mathcal{D}_r \tilde{p}^* - \mathcal{D}_r p^*\|_1 \leq 2\|\mathbf{c}_r(p^*) - c\|_1. \tag{33}$$

Since $\mathcal{D}_r \tilde{p}^*$ is column feasible (hence $\|\mathbf{c}_r(\tilde{p}^*) - c\|_1 = 0$), we have

$$P(\mathcal{D}_r \tilde{p}^*) - P(\mathcal{D}_r p^*) = \langle C, \mathcal{D}_r(\tilde{p}^* - p^*)\rangle - 2\|C\|_\infty \|\mathbf{c}_r(p^*) - c\|_1.$$

Applying Hölder's inequality, we have

$$\begin{aligned} P(\mathcal{D}_r \tilde{p}^*) - P(\mathcal{D}_r p^*) &\leq \|C\|_\infty \|\mathcal{D}_r \tilde{p}^* - \mathcal{D}_r p^*\|_1 - 2\|C\|_\infty \|\mathbf{c}_r(p^*) - c\|_1 \\ &\overset{(33)}{\leq} 2\|C\|_\infty \|\mathbf{c}_r(p^*) - c\|_1 - 2\|C\|_\infty \|\mathbf{c}_r(p^*) - c\|_1 = 0. \end{aligned}$$

Then $P(\mathcal{D}_r \tilde{p}^*) - P(\mathcal{D}_r p^*) \leq 0$, which implies that $P(\mathcal{D}_r \tilde{p}^*)$ is also optimal. Since $\Pi(r, c) \subseteq \{\mathcal{D}_r p : p \in \{\Delta^m\}^n\}$, we have

$$P(\mathcal{D}_r \tilde{p}^*) = \min_{p \in \{\Delta^m\}^n} P(\mathcal{D}_r p) \leq \min_{X \in \Pi(r,c)} \langle C, X\rangle .$$

Since $\mathcal{D}_r \tilde{p}^* := \tilde{X}^* \in \Pi(r, c)$, we clearly have $\min_{X \in \Pi(r,c)} \langle C, X\rangle \leq \left\langle C, \tilde{X}^* \right\rangle$, hence $\left\langle C, \tilde{X}^* \right\rangle = \min_{X \in \Pi(r,c)} \langle C, X\rangle$ and $\tilde{X}^*$ is therefore an optimizer to (1).

Now, let $p^* \in \{\Delta^m\}^n$ be an optimizer to (3) satisfying $\mathbf{c}_r(p^*) = c$ (whose existence we have just proved). Let p be an ε-solution to (3) and $\mathcal{D}_r \tilde{p} = \text{Round}(\mathcal{D}_r p, r, c)$. Then

$$\varepsilon \geq P(\mathcal{D}_r p) - P(\mathcal{D}_r p^*) = P(\mathcal{D}_r p) - P(\mathcal{D}_r \tilde{p}) + P(\mathcal{D}_r \tilde{p}) - P(\mathcal{D}_r p^*).$$

We have the lower bound

$$\begin{aligned} P(\mathcal{D}_r p) - P(\mathcal{D}_r \tilde{p}) &= \langle C, \mathcal{D}_r(p - \tilde{p})\rangle + 2\|C\|_\infty \|\mathbf{c}_r(p) - c\|_1 \\ &\geq -\|C\|_\infty \|\mathcal{D}_r(p - \tilde{p})\|_1 + 2\|C\|_\infty \|\mathbf{c}_r(p) - c\|_1 \overset{(27)}{\geq} 0, \end{aligned}$$

where the first inequality follows by Hölder's inequality and the second by Lemma C.1. Then we have

$$P(\mathcal{D}_r \tilde{p}) - P(\mathcal{D}_r p^*) \leq \varepsilon,$$

proving the claim.

b) Define ψ^* as some optimal Lagrange multipliers (i.e., part of an optimal pair (φ^*, ψ^*) for (24)) for the c-marginal constraint. We then rewrite the constrained problem (2) as $\min_{p \in Q} f(p)$ s.t. $Ap = c$, where

$$f(p) = \langle C, \mathcal{D}_r p\rangle - \eta H(\mathcal{D}_r p),$$

A is the linear mapping implementing the column sum $Ap := \mathbf{c}_r(p)$, and $Q = \{\Delta^m\}^n$. Let $p^* \in \text{Argmin}_{p \in Q} f(p)$ s.t. $Ap = c$.

Then, by Lemma C.3 (noting that Slater's condition is satisfied for OT problems, since the product coupling is feasible), choosing the penalty coefficient $\mu \geq \|\psi^*\|_\infty$ implies $p^* \in \text{Argmin}_{p \in Q}\{f(p) + \mu\|Ap - c\|_1\}$. Since r has full support, $-H(\mathcal{D}_r p)$ (and therefore f) is strongly convex on $\{\Delta^m\}^n$ [4, Example 5.27], which implies that p^* is the unique minimizer of both the penalized and constrained problems.

Therefore, if $\|\psi^*\|_\infty \leq 2\|C\|_\infty$, (2) and (3) have the same unique minimizer. Applying the bound from Lemma C.4, we have the condition

$$\|C\|_\infty - \eta \log \min_j\{c_j\} \leq 2\|C\|_\infty$$

or

$$-\eta \log \min_j\{c_j\} \leq \|C\|_\infty.$$

The condition on η follows from rearranging. □

Next, we prove the closed-form solutions for the primal and dual mirror maps introduced in Subsection 3.1.

Proof of Equation (6)**:** Beginning with the primal mirror map, we have

$$\begin{aligned} p^+ =& \mathcal{M}_\tau^\eta(p^0;\theta) \\ \overset{(5)}{=}& \operatorname*{argmin}_{p\in\{\Delta^m\}^n} \left\{ \left\langle \mathcal{D}_r(C+\eta(\log p^0 + \mathbf{1}_n\mathbf{1}_m^\top) + 2\|C\|_\infty\theta), p-p^0\right\rangle + \frac{1}{\tau}\mathrm{D}_{\mathrm{KL}}(\mathcal{D}_r p\|\mathcal{D}_r p^0)\right\}. \end{aligned}$$

Writing the optimality conditions for each p_{ij}^+, we have

$$0 = \tau r_i(C_{ij} + 2\|C\|_\infty\theta_j + \eta) + r_i\eta\tau\log p_{ij}^0 + r_i\log\frac{p_{ij}^+}{p_{ij}^0} + \lambda_i,$$

where the λ_i terms enforce row normalization. Solving for p gives

$$p_{ij}^+ = (p^0)_{ij}^{1-\tau\eta}\exp\left[-\tau(C_{ij} + 2\|C\|_\infty\theta_j) - \log(Z_i)\right],$$

where the log-normalization terms $\log Z_i$ absorb all row constants. We therefore obtain the form in the first part of (6).

We now prove the closed form solution for the dual mirror map. By definition, we have

$$\theta^+ = \mathcal{M}_\tau^\eta(\theta^0;p) \overset{(5)}{=} \operatorname*{argmax}_{\theta\in\mathbb{R}^m}\left\{\left\langle 2\|C\|_\infty(\mathbf{c}_r(p)-c), \theta-\theta^0\right\rangle - \frac{1}{\tau}\,\mathrm{D}_{H_c^\alpha}(\theta\|\theta^0)\right\}.$$

Then for each index $j\in\{1,\dots,m\}$ we have the optimality condition

$$0 = 2\tau\|C\|_\infty(\mathbf{c}_r(p)_j - c_j) - \frac{c_j^\alpha}{2}\log\left(\frac{\theta_j^+ + 1}{1-\theta_j^+}\frac{1-\theta_j^0}{1+\theta_j^0}\right)$$

which gives

$$\frac{\theta_j^+ + 1}{1-\theta_j^+} = \frac{1+\theta_j^0}{1-\theta_j^0}\exp\left[4\tau\|C\|_\infty\frac{\mathbf{c}_r(p)_j - c_j}{c_j^\alpha}\right].$$

After some algebraic manipulation, we obtain

$$\theta_j^+ = \frac{\left(\frac{1+\theta_j^0}{1-\theta_j^0}\right)e^{4\tau\|C\|_\infty(\mathbf{c}_r(p)_j - c_j)/c_j^\alpha} - 1}{\left(\frac{1+\theta_j^0}{1-\theta_j^0}\right)e^{4\tau\|C\|_\infty(\mathbf{c}_r(p)_j - c_j)/c_j^\alpha} + 1} = \tanh\left[\frac{2\tau\|C\|_\infty}{c_j^\alpha}(\mathbf{c}_r(p)_j - c_j) + \frac{1}{2}\log\frac{1+\theta_j^0}{1-\theta_j^0}\right],$$

as claimed, where the second equality follows from $\tanh(x) = (e^{2x}-1)/(e^{2x}+1)$. □

Proof of Lemma 3.1 By our assumption on η and γ, it follows that $\tau\eta' \le 1$, so θ' is a convex combination of θ^a and θ^b, and so $\theta'\in[-1,1]^m$.

It then follows from simple algebra and the definition of η' that $(1-\tau\eta)\gamma^{-1} = (\eta')^{-1} - \tau$. Then, using the definition of the dual-to-primal map, we have

$$\begin{aligned} p^\gamma(\theta^a))_{ij}^{1-\tau\eta} &\overset{(7)}{\propto} \exp[-(1-\tau\eta)\gamma^{-1}(C_{ij} + 2\|C\|_\infty\theta_j^a)] \\ &= \exp\left[-\left(\frac{1}{\eta'} - \tau\right)(C_{ij} + 2\|C\|_\infty\theta_j^a)\right]. \end{aligned} \tag{34}$$

Then, using the primal mirror map $\mathcal{M}_\tau^\gamma(\cdot;\theta)$ in (6), we obtain

$$\begin{aligned} \mathcal{M}_\tau^\eta(p^\gamma(\theta^a);\theta^b)_{ij} &\overset{(6)}{\propto} (p^\gamma(\theta^a))_{ij}^{1-\tau\eta}\exp\left[-\tau(C_{ij} + 2\|C\|_\infty\theta_j^b)\right] \\ &\overset{(34)}{\propto} \exp\left[-\left(\frac{1}{\eta'} - \tau\right)(C_{ij} + 2\|C\|_\infty\theta_j^a) - \tau(C_{ij} + 2\|C\|_\infty\theta_j^b)\right] \\ &= \exp\left[-\frac{1}{\eta'}(C_{ij} + 2\|C\|_\infty\theta_j')\right] \overset{(7)}{\propto} p^{\eta'}(\theta')_{ij}, \end{aligned}$$

where the last identity follows by the definition of θ'. □

Proof of Proposition 3.2 We prove the claim by induction. First, we consider the base case $t = 0$. Since $\eta_0 = \infty$ and $\theta_t = \nu_t = \mathbf{0}_m$, we have

$$p^{\eta_0}(\nu^0)_{ij} \propto \exp[-\eta_0^{-1}(C_{ij} + 2\|C\|_\infty \nu_j)] = 1,$$

therefore, after row-wise normalization, $p^{\eta_0}(\nu^0)_{ij} = 1/m = p^0_{ij}$.

For the inductive step, suppose that the claim holds for iteration $t \geq 0$, i.e., $p^t = p^{\eta_t}(\nu^t)$. Then, using Lemma 3.1 with $(\eta, \gamma, \tau, \theta^a, \theta^b) = (\eta, \eta_t, \tau_1, \nu^t, \theta^t)$, and the choice of η_{t+1} in (12), we have

$$\bar{p}^{t+1} \overset{(9)}{=} \mathcal{M}^{\eta}_{\tau_1}(p^t; \theta^t) \overset{(8),(13)}{=} p^{\eta_{t+1}}(\bar{\nu}^{t+1}).$$

Similarly, using Lemma 3.1 with $(\eta, \gamma, \tau, \theta^a, \theta^b) = (\eta, \eta_t, \tau_1, \nu^t, \bar{\theta}^{t+1})$, and the choice of η_{t+1} in (12) we have

$$p^{t+1} \overset{(10)}{=} \mathcal{M}^{\eta}_{\tau_1}(p^t; \bar{\theta}^{t+1}) \overset{(8),(15)}{=} p^{\eta_{t+1}}(\nu^{t+1}),$$

therefore the claim holds for all $t \geq 0$. □

D Analysis of Algorithm 2

In this section, we analyze Algorithm 2 in the performant parameter regime ($\eta = 0$, $\beta = \log 3$, $\tau_1 = \tau_2 = 1/(2\|C\|_\infty)$). Under this setting, we have $\eta_0 = \infty$ and $\eta_t = 2\|C\|_\infty/t$ for $t \geq 1$.

For convenience, we restate the LAMP algorithm with our empirical parameter choices in Algorithm 4, where we use $\tanh(\log(3)/2) = 1/2$.

Algorithm 4 LAMP (Performant Parameters)

Require: $r \in \Delta^n$, $c \in \Delta^m$, $\alpha > 0$, set $\nu_0 = \theta_0 = \mathbf{0}_m$.

for $t \geq 0$ **do**

Set $\eta_t = 2\|C\|_\infty/t$ (or $\eta_t = \infty$ if $t = 0$) and compute the midpoints

$$\bar{\nu}^{t+1} = \frac{t}{t+1}\nu^t + \frac{1}{t+1}\theta^t \tag{35}$$

$$\bar{\theta}^{t+1} = \operatorname*{argmax}_{\theta \in \mathbb{R}^m} \{\langle \mathbf{c}_r(p^{\eta_t}(\nu^t)) - c, \theta \rangle - \mathrm{D}_{H_c^\alpha}(\theta \| \theta^t)\} \tag{36}$$

Set $\eta_{t+1} = 2\|C\|_\infty/(t+1)$ and compute the main sequences

$$\nu^{t+1} = \frac{t}{t+1}\nu^t + \frac{1}{t+1}\bar{\theta}^{t+1} \tag{37}$$

$$\hat{\theta}^{t+1} = \operatorname*{argmax}_{\theta \in \mathbb{R}^m} \{\langle \mathbf{c}_r(p^{\eta_{t+1}}(\bar{\nu}^{t+1})) - c, \theta \rangle - \mathrm{D}_{H_c^\alpha}(\theta \| \theta^t)\} \tag{38}$$

$$\theta^{t+1} = \mathrm{clip}(\hat{\theta}^{t+1}, -1/2, 1/2). \tag{39}$$

end for

Note that by our initialization $\theta_0 = \mathbf{0}_m$ and the definition of the dual mirror map in (6), we have $\bar{\theta}^{t+1} \in (-1, 1)^m$ and $\hat{\theta}^{t+1} \in (-1, 1)^m$ for all $t \geq 0$. Since $\bar{\nu}^{t+1}$ and ν^{t+1} are convex combinations of the $\{\theta^t\}$ and $\{\bar{\theta}_t\}$ sequences, we then have $\bar{\nu}^{t+1} \in (-1, 1)^m$ and $\nu^{t+1} \in (-1, 1)^m$ for all $t \geq 0$.

With our parameter choices, the dual-to-primal map on each iteration $t \geq 1$ is

$$p^{\eta_t}(\nu)_{ij} \propto \exp\left[-t\left(\frac{C_{ij}}{2\|C\|_\infty} + \nu_j\right)\right] \tag{40}$$

We therefore observe that the choice $\tau_1 = \tau_2 = 1/(2\|C\|_\infty)$ effectively normalizes and halves C in all step computations, hence for simplicity of notation we assume that $\|C\|_\infty = 1$ for intermediate results.

We observe that the gradient of the dual entropy $H_c^\alpha(\nu)$ is

$$\nabla H_c^\alpha(\nu) = \frac{1}{2} c^\alpha \odot \log \frac{1+\nu}{1-\nu}.^5 \tag{41}$$

where $\odot$ is an elementwise Hadamard product. Then, for $\theta \in [-1,1]^m$, $\nu \in (-1,1)^m$, the Bregman divergence $\mathrm{D}_{H_c^\alpha}(\theta\|\nu)$ has the form

$$\mathrm{D}_{H_c^\alpha}(\theta\|\nu) = \left\langle c^\alpha, \frac{\theta+1}{2} \log\left[\frac{\theta+1}{\nu+1}\right] + \frac{1-\theta}{2}\log\left[\frac{1-\theta}{1-\nu}\right] \right\rangle \tag{42}$$

$$= \left\langle \frac{c^\alpha}{2}, \log\left[\frac{1-\theta^2}{1-\nu^2}\right] \right\rangle + \langle \theta, \nabla H_c^\alpha(\theta) - \nabla H_c^\alpha(\nu)\rangle, \tag{43}$$

where the second equality can be shown by algebra.

Furthermore, define the vector-valued function $Z^t : [-1,1]^m \to \mathbb{R}^n$ as

$$Z^t(\theta)_i = \sum_{j=1}^m \exp\left[-t\left(\frac{C_{ij}}{2\|C\|_\infty} + \theta_j\right)\right], \tag{44}$$

which is the set of row normalization constants for $p^{\eta_t}(\theta)$.

For further convenience, we denote

$$\begin{aligned} X^t &:= \mathcal{D}_r p^{\eta_t}(\nu^t),\ \bar{X}^{t+1} := \mathcal{D}_r p^{\eta_{t+1}}(\bar{\nu}^{t+1}); \\ \mathbf{c}(X^t) &:= \mathbf{c}_r(p^{\eta_t}(\nu^t)),\ \mathbf{c}(\bar{X}^{t+1}) := \mathbf{c}_r(p^{\eta_{t+1}}(\bar{\nu}^{t+1})). \end{aligned} \tag{45}$$

By definition, then, we have for all $t \geq 0$ that

$$\mathbf{r}(X^t) = \mathbf{r}(\mathcal{D}_r p^{\eta_t}(\nu^t)) = r, \quad \mathbf{r}(\bar{X}^{t+1}) = \mathbf{r}(\mathcal{D}_r p^{\eta_{t+1}}(\bar{\nu}^{t+1})) = r. \tag{46}$$

For further notational convenience, we rewrite the unregularized ($\eta = 0$) saddle-point problem (4) as a problem over the coupling X rather than the row-stochastic matrix p

$$\min_{X \in \Delta_r^{n\times m}} \max_{\theta \in [-1,1]^m} \left\{K(X,\theta) := \langle C, X\rangle + 2\|C\|_\infty \langle \theta, \mathbf{c}(X) - c\rangle\right\}, \tag{47}$$

where $\Delta_r^{n\times m} = \{X \in \Delta^{n\times m} : \mathbf{r}(X) = r\}$ is the set of row-feasible couplings.

To begin, we show that there exists a dual optimizer in the set $[-1/2, 1/2]^m$, making the clipping step in (39) benign.

Lemma D.1. *Suppose that $\eta = 0$. There exists a saddle-point of (47) (X^*, θ^*) where $\theta^* \in [-1/2, 1/2]^m$ and X^* is a minimizer of (1).*

Proof. We recall that, when $\|C\|_\infty < \infty$, there exists an optimal dual pair $\varphi^* \in \mathbb{R}^n$, $\psi^* \in \mathbb{R}^m$ to (23) satisfying $\|\varphi^*\|_\infty \leq \|C\|_\infty$, $\|\psi^*\|_\infty \leq \|C\|_\infty$ (see [41, Remark 1.13]). Let $X^* \in \Pi(r,c)$ be the corresponding primal solution, which, by strong duality, is an optimizer to (1). Then, note that the following saddle-point problems are equivalent

$$\begin{aligned} &\min_{X\in\Delta^{n\times m}} \max_{\varphi\in\mathbb{R}^n, \psi\in\mathbb{R}^m} \{\langle C, X\rangle + \langle \varphi, \mathbf{r}(X) - r\rangle + \langle \psi, \mathbf{c}(X) - c\rangle\} \\ &\quad = \min_{X\in\Delta_r^{n\times m}} \max_{\psi\in\mathbb{R}^m} \{\langle C, X\rangle + \langle \psi, \mathbf{c}(X) - c\rangle\} \qquad (48) \\ &\quad = \min_{X\in\Delta_r^{n\times m}} \max_{\theta\in\mathbb{R}^m} \left\{K(X,\theta) := \langle C, X\rangle + 2\|C\|_\infty \langle \theta, \mathbf{c}(X) - c\rangle\right\}. \qquad (49) \end{aligned}$$

Define $\theta^* = \psi^*/(2\|C\|_\infty) \in [-1/2, 1/2]^m$, which satisfies $\theta^* \in [-1/2, 1/2]^m$. Since (X^*, ψ^*) is a saddle-point of (48), (X^*, θ^*) is a saddle-point of (49). Finally, by the saddle-point property, for all $X \in \Delta_r^{n\times m}$ and $\theta \in \mathbb{R}^m$, we have

$$K(X^*, \theta) \leq K(X^*, \theta^*) \leq K(X, \theta^*).$$

Since this is also true for all $\theta \in [-1,1]^m$, then (X^*, θ^*) is a saddle-point of (47), completing the proof. □

[5] We note that this can be simplified to $\nabla H_c^\alpha(\nu) = c^\alpha \odot \operatorname{arctanh}(\nu)$, however this form is not explicitly used in our analysis.

Next, we show that the clip operation in (39) can only decrease the divergence from another point in the subset $[-1/2, 1/2]^m$.

Lemma D.2. *Let $\theta \in [-1/2, 1/2]^m$ and $\hat{\nu} \in (-1, 1)^m$, and define*

$$\nu = \mathrm{clip}(\hat{\nu}, -1/2, 1/2).$$

Then,

$$\mathrm{D}_{H_c^\alpha}(\theta \| \nu) \le \mathrm{D}_{H_c^\alpha}(\theta \| \hat{\nu}). \tag{50}$$

Proof. We begin by recalling the optimality condition for a convex optimization problem $\min_{x \in \mathcal{X}} f(x)$ (see, e.g., [6, Equation 4.21])

$$x^* = \operatorname*{argmin}_{x \in \mathcal{X}} f(x) \textbf{ if and only if } x^* \in \mathcal{X} \text{ and } \langle \nabla f(x^*), x - x^* \rangle \ge 0 \text{ for all } x \in \mathcal{X}. \tag{51}$$

Now, we claim that $\nu = \operatorname{argmin}_{\mu \in [-1/2, 1/2]^m} \mathrm{D}_{H_c^\alpha}(\mu \| \hat{\nu})$. Suppose for some j that $\hat{\nu}_j > 1/2$, so $\nu_j = 1/2$. Then, consider the convex univariate function

$$\phi_j(\gamma) = \frac{1+\gamma}{2} \log\left(\frac{1+\gamma}{1+\hat{\nu}_j}\right) + \frac{1-\gamma}{2} \log\left(\frac{1-\gamma}{1-\hat{\nu}_j}\right). \tag{52}$$

The function $\phi_j(\gamma)$ is nonnegative over $\gamma \in [-1, 1]$, with its minimum at $\hat{\nu}_j$ (as one can verify). Therefore, if $\hat{\nu}_j \in [-1/2, 1/2]$, then $\nu_j = \hat{\nu}_j = \operatorname{argmin}_{\gamma \in [-1/2, 1/2]} \phi_j(\gamma)$. Now, assume that $\hat{\nu}_j \notin [-1/2, 1/2]$. Taking the derivative of ϕ_j, we obtain

$$\phi_j'(\gamma) = \frac{1}{2} \log\left(\frac{1+\gamma}{1-\gamma} \frac{1-\hat{\nu}_j}{1+\hat{\nu}_j}\right). \tag{53}$$

Note that if $\hat{\nu}_j > 1/2$, we have $1 + \hat{\nu}_j > 3/2$ and $1 - \hat{\nu}_j < 1/2$, so

$$\phi_j'(1/2) = \frac{1}{2} \log\left(\frac{3/2}{1/2} \frac{1-\hat{\nu}_j}{1+\hat{\nu}_j}\right) = \frac{1}{2} \log\left(\frac{3/2}{1+\hat{\nu}_j}\right) + \frac{1}{2} \log\left(\frac{1-\hat{\nu}_j}{1/2}\right) < 0. \tag{54}$$

Therefore, at $1/2$, $-\phi_j'(1/2) = -\phi_j'(\nu_j)$ and therefore $\phi_j'(\nu_j)(x - \nu_j) \ge 0$ for all $x \in [-1/2, 1/2]$, so $\nu_j = \operatorname{argmin}_{\gamma \in [-1/2, 1/2]} \phi_j(\gamma)$ by (51). A symmetric argument shows that $\nu_j = -1/2 = \operatorname{argmin}_{\gamma \in [-1/2, 1/2]} \phi_j(\gamma)$ for the case where $\hat{\nu}_j < -1/2$.

Repeating for every coordinate $1 \le j \le m$ and using the elementwise non-negativity of c^α, we have that

$$\nu = \operatorname*{argmin}_{\mu \in [-1/2, 1/2]^m} \left\{ \sum_{j=1}^m c_j^\alpha \phi_j(\mu_j) = \mathrm{D}_{H_c^\alpha}(\mu \| \hat{\nu}) \right\}. \tag{55}$$

Therefore, $\mathrm{clip}(\hat{\nu}, -1/2, 1/2)$ can be understood as a Bregman projection onto the subset $[-1/2, 1/2]^m$. Now, using the three-points identity for Bregman divergences, we have

$$\begin{aligned} \mathrm{D}_{H_c^\alpha}(\theta \| \nu) &= \mathrm{D}_{H_c^\alpha}(\theta \| \hat{\nu}) - \mathrm{D}_{H_c^\alpha}(\nu \| \hat{\nu}) - \langle \nabla H_c^\alpha(\nu) - \nabla H_c^\alpha(\hat{\nu}), \theta - \nu \rangle \\ &= \mathrm{D}_{H_c^\alpha}(\theta \| \hat{\nu}) - \mathrm{D}_{H_c^\alpha}(\nu \| \hat{\nu}) - \left\langle \nabla_\nu \mathrm{D}_{H_c^\alpha}(\nu \| \hat{\nu}), \theta - \nu \right\rangle. \end{aligned}$$

By (51) and (55), $\left\langle \nabla_\nu \mathrm{D}_{H_c^\alpha}(\nu \| \hat{\nu}), \theta - \nu \right\rangle \ge 0$ for all $\theta \in [-1/2, 1/2]^m$. Then, we obtain

$$\mathrm{D}_{H_c^\alpha}(\theta \| \nu) \le \mathrm{D}_{H_c^\alpha}(\theta \| \hat{\nu}) - \mathrm{D}_{H_c^\alpha}(\nu \| \hat{\nu}) \le \mathrm{D}_{H_c^\alpha}(\theta \| \hat{\nu}),$$

where the second inequality follows from the nonnegativity of Bregman divergences for convex functions. We therefore conclude the proof. □

Next, we note a simple identity following from the definition of $p^{\eta_t}(\theta)$ and Z^t.

Lemma D.3. *Let $\theta \in [-1, 1]^m$ and $t \ge 1$ and define $\tilde{X} = \mathcal{D}_r p^{\eta_t}(\theta)$. Then, we have the following identity*

$$-\left\langle r, \log Z^t(\theta) \right\rangle - t \left\langle \mathbf{c}(\tilde{X}), \theta \right\rangle = \frac{t}{2} \left\langle C, \tilde{X} \right\rangle - H(\tilde{X}) + H(r). \tag{56}$$

Proof. First, note that $Z^t(\theta)_i$ is the normalization constant for the i^{th} row of $p^{\eta_t}(\theta)$. Then, from (40), we have

$$\begin{aligned}
-H(\tilde{X}) &= \langle \mathcal{D}_r p^{\eta_t}(\theta), \log \mathcal{D}_r p^{\eta_t}(\theta) \rangle \\
&= \langle \mathcal{D}_r p^{\eta_t}(\theta), \log p^{\eta_t}(\theta) \rangle + \langle r, \log r \rangle \\
&\overset{(40)}{=} -t \left\langle \mathcal{D}_r p^{\eta_t}(\theta), \frac{C}{2} + \mathbf{1}\theta^\top \right\rangle - \left\langle \mathcal{D}_r p^{\eta_t}(\theta), \log Z^t(\theta) \right\rangle - H(r) \\
&= -\frac{t}{2} \left\langle \tilde{X}, C \right\rangle - t \left\langle \mathbf{c}(\tilde{X}), \theta \right\rangle - \left\langle r, \log Z^t(\theta) \right\rangle - H(r).
\end{aligned}$$

Rearranging yields the result. □

We start by stating consequences of the optimality conditions in (35)-(39).

Lemma D.4. *For all $t \geq 0$, the following equalities hold*

a)

$$\log X^{t+1} - \log X^t = -\frac{1}{2}C - \mathbf{1}(\bar{\theta}^{t+1})^\top - \log Z^{t+1}(\nu^{t+1})\mathbf{1}^\top + \log Z^t(\nu^t)\mathbf{1}^\top, \tag{57}$$

b)

$$\nabla H_c^\alpha(\bar{\theta}^{t+1}) - \nabla H_c^\alpha(\theta^t) = \mathbf{c}(X^t) - c, \tag{58}$$

c)

$$\nabla H_c^\alpha(\hat{\theta}^{t+1}) - \nabla H_c^\alpha(\theta^t) = \mathbf{c}(\bar{X}^{t+1}) - c, \tag{59}$$

Proof. a) Using the definition of the dual-to-primal map (40) and canceling the common $\log(r\mathbf{1}^\top)$ terms gives

$$\begin{aligned}
\log X^{t+1} - \log X^t &\overset{(45)}{=} \log p^{\eta_{t+1}}(\nu^{t+1}) - \log p^t(\nu^t) \\
&\overset{(40)}{=} -(t+1)\left(\frac{1}{2}C + \mathbf{1}(\nu^{t+1})^\top\right) + t\left(\frac{1}{2}C + \mathbf{1}(\nu^t)^\top\right) \\
&\quad - \log Z^{t+1}(\nu^{t+1})\mathbf{1}^\top + \log Z^t(\nu^t)\mathbf{1}^\top \\
&= -\frac{1}{2}C + \mathbf{1}\left(t\nu^t - (t+1)\nu^{t+1}\right)^\top - \log Z^{t+1}(\nu^{t+1})\mathbf{1}^\top + \log Z^t(\nu^t)\mathbf{1}^\top \\
&\overset{(37)}{=} -\frac{1}{2}C - \mathbf{1}(\bar{\theta}^{t+1})^\top - \log Z^{t+1}(\nu^{t+1})\mathbf{1}^\top + \log Z^t(\nu^t)\mathbf{1}^\top,
\end{aligned}$$

where the final line follows from the ν^{t+1} update.

b) The optimality conditions for problem (36) give

$$0 = \mathbf{c}(X^t) - c - \nabla H_c^\alpha(\bar{\theta}^{t+1}) + \nabla H_c^\alpha(\theta^t). \tag{60}$$

Rearranging gives the result. Statement c) follows from the same logic, noting that the optimality conditions for problem (38) give

$$0 = \mathbf{c}(\bar{X}^{t+1}) - c - \nabla H_c^\alpha(\hat{\theta}^{t+1}) + \nabla H_c^\alpha(\theta^t). \tag{61}$$

□

Using the primal optimality conditions in Lemma D.4, we can show the following identity for differences of KL-divergence terms.

Lemma D.5. *For all $t \geq 0$, the following identities hold*

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^t) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^{t+1}) = &- \left\langle \mathbf{c}(\bar{X}^{t+1}), \bar{\theta}^{t+1} \right\rangle - \frac{1}{2}\left\langle C, \bar{X}^{t+1} \right\rangle \\
&- \left\langle r, \log Z^{t+1}(\nu^{t+1}) \right\rangle + \left\langle r, \log Z^t(\nu^t) \right\rangle.
\end{aligned} \tag{62}$$

Proof. Note that by the definition of the KL-divergence D_{KL}, we have

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^t)-\mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^{t+1}) &= \left\langle \bar{X}^{t+1}, \log X^{t+1} - \log X^t \right\rangle \\ &\overset{(57)}{=} \left\langle \bar{X}^{t+1}, -\frac{1}{2}C - \mathbf{1}(\bar{\theta}^{t+1})^\top - \log Z^{t+1}(\nu^{t+1})\mathbf{1}^\top + \log Z^t(\nu^t)\mathbf{1}^\top \right\rangle,\end{aligned}$$

where the second line follows by Lemma D.4. Rearranging and using $\bar{X}^{t+1}\mathbf{1} = r$ and $(\bar{X}^{t+1})^\top\mathbf{1} = \mathbf{c}(\bar{X}^{t+1})$ gives the result. □

Lemma D.6. *For all iteration $t \geq 0$, we have the following identity*

$$\begin{aligned}\mathrm{D}_{H_c^\alpha}(\theta^*\|\hat{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^t) + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\theta^t) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\hat{\theta}^{t+1}) \\ = \left\langle \bar{\theta}^{t+1} - \theta^*, \mathbf{c}(\bar{X}^{t+1}) - c \right\rangle. \end{aligned} \tag{63}$$

Proof. Using the definition of $\mathrm{D}_{H_c^\alpha}$ in (43) and expanding terms, we obtain

$$\begin{aligned}&\mathrm{D}_{H_c^\alpha}(\theta^*\|\hat{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^t) + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\theta^t) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\hat{\theta}^{t+1}) \\ &\overset{(43)}{=} \left\langle \frac{c^\alpha}{2}, \log\frac{1-(\theta^*)^2}{1-(\hat{\theta}^{t+1})^2} \right\rangle - \left\langle \frac{c^\alpha}{2}, \log\frac{1-(\theta^*)^2}{1-(\theta^t)^2} \right\rangle + \left\langle \theta^*, \nabla H_c^\alpha(\theta^t) - \nabla H_c^\alpha(\hat{\theta}^{t+1}) \right\rangle \\ &\quad + \left\langle \frac{c^\alpha}{2}, \log\frac{1-(\bar{\theta}^{t+1})^2}{1-(\theta^t)^2} \right\rangle - \left\langle \frac{c^\alpha}{2}, \log\frac{1-(\bar{\theta}^{t+1})^2}{1-(\hat{\theta}^{t+1})^2} \right\rangle + \left\langle \bar{\theta}^{t+1}, \nabla H_c^\alpha(\hat{\theta}^{t+1}) - \nabla H_c^\alpha(\theta^t) \right\rangle \\ &= \left\langle \bar{\theta}^{t+1} - \theta^*, \nabla H_c^\alpha(\hat{\theta}^{t+1}) - \nabla H_c^\alpha(\theta^t) \right\rangle \\ &\overset{(59)}{=} \left\langle \bar{\theta}^{t+1} - \theta^*, \mathbf{c}(\bar{X}^{t+1}) - c \right\rangle,\end{aligned}$$

where the second equality follows by algebra and the final line follows by Lemma D.4(c). □

With the primary identities/inequalities established, we now prove a single-step bound for Algorithm 4.

Lemma D.7. *Let $(X^*, \theta^*) \in \Pi(r,c) \times [-1/2, 1/2]^m$ be a saddle-point of (47) where X^* is an optimizer of (1). Then, for all $t \geq 0$, we have the inequality*

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(X^*\|X^t) - \mathrm{D}_{\mathrm{KL}}(X^*\|X^{t+1}) \geq \frac{1}{2}\left\langle \bar{X}^{t+1} - X^*, C \right\rangle + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \theta^* \right\rangle \\ - \mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^t) + \mathrm{D}_{H_c^\alpha}(\theta^*\|\hat{\theta}^{t+1}) + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\theta^t) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1}\|\hat{\theta}^{t+1}) \\ + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^t) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^{t+1}).\end{aligned} \tag{64}$$

Proof. First, we note that Lemma D.1 implies that such a saddle-point exists. Then, expanding the definition of the KL-divergence gives

$$\begin{aligned}&\mathrm{D}_{\mathrm{KL}}(X^*\|X^t) - \mathrm{D}_{\mathrm{KL}}(X^*\|X^{t+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^t) + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^{t+1}) \\ &\quad = \left\langle \bar{X}^{t+1} - X^*, \log X^t - \log X^{t+1} \right\rangle \\ &\quad \overset{(57)}{=} \frac{1}{2}\left\langle \bar{X}^{t+1} - X^*, C \right\rangle \\ &\qquad + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \bar{\theta}^{t+1} \right\rangle + \left\langle \bar{X}^{t+1} - X^*, (\log Z^{t+1}(\nu^{t+1}) - \log Z^t(\nu^t))\mathbf{1}^\top \right\rangle \\ &\quad = \frac{1}{2}\left\langle \bar{X}^{t+1} - X^*, C \right\rangle + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \hat{\theta}^{t+1} \right\rangle + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \bar{\theta}^{t+1} - \hat{\theta}^{t+1} \right\rangle,\end{aligned}$$

where the second line follows by Lemma D.4 and the third line by algebra and the fact that $(\bar{X}^{t+1} - X^*)\mathbf{1} = 0$ by (46) and the row feasibility of X^*. Adding and subtracting $\mathbf{c}(X^t)$ in the second inner product term then yields

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(X^*\|X^t) - \mathrm{D}_{\mathrm{KL}}(X^*\|X^{t+1}) = \frac{1}{2}\left\langle \bar{X}^{t+1} - X^*, C \right\rangle + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \hat{\theta}^{t+1} \right\rangle \\ + \left\langle \mathbf{c}(\bar{X}^{t+1}) - \mathbf{c}(X^t), \bar{\theta}^{t+1} - \hat{\theta}^{t+1} \right\rangle + \left\langle \mathbf{c}(X^t) - c, \bar{\theta}^{t+1} - \hat{\theta}^{t+1} \right\rangle \\ + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^t) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1}\|X^{t+1}).\end{aligned} \tag{65}$$

Now we can begin to utilize the dual optimality conditions. First, from (36) and (38), we have

$$\begin{aligned}\left\langle \mathbf{c}(\bar{X}^{t+1}) - \mathbf{c}(X^t), \bar{\theta}^{t+1} - \hat{\theta}^{t+1} \right\rangle &\overset{(58)(59)}{=} -\left\langle \nabla H_c^\alpha(\hat{\theta}^{t+1}) - \nabla H_c^\alpha(\bar{\theta}^{t+1}), \hat{\theta}^{t+1} - \bar{\theta}^{t+1} \right\rangle \\ &= -\mathrm{D}_{H_c^\alpha}(\hat{\theta}^{t+1} \| \bar{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \hat{\theta}^{t+1}), \end{aligned} \tag{66}$$

where the second equality can be verified by simple algebra. Next, we have

$$\begin{aligned} -\left\langle \mathbf{c}(X^t) - c, \hat{\theta}^{t+1} - \bar{\theta}^{t+1} \right\rangle &\overset{(58)}{=} -\left\langle \nabla H_c^\alpha(\bar{\theta}^{t+1}) - \nabla H_c^\alpha(\theta^t), \hat{\theta}^{t+1} - \bar{\theta}^{t+1} \right\rangle \\ &= \mathrm{D}_{H_c^\alpha}(\hat{\theta}^{t+1} \| \bar{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\hat{\theta}^{t+1} \| \theta^t) + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \theta^t), \end{aligned} \tag{67}$$

Finally, using the three-point inequality from [10, Lemma 3.2] applied to the negation of (38) using the definitions in (45), we obtain

$$\begin{aligned}\left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \hat{\theta}^{t+1} \right\rangle &\geq \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \theta^* \right\rangle - \mathrm{D}_{H_c^\alpha}(\theta^* \| \theta^t) \\ &\quad + \mathrm{D}_{H_c^\alpha}(\hat{\theta}^{t+1} \| \theta^t) + \mathrm{D}_{H_c^\alpha}(\theta^* \| \hat{\theta}^{t+1}). \end{aligned} \tag{68}$$

Combining (66), (67), and (68) with (65), we obtain the result (64). □

We now prove Proposition 3.4. For expository purposes, we state the two conclusions (a) and (b) as separate lemmas, each following from Lemma D.7.

Lemma D.8. *For all $t \geq 1$, we have the inequality*

$$\left\langle \widetilde{X}^t, C \right\rangle - \min_{X \in \Pi(r,c)} \langle X, C \rangle \leq 4\|C\|_\infty \|\mathbf{c}(X^t) - c\|_1 + \frac{2\|C\|_\infty \log m}{t}. \tag{69}$$

where $\widetilde{X}^t = \mathrm{Round}(X^t, r, c)$.

Proof. Starting from Lemma D.7 and applying Lemmas D.5 and D.6 and algebra, we obtain

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(X^* \| X^t) - \mathrm{D}_{\mathrm{KL}}(X^* \| X^{t+1}) &\overset{(64)}{\geq} \frac{1}{2}\left\langle \bar{X}^{t+1} - X^*, C \right\rangle + \left\langle \mathbf{c}(\bar{X}^{t+1}) - c, \theta^* \right\rangle \\ &\quad - \mathrm{D}_{H_c^\alpha}(\theta^* \| \theta^t) + \mathrm{D}_{H_c^\alpha}(\theta^* \| \hat{\theta}^{t+1}) + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \theta^t) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \hat{\theta}^{t+1}) \\ &\quad + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^t) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^{t+1}) \\ &\overset{(62)(63)}{=} -\frac{1}{2}\langle X^*, C \rangle - \left\langle \bar{\theta}^{t+1}, c \right\rangle + \left\langle r, \log Z^t(\nu^t) - \log Z^{t+1}(\nu^{t+1}) \right\rangle \\ &\overset{(37)}{=} -\frac{1}{2}\langle X^*, C \rangle + \left\langle c, t\nu^t - (t+1)\nu^{t+1} \right\rangle + \left\langle r, \log Z^t(\nu^t) - \log Z^{t+1}(\nu^{t+1}) \right\rangle \end{aligned}$$

where the last two equalities follow from algebra.

Summing from 0 to $t-1$ for $t \geq 1$, we have

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(X^* \| X^0) - \mathrm{D}_{\mathrm{KL}}(X^* \| X^t) &\geq -\frac{t}{2}\langle X^*, C \rangle + t\left\langle \mathbf{c}(X^t) - c, \nu^t \right\rangle \\ &\quad - t\left\langle \mathbf{c}(X^t), \nu^t \right\rangle - \left\langle r, \log Z^t(\nu^t) \right\rangle + \left\langle r, \log Z^0(\nu^0) \right\rangle \\ &\overset{(45)(56)}{=} \frac{t}{2}\left\langle X^t - X^*, C \right\rangle + t\left\langle \mathbf{c}(X^t) - c, \nu^t \right\rangle - H(X^t) + H(r) + \log m \\ &\geq \frac{t}{2}\left\langle X^t - X^*, C \right\rangle + t\left\langle \mathbf{c}(X^t) - c, \nu^t \right\rangle, \end{aligned}$$

where the equality follows from Lemma D.3 with $(\theta, \tilde{X}) = (\nu^t, X^t)$, and the facts that $\eta_0^{-1} = 0$ and

$$\left\langle r, \log Z^0(\nu^0) \right\rangle = \sum_{i=1}^n r_i \mathrm{LSE}(\mathbf{0}_m) = \sum_{i=1}^n r_i \log m = \log m.$$

The final line follows from

$$-H(X^t) + H(r) \geq -H(X^t) = -\sum_{i=1}^{n} r_i H(p_{i:}) \geq -\sum_{i=1}^{n} r_i \log m = -\log m,$$

where we use the fact that each row $p_{i:} \in \Delta^m$.

Rearranging, discarding the negative $\mathrm{D}_{\mathrm{KL}}(X^* \| X^t)$ term and dividing both sides by $t/2$ gives,

$$\begin{aligned}
\left\langle X^t - X^*, C \right\rangle &\leq 2 \left\langle -\nu^t, \mathbf{c}(X^t) - c \right\rangle + \frac{2\mathrm{D}_{\mathrm{KL}}(X^* \| X^0)}{t} \\
&\leq 2\|\mathbf{c}(X^t) - c\|_1 + \frac{2\mathrm{D}_{\mathrm{KL}}(X^* \| X^0)}{t} \\
&\leq 2\|\mathbf{c}(X^t) - c\|_1 + \frac{2\log m}{t}, \qquad (70)
\end{aligned}$$

where the second inequality follows from Hölder's inequality and using $\|\nu^t\|_\infty \leq 1$ and the final line from the choice $X^0 = \mathcal{D}_r(1/m)^{n \times m}$.

The lower bound follows from defining $\widetilde{X}^t = \mathrm{Round}(X^t, r, c)$ and applying Lemma C.1 to obtain

$$\begin{aligned}
\left\langle X^t - X^*, C \right\rangle &= \left\langle \widetilde{X}^t - X^*, C \right\rangle + \left\langle X^t - \widetilde{X}^t, C \right\rangle \\
&\geq \left\langle \widetilde{X}^t - X^*, C \right\rangle - \|C\|_\infty \|X^t - \widetilde{X}^t\|_1 \\
&\overset{(27)}{\geq} \left\langle \widetilde{X}^t - X^*, C \right\rangle - 2\|\mathbf{c}(X^t) - c\|_1, \qquad (71)
\end{aligned}$$

where the first inequality follows from Hölder's inequality and the second from Lemma C.1. Combining (70) with (71) gives the result for $\|C\|_\infty = 1$. If $\|C\|_\infty \neq 1$, then applying the analysis to the normalized cost $C/\|C\|_\infty$ and multiplying both sides by $\|C\|_\infty$ gives the claimed inequality. □

Lemma D.9. *Let $(X^*, \theta^*) \in \Pi(r, c) \times [-1/2, 1/2]^m$ be a saddle-point of problem (47). Then, defining*

$$D_t := \sum_{s=0}^{t-1} \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \hat{\theta}^{s+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \theta^s) + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^{s+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^s), \qquad (72)$$

then for all $t \geq 1$

$$\min_{1 \leq s \leq t} \{K(\bar{X}^s, \theta^*) - K(X^*, \bar{\theta}^s)\} \leq \frac{2\|C\|_\infty [\log m + (1+\alpha)\log 2 + D_t]}{t}. \qquad (73)$$

Proof. Assume that $\|C\|_\infty = 1$. First, we note that, by the definition in (47)

$$K(\bar{X}^{t+1}, \theta^*) = \left\langle \bar{X}^{t+1}, C \right\rangle + 2\left\langle \theta^*, \mathbf{c}(\bar{X}^{t+1}) - c \right\rangle, \quad K(X^*, \bar{\theta}^{t+1}) = \left\langle X^*, C \right\rangle.$$

Then, for $t \geq 0$ Lemma D.7 can be rewritten as

$$\begin{aligned}
\frac{1}{2}[K(\bar{X}^{t+1}, \theta^*) - K(X^*, \bar{\theta}^{t+1})] &\overset{(64)}{\leq} \mathrm{D}_{\mathrm{KL}}(X^* \| X^t) - \mathrm{D}_{\mathrm{KL}}(X^* \| X^{t+1}) \\
&\quad + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \hat{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \theta^t) + \mathrm{D}_{H_c^\alpha}(\theta^* \| \theta^t) - \mathrm{D}_{H_c^\alpha}(\theta^* \| \hat{\theta}^{t+1}) \\
&\quad + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^{t+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^t) \\
&\overset{(39)(50)}{\leq} \mathrm{D}_{\mathrm{KL}}(X^* \| X^t) - \mathrm{D}_{\mathrm{KL}}(X^* \| X^{t+1}) \\
&\quad + \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \hat{\theta}^{t+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{t+1} \| \theta^t) + \mathrm{D}_{H_c^\alpha}(\theta^* \| \theta^t) - \mathrm{D}_{H_c^\alpha}(\theta^* \| \theta^{t+1}) \\
&\quad + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^{t+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{t+1} \| X^t)
\end{aligned}$$

where the second inequality uses Lemma D.2 and $\theta^* \in [-1/2, 1/2]^m$.

Summing from 0 to $t-1$ and using the definition of D_t in (72) gives

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(X^*\|X^0) - \mathrm{D}_{\mathrm{KL}}(X^*\|X^t) + \mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^0) - \mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^t) + D_t \\ &\geq \frac{1}{2}\sum_{s=1}^{t} K(\bar{X}^s, \theta^*) - K(X^*, \bar{\theta}^s) \\ &\geq \frac{t}{2}\min_{1\leq s\leq t}\{K(\bar{X}^s, \theta^*) - K(X^*, \bar{\theta}^s)\},\end{aligned}$$

where the final inequality follows from the saddle-point property, which implies that

$$K(Y, \theta^*) - K(X^*, \theta) \geq 0 \tag{74}$$

for all $Y \in \{X \in \Delta^{n\times m} : \mathbf{r}(X) = r\}$ and $\theta \in [-1, 1]^m$. Note that

$$\begin{aligned}\mathrm{D}_{H_c^\alpha}(\theta^*\|\theta^0) &\overset{(42)}{=} \left\langle c^\alpha, \frac{\theta^*+1}{2}\log(\theta^*+1) + \frac{1-\theta^*}{2}\log(1-\theta^*)\right\rangle \\ &\leq \left\langle c^\alpha, \frac{\theta^*+1}{2}\log 2 + \frac{1-\theta^*}{2}\log 2\right\rangle = (1+\alpha)\log 2\end{aligned} \tag{75}$$

since $\theta^0 = \mathbf{0}_m$ and $\theta^* \in [-1/2, 1/2]^m \subset [-1, 1]^m$. The result for $\|C\|_\infty = 1$ then directly follows from dropping the nonnegative terms and using $\mathrm{D}_{\mathrm{KL}}(X^*\|X^0) \leq \log m$ combined with (75). As in Lemma D.8, applying the analysis to the normalized cost and multiplying both sides by $\|C\|_\infty$ gives the result. □

Proposition 3.4 then follows from combining Lemmas D.8 and D.9. These conditions can be understood as reducing convergence to controlling two separate, computable terms. Lemma D.8 provides last-iterate sub-optimality certificates, but requires convergence in the column infeasibility $\|\mathbf{c}(X^t) - c\|_1$ to imply convergence in objective value. As observed in Section 5 (see Fig. 2), LAMP rapidly finds feasible solutions, with convergence rates then limited by the sublinear term.

Lemma D.9 further shows that LAMP finds an ε-approximate saddle-point in $\mathcal{O}(\varepsilon^{-1}(\log m + D_t))$ iterations. As stated in the main text, proving an $o(t)$ bound on D_t would then imply the convergence of LAMP, with non-asymptotic complexity dependent on the specific bound. Note that D_t is computable, and therefore can be tracked along iterate trajectories. Fig. 6 shows the empirical behavior of $\{D_t\}_{t\geq 0}$ for cell similarity (Fig. 6a) and DOTmark (Fig. 6b). For both problem sets, the D_t rapidly becomes negative, then approaches 0 from below. While we observe some nonmonotonicity, the behavior is relatively simple. For the cell similarity problems, the choice of metric has little effect on the behavior of the D_t curve. In contrast, the behavior of D_t is more metric dependent for the DOTmark instances, as shown in in Fig. 6b. Each curve appears to converge to zero, but the convergence rate appears to decrease as the $\|C\|_\infty$ value increases. In the case of 2-dimensional DOTmark problems, we have $\|C\|_\infty = \sqrt{n} - 1$ for ℓ_∞, $\|C\|_\infty = 2(\sqrt{n} - 1)$ for ℓ_1, and $\|C\|_\infty = 2(\sqrt{n} - 1)^2$ for ℓ_2^2. It may be possible to show that the summand of D_t is nonpositive after some number of iterations, however we leave investigations to future work.

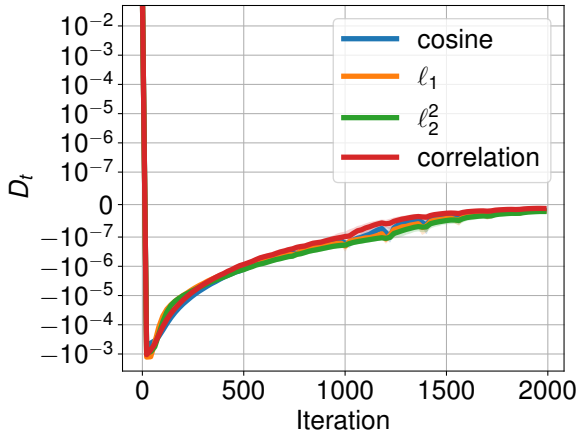
E Numerical and Experimental Details

In this section we provide further details on our GPU-accelerated implementation of LAMP and the numerical experiments in Sections 3, 4, and 5.

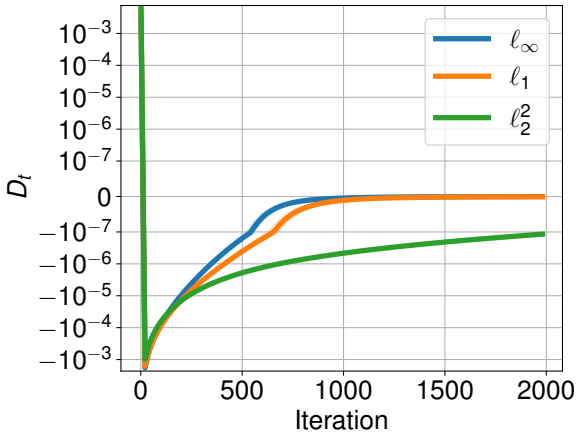
E.1 Fused/Warp-tiled CUDA Reductions

For both LAMP and Sinkhorn, we perform the max and LSE reductions in a single loop-fused and warp-tiled kernel `warp_lse_reduction`.

Loop fusion replaces the separate reduction loops with a single loop, which performs an on-the-fly variant of the `LogSumExp` trick. Consider the generic problem of computing $\mathrm{LSE}(x_{i:})$ over $x \in \mathbb{R}^{n\times m}$ (i.e., a row-wise LSE reduction). The fused loop maintains two sequences, m_i^k and s_i^k, for storing the maximum value and



(a) Behavior of the summed sequence D_t for the cell similarity OT problems ($n \in \{256, 576, 1024, 1600\}$ and $n = m$).



(b) Behavior of the summed sequence D_t for DOTmark OT problems ($n \in \{256, 576, 1024, 1600\}$ and $n = m$).

Figure 6: Behavior of the summed sequence D_t in OT benchmark problems from the cell similarity database [left] and DOTmark [right].

accumulated exponential sums for row i respectively. We initialize $m_i^0 = -\infty$[6], $s_i^0 = 0$. For $1 \le j \le m$, if $x_{ij} \ge m_i^{j-1}$, then we perform the update

$$m_i^j = x_{ij}, \quad s_i^j = 1 + (s_i^{j-1})^{-m_i^j + m_i^{j-1}}.$$

Otherwise, we simply set $m_i^j = m_i^{j-1}$ and update

$$s_i^j = s_i^{j-1} + \exp[x_{ij} - m_i^{j-1}].$$

Warp-tiling splits the computation into a two-loop structure. The outer loop iterates n/n_{Warps} times and performs the row-wise LSE and max reductions for n_{Warps} rows simultaneously. The reduction operations are warp-tiled, meaning that each warp computes 32 terms of each reduction sum in parallel. After iterating through all m elements, the warp threads perform synchronized reductions/broadcasting using `__shfl_down_sync()` and `__shfl_sync()` primitives. Since Julia uses column-major storage, we store a transposed copy of the C matrix (for non-kernelized costs) to ensure that memory accesses are contiguous within each warp.

E.2 Experimental Details

We now provide additional details on problem selection, termination criteria, and problem parameters.

We compare LAMP to a log-domain stabilized implementation of Sinkhorn's algorithm [39, 1, 37], Dual Extrapolation [19], APDAMD [26], Accelerated Sinkhorn [26], and HPD [8]. We implement each solver in Julia with GPU-accelerated operations using the CUDA.jl package. The implementations, data, and experiment/plotting code are available in a public repository[7].

Each solver was run with a maximum of T iterations and a wall clock limit of S seconds. The values of T and S varied by experiment, and are given for each figure below. Unless otherwise specified, if a solver reached 10^{-10} primal-dual gap (checked every 25-200 iterations, depending on the experiment), then the solver terminates early. For all EOT solvers, the dual function is given by (24). For LAMP the dual functions for $\eta = 0$ and $\eta > 0$ are given by (25) and (26), respectively. Dual extrapolation [19] used the $\eta = 0$ dual function in (25).

[6] AKA `-DBL_MAX`

[7] https://github.com/mxburns2022/CuLAMP.jl.

For all LAMP testing, we set $\beta \approx \log 3$ (specifically 1.1), $\alpha = 0.01$, $\tau_1 = \tau_2 = 1/(2\|C\|_\infty)$. Algorithm comparisons were performed on an HPC cluster running a Xeon Gold 6548Y+ CPU with an NVIDIA L40s GPU.

To ensure that each marginal has full support, we first add a small 10^{-6} additive perturbation to the marginals and then renormalize.

Details for Figure 1 We choose five 32×32 problems each from the DOTmark [38] classes "ClassicImages", "GRFsmooth", and "GRFrough" for a total of 15 problems. Each solver is given a 20 minute/10^6 iteration limit. Elapsed wall time, objective value, infeasibility, and primal/dual values are reported every 25 iterations. We set the hyperparameters for each algorithm using η as described in each original reference.

Details for Figure 2 We choose ten 64×64 image pairs each from the DOTmark [38] classes "ClassicImages", "GRFsmooth", and "MicroscopyImages" for a total of 30 problem instances. Furthermore, we run each problem using ℓ_2^2, ℓ_1, and ℓ_∞ ground costs. Kernel-based implementations of log-domain Sinkhorn and LAMP are each given a 10 minute/10^6 iteration limit. Unlike the non-kernelized Sinkhorn, we terminate the solver based on the condition $\|\mathbf{c}(X^t) - c\|_1 + \|\mathbf{r}(X^t) - r\|_1 \leq \varepsilon/2$ (which is 5×10^{-11}) rather than a primal-dual gap. Elapsed wall time, objective value, infeasibility, and dual values are reported every 200 iterations for LAMP and Sinkhorn, while annealed Sinkhorn reports metrics after every call to the inner Sinkhorn routine. For annealed Sinkhorn, we use a multiplicative annealing schedule with $\eta_t = \max\{0.8^t \eta_i, \eta_f\}$ with $\eta_i = 0.1$, $\eta_f = 10^{-10}$ after brief tuning, and each subproblem is solved to $\varepsilon = 10^{-10}$ accuracy.

Details for Figure 3: We utilize the cell omics dataset collected by [27] (published under a CC-BY-4.0 license) and preprocessed by [18]. The preprocessed dataset used can be found in the publicly available repository https://github.com/cantinilab/OT-scOmics as `data/liu_scatac_preprocessed.csv.gz`. Each row of the dataset represents a genetic feature, with columns representing individual cells. Following [18], we model the similarity between two cells by an optimal transport problem. The problem dimension $n = m$ is the number of genetic features. The marginals r and c are computed as the normalized feature vectors for each cell. For a k-cell database, costs are computed as

$$C_{ij} = \sum_{\ell=1}^{k} d(i_\ell, j_\ell), \tag{76}$$

where $d(i_\ell, j_\ell)$ is a metric between feature i of cell ℓ and feature j of cell ℓ. Therefore, the cost of computing each entry of C scales $\mathcal{O}(k)$ (i.e., with the number of cells). We consider four different metrics d,

1. ℓ_1 distance $d(x, y) = \|x - y\|_1$,
2. ℓ_2^2 distance $d(x, y) = \|x - y\|_2^2$,
3. cosine similarity metric
$$d(x, y) = 1 - \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2},$$
4. Pearson correlation metric
$$d(x, y) = 1 - \frac{\langle x - \bar{x}, y - \bar{y} \rangle}{\|x - \bar{x}\|_2 \|y - \bar{y}\|_2},$$
where the sample means $\bar{x}$ and $\bar{y}$ as well as second moments are precomputed for each feature for efficiency.

From the dataset, we select the $n = m = 5000$ features with the highest variance and randomly subsample $k = 20$ cells at random to form the dataset. We then randomly choose a pair of cells to create an OT problem instance. We repeat this process 10 times to generate 10 problem instances, which we then run with a 1 hour/100000 iteration timeout. The accuracy target was additionally set at 10^{-4}, with algorithms terminating early if a 10^{-4}-solution was detected using a primal-dual gap bound. Trajectories are averaged over the 10 problems. For Sinkhorn, we use $\eta = 10^{-4}$. For annealed Sinkhorn, we use a multiplicative annealing schedule

with $\eta_t = \max\{0.95^t \eta_i, \eta_f\}$ with $\eta_i = 0.01$, $\eta_f = 10^{-4}$ after brief tuning, and each subproblem is solved to $\varepsilon = 10^{-4}$ accuracy.

Details for Figure 4 The original images were generated using Google Gemini 2 at a resolution of 1024×1024. Note that our use of the images is in compliance with the Google Generative AI Terms of Service. The images were then rescaled using the Images.jl package and the RGB distributions were computed using a 1D kernel density estimator with $2^{\lceil \log_2 n \rceil}$ sample points as described in [36]. Denoting γ_R, γ_G, and γ_B as the normalized KDE densities, we then compute each image marginal γ as

$$\gamma_i \propto \gamma_R(R_i) + \gamma_G(G_i) + \gamma_B(B_i),$$

where R_i, G_i, and B_i are respectively the RGB values of pixel i. KDE densities are computed using the KernelDensity.jl package. Transfer maps are computed using ℓ_2^2 ground costs. Upon terminating at iteration T, the color channels for the row-marginal transfer image is then computed as

$$\begin{aligned} R^r_{\text{out}} &= p^{\eta_T}(\nu^T) R^c, \quad G^r_{\text{out}} = p^{\eta_T}(\nu^T) G^c, \quad B^r_{\text{out}} = p^{\eta_T}(\nu^T) B^c, \\ R^c_{\text{out}} &= (p^{\eta_T}(\nu^T))^\top R^r, \quad G^c_{\text{out}} = (p^{\eta_T}(\nu^T))^\top G^r, \quad B^c_{\text{out}} = (p^{\eta_T}(\nu^T))^\top B^r, \end{aligned}$$

where R^r, G^r, and B^r are the input RGB values for the row-marginal image, and R^c, G^c, and B^c are the input RGB values for the column-marginal image.

Details for Figure 5 We choose five 32×32 problems each from the DOTmark classes "ClassicImages", "GRFsmooth", and "MicroscopyImages" for a total of 15 problems. Prior to running the experiment, we performed brief parameter tuning to optimize the primal and dual stepsizes (both set to 1) and the number of inner iterations (set to 4) for Dual Extrapolation.

Both solvers were given a 10 minute/10^6 iteration limit. Since Dual Extrapolation has a two-loop structure with 4 inner iterations per outer iteration, we set a limit of 2.5×10^5 total "outer" iterations. Elapsed wall time, objective value, infeasibility, and primal/dual values are reported every 200 outer iterations.

Details for Figure 6: For Fig. 6a, we construct 10 cell similarity problem instances each with $k = 50$ cells, $n = m \in \{256, 576, 1024, 1600\}$ features, and costs computed using one of the four similarity measures described for Fig. 3 (ℓ_1, ℓ_2^2, cosine, Pearson correlation). The summand of D_t,

$$\mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \hat{\theta}^{s+1}) - \mathrm{D}_{H_c^\alpha}(\bar{\theta}^{s+1} \| \theta^s) + \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^{s+1}) - \mathrm{D}_{\mathrm{KL}}(\bar{X}^{s+1} \| X^s),$$

is logged at every iteration, with Fig. 6a reporting the cumulative sum averaged over. The cost/marginal computations are identical to Fig. 3. For Fig. 6b, we select 10 image pairs each from 3 DOTmark classes ("ClassicImages", "GRFsmooth", and "GRFrough") with images sized to r x r, where $r \in \{16, 24, 32, 40\}$, which gives $n = m \in \{256, 576, 1024, 1600\}$. Costs are computed using ℓ_1, ℓ_2^2, or ℓ_∞ ground costs. As in Fig. 6a, the summand of D_t is logged at every iteration, with the cumulative sum averaged over the problem instances for each cost metric. Each solver is given a 10 minute/100000 iteration time limit.

Details for Table 1: All data for Table 1 were collected on an RTX 4090 GPU with an Intel i9-13900k CPU and 64 GB of memory running NVIDIA driver 595.58.03. Kernel benchmarks were collected using Julia 1.12.6 with CUDA.jl version 5.11.0, and PyKeOps data was collected using Python 3.12.0, PyKeOps v2.3, and PyTorch 2.9.1 packaged with CUDA 12.6.

We generated synthetic data for each kernel of the specified size, then obtained the median and standard error kernel latency using the BenchmarkTools.jl package.

For PyKeOps, we instantiated each matrix/vector using PyTorch [34] CUDA utilities. Kernel operations were performed using the `Genred` function with the "GPU" backend. A sample execution time was measured using the `timeit` Python package with 500 samples. 50 samples for each size were collected, from which the median and standard error were computed. Kernel compilation was performed in the setup phase, and therefore did not contribute to the runtime.

F Review of PDMP Guarantees

In this section, we recall the convergence guarantees of [25] for PDMP. We first provide the formal statement of Theorem 3.3 and provide a translation of our notation to that of [25] to ease comparison.

Instead of maximizing over $[-1,1]^m$, the authors in [25] dualize the ℓ_1 norm of $x \in \mathbb{R}^m$ using the identity

$$\|x\|_1 = \max_{\mu \in \{\Delta^2\}^m} \left\langle x, \mu^+ - \mu^- \right\rangle, \tag{77}$$

where $\{\Delta^2\}^m$ is the set of $m \times 2$ matrices where each row lies in the 2-dimensional simplex. We then use the notation $\mu = (\mu^+, \mu^-)$ with $\mu^+, \mu^- \in [0,1]^m$ and so $\mu_j^+ + \mu_j^- = 1$ for all $j \in \{1 \dots m\}$. It is simple to verify that $\mu_j^+ - \mu_j^- \in [-1,1]$, hence we have the one-to-one translation

$$\theta(\mu)_j := \mu_j^+ - \mu_j^- = \theta_j \tag{78}$$

for some $\theta \in [-1,1]^m$. Since $\mu_j^+ + \mu_j^- = 1$, we equivalently have

$$\mu_j^+ = \frac{1+\theta(\mu)_j}{2}, \quad \mu_j^- = \frac{1-\theta(\mu)_j}{2}. \tag{79}$$

Using the simplex dual notation, the authors of [25] target the strongly-convex strongly-concave saddle-point problem

$$\min_{p \in \{\Delta^m\}^n} \max_{\mu \in \{\Delta^2\}^m} \Big\{ \widetilde{K}^\eta(\mathcal{D}_r p, \theta) := \frac{1}{2} \langle C, \mathcal{D}_r p \rangle - \eta H_r(p) \\ - \eta_\mu \widetilde{H}_d^\alpha(\mu) + \|C\|_\infty \left\langle \mu^+ - \mu^-, \mathbf{c}_r(p) - c \right\rangle \Big\}, \tag{80}$$

where the authors also rescale by $1/2$ and we define the dual entropy

$$H_d^\alpha(\mu) = \sum_{j=1}^m c_j^\alpha (\mu_j^+ \ln \mu_j^+ + \mu_j^- \ln \mu_j^-),$$

which is $\min_j c_j^\alpha$-strongly convex (hence $-H_d^\alpha(\mu)$ is strongly concave) and can easily be shown to satisfy $H_d^\alpha(\mu) = H_c^\alpha(\theta(\mu))$ using (79). Using the μ notation, Algorithm 5 states PDMP more closely aligned with [25]. One can verify that the clipping step in (81) is equivalent to the simplified, tanh-based clipping step in (11).

We now state formal version of Theorem 3.3, which is the primary result of [25]. Note that our parameters are slightly different from those given in [25, Equation 2.19] to account for the difference in formulation. The authors in [25] pre-divide (80) by $2\|C\|_\infty$ and assume for their analysis that $\|C\|_\infty = 1$. Accordingly, the ε used to set parameters below scaled by $1/\|C\|_\infty$ and the primal/dual stepsizes τ_1/τ_2 are scaled by $1/(2\|C\|_\infty)$ compared with the statement in [25].

Theorem F.1 ([25, Theorem 2.2])**.** *Given $\varepsilon \in (0, \|C\|_\infty)$, set*

$$\beta = C_1 \log \frac{\|C\|_\infty m}{\varepsilon}, \quad \hat{\eta} = \frac{C_2^2 \varepsilon}{\sqrt{\beta}\|C\|_\infty \log m}, \quad \tau_1 = \frac{C_2}{2\|C\|_\infty \sqrt{\beta}}, \quad \tau_2 = \frac{15 C_2 \sqrt{\beta}}{2\|C\|_\infty}, \quad \alpha = C_3, \\ \eta = \frac{2\hat{\eta}}{\tau_1}, \quad \eta_\mu = \frac{2\hat{\eta}}{\tau_2}, \tag{82}$$

where $C_1 > 0$, $C_2 > 0$, and $0 < C_3 \le 1$ are some universal constants such that C_1 and $C_2\sqrt{C_1}$ are sufficiently large and C_2, C_3, and C_2^2/C_3 are sufficiently small, the number of iterations for Algorithm 5 to obtain an ε-solution to (1) is at most

$$T = \mathcal{O}\left(\frac{1}{\hat{\eta}} \log \frac{m\|C\|_\infty}{\varepsilon}\right) = \mathcal{O}\left(\frac{\|C\|_\infty \log m}{\varepsilon} \log \frac{m\|C\|_\infty}{\varepsilon}\right).$$

We refer interested readers to [25] for the full proof of Theorem F.1, which is quite technical and beyond the scope of this work.

Algorithm 5 Primal-Dual Mirror Prox

Require: $C \in \mathbb{R}_+^{n\times m}$, $r \in \Delta^n$, $c \in \Delta^m$, $\alpha > 0$, $\tau_1 > 0$, $\tau_2 > 0$, $\eta \geq 0$, $\eta_\mu \geq 0$, set $p^0 = (1/m)^{n\times m}$, $\mu^0 = (1/2)^{m\times 2}$, $c^\alpha = c + \alpha m^{-1}\mathbf{1}_m$.

for $t \geq 0$ **do**

Step 1) Compute the midpoints

$$\bar{p}^{t+1} = \underset{p\in\{\Delta^m\}^n}{\operatorname{argmin}} \left\{ \left\langle \nabla_p \widetilde{K}^\eta(p^t, \mu^t), p \right\rangle + \frac{1}{\tau_1} \mathrm{D}_{\mathrm{KL}}(\mathcal{D}_r p \| \mathcal{D}_r p^t) \right\},$$

$$\bar{\mu}^{t+1} = \underset{\mu\in\{\Delta^2\}^m}{\operatorname{argmax}} \left\{ \left\langle \nabla_\mu \widetilde{K}^\eta(p^t, \mu^t), \mu \right\rangle - \frac{1}{\tau_2} \mathrm{D}_{H_d^\alpha(\mu)}(\mu \| \mu^t) \right\}.$$

Step 2) Compute the main sequence

$$p^{t+1} = \underset{p\in\{\Delta^m\}^n}{\operatorname{argmin}} \left\{ \left\langle \nabla_p \widetilde{K}^\eta(p^t, \bar{\mu}^{t+1}), p \right\rangle + \frac{1}{\tau_1} \mathrm{D}_{\mathrm{KL}}(\mathcal{D}_r p \| \mathcal{D}_r p^t) \right\},$$

$$\hat{\mu}^{t+1} = \underset{\mu\in\{\Delta^2\}^m}{\operatorname{argmax}} \left\{ \left\langle \nabla_\mu \widetilde{K}^\eta(\bar{p}^{t+1}, \mu^t), \mu \right\rangle - \frac{1}{\tau_2} \mathrm{D}_{H_d^\alpha(\mu)}(\mu \| \mu^t) \right\}.$$

Step 3) Clip the dual variables

$$\mu^{t+1} = \mathrm{clip}\left(\hat{\mu}^{t+1}, \left[\frac{1}{1+e^\beta}, \frac{e^\beta}{1+e^\beta} \right] \right). \tag{81}$$

end for