Improved Rate of First Order Algorithms for Entropic Optimal Transport

Yiling Luo
Georgia Institute of Technology

Yiling Xie
Georgia Institute of Technology

Xiaoming Huo
Georgia Institute of Technology

Abstract

This paper improves the state-of-the-art rate of a first-order algorithm for solving entropy regularized optimal transport. The resulting rate for approximating the optimal transport (OT) has been improved from $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ to $\widetilde{\mathcal{O}}(n^2/\epsilon)$, where n is the problem size and ϵ is the accuracy level. In particular, we propose an accelerated primal-dual stochastic mirror descent algorithm with variance reduction. Such special design helps us improve the rate compared to other accelerated primal-dual algorithms. We further propose a batch version of our stochastic algorithm, which improves the computational performance through parallel computing. To compare, we prove that the computational complexity of the Stochastic Sinkhorn algorithm is $\widetilde{\mathcal{O}}(n^2/\epsilon^2)$, which is slower than our accelerated primal-dual stochastic mirror algorithm. Experiments are done using synthetic and real data, and the results match our theoretical rates. Our algorithm may inspire more research to develop accelerated primal-dual algorithms that have rate $\widetilde{\mathcal{O}}(n^2/\epsilon)$ for solving OT.

1 INTRODUCTION

The *Optimal Transport* (OT) (Monge, 1781; Kantorovich, 1942; Villani, 2009) is an optimization problem that has been actively studied. In this section, we review the OT problem. In Section 1.1, we review the OT formulation and its related concepts. In Section 1.2, we survey the existing algorithms for solving OT and summarize our contribution given the literature background.

1.1 Optimal Transport

We review the definition of OT. Given a cost matrix $C \in \mathbb{R}_+^{n\times n}$ and two vectors $\boldsymbol{p}, \boldsymbol{q} \in \Delta_n$, where $\Delta_n := \{\boldsymbol{a} \in \mathbb{R}_+^n : \boldsymbol{a}^T\mathbf{1} = 1\}$ is the standard simplex, OT is defined as follows:

$$\min_{X\in\mathcal{U}(\boldsymbol{p},\boldsymbol{q})} \langle C, X\rangle, \tag{1}$$

where $\mathcal{U}(\boldsymbol{p},\boldsymbol{q}) := \left\{X \in \mathbb{R}_+^{n\times n} \,\middle|\, X\mathbf{1} = \boldsymbol{p}, X^T\mathbf{1} = \boldsymbol{q}\right\}$, and $\langle C, X\rangle := \sum_{i,j=1}^n C_{i,j}X_{i,j}$.

The ϵ*-solution* is always used when evaluating algorithm efficiency for solving OT, so we review its definition as follows. Denote the optimal solution of problem (1) as X^*, an $\epsilon-$solution $\widehat{X}$ is such that:

$$\begin{aligned} &\widehat{X} \in \mathcal{U}(\boldsymbol{p},\boldsymbol{q});\\ &\langle C, \widehat{X}\rangle \le \langle C, X^*\rangle + \epsilon. \end{aligned}$$

Note that for a stochastic algorithm, the second condition is replaced by $\mathbb{E}\langle C, \widehat{X}\rangle \le \langle C, X^*\rangle + \epsilon$.

Our paper adopts a two-step approach (Altschuler et al., 2017) for finding an ϵ-solution to problem (1). In the first step, one finds an approximate solution $\widetilde{X}$ to the *entropic OT* problem (2).

$$\min_{X\in\mathcal{U}(\boldsymbol{p}',\boldsymbol{q}')} \langle C, X\rangle - \eta H(X), \tag{2}$$

where $H(X) = -\sum_{i,j} X_{i,j}\log(X_{i,j})$ is the entropy, $\eta = \frac{\epsilon}{4\log(n)}$, and $\begin{pmatrix}\boldsymbol{p}'\\ \boldsymbol{q}'\end{pmatrix} = \left(1 - \frac{\epsilon}{64\|C\|_\infty}\right)\begin{pmatrix}\boldsymbol{p}\\ \boldsymbol{q}\end{pmatrix} + \frac{\epsilon}{64n\|C\|_\infty}\begin{pmatrix}\mathbf{1}_n\\ \mathbf{1}_n\end{pmatrix}$. In the second step, one rounds $\widetilde{X}$ to the original feasible region $\mathcal{U}(\boldsymbol{p},\boldsymbol{q})$. Once a certain accuracy level is achieved in the first step, Altschuler et al. (2017) guarantees the final solution to be an ϵ-solution to problem (1).

1.2 Literature Review

We review the state-of-the-art algorithms that solve OT and summarize their computational complexity (measured by the number of numerical operations) for giving an ϵ-solution to OT in Table 1. We list the year of the relevant publication, the names of the methods, the computational complexities, and whether (a $\surd$ sign) or not (an $\times$ sign) the method solves entropic OT as an intermediate step for approximating OT in columns. In particular, the computational complexities

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

Table 1: Order of Complexity of OT Algorithms.

Year	Algorithm	Order of Complexity	Solves Entropic OT
2013	Sinkhorn (Cuturi, 2013)	n^2/ϵ^2 (Dvurechensky et al., 2018)	√
2017	Greenkhorn (Altschuler et al., 2017)	n^2/ϵ^3 (Altschuler et al., 2017); n^2/ϵ^2 (Lin et al., 2019)	√
2018	Stochastic Sinkhorn (Abid and Gower, 2018)	n^2/ϵ^3; n^2/ϵ^2 (This paper)	√
2018	APDAGD (Dvurechensky et al., 2018)	$n^{2.5}/\epsilon$	√
2018	Packing LP (Blanchet et al., 2018; Quanrud, 2018)	n^2/ϵ	×
2018	Box Constrained Newton (Blanchet et al., 2018)	n^2/ϵ	√
2019	APDAMD (Lin et al., 2019)	$n^{2.5}/\epsilon$	√
2019	Dual Extrapolation (Jambulapati et al., 2019)	n^2/ϵ	×
2019	Accelerated Sinkhorn (Lin et al., 2022)	$n^{7/3}/\epsilon^{4/3}$	√
2019	Dijkstra's search + DFS (Lahn et al., 2019)	$n^2/\epsilon + n/\epsilon^2$	×
2020	APDRCD (Guo et al., 2020)	$n^{2.5}/\epsilon$	√
2021	AAM (Guminov et al., 2021)	$n^{2.5}/\epsilon$	√
2022	Hybrid Primal-Dual (Chambolle and Contreras, 2022)	$n^{2.5}/\epsilon$	√
2022	PDASGD (Xie et al., 2022)	$n^{2.5}/\epsilon$	√
2022	PDASMD	$\boldsymbol{n^2/\epsilon}$ (This paper)	√

are shown in their order of n and ϵ, where the $\log(n)$ term is omitted. The mark of "(This Paper)" indicates a rate derived in this paper. It is clear that among the methods that solve entropic OT, our PDASMD algorithm achieves the lowest rate.

There are four main techniques to solve problem (2) in current literature:

- The first technique solves the dual problem of problem (2) by the Bregman projection technique. Specifically, this technique partitions the dual variables into blocks and iteratively updates each block. Algorithms that use this technique include Sinkhorn algorithm (Cuturi, 2013), Greenkhorn algorithm (Altschuler et al., 2017) and Stochastic Sinkhorn algorithm (Abid and Gower, 2018).
- The second technique also solves the dual problem of problem (2) but uses accelerated first-order methods. Algorithms that use this technique include accelerated gradient descent (APDAGD) (Dvurechensky et al., 2018), accelerated mirror descent (APDAMD) (Lin et al., 2019), accelerated alternating minimization (AAM) (Guminov et al., 2021), accelerated randomized coordinate descent (APDRCD) (Guo et al., 2020) and accelerated stochastic gradient descent (PDASGD) (Xie et al., 2022). This technique can also be combined with the first technique. See, for example, the accelerated Sinkhorn algorithm in Lin et al. (2022).
- The third technique solves the dual problem of problem (2) by second-order algorithms. An instance that uses this technique is the box-constrained Newton algorithm (Blanchet et al., 2018).
- The fourth technique minimizes the primal-dual gap of problem (2). An instance that uses this technique is the hybrid primal-dual algorithm (Chambolle and Contreras, 2022).

Besides works that use the two-step approach to solve the entropic OT first, some works directly solve the unpenalized OT problem (1) by linear programming (Blanchet et al., 2018; Quanrud, 2018), dual-extrapolation (Jambulapati et al., 2019), or graph-based search algorithm (Lahn et al., 2019).

We compare the computational complexity in Table 1 of our algorithm with other state-of-the-art algorithms as follows.

First, our PDASMD algorithm belongs to the second class of algorithms to solve the entropic OT problem (2). All other algorithms in this class reported a rate of $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ for approximating OT, while our algorithm has a better rate of $\widetilde{\mathcal{O}}(n^2/\epsilon)$. Thus our algorithm improves the rate for this class. The advantage of our algorithm mainly comes from the special technique that we use: though all the algorithms in this class use the acceleration technique, no accelerated variance reduction version of stochastic mirror descent has been tried in the previous algorithms. We apply those techniques to entropic OT and find that they lead to a better theoretical rate.

Second, among all algorithms for solving entropic OT, our PDASMD algorithm still reports the best rate. There is only one algorithm on entropic OT that achieved the same rate: the box-constrained Newton algorithm. However, we note that the Newton algorithm is a second-order algorithm, which requires computing the Hessian of the objective function. By its second-order nature, each step of the Newton algorithm will be expensive in terms of computation and memory. On the other hand, our PDASMD algorithm is based on mirror descent, which is a first-order algorithm. Our PDASMD algorithm is thus easier to implement.

Finally, the algorithms that directly solve the original OT problem also report the same optimal rate as our PDASMD algorithm, including the packing LP algorithm, the dual extrapolation algorithm, and the graph-based Dijkstra DFS algorithm (when $\epsilon \gtrsim 1/n$). Compared with those algorithms, we have the extra advantage that our algorithm can not only approximate the OT problem but also solve the entropic OT. Thus, when one wants to solve the entropic OT, our algorithm is still preferred.

Our Contribution We summarize two main contributions in this work as follows.

- We propose an accelerated primal-dual stochastic algorithm that has computational complexity $\widetilde{\mathcal{O}}(n^2/\epsilon)$ for solving OT. Every step of our algorithm is defined by simple arithmetic operations and is counted in the complexity calculation. Thus our algorithm is practical. Moreover, compared with other algorithms that achieve the same rate for solving OT: our algorithm has the extra advantage that it can also be applied to entropic OT; it is a first-order algorithm, so it can be easily implemented without computing the Hessian. We also propose a batch version of our algorithm to increase the computational power.
- We prove that the computational complexity of the Stochastic Sinkhorn algorithm is $\widetilde{\mathcal{O}}(n^2/\epsilon^2)$, instead of the $\widetilde{\mathcal{O}}(n^2/\epsilon^3)$ rate in the literature. Our proved rate for Stochastic Sinkhorn matches the state-of-the-art rate of Sinkhorn and Greenkhorn. Moreover, the provable rate by our accelerated primal-dual stochastic algorithm is better than that of the Stochastic Sinkhorn, which again illustrates the advantage of our algorithm.

Paper Organization The rest of the paper is organized as follows. In Section 2, we present our main algorithm of Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD), show its convergence, and analyze its complexity for solving OT; as a comparison, we also prove the rate of Stochastic Sinkhorn, which is improved over the existing result. In Section 3, we develop a batch version of PDASMD and show its convergence and computational complexity. In Section 4, we run numerical examples to support our theorems. In Section 5, we discuss the findings in this work and some future research.

2 PRIMAL-DUAL ACCELERATED STOCHASTIC PROXIMAL MIRROR DESCENT (PDASMD)

In this section, we present our PDASMD algorithm for solving a linear constrained convex problem, which includes the entropic OT as a special case. We analyze the convergence rate of the PDASMD algorithm, then apply it to OT and derive the computational complexity. As a comparison, we also analyze the computational complexity of the Stochastic Sinkhorn. Since our algorithm uses the Proximal Mirror Descent technique, we review the background of such a technique in Appendix A and briefly explain why it is suitable for entropic OT.

2.1 Definition and Notation

We first introduce some notations that we will use throughout the rest of this paper.

Notations: For a vector $\boldsymbol{a}$: let $sign(\boldsymbol{a})$ be such that $(sign(\boldsymbol{a}))_i = 1$ if $a_i > 0$ and -1 otherwise. Let $\mathbf{1}_n$ be the n-dimensional vector where each element is 1. For matrices $X \in \mathbb{R}^{n\times o}, Y \in \mathbb{R}^{p\times q}$: let $X \otimes Y$ denote the standard Kronecker product; let $\exp(X)$ and $\log(X)$ be the elementwise exponential and logarithm of X; let $\|X\|_2$ be the operator norm of X and $\|X\|_\infty$ be $\max_{i,j}|X_{i,j}|$; denote the matrix norm induced by two arbitrary vector norms $\|\cdot\|_H$ and $\|\cdot\|_E$ as $\|X\|_{E\to H} := \max_{\boldsymbol{a}:\|\boldsymbol{a}\|_E\le 1}\|X\boldsymbol{a}\|_H$; denote the vectorization of X as $\text{Vec}(X) = (X_{11},...,X_{n1},X_{12},...,X_{n2},...,X_{1o},...,X_{no})^T$. For two non-negative real values $s(\kappa)$ and $t(\kappa)$, denote $s(\kappa) = \Theta(t(\kappa))$ if $\exists k > 0$ and $K > 0$ such that $kt(\kappa) \le s(\kappa) \le Kt(\kappa)$; denote $s(\kappa) = \mathcal{O}(t(\kappa))$ if $\exists K > 0$ such that $s(\kappa) \le Kt(\kappa)$; denote $s(\kappa) = \widetilde{\mathcal{O}}(t(\kappa))$ to indicate the previous inequality where K depends on some logarithmic function of κ.

Next, we review some key definitions that will be useful. [1]

Definition 1 (Strong convexity). *$f : \mathcal{Q} \to \mathbb{R}$ is α-strongly convex w.r.t. $\|\cdot\|_H$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Q}$:*

$$f(\boldsymbol{y}) \ge f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{\alpha}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_H^2.$$

Definition 2 (Smoothness). *A convex function $f : \mathcal{Q} \to \mathbb{R}$ is β-smooth w.r.t. $\|\cdot\|_H$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Q}$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_{H,*} \le \beta\|\boldsymbol{x}-\boldsymbol{y}\|_H,$$

where $\|\boldsymbol{u}\|_{H,*} := \max_{\boldsymbol{v}}\{\langle \boldsymbol{u}, \boldsymbol{v}\rangle : \|\boldsymbol{v}\|_H \le 1\}$ is the dual norm of $\|\cdot\|_H$. Or equivalently,

$$f(\boldsymbol{y}) \le f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{\beta}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_H^2.$$

[1] Our definitions follow those in Allen-Zhu (2017).

Definition 3 (Bregman divergence)**.** *For a mirror function $w(\cdot)$ that is 1-strongly convex w.r.t. $\|\cdot\|_H$, we denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the Bregman divergence w.r.t. $\|\cdot\|_H$ generated by $w(\cdot)$, where*

$$V_{\boldsymbol{x}}(\boldsymbol{y}) := w(\boldsymbol{y}) - w(\boldsymbol{x}) - \langle \nabla w(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle.$$

One can conclude from the definition that

$$V_{\boldsymbol{x}}(\boldsymbol{y}) \geq \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

If we further assume that the mirror function $w(\cdot)$ is γ-smooth w.r.t. $\|\cdot\|_H$, we then have

$$V_{\boldsymbol{x}}(\boldsymbol{y}) \leq \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

2.2 General Formulation and PDASMD Algorithm

In this section, we first state a general linear constrained problem and explain how it includes entropic OT as a special case. We then propose our algorithm to solve this general problem. Finally, we show the convergence rate of our algorithm.

We consider a linear constrained problem as follows:

$$\min_{\boldsymbol{x} \in \mathbb{R}^m} f(\boldsymbol{x}) \qquad s.t. \; A\boldsymbol{x} = \boldsymbol{b} \in \mathbb{R}^l, \tag{3}$$

where f is strongly convex. One observes that the entropic OT (2) is a special case of problem (3) with $\boldsymbol{x} = \mathrm{Vec}(X)$, $f(\boldsymbol{x}) = \langle \mathrm{Vec}(C), \boldsymbol{x} \rangle + \eta \sum_{i=0}^{n-1} \sum_{j=1}^{n} x_{in+j} \log(x_{in+j})$, $\boldsymbol{b} = (\boldsymbol{p}^T, \boldsymbol{q}^T)^T$, $A = \begin{bmatrix} \mathbf{1}^T \otimes I_n \\ I_n \otimes \mathbf{1}^T \end{bmatrix}$.

A standard approach for solving the constrained problem (3) is to optimize its Lagrange dual problem (4):

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \{\phi(\boldsymbol{\lambda}) :=& \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle + \max_{\boldsymbol{x}} (-f(\boldsymbol{x}) - \langle A^T \boldsymbol{\lambda}, \boldsymbol{x} \rangle) \\ =& \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle - f(\boldsymbol{x}(\boldsymbol{\lambda})) - \langle A^T \boldsymbol{\lambda}, \boldsymbol{x}(\boldsymbol{\lambda}) \rangle\}, \end{aligned} \tag{4}$$

where by first-order condition $\boldsymbol{x}(\boldsymbol{\lambda})$ is such that

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\boldsymbol{\lambda})) = -A^T \boldsymbol{\lambda}. \tag{5}$$

Since problem (3) is a linear constrained convex problem, the strong duality holds. Thus solving problem (3) is equivalent to solving its dual problem (4). In particular, we develop a stochastic algorithm for the case that the dual is of finite sum form. We further assume that all terms in the finite sum are smooth for convergence analysis. The conditions on the dual are formalized as follows:

Assumption 1 (Finite-sum dual)**.** *Assume that the dual can be written as $\phi(\boldsymbol{\lambda}) = \frac{1}{m}\sum_{i=1}^m \phi_i(\boldsymbol{\lambda})$, where ϕ_i is convex and L_i-Lipchitz smooth w.r.t. an arbitrary $\|\cdot\|_H$ norm.*

Note that the assumption on the dual is reasonable and can be satisfied by some problems, including entropic OT. We now give a concrete example that the assumption holds. Consider a primal objective $f(\boldsymbol{x}) = \sum_{i=1}^m f_i(x_i)$ where each f_i is $\nu-$strongly convex w.r.t. another arbitrary norm $\|\cdot\|_E$ (note that it can be different from the $\|\cdot\|_H$ norm). In this case, we can solve the primal-dual relationship in equation (5) to get:

$$x_i(\boldsymbol{\lambda}) = (\nabla f_i)^{-1}(-\boldsymbol{a}_i^T \boldsymbol{\lambda}), i = 1, \dots, m,$$

where $\boldsymbol{a}_i$ is the ith column of A. As a consequence, the dual problem (4) can be written as a finite sum:

$$\begin{aligned} \phi(\boldsymbol{\lambda}) &= \frac{1}{m}\sum_{i=1}^m (\langle \boldsymbol{\lambda}, \boldsymbol{b}_i \rangle - m f_i(x_i(\boldsymbol{\lambda})) - m \boldsymbol{a}_i^T \boldsymbol{\lambda} x_i(\boldsymbol{\lambda})) \\ &:= \frac{1}{m}\sum_{i=1}^m \phi_i(\boldsymbol{\lambda}), \end{aligned}$$

where $\boldsymbol{b}_i$'s are arbitrarily chosen vectors satisfying the constraint $\sum_{i=1}^m \boldsymbol{b}_i = m\boldsymbol{b}$. One can check that $\nabla \phi_i(\boldsymbol{\lambda}) = \boldsymbol{b}_i - m x_i(\boldsymbol{\lambda}) \boldsymbol{a}_i$. By Nesterov (2005), ϕ_i is convex and L_i-Lipchitz smooth w.r.t. $\|\cdot\|_H$ norm, where $L_i \leq \frac{m}{\nu}\|\boldsymbol{a}_i\|_{E \to H,*}$.

With the finite sum representation of ϕ, we propose a PDASMD algorithm (Algorithm 1) to solve problem (3). We add a few remarks to explain the algorithm as follows.

Remark 1. *One should choose a specific $\|\cdot\|_H$ norm and a mirror function $w(\cdot)$ to run the algorithm. Those choices have a direct impact on the mirror descent step 10 and proximal gradient descent step 11: if we let $\|\cdot\|_H = \|\cdot\|_2$ and $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, both steps reduce to stochastic gradient descent, then the algorithm essentially reduces to the PDASGD algorithm in Xie et al. (2022).*

Remark 2. *The primal variables $\boldsymbol{x}$'s in Algorithm 1 are updated by Steps 14 through 16, and we explain those steps as follows: The iterates in Steps 14 through 16 essentially leads to $\boldsymbol{x}^{S-1} = \left(\sum_{s=0}^{S-1} \boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)/\tau_{1,s}\right) \Big/ \left(\sum_{s=0}^{S-1} (1/\tau_{1,s})\right)$. We express such updates in $\boldsymbol{x}^s$ in an iterative way to avoid storing all updates of $\widetilde{\boldsymbol{y}}_s$'s. In this way, our algorithm is memory efficient.*

Remark 3. *The dual variables $\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{y}$'s are updated by Steps 2 through 13. The update consists of outer loops indexed by s and inner loops indexed by j, which uses the variance reduction and acceleration technique in Allen-Zhu (2017) (Algorithm 5 in that paper). We now summarize the variance reduction and acceleration technique for a better understanding of our algorithm.*

The ***variance reduction*** *in Algorithm 1 is step 9, which works as follows: For the finite-sum dual $\phi(v) = \frac{1}{m}\sum_{i=1}^m \phi_i(v)$, a stochastic algorithm without variance reduction updates the parameter estimation using $\nabla \phi_i(v)$, which in general has $Var[\nabla \phi_i(v)] \neq 0, \forall v$ and thus needs the step size $\to 0$ for convergence. A variance reduced algorithm replaces $\nabla \phi_i(v)$ by $A_k = \nabla \phi_i(v) -$*

Algorithm 1: Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD)

```
1: Initialize $l$ the number of inner iterations; $\tau_2 = \frac{1}{2}$,
   $\boldsymbol{y}_0 = \boldsymbol{z}_0 = \widetilde{\boldsymbol{v}}_0 = \boldsymbol{v}_0 = \boldsymbol{0}$, $C_0 = D_0 = 0$; choose a
   mirror function $w(\cdot)$ that is 1-strongly convex and
   $\gamma$-smooth w.r.t. $\|\cdot\|_H$, and denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the
   Bregman divergence generated by $w(\cdot)$; take
   $\bar{L} = (\sum_{i=1}^{m} L_i)/m$, where $L_i$ is the smoothness (w.r.t.
   $\|\cdot\|_H$) for each component $\phi_i$ of the dual function $\phi(\cdot)$
   in Assumption 1.
2: for s = 0,...,S-1 do
3:     $\tau_{1,s} \leftarrow 2/(s+4)$; $\alpha_s \leftarrow 1/(9\tau_{1,s}\bar{L})$;
4:     $\boldsymbol{\mu}^s \leftarrow \nabla\phi(\widetilde{\boldsymbol{v}}^s)$.
5:     for $j = 0$ to $l-1$ do
6:         $k \leftarrow (sl) + j$;
7:         $\boldsymbol{v}_{k+1} \leftarrow \tau_{1,s}\boldsymbol{z}_k + \tau_2\widetilde{\boldsymbol{v}}^s + (1-\tau_{1,s}-\tau_2)\boldsymbol{y}_k$;
8:         Pick i randomly from $\{1, 2, \ldots, m\}$, each with
           probability $p_i := L_i/m\bar{L}$;
9:         $\widetilde{\boldsymbol{\nabla}}_{k+1} \leftarrow \boldsymbol{\mu}^s + \frac{1}{mp_i}(\nabla\phi_i(\boldsymbol{v}_{k+1}) - \nabla\phi_i(\widetilde{\boldsymbol{v}}^s))$;
10:        $\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}}\{\frac{1}{\alpha_s}V_{\boldsymbol{z}_k}(\boldsymbol{z}) + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{z}\rangle\}$;
11:        $\boldsymbol{y}_{k+1} =$
           $\arg\min_{\boldsymbol{y}}\{\frac{9\bar{L}}{2}\|\boldsymbol{y} - \boldsymbol{v}_{k+1}\|_H^2 + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{y}\rangle\}$.
12:    end for
13:    $\widetilde{\boldsymbol{v}}^{s+1} \leftarrow \frac{1}{l}\sum_{j=1}^{l}\boldsymbol{y}_{sl+j}$;
14:    $C_s \leftarrow C_s + \frac{1}{\tau_{1,s}}$;
15:    Pick $\widetilde{\boldsymbol{y}}_s$ uniform randomly from $\{\boldsymbol{y}_{sl+j}\}_{j=1}^{l}$, update
       $D_s \leftarrow D_s + \frac{1}{\tau_{1,s}}\boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)$, where $\boldsymbol{x}(\cdot)$ is given by
       equation (5);
16:    $\boldsymbol{x}^s = D_s/C_s$.
17: end for
18: Output: $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{S-1}$.
```

$B_k + \mathbb{E}[B_k]$. When B_t and $\nabla\phi_i(v)$ have correlation $r > 0.5$ and $Var[B_t] \approx Var[\nabla\phi_i(v)]$, one can check that $Var[A_t] = Var[\nabla\phi_i(v) - B_k] = Var[\nabla\phi_i(v)] - 2r\sqrt{Var[\nabla\phi_i(v)]Var[B_k]} + Var[B_k] < Var[\nabla\phi_i(v)]$ (so the variance is reduced). Step 9 in Algorithm 1 uses this variance reduction technique by taking $B_k = \nabla\phi_i(\widetilde{v}^s)$.

*The **acceleration** in Algorithm 1 are steps 7, 10, 11, namely the Katyusha acceleration in Allen-Zhu (2017). We summarize this technique and compare it with a classical method in Allen-Zhu and Orecchia (2014) that uses Nesterov's momentum. To simplify explanation, consider the special case $\|\cdot\|_H = \|\cdot\|_2, w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, steps 7, 10, 11 of Algorithm 1 are:*

$v_{k+1} = \tau_1 z_k + \tau_2\widetilde{v} + (1-\tau_1-\tau_2)y_k; y_{k+1} = v_{k+1} - \frac{1}{3L}\widetilde{\nabla}_{k+1}; z_{k+1} = z_k - \alpha\widetilde{\nabla}_{k+1}$,

where $\mathbb{E}\widetilde{\nabla}_{k+1} = \nabla\phi(v_{k+1})$. On the other hand, the method in Allen-Zhu and Orecchia (2014) updates as

$v_{k+1} = \tau_1 z_k + (1-\tau_1)y_k; y_{k+1} = v_{k+1} - \frac{1}{L}\nabla\phi(v_{k+1}); z_{k+1} = z_k - \alpha\nabla\phi(v_{k+1})$.

The two updating schemes both have a "gradient descent" step in y_{k+1} and "momentum" term z_{k+1} that accumulates the gradient history; the difference is in v_{k+1}: the classical method takes a weighted average of z_k and y_k (that is, Nesterov's momentum), while Katyusha acceleration has one more term $\widetilde{v}$ (which is called Katyusha momentum (Allen-Zhu, 2017)). Such Katyusha momentum serves as a "magnet" to retract the estimation to $\widetilde{v}$, which is the average of past l estimates. Since our algorithm is a stochastic algorithm, such a "magnet" helps the algorithm to stabilize. Thus, the Katyusha acceleration works well.

We prove the convergence rate of the PDASMD algorithm as follows:

Theorem 1. *Under Assumption 1, we apply Algorithm 1 to solve problem* (3). *Choose a mirror function $w(\cdot)$ that is* 1-*strongly convex and γ-smooth w.r.t. $\|\cdot\|_H$ norm. Denote the primal and dual optimal solution as $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$, respectively. Assume that $\|\lambda^*\|_H \le R$. We have the convergence of the algorithm as follows:*

$$\|\mathbb{E}[\boldsymbol{b} - A\boldsymbol{x}^{S-1}]\|_{H,*} \le \frac{2}{S^2 l}\left[l\bar{L}R + 18\bar{L}R\gamma\right], \tag{6}$$

$$f(\mathbb{E}(\boldsymbol{x}^{S-1})) - f(\boldsymbol{x}^*) \le \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right]. \tag{7}$$

The proof of the theorem is deferred to Appendix B.

2.3 Applying to Optimal Transport

In this section, we give the detailed procedure of applying PDASMD to get an approximation solution to the OT. Especially, we consider two cases: in the first case, we use $\|\cdot\|_H = \|\cdot\|_2$ and PDASMD reduce to PDASGD; in the second case, we use $\|\cdot\|_H = \|\cdot\|_\infty$ and prove an improved computational complexity over the first case. For the latter case, our algorithm achieves the best possible rate in the current literature. Our algorithm improves the rate of the first-order algorithms for solving entropic OT.

We apply the PDASMD algorithm to solve the entropic OT (2) as follows. Since problem (2) a special case of problem (3), we plug $A, \boldsymbol{b}, f(\cdot)$ into the general dual formula (4) to get the dual problem of problem (2). With a little abuse of notation, we split the dual variables as $(\boldsymbol{\tau}^T, \boldsymbol{\lambda}^T)^T$ for $\boldsymbol{\tau}, \boldsymbol{\lambda} \in \mathbb{R}^n$. The dual problem of problem (2) is:

$$\phi(\boldsymbol{\tau}, \boldsymbol{\lambda}) = \eta\langle\boldsymbol{1}_{n^2}, \boldsymbol{x}(\boldsymbol{\tau}, \boldsymbol{\lambda})\rangle - \langle\boldsymbol{p}', \boldsymbol{\tau}\rangle - \langle\boldsymbol{q}', \boldsymbol{\lambda}\rangle, \tag{8}$$

where the relationship between primal-dual variables is

$$\boldsymbol{x}(\boldsymbol{\tau}, \boldsymbol{\lambda}) = \exp\left(\frac{A^T(\boldsymbol{\tau}^T, \boldsymbol{\lambda}^T)^T - \mathrm{Vec}(C) - \eta\boldsymbol{1}_{n^2}}{\eta}\right). \tag{9}$$

Moreover, to get a dual with the finite-sum structure, we follow Genevay et al. (2016) to transfer the dual objective to semi-dual by fixing $\boldsymbol{\lambda}$ and solving the first-order condition w.r.t. $\boldsymbol{\tau}$ in objective (8). This gives us the relationship between the dual variables:

$$\tau_i(\boldsymbol{\lambda}) = \eta \log p_i' - \eta \log \left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta) \right).$$

Plugging the relationship above into the dual objective (8) gives us the semi-dual objective. With a little abuse of notation, we denote the semi-dual objective function as $\phi(\boldsymbol{\lambda})$, which is:

$$\begin{aligned}
\phi(\boldsymbol{\lambda}) &= -\langle \boldsymbol{q}', \boldsymbol{\lambda} \rangle - \eta \sum_{i=1}^{n} p_i' \log p_i' \\
&\quad + \eta \sum_{i=1}^{n} \log \left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta) \right) + \eta \\
&= \frac{1}{n} \sum_{i=1}^{n} n p_i' \Bigg[- \langle \boldsymbol{q}', \boldsymbol{\lambda} \rangle - \eta \log p_i' \\
&\quad + \eta \log \left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta) \right) + \eta \Bigg] \\
&:= \frac{1}{n} \sum_{i=1}^{n} \phi_i(\boldsymbol{\lambda}). \qquad (10)
\end{aligned}$$

It is easy to check that each $\phi_i(\boldsymbol{\lambda})$ is convex. To apply our algorithm, we further check the smoothness of $\phi_i(\boldsymbol{\lambda})$ in the following lemma:

Lemma 1. *$\phi_i(\cdot)$ in the semi-dual objective (10) is $\frac{np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_2$ norm, and is $\frac{5np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_\infty$ norm.*

Lemma 1 is proved in Appendix C. By Lemma 1, we can calculate the parameter in PDASMD Algorithm 1 as $\bar{L} = 1/\eta$ for $\|\cdot\|_H = \|\cdot\|_2$, and $\bar{L} = 5/\eta$ for $\|\cdot\|_H = \|\cdot\|_\infty$. For these two cases, we can apply Algorithm 1 to approximate problem (2). We further round the approximating solution of problem (2) to feasible region of problem (1). In this way, we get an $\epsilon-$solution to problem (1). The full procedure is deferred to Appendix D due to page limit. We state the computational complexity of the full procedure in the following theorem:

Theorem 2. *Set $l = \Theta(n)$ in the PDASMD algorithm, the overall number of arithmetic operations for finding a solution $\widehat{X}$ such that $\mathbb{E}\langle C, \widehat{X} \rangle \leq \langle C, X^* \rangle + \epsilon$ is*

- $\widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty (1+\sqrt{\gamma/n})}{\epsilon} \right)$ *for* $\|\cdot\|_H = \|\cdot\|_2$;
- $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty (1+\sqrt{\gamma/n})}{\epsilon} \right)$ *for* $\|\cdot\|_H = \|\cdot\|_\infty$.

The proof of Theorem 2 is in Appendix D.

Remark 4. *The complexities still depend on γ, the smoothness of $w(\cdot)$ w.r.t. $\|\cdot\|_H$. For example, when taking $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, we have $\gamma = 1$ for $\|\cdot\|_H = \|\cdot\|_2$, and $\gamma = n$ for $\|\cdot\|_H = \|\cdot\|_\infty$. The corresponding computational complexity is then $\widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty}{\epsilon} \right)$ and $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty}{\epsilon} \right)$. Now for $\|\cdot\|_H = \|\cdot\|_\infty$, as long as we choose a proper $w(\cdot)$ such that $\gamma = \mathcal{O}(n)$, the rate $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty}{\epsilon} \right)$ is achieved. One may further improve the rate by a constant by improving the dependency of γ on n. Such improvement is an open question in optimization; though we make no effort to do it in this paper, we still note this opportunity.*

Remark 5. *If we choose $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, we have closed-form solutions for each step of PDASMD.*

- *For both settings, step 10 of PDASMD algorithm becomes* $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \alpha_s \widetilde{\boldsymbol{\nabla}}_{k+1}$;
- *For* $\|\cdot\|_H = \|\cdot\|_2$, *step 11 of PDASMD is* $\boldsymbol{y}_{k+1} = \boldsymbol{v}_{k+1} - \frac{1}{9\bar{L}} \widetilde{\boldsymbol{\nabla}}_{k+1}$;
- *For* $\|\cdot\|_H = \|\cdot\|_\infty$, *step 11 of PDASMD becomes* $\boldsymbol{y}_{k+1} = \boldsymbol{v}_{k+1} - \frac{\|\widetilde{\boldsymbol{\nabla}}_{k+1}\|_1}{9\bar{L}} sign(\widetilde{\boldsymbol{\nabla}}_{k+1})$.

It is clear that in both settings, each step of PDASMD is defined by simple arithmetic operations and thus is easy to implement. There is no gap between our theory and practice.

2.4 Computational Complexity of the Stochastic Sinkhorn

In this section, we prove that the computational complexity of the Stochastic Sinkhorn for finding an ϵ-solution to OT is $\widetilde{\mathcal{O}}(\frac{n^2}{\epsilon^2})$, which is improved over the known rate of $\widetilde{\mathcal{O}}(\frac{n^2}{\epsilon^3})$ (Abid and Gower, 2018) and matches the state-of-the-art rate of Sinkhorn and Greenkhorn (Dvurechensky et al., 2018; Lin et al., 2019). Moreover, our PDASMD algorithm beats the provable rate of Stochastic Sinkhorn. This illustrates the advantage of our PDASMD algorithm.

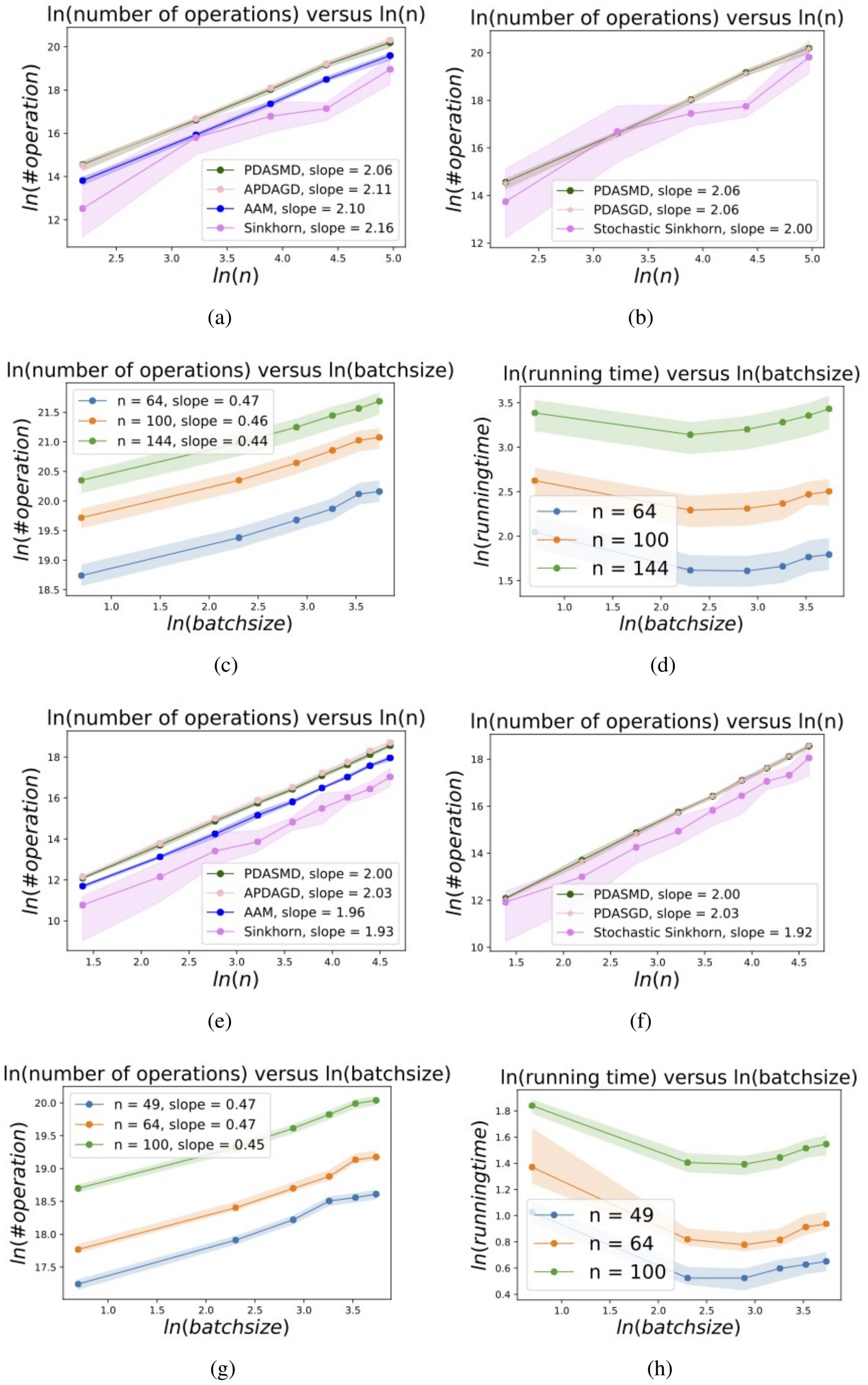
The Stochastic Sinkhorn algorithm is proposed by Abid and Gower (2018). One can check Appendix E for a full algorithm description. We show the computational complexity of Stochastic Sinkhorn as follows:

Theorem 3. *Stochastic Sinkhorn finds a solution $\widehat{X}$ such that $\mathbb{E}\langle C, \widehat{X} \rangle \leq \langle C, X^* \rangle + \epsilon$ in*

$$\mathcal{O}\left(\frac{n^2 \|C\|_\infty^2 \log n}{\epsilon^2} \right)$$

arithmetic operations.

The proof of Theorem 3 is in Appendix E.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 1: Computational complexity comparison of different algorithms for finding an ϵ-solution of OT. The logarithmic of the total number of numerical operations to achieve a given ϵ approximation error is plotted against either the logarithmic transform of the sample size n in the PDASMD algorithm (rows 1 and 3) or the batch size in the PDASMD-B algorithm (rows 2 and 4). The first two rows use synthetic data, and the last two are for the MNIST data. The relevant discussion can be seen in Section 4 Numerical Studies. The error bars in all the plots come from repeating the experiment using 5 pairs of randomly generated/chosen marginals.

3 PDASMD WITH BATCH IMPLEMENTATION (PDASMD-B)

In this section, we propose a batch version of PDASMD, namely the PDASMD-B algorithm. The batch implementation of the stochastic step in PDASMD-B allows parallel computing. This further improves the computational power of our algorithm.

Algorithm 2: Batch PDASMD (PDVRASMD-B)

```
1:  Initialize $l$ the number of inner iterations, $B$ the batch size; set $\tau_2 \leftarrow \frac{1}{2B}$, $C_0 = D_0 = 0$,
    $\boldsymbol{y}_0 = \boldsymbol{z}_0 = \widetilde{\boldsymbol{v}}_0 = \boldsymbol{v}_0 = \boldsymbol{0}$; choose a mirror function $w(\cdot)$ that is 1-strongly convex and $\gamma$-smooth w.r.t.
    $\|\cdot\|_H$, and denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the Bregman divergence generated by $w(\cdot)$; take $\bar{L} = (\sum_{i=1}^m L_i)/m$, where $L_i$
    is the smoothness (w.r.t. $\|\cdot\|_H$) for each component $\phi_i$ of the dual function $\phi(\cdot)$ in Assumption 1.
2:  for s = 0,...,S-1 do
3:      $\tau_{1,s} \leftarrow 2/(s+4)$; $\alpha_s \leftarrow 1/(9\tau_{1,s}\bar{L})$;
4:      $\boldsymbol{\mu}^s \leftarrow \nabla\phi(\widetilde{\boldsymbol{v}}^s)$.
5:      for $j = 0$ to $l-1$ do
6:          $k \leftarrow (sl) + j$;
7:          $\boldsymbol{v}_{k+1} \leftarrow \tau_{1,s}\boldsymbol{z}_k + \tau_2\widetilde{\boldsymbol{v}}^s + (1 - \tau_{1,s} - \tau_2)\boldsymbol{y}_k$;
8:          Pick $B$ samples independently from $\{1, 2, \dots, m\}$ with replacement, where sample $i$
            is picked with probability $p_i = L_i/m\bar{L}$; denote the sampled index set as $I$;
9:          $\widetilde{\boldsymbol{\nabla}}_{k+1} \leftarrow$
            $\boldsymbol{\mu}^s + \frac{1}{B}\sum_{i\in I}\frac{1}{mp_i}(\nabla\phi_i(\boldsymbol{v}_{k+1}) - \nabla\phi_i(\widetilde{\boldsymbol{v}}^s))$;
10:         $\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}}\{\frac{1}{\alpha_s}V_{\boldsymbol{z}_k}(\boldsymbol{z}) + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{z}\rangle\}$;
11:         $\boldsymbol{y}_{k+1} =$
            $\arg\min_{\boldsymbol{y}}\{\frac{9\bar{L}}{2}\|\boldsymbol{y} - \boldsymbol{v}_{k+1}\|_H^2 + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{y}\rangle\}$.
12:     end for
13:     $\widetilde{\boldsymbol{v}}^{s+1} \leftarrow \frac{1}{l}\sum_{j=1}^{l}\boldsymbol{y}_{sl+j}$;
14:     $C_s \leftarrow C_s + \frac{1}{\tau_{1,s}}$;
15:     Pick $\widetilde{\boldsymbol{y}}_s$ uniform randomly from $\{\boldsymbol{y}_{sl+j}\}_{j=1}^{l}$, update
        $D_s \leftarrow D_s + \frac{1}{\tau_{1,s}}\boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)$, where $\boldsymbol{x}(\cdot)$ is given by equation (5);
16:     $\boldsymbol{x}^s = D_s/C_s$.
17: end for
18: Output: $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{S-1}$.
```

We give PDASMD-B in Algorithm 2 and briefly explain it as follows. As compared to the non-batch version PDASMD in Algorithm 1, Step 8 of PDASMD-B now samples a small batch of samples and calculates $\widetilde{\boldsymbol{\nabla}}_{k+1}$ based on the gradient of this small batch. Other hyper-parameters in the algorithm are changed accordingly to ensure convergence.

We apply PDASMD-B to solve OT. The main steps are the same as those in Subsection 2.3; thus, we omit the details. To compute the computational complexity for giving an ϵ-solution to OT, one needs the convergence result of PDASMD-B, which we include in Appendix F. And the computational complexity for solving OT is stated in the following corollary.

Corollary 1. *Run PDASMD-B with batch size B, $\|\cdot\|_H = \|\cdot\|_\infty$ and inner loop size $l = n/B$ (assume w.l.o.g. that l is an integer), the overall number of arithmetic operations to find a solution $\widehat{X}$ such that $\mathbb{E}\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon$ is*

$$\widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty\sqrt{1/B + B\gamma/n}}{\epsilon}\right).$$

Remark 6. *Corollary 1 shows the speed-up of PDASMD-B from parallel computing. We analyzed the speed-up for two cases of γ as follows. The first case is similar to the one in Remark 4: taking $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, then we have $\gamma = n$. This gives us the total computation of $\widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty\sqrt{B}}{\epsilon}\right)$, which is $\sqrt{B}$ times that of non-batch version. There are B batches of parallel computation, so if we ignore the communication time, our batch algorithm enjoys a sublinear speed-up of $\mathcal{O}(\sqrt{B})$. The second case assumes one can further improve the rate $\gamma \sim \mathcal{O}(n)$ to $\gamma \sim \mathcal{O}(\sqrt{n})$. Then for $B \leq \sqrt{n}$, the number of total computations does not increase with B, which indicates a linear speed-up of $\mathcal{O}(B)$ using parallel computing. Though such an improvement in γ is still an open question in optimization, this implies a potentially huge advantage of the batch algorithm.*

4 NUMERICAL STUDIES

In this section, we discuss the result of our numerical studies. The goals of our experiment are to check our theoretical computational complexity of the PDASMD algorithm w.r.t. the marginal size n in Theorem 2, and to check the theoretical computational complexity of the PDASMD-B algorithm w.r.t. the batch size B in Corollary 1. We use both synthetic and real grey-scale images [2] as the marginal distribution for our experiment. Due to the page limit, our data description and algorithm implementation are deferred to Appendix G. We have more applications of our algorithm, including domain adaptation and color transfer, in Appendix H.

Our experiment results are given in Figure 1. We now explain the plots and summarize the results from the plots as follows.

Figures 1(a), 1(b), 1(e) and 1(f) check the computational complexity of PDASMD on the marginal size n. In our experiment, we run PDASMD with $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$. By Theorem 2, for this case, when fixing the accuracy level ϵ, we should have the computational complexity $\sim \mathcal{O}(n^2)$. That is, fixing a ϵ and plotting the logarithm of computation count versus the logarithm of n, we expect to see a line with slope 2. Figures 1(a), 1(b)

[2] The MNIST dataset (LeCun, 1998).

(using synthetic data as marginals) and Figures 1(e) and 1(f) (using real data as marginals) have the lines corresponding to the PDASMD algorithm have slopes that are close to 2, which supports our theoretical rate.

In Figures 1(a), 1(b), 1(e) and 1(f) we also include lines that correspond to other state-of-the-art algorithms. The goal is to compare the practical performance of the PDASMD algorithm with deterministic algorithms (Figures 1(a) and 1(e)) and other stochastic algorithms (Figure 1(b) and 1(f)). We conclude from the plots that the total computation numbers of the AAM, Sinkhorn and Stochastic Sinkhorn are less than that of the PDASMD, which illustrates the practical advantage of those algorithms. However, such an observation does not disqualify our PDASMD algorithm since we still have a provable complexity that is better than those algorithms. Inspired by such an observation, one may further improve the PDASMD in practice. One possible way is to combine the PDASMD algorithm with the Sinkhorn to take advantage of the better theoretical rate of PDASMD and the good empirical performance of the Sinkhorn.

Figures 1(c) and 1(g) check the computational complexity of PDASMD-B on the batch size B. We fix the accuracy level ϵ and run PDASMD-B with $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. By Corollary 1, for a given marginal size n, we have the number of total computation $\sim \mathcal{O}(\sqrt{B})$. Thus, when plotting the logarithm of computation count versus the logarithm of B, we should get a line with slope 0.5. In Figures 1(c) (using synthetic data as marginals) and 1(g) (using real data as marginals), we see that for different marginal size n, the slopes are all close to $.5$. Such an observation matches our theory.

With such computational complexity of PDASMD-B on the batch size B, if we can fully parallelize, the running time of PDASMD-B should be $\sim \mathcal{O}(B^{-0.5})$. To check this, we plot the logarithm of running time versus the logarithm of B in Figures 1(d) and 1(h). The lines fail to have slope -0.5. This is not surprising to see in practice because of the commutation time and limit in the computational resource. But from the plots, we can still benefit from the batch algorithm: when the batch size is not too large ($<= \exp(2.5)$), the running time decreases as the batch size increases. This illustrates the usefulness of the batch version algorithm in practice.

To summarize, our computational complexity of PDASMD on n and PDASMD-B on B are supported by numerical studies.

5 DISCUSSION AND FUTURE STUDIES

This paper proposes a new first-order algorithm for solving entropic OT. We call our algorithm the PDASMD algorithm. We prove that our algorithm finds an ϵ-solution to OT using $\widetilde{\mathcal{O}}(n^2/\epsilon)$ arithmetic operations. Such a rate improves the previously state-of-the-art rate of $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ among the first-order algorithms applied to entropic OT. We perform numerical studies, and the results match our theory.

We discuss some future directions for improving the computational efficiency of OT.

One direction is to revisit other first-order algorithms that are proved to have $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ computational complexity, and see if they can be improved to $\widetilde{\mathcal{O}}(n^2/\epsilon)$. Some algorithms show the $\widetilde{\mathcal{O}}(n^2/\epsilon)$ rate in practice, but there is no proof for such a rate. The techniques in our paper may inspire proper modifications to those algorithms to get a better provable rate. In this way, one may further prove a computational complexity better than that of the PDASMD algorithm by a constant.

Another direction is to combine our algorithm with iterative projection-based algorithms such as the Sinkhorn. This direction is motivated by the Accelerated Sinkhorn algorithm in Lin et al. (2022), which updates the dual variables of entropic OT by Nesterov's estimate sequence (for acceleration) and two Sinkhorn steps. Now our PDASMD algorithm also uses an acceleration technique (Katyusha momentum), it would be interesting to analyze a stochastic Accelerated Sinkhorn by replacing its Nesterov's estimate sequence with the Katyusha momentum.

The third direction is to improve the batch version of our PDASMD algorithm. Our batch-version algorithm has a sub-linear speed-up when fully parallelized and ignores the communication time. In such a setting, one may expect an optimally designed batch algorithm to speed up linearly. That is, the total number of computations does not scale up with the batch size, and the computing time is $1/B$ that of the non-batch version when the batch size is B. If one can improve our batch version algorithm to achieve a linear speed-up, the computational advantage will be huge.

Besides computing for OT, the broader applications of our PDASMD algorithm are also interesting. Our PDASMD algorithm can be applied to a linear constrained strongly convex problem as long as its dual is of a finite-sum form. This motivates one to apply our algorithm to solve other problems such as the unbalanced OT (Pham et al., 2020) and the Wasserstein barycenter (Cuturi and Doucet, 2014) for better computational complexity.

Acknowledgements

This research is supported by the National Science Foundation under Grant No.CCF-1740776 and No.DMS-2015363.

References

Abid, B. K. and Gower, R. (2018). Stochastic Algorithms for Entropy-Regularized Optimal Transport Problems. In *International Conference on Artificial Intelligence and Statistics*, pages 1505–1512. PMLR.

Allen-Zhu, Z. (2017). Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. *The Journal of Machine Learning Research*, 18(1):8194–8244.

Allen-Zhu, Z. and Orecchia, L. (2014). Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*.

Altschuler, J., Weed, J., and Rigollet, P. (2017). Near-Linear Time Approximation Algorithms for Optimal Transport via Sinkhorn Iteration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1961–1971.

Blanchet, J., Jambulapati, A., Kent, C., and Sidford, A. (2018). Towards Optimal Running Times for Optimal Transport. *arXiv preprint arXiv:1810.07717*.

Chambolle, A. and Contreras, J. P. (2022). Accelerated Bregman Primal-Dual methods applied to Optimal Transport and Wasserstein Barycenter problems. *arXiv preprint arXiv:2203.00802*.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2015). Optimal transport for domain adaptation. *arXiv preprint arXiv:1507.00504*.

Cuturi, M. (2013). Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Cuturi, M. and Doucet, A. (2014). Fast computation of Wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR.

Dvurechensky, P., Gasnikov, A., and Kroshnin, A. (2018). Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm. In *35th International Conference on Machine Learning, ICML 2018*, pages 2196–2220.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014). Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic Optimization for Large-Scale Optimal Transport. In *NIPS 2016-Thirtieth Annual Conference on Neural Information Processing System*.

Guminov, S., Dvurechensky, P., Tupitsa, N., and Gasnikov, A. (2021). On a Combination of Alternating Minimization and Nesterov's Momentum. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3886–3898. PMLR.

Guo, W., Ho, N., and Jordan, M. (2020). Fast Algorithms for Computational Optimal Transport and Wasserstein Barycenter. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2088–2097. PMLR.

Jambulapati, A., Sidford, A., and Tian, K. (2019). A Direct $\tilde{O}(1/\epsilon)$ Iteration Parallel Algorithm for Optimal Transport. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

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

LeCun, Y. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Lin, T., Ho, N., and Jordan, M. (2019). On Efficient Optimal Transport: An Analysis of Greedy and Accelerated Mirror Descent Algorithms. In *International Conference on Machine Learning*, pages 3982–3991. PMLR.

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

Mishchenko, K. (2019). Sinkhorn Algorithm as a Special Case of Stochastic Mirror Descent. *arXiv preprint arXiv:1909.06918*.

Monge, G. (1781). Mémoire sur la Théorie des Déblais et des Remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Nemirovskii, A. and Yudin, D. (1983). *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley.

Nesterov, Y. (2003). *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media.

Nesterov, Y. (2005). Smooth Minimization of Non-Smooth Functions. *Mathematical programming*, 103(1):127–152.

Pham, K., Le, K., Ho, N., Pham, T., and Bui, H. (2020). On unbalanced optimal transport: An analysis of sinkhorn algorithm. In *International Conference on Machine Learning*, pages 7673–7682. PMLR.

Quanrud, K. (2018). Approximating Optimal Transport With Linear Programs. In Fineman, J. T. and Mitzenmacher, M., editors, *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, volume 69 of *OpenAccess Series in Informatics (OASIcs)*, pages 6:1–6:9, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Rabin, J., Ferradans, S., and Papadakis, N. (2014). Adaptive color transfer with relaxed optimal transport. In

2014 IEEE international conference on image processing (ICIP), pages 4852–4856. IEEE.

Villani, C. (2009). *Optimal Transport: Old and New*, volume 338. Springer.

Xie, Y., Luo, Y., and Huo, X. (2022). An Accelerated Stochastic Algorithm for Solving the Optimal Transport Problem. *arXiv preprint arXiv:2203.00813*.

Improved Rate of First Order Algorithms for Entropic Optimal Transport: Supplementary Materials

A Proximal Mirror Descent

In this section, we review the technique of stochastic proximal mirror descent.

Let us start with the objective function:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}). \tag{11}$$

A popular way to minimize problem (11) is the Stochastic Gradient Descent (SGD). At time t, the SGD algorithm randomly samples i_t from $\{1,\ldots,n\}$ and updates as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - b_t \nabla f_{i_t}(\boldsymbol{x}_t), \tag{12}$$

where b_t is the step size. Note that formula (12) is essentially the solution to the following ℓ_2 penalized problem:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t)\rangle + \frac{1}{2b_t}\|\boldsymbol{x}-\boldsymbol{x}_t\|_2^2 \right\}. \tag{13}$$

The proximal/mirror descent is proposed by Nemirovskii and Yudin (1983), where they generalize the SGD by replacing the $\|\cdot\|_2^2$ term in problem (13) by some proximity function. There are two popular choices of proximity functions, and they lead to stochastic proximal and mirror descent, respectively. In this paper, we use stochastic proximal mirror descent to represent both cases.

The choice of proximity function that leads to stochastic proximal gradient descent is the square of an arbitrary norm $\|\cdot\|_H$ (as compared to the $\|\cdot\|_2$ norm in problem (13)). This results in the update

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t)\rangle + \frac{1}{2b_t}\|\boldsymbol{x}-\boldsymbol{x}_t\|_H^2 \right\}. \tag{14}$$

The choice of proximity function that gives stochastic mirror descent is the Bregman divergence. Recall that for a mirror map $w(\cdot)$, the Bregman divergence is

$$V_{\boldsymbol{x}}(\boldsymbol{y}) := w(\boldsymbol{y}) - w(\boldsymbol{x}) - \langle \nabla w(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle.$$

The stochastic mirror descent then updates as:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t)\rangle + \frac{1}{2b_t}V_{\boldsymbol{x}_t}(\boldsymbol{x}) \right\}. \tag{15}$$

Note that the popular KL-divergence $\mathcal{KL}(\boldsymbol{x}||\boldsymbol{x}') := \sum_i x_i \log(x_i/x_i') - \sum_i x_i + \sum_i x_i'$ is a special case of Bregman divergence by choosing w to be the negative entropy $w(\boldsymbol{x}) = \sum_i x_i \log x_i$.

Recall that the objective is to solve the (entropic) OT, so we explain why the proximal mirror algorithm might be suitable for optimizing the entropic OT compared with the SGD.

First, the objective function of entropic OT coincides with the proximal mirror descent formulation in that each step of proximal mirror descent minimizes an inner product term plus a divergence term other than the ℓ_2 norm. In this way, the proximal mirror descent may help to prove a faster convergence when solving OT.

Second, it is pointed out that the popular Sinkhorn algorithm to solve entropic OT can be interpreted as a special case of the stochastic proximal mirror descent algorithm (Mishchenko, 2019). We briefly summarize their statement as follows. The Sinkhorn algorithm iteratively updates the dual variables $\boldsymbol{u}, \boldsymbol{v}$ of problem (2) by:

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + log\boldsymbol{p}' - log(X(\boldsymbol{u}^k, \boldsymbol{v}^k)\mathbf{1}), \boldsymbol{v}^{k+1} = \boldsymbol{v}^k, \tag{16}$$

and

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k, \boldsymbol{v}^{k+1} = \boldsymbol{v}^k + log\boldsymbol{q}' - log(X(\boldsymbol{u}^k, \boldsymbol{v}^k)^T\mathbf{1}), \tag{17}$$

where the relationship between the primal-dual variables is

$$X(\boldsymbol{u}, \boldsymbol{v}) = diag(\exp(\boldsymbol{u}))\exp(-C/\eta)diag(\exp(\boldsymbol{v})).$$

Notice that the dual variables $\boldsymbol{u}, \boldsymbol{v}$ are equivalent to the dual variables $\boldsymbol{\lambda}, \boldsymbol{\tau}$ we use in formulation (9) plus constants.

To interpret Sinkhorn as a Stochastic Mirror Descent, one considers the objective function:

$$\min_{X \in \mathbb{R}^{n\times n}} f(X) := \frac{1}{2}(f_1(X) + f_2(X)) \tag{18}$$

$$f_1(X) = \mathcal{KL}(X\mathbf{1}||\boldsymbol{p}'), f_2(X) = \mathcal{KL}(X^T\mathbf{1}||\boldsymbol{q}'). \tag{19}$$

Now the objective function is a finite-sum of two functions: $f_1(\cdot)$ and $f_2(\cdot)$, then we can run SMD on it. Suppose that the SMD is initialized at $X_0 = \exp(-C/\eta)$, choose the step size $\eta = 1$ and mirror map $w(X) = \sum_{i,j} X_{i,j}(\log X_{i,j} - 1)$. When the first sample is used (i.e. the sub-gradient of f_1 is used), SMD updates as

$$\nabla w(X^{k+1}) = \nabla w(X^k) - \nabla f_1(X^k).$$

One can check that it is exactly equivalent to one step Sinkhorn update in $\boldsymbol{u}$ as step (16). Similarly, SMD using f_2 is equivalent to one step Sinkhorn update in v as step (17).

From the above, the Sinkhorn is a special case of SMD, which suggests that mirror-based algorithms may be proper for solving the entropic OT. Given the success of the Sinkhorn algorithm, it would be interesting to discover more general stochastic proximal mirror descent algorithms and study their performance for solving OT.

B Proof for Theorem 1

To prove Theorem 1, the following lemmas are established:

Lemma 2 (Coupling step 1)**.** *Consider one inner loop of Algorithm 1, where the randomness only comes from the choice of i. It satisfies that for $\forall u$:*

$$\begin{aligned}&\alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle\\ \le&\frac{\alpha_s}{\tau_{1,s}}\{\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k-1}\rangle\}\\ &+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].\end{aligned}$$

Proof. Note that the inner loop of Algorithm 1 is the same as Algorithm 5 in Allen-Zhu (2017). We can take $F(\cdot) = f(\cdot) = \phi(\cdot), \psi(\cdot) = 0$ and $\sigma = 0$ in Lemma E.4. of Allen-Zhu (2017) to get:

$$\begin{aligned}&\alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle\\ \le&\frac{\alpha_s}{\tau_{1,s}}\{\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k-1}\rangle\}\\ &+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].\end{aligned}$$

We also provide full proof as follows for a better understanding of the algorithm. Abbreviate $\widetilde{v} = \widetilde{v}^s$, $\alpha = \alpha_s$, $\tau_1 = \tau_{1,s}$, and denote $\sigma_{k+1}^2 = \|\nabla\phi(v_{k+1}) - \widetilde{\nabla}_{k+1}\|_{H,*}$, $Prog(v_{k+1}) := -\min_y\{\frac{9\bar{L}}{2}\|y - v_{k+1}\|_H^2 + \langle\widetilde{\nabla}_{k+1}, y - v_{k+1}\rangle\}$. We claim the following bounds hold, which we will prove later:

$$\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] \ge \mathbb{E}[Prog(v_{k+1})] - \frac{1}{16\bar{L}}\mathbb{E}[\sigma_{k+1}^2], \tag{20}$$

$$\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2] \leq 8\bar{L}(\phi(\widetilde{v}) - \phi(v_{k+1}) - \langle \nabla\phi(v_{k+1}), \widetilde{v} - v_{k+1}\rangle), \tag{21}$$

$$\alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \leq -\frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u), \forall u. \tag{22}$$

Then we have:

$$\begin{aligned}
&\alpha\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle \\
&= \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle + \alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \\
&\overset{(22)}{\leq} \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u).
\end{aligned} \tag{23}$$

Let $w = \tau_1 z_{k+1} + \tau_2\widetilde{v} + (1 - \tau_1 - \tau_2)y_k$, then $v_{k+1} - w = \tau_1(z_k - z_{k+1})$, and

$$\begin{aligned}
&\mathbb{E}\left[\alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2\right] \\
=&\mathbb{E}\left[\frac{\alpha}{\tau_1}\left(\langle \widetilde{\nabla}_{k+1}, v_{k+1} - w\rangle - \frac{1}{2\alpha\tau_1}\|v_{k+1} - w\|_H^2\right)\right] \\
=&\mathbb{E}\left[\frac{\alpha}{\tau_1}\left(\langle \widetilde{\nabla}_{k+1}, v_{k+1} - w\rangle - \frac{9\bar{L}}{2}\|v_{k+1} - w\|_H^2\right)\right] \\
\overset{(20)}{\leq}&\mathbb{E}\left[\frac{\alpha}{\tau_1}\left(\phi(v_{k+1}) - \phi(y_{k+1}) + \frac{1}{16\bar{L}}\sigma_{k+1}^2\right)\right] \\
\overset{(21)}{\leq}&\mathbb{E}\left[\frac{\alpha}{\tau_1}\left(\phi(v_{k+1}) - \phi(y_{k+1}) + \frac{1}{2}(\phi(\widetilde{v}) - \phi(v_{k+1}) - \langle \nabla\phi(v_{k+1}), \widetilde{v} - v_{k+1}\rangle)\right)\right].
\end{aligned} \tag{24}$$

Take expectation on both sides of inequality (23) and plug in (24), we have the lemma claim.

It remains to check inequalities (20), (21) and (22), which we do as follows.

For inequality (20):

$$\begin{aligned}
&Prog(v_{k+1}) \\
&= -\min_y\{\frac{9\bar{L}}{2}\|y - v_{k+1}\|_H^2 + \langle \widetilde{\nabla}_{k+1}, y - v_{k+1}\rangle\} \\
&= -\left(\frac{9\bar{L}}{2}\|y_{k+1} - v_{k+1}\|_H^2 + \langle \widetilde{\nabla}_{k+1}, y_{k+1} - v_{k+1}\rangle\right) \\
&= -\left(\frac{\bar{L}}{2}\|y_{k+1} - v_{k+1}\|_H^2 + \langle \nabla\phi(v_{k+1}), y_{k+1} - v_{k+1}\rangle\right) \\
&\quad + \left(\langle \nabla\phi(v_{k+1}) - \widetilde{\nabla}_{k+1}, y_{k+1} - v_{k+1}\rangle - 4\bar{L}\|y_{k+1} - v_{k+1}\|_H^2\right) \\
&\leq -(\phi(y_{k+1}) - \phi(v_{k+1})) + \left(\langle \nabla\phi(v_{k+1}) - \widetilde{\nabla}_{k+1}, y_{k+1} - v_{k+1}\rangle - 4\bar{L}\|y_{k+1} - v_{k+1}\|_H^2\right) && (25) \\
&\leq -(\phi(y_{k+1}) - \phi(v_{k+1})) + \frac{1}{16\bar{L}}\|\nabla\phi(v_{k+1}) - \widetilde{\nabla}_{k+1}\|_{H,*}^2, && (26)
\end{aligned}$$

where inequality (25) comes from the smoothness of ϕ, and inequality (26) uses the Young's inequality $\langle a, b\rangle - \frac{1}{2}\|b\|_H^2 \leq \frac{1}{2}\|a\|_{H,*}^2$. Take expectation on both sides and rearrange the terms we have inequality (20).

For inequality (21): Since each $\phi_i(\cdot)$ is convex and L_i- smooth, by Theorem 2.1.5 in Nesterov (2003), we have

$$\|\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v})\|_{H,*}^2 \leq 2L_i[\phi_i(\widetilde{v}) - \phi_i(v_{k+1}) - \langle \nabla\phi_i(v_{k+1}), \widetilde{v} - v_{k+1}\rangle]. \tag{27}$$

Thus

$$\begin{aligned}
&\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2] \\
=&\mathbb{E}_i\left[\left\|\frac{1}{mp_i}(\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v})) - (\nabla\phi(v_{k+1}) - \nabla\phi(\widetilde{v}))\right\|_{H,*}^2\right] \\
\leq&2\mathbb{E}_i\left[\frac{1}{m^2p_i^2}\|(\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v})\|_{H,*}^2\right] + 2\|\nabla\phi(v_{k+1}) - \nabla\phi(\widetilde{v}))\|_{H,*}^2 \\
\overset{(27)}{\leq}&4\mathbb{E}_i\left[\frac{L_i}{m^2p_i^2}\left(\phi_i(\widetilde{v}) - \phi_i(v_{k+1}) - \langle\nabla\phi_i(v_{k+1}), \widetilde{v} - v_{k+1}\rangle\right)\right] + 2\|\nabla\phi(v_{k+1}) - \nabla\phi(\widetilde{v}))\|_{H,*}^2 \\
=&4\bar{L}\left(\phi(\widetilde{v}) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v} - v_{k+1}\rangle\right) + 2\|\nabla\phi(v_{k+1}) - \nabla\phi(\widetilde{v}))\|_{H,*}^2 \\
\overset{(27)}{\leq}&8\bar{L}\left(\phi(\widetilde{v}) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v} - v_{k+1}\rangle\right). \qquad (28)
\end{aligned}$$

For inequality (22): By definition of z_{k+1}, we have

$$\nabla V_{z_k}(z_{k+1}) + \alpha\widetilde{\nabla}_{k+1} = 0,$$

then

$$\langle\nabla V_{z_k}(z_{k+1}) + \alpha\widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle = 0, \forall u. \qquad (29)$$

One has the “three-point equality of Bregman divergence" that

$$\langle\nabla V_{z_k}(z_{k+1}), z_{k+1} - u\rangle = V_{z_k}(z_{k+1}) - V_{z_k}(u) + V_{z_{k+1}}(u). \qquad (30)$$

We can check

$$\begin{aligned}
&\alpha\langle\widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \\
\overset{(29)}{=}&-\langle\nabla V_{z_k}(z_{k+1}), z_{k+1} - u\rangle \\
\overset{(30)}{=}&-V_{z_k}(z_{k+1}) + V_{z_k}(u) - V_{z_{k+1}}(u) \\
\leq&-\frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u), \qquad (31)
\end{aligned}$$

where the inequality comes from the strong convexity of the mirror function $w(\cdot)$. □

Lemma 3 (Coupling step 2)**.** *Using the Lemma 2, we further have*

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - u\rangle \\
\leq&\alpha_s\phi(v_{k+1}) + \frac{\alpha_s(1 - \tau_{1,s} - \tau_2)}{\tau_{1,s}}\phi(y_k) + \frac{\alpha_s}{\tau_{1,s}}\left(\tau_2\phi(\widetilde{v}^s) - \mathbb{E}[\phi(y_{k+1})]\right) + V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

Proof. First compute that

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - u\rangle = \alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - z_k\rangle + \alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle \\
=&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle + \frac{\alpha_s(1 - \tau_{1,s} - \tau_2)}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}), y_k - v_{k+1}\rangle \\
&+ \alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle \\
\leq&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle + \frac{\alpha_s(1 - \tau_{1,s} - \tau_2)}{\tau_{1,s}}(\phi(y_k) - \phi(v_{k+1})) + \alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle,
\end{aligned}$$

where the second equality by the updating rule $v_{k+1} = \tau_{1,s}z_k + \tau_2\widetilde{v}^s + (1 - \tau_{1,s} - \tau_2)y_k$, and the inequality by convexity of ϕ. Next, we apply Lemma 2 to get

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle \\
\leq&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}(\phi(y_k)-\phi(v_{k+1})) \\
&+\frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle\right) \\
&+V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)] \\
=&\alpha_s\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}\phi(y_k) + \frac{\alpha_s}{\tau_{1,s}}\left(\tau_2\phi(\widetilde{v}^s) - \mathbb{E}[\phi(y_{k+1})]\right) + V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

□

Lemma 4 (One outer loop)**.** *Consider the sth epoch, assume that all randomness in the first $s-1$ epochs are fixed, we have*

$$\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_2}{\tau_{1,s+1}^2}\sum_{k=sl}^{sl+l-2}\mathbb{E}(\phi(y_{k+1})-\phi^*) \\
&+\frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2}\mathbb{E}(\phi(y_{(s+1)l})-\phi^*) \\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\phi(v_{k+1})-\phi^*) + \frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}(\phi(y_{sl})-\phi^*) + \frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-2}(\phi(y_{k+1})-\phi^*) \\
&+9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)),
\end{aligned}$$

where $\phi^* = \min\phi(\cdot)$.

Proof. Sum up the inequality in Lemma 3 for $k = sl+j, j = 0,\ldots,l-1$, we have:

$$\begin{aligned}
&\alpha_s\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle \\
\leq&\sum_{k=sl}^{sl+l-1}\left\{\alpha_s\mathbb{E}\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}\mathbb{E}\phi(y_k) + \frac{\alpha_s}{\tau_{1,s}}\left(\tau_2\phi(\widetilde{v}^s) - \mathbb{E}[\phi(y_{k+1})]\right)\right. \\
&\left.+\mathbb{E}V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)]\right\} \\
=&\sum_{k=sl}^{sl+l-1}\left\{\alpha_s\mathbb{E}\phi(v_{k+1}) - \frac{\alpha_s(\tau_{1,s}+\tau_2)}{\tau_{1,s}}\mathbb{E}\phi(y_{k+1})\right\} \\
&+\frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}[\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\alpha_s\tau_2 l}{\tau_{1,s}}\phi(\widetilde{v}^s) + \mathbb{E}V_{z_{sl}}(u) - \mathbb{E}[V_{z_{(s+1)l}}(u)].
\end{aligned} \tag{32}$$

By convexity of ϕ, using Jensen's inequality, we have $\frac{1}{l}\sum_{k=sl-l}^{sl-1}\phi(y_{k+1}) \geq \phi(\frac{1}{l}\sum_{k=sl-l}^{sl-1}y_{k+1}) = \phi(\widetilde{v}^s)$. Thus

$$\begin{aligned}
&\alpha_s\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\alpha_s(\tau_{1,s}+\tau_2)}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\phi(y_{k+1}) \\
\leq&\alpha_s\sum_{k=sl}^{sl+l-1}\mathbb{E}\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}[\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\alpha_s\tau_2}{\tau_{1,s}}\sum_{k=sl-l}^{sl-1}\phi(y_{k+1}) \\
&+\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u).
\end{aligned}$$

Recall that $\alpha_s = 1/(9\bar{L}\tau_{1,s})$, we have

$$\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2}\sum_{k=sl}^{sl+l-1}\mathbb{E}\phi(y_{k+1})\\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\phi(v_{k+1}) + \frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}^2}[\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-1}\phi(y_{k+1})\\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

Deducting $\phi^* = \min\phi(\cdot)$ from both sides and rearranging terms, we get

$$\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2}\sum_{k=sl}^{sl+l-2}\mathbb{E}(\phi(y_{k+1})-\phi^*)\\
&+ \frac{1}{\tau_{1,s}^2}\mathbb{E}(\phi(y_{(s+1)l})-\phi^*)\\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\phi(v_{k+1})-\phi^*) + \frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}(\phi(y_{sl})-\phi^*) + \frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-2}(\phi(y_{k+1})-\phi^*)\\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

By our choice of $\tau_{1,s}$ and τ_2, one can check

$$\frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2} \leq \frac{1}{\tau_{1,s}^2}, \quad \frac{\tau_2}{\tau_{1,s+1}^2} \leq \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2}.$$

So we further have

$$\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_2}{\tau_{1,s+1}^2}\sum_{k=sl}^{sl+l-2}\mathbb{E}(\phi(y_{k+1})-\phi^*)\\
&+ \frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2}\mathbb{E}(\phi(y_{(s+1)l})-\phi^*)\\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\phi(v_{k+1})-\phi^*) + \frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}(\phi(y_{sl})-\phi^*) + \frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-2}(\phi(y_{k+1})-\phi^*)\\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

□

Finally, we can prove our main Theorem 1 as follows.

Proof. By Lemma 4, for $s = 1, \ldots, S-1$, denote $\delta(\cdot) := \phi(\cdot) - \phi^*$ we have:

$$\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_2}{\tau_{1,s+1}^2}\sum_{k=sl}^{sl+l-2}\mathbb{E}\delta(y_{k+1}) + \frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2}\mathbb{E}\delta(y_{(s+1)l})\\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\delta(v_{k+1}) + \frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}\delta(y_{sl}) + \frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-2}\mathbb{E}\delta(y_{k+1})\\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned} \tag{33}$$

For $s = 0$, apply similar proof as Lemma 4 on inequality (32), we have:

$$\begin{aligned}
&\frac{1}{\tau_{1,0}}\sum_{k=0}^{l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle + \frac{\tau_2}{\tau_{1,1}^2}\sum_{k=0}^{l-2}\mathbb{E}\delta(y_{k+1}) + \frac{1-\tau_{1,1}}{\tau_{1,1}^2}\mathbb{E}\delta(y_l)\\
\leq&\frac{1}{\tau_{1,0}}\sum_{k=0}^{l-1}\mathbb{E}\delta(v_{k+1}) + \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0) + \frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}^0) + 9\bar{L}(V_{z_0}(u) - \mathbb{E}V_{z_l}(u)).
\end{aligned} \tag{34}$$

Telescope inequality (33) for $s = 1, \dots, S-1$ and add inequality (34), we have following bound:

$$\begin{aligned}&\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle-\delta(v_{k+1}))+\frac{\tau_2}{\tau_{1,S}^2}\sum_{k=(S-1)l}^{Sl-2}\mathbb{E}\delta(y_{k+1})\\ \le&\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0)+\frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0)+9\bar{L}(V_{z_0}(u)-\mathbb{E}V_{z_{Sl}}(u))-\frac{1-\tau_{1,S}}{\tau_{1,S}^2}\mathbb{E}\delta(y_{Sl}).\end{aligned}\tag{35}$$

Now for the term $\langle\nabla\phi(v), v-u\rangle - \phi(v)$, we note that

$$\begin{aligned}\langle\nabla\phi(v), v-u\rangle-\phi(v)&=\langle\nabla\phi(v), v-u\rangle-(\langle v, b\rangle-f(x(v))-\langle A^T v, x(v)\rangle)\\ &=\langle b-Ax(v), v-u\rangle-(\langle v, b-Ax(v)\rangle-f(x(v)))\\ &=\langle b-Ax(v), -u\rangle+f(x(v)).\end{aligned}\tag{36}$$

Thus

$$\begin{aligned}&\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\langle\nabla\phi(v_{k+1}), v_{k+1}-u\rangle-\delta(v_{k+1}))\\ =&\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\left[\langle b-Ax(v_{k+1}), -u\rangle+f(x(v_{k+1}))-f(x(\lambda^*))\right]\\ =&\left\langle\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}b-A\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right], -u\right\rangle\\ &\quad+\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}f(x(v_{k+1}))-\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f(x(\lambda^*))\\ \ge&\left\langle\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}b-A\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right], -u\right\rangle\\ &\quad+\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f\left(\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right]/\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)-\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f(x(\lambda^*))\\ =&\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\left[f(\mathbb{E}(x^{S-1}))-f(x(\lambda^*))+\langle b-A\mathbb{E}(x^{S-1}), -u\rangle\right],\end{aligned}\tag{37}$$

where the inequality applies Jensen's inequality on convex function f. Plugging inequality (37) into inequality (35) and using the fact that $0 < \frac{\tau_2}{\tau_{1,S}^2} \le \frac{1-\tau_{1,S}}{\tau_{1,S}^2}$, we have

$$\begin{aligned}&\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(f(\mathbb{E}(x^{S-1}))-f(x(\lambda^*)))\\ \le&\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0)+\frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0)+9\bar{L}(V_{z_0}(u)-\mathbb{E}V_{z_{Sl}}(u))-\frac{\tau_2}{\tau_{1,S}^2}\sum_{k=(S-1)l}^{Sl-1}\mathbb{E}\delta(y_{k+1})\\ &+\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(\langle b-A\mathbb{E}(x^{S-1}), u\rangle)\\ \le&\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0)+\frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0)+9\bar{L}V_{z_0}(u)-\frac{\tau_2 l}{\tau_{1,S}^2}\mathbb{E}\delta(\widetilde{v}^S)\\ &+\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(\langle b-A\mathbb{E}(x^{S-1}), u\rangle),\end{aligned}\tag{38}$$

where the second inequality comes from the definition of $\widetilde{v}^S$ and Jensen's inequality. Recall that inequality (38) holds for any u, including the one that minimizes the R.H.S.. We can further upper bound $\min_u R.H.S.$ by restricting $u \in B_H(2R) :=$

$\{u : \|u\|_H \leq 2R\}$:

$$\begin{aligned}
&\min_{u\in B_H(2R)} 9\bar{L}V_0(u) + \sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\langle b - A\mathbb{E}(x^{S-1}), u\rangle \\
\leq &\min_{u\in B_H(2R)} 9\bar{L}\gamma\|u\|_H^2/2 + \sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\langle b - A\mathbb{E}(x^{S-1}), u\rangle \\
\leq &\min_{u\in B_H(2R)} \langle\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}(b - A\mathbb{E}(x^{S-1})), u\rangle + 18\bar{L}R^2\gamma \\
= &- 2R\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)\|b - A\mathbb{E}(x^{S-1})\|_{H,*} + 18\bar{L}R^2\gamma.
\end{aligned} \tag{39}$$

Plugging the bound (39) into inequality (38), we have

$$\begin{aligned}
&\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(f(\mathbb{E}(x^{S-1})) - f(x(\lambda^*))) + \frac{\tau_2 l}{\tau_{1,S}^2}\mathbb{E}\delta(\widetilde{v}^S) + 2R\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)\|b - A\mathbb{E}(x^{S-1})\|_{H,*} \\
\leq &\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0) + \frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0) + 18\bar{L}R^2\gamma \\
= &2l\delta(0) + 18\bar{L}R^2\gamma.
\end{aligned} \tag{40}$$

Calculate $\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}} = \sum_{s=0}^{S-1}(s+4)/2 = (2S+3)S/4 \geq S^2/2$, then

$$f(\mathbb{E}(x^{S-1})) - f(x^*) \leq \frac{4}{S^2 l}\left[l\delta(0) + 9\bar{L}R^2\gamma\right]. \tag{41}$$

On the other hand, notice that $f(x(\lambda^*)) = -\phi(\lambda^*) := \phi^*$ and

$$\begin{aligned}
&f(\mathbb{E}(x^{S-1})) - f(x(\lambda^*)) \\
= &f(\mathbb{E}(x^{S-1})) + \phi^* \\
= &f(\mathbb{E}(x^{S-1})) + \langle\lambda^*, b\rangle + \max_x(-f(x) - \langle A^T\lambda^*, x\rangle) \\
\geq &f(\mathbb{E}(x^{S-1})) + \langle\lambda^*, b\rangle - f(\mathbb{E}(x^{S-1})) - \langle A^T\lambda^*, \mathbb{E}(x^{S-1})\rangle \\
= &\langle\lambda^*, b - A\mathbb{E}(x^{S-1})\rangle \geq -R\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*}.
\end{aligned} \tag{42}$$

Plugging inequality (42) into inequality (40), we have

$$R\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*} \leq 2l\delta(0) + 18\bar{L}R^2\gamma. \tag{43}$$

Thus

$$\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*} \leq \frac{4\left[l\delta(0) + 9\bar{L}R^2\gamma\right]}{S^2 lR}. \tag{44}$$

Further check that

$$\delta(0) = \phi(0) - \phi^* \leq \langle\nabla\phi(\lambda^*), 0 - \lambda^*\rangle + \frac{\bar{L}}{2}\|0 - \lambda^*\|_H^2 = \frac{\bar{L}}{2}\|\lambda^*\|_H^2 \leq \frac{\bar{L}}{2}R^2. \tag{45}$$

Plugging the bound (45) into inequalities (41) and (44), we get the theorem claim. □

C Proof for Lemma 1

Proof. By Proposition 2 of Xie et al. (2022), $\phi_i(\cdot)$ is $\frac{np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_2$. So here we only show the second part of the statement. That is, prove the smoothness w.r.t. $\|\cdot\|_\infty$.

By

$$\phi_i(\lambda) = np_i' \left(-\langle q', \lambda \rangle - \eta \log p_i' + \eta \log \left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j} - \eta)/\eta) \right) + \eta \right),$$

we calculate that

$$\nabla \phi_i(\lambda) = np_i' \left(-q' + \frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j})/\eta)\right)} \right).$$

The goal is $\forall \lambda, \lambda'$, bound the $\|\cdot\|_{\infty,*} = \|\cdot\|_1$ of following difference in the gradient:

$$\nabla \phi_i(\lambda) - \nabla \phi_i(\lambda') = np_i' \left(\frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k' - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j' - c_{i,j})/\eta)\right)} \right).$$

Further denote $\Delta\lambda = \lambda' - \lambda$, then

$$\begin{aligned}
&\|\nabla \phi_i(\lambda) - \nabla \phi_i(\lambda')\|_1 \\
=&np_i' \left\| \frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k + (\Delta\lambda)_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j + (\Delta\lambda)_j - c_{i,j})/\eta)\right)} \right\|_1 \\
=&np_i' \left\| \frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k - c_{i,k})/\eta) * \exp((\Delta\lambda/\eta)_k))_{k=1,\dots,n}}{\left(\sum_{j=1}^{n} \exp((\lambda_j - c_{i,j})/\eta) \exp((\Delta\lambda/\eta)_j)\right)} \right\|_1.
\end{aligned}$$

Taking $\boldsymbol{a} = (\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}$ and $\boldsymbol{b} = \Delta\lambda/\eta$ in Lemma 5, we immediately have

$$\|\nabla \phi_i(\lambda) - \nabla \phi_i(\lambda')\|_1 \le np_i' 5 \|\Delta\lambda/\eta\|_\infty = \frac{5np_i'}{\eta} \|\lambda - \lambda'\|_\infty. \tag{46}$$

Thus, $\phi_i(\cdot)$ is $\frac{5np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_\infty$ norm. □

The following lemma is used in the proof of Lemma 1.

Lemma 5. *Consider two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, and let $\exp(\boldsymbol{b})$ be the element-wise exponential of $\boldsymbol{b}$. When $\boldsymbol{a} > \mathbf{0}$, we have*

$$\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 \le 5\|\boldsymbol{b}\|_\infty.$$

Proof. Consider two cases:

First, when $\|\boldsymbol{b}\|_\infty > 0.5$:

$$\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 \le \left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} \right\|_1 + \left\| \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 = 2 < 5 * 0.5 < 5\|\boldsymbol{b}\|_\infty.$$

Second, when $\|\boldsymbol{b}\|_\infty \le 0.5$:

$$
\begin{aligned}
&\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 \\
=&\sum_{i=1}^{d} \frac{|\sum_{j=1}^{d} a_i a_j \exp(b_j) - \sum_{j=1}^{d} a_i \exp(b_i) a_j|}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \\
\le&\sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j |\exp(b_j) - \exp(b_i)|}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \\
\le&\sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j (|\exp(b_j) - 1| + |\exp(b_i) - 1|)}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1 \exp(-0.5)} \\
\le&2\exp(0.5)(\exp(0.5) - 1) \sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j (|b_j| + |b_i|)}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1} \\
\le&4\exp(0.5)(\exp(0.5) - 1) \sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j \|\boldsymbol{b}\|_\infty}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1} \\
=&4\exp(0.5)(\exp(0.5) - 1)\|\boldsymbol{b}\|_\infty < 5\|\boldsymbol{b}\|_\infty.
\end{aligned}
$$

Combine two cases we have the lemma holds. □

D Proof for Theorem 2

The full procedure for finding an ϵ-solution to OT using PDASMD is given in the following algorithm:

```
Algorithm 3: Approximating OT by PDASMD
  Input: Accuracy $\epsilon > 0$, $\eta = \frac{\epsilon}{4\log(n)}$ and $\epsilon' = \frac{\epsilon}{8\|C\|_\infty}$.
  Step 1: Let $\boldsymbol{p}' \in \Delta_n$ and $\boldsymbol{q}' \in \Delta_n$ be
      $\begin{pmatrix} \boldsymbol{p}' \\ \boldsymbol{q}' \end{pmatrix} = \left(1 - \frac{\epsilon'}{8}\right) \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{pmatrix} + \frac{\epsilon'}{8n} \begin{pmatrix} \mathbf{1}_n \\ \mathbf{1}_n \end{pmatrix}.$
  Step 2: Compute $\widetilde{\boldsymbol{X}}$ by PDASMD on objective (2) long enough such that $f(\mathbb{E}\widetilde{\boldsymbol{x}}) - f(\boldsymbol{x}^*) \le \frac{\epsilon}{4}$ and $\|A\mathbb{E}\widetilde{\boldsymbol{x}} - b\|_1 \le \frac{\epsilon'}{2}$.
  Step 3: Round $\widetilde{X}$ to $\widehat{X}$ by Algorithm 2 in Altschuler et al. (2017) such that $\widehat{X}\mathbf{1}_n = \boldsymbol{p}$, $\widehat{X}^T\mathbf{1}_n = \boldsymbol{q}$.
  Output: $\widehat{X}$
```

The total computational cost of Algorithm 3 is given in Theorem 2, and we prove it as follows:

Proof. We have the convergence result in Theorem 1 holds for the dual formulation. To extend the proof of Theorem 1 to the semi-dual formulation for the OT problem, we just need the following equality to hold:

$$
[Ax(v) - b] = \begin{bmatrix} \mathbf{0}_n \\ \nabla\phi(v) \end{bmatrix}.
$$

One can easily check it is true for the semi-dual of OT. Moreover, Xie et al. (2022) shows that the stopping criteria in step 2 of Algorithm 3 guarantees

$$
\mathbb{E}\langle C, \widehat{X}\rangle \le \langle C, X^*\rangle + \epsilon.
$$

That is, the output of Algorithm 3 is an $\epsilon-$solution. We now focus on the computational complexity of Algorithm 3.

Case 1: $\|\cdot\|_H = \|\cdot\|_2$. By Theorem 1 we have

$$
\|\mathbb{E}[b - A\widetilde{x}]\|_2 \le \frac{2\left[l\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{47}
$$

$$
f(\mathbb{E}(\widetilde{x})) - f(x^*) \le \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right], \tag{48}
$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n \frac{np'_i}{\eta} = \frac{1}{\eta}$, and R is an upper bound for $\|\lambda^*\|_2$. By Lemma 3.2 in Lin et al. (2019), $R = \eta\sqrt{n}(R' + .5)$ for $R' = \|C\|_\infty/\eta + \log(n) - 2\log(\min_{1\leq i,j\leq n}\{p'_i, q'_j\}) \leq 4\|C\|_\infty \log(n)/\epsilon + \log(n) - 2\log(\epsilon) + 2\log(64n\|C\|_\infty) = \mathcal{O}(\|C\|_\infty \log(n)/\epsilon)$.

Using the bound $\|A\mathbb{E}\widetilde{x} - b\|_1 \leq \sqrt{2n}\|A\mathbb{E}\widetilde{x} - b\|_2$ we have the stopping criteria in step 2 satisfied for

$$\begin{aligned}
S =& \max\left\{\mathcal{O}\left(\sqrt{\frac{\bar{L}R\sqrt{n}}{\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}\gamma R\sqrt{n}}{l\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2}{\epsilon}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2\gamma}{l\epsilon}}\right)\right\} \\
=& \max\left\{\mathcal{O}\left(\sqrt{\frac{n\log(n)\|C\|_\infty^2}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\gamma n\|C\|_\infty^2}{l\epsilon^2}}\right),\right. \\
&\left.\mathcal{O}\left(\frac{n^{.5}\|C\|_\infty\sqrt{\log n}}{\epsilon}\right), \mathcal{O}\left(\sqrt{\frac{n\log(n)\|C\|_\infty^2\gamma}{l\epsilon^2}}\right)\right\} \\
=& \mathcal{O}\left(\frac{n^{.5}}{\epsilon}\max\left(\|C\|_\infty\sqrt{\log n}, \sqrt{\log(n)\|C\|_\infty^2\gamma/l}\right)\right) \\
=& \mathcal{O}\left(\frac{n^{.5}\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/l}\right)\right).
\end{aligned}$$

In Algorithm 3, step 1 and step 3 has total number of $\mathcal{O}(n^2)$ operations, and the algorithm complexity is dominated by step 2. Now each outer loop of PDASMD has $\mathcal{O}(n^2 + nl)$ operations, thus the total number of arithmetic operations of Algorithm 3 is

$$\mathcal{O}\left(\frac{n^{1.5}\|C\|_\infty}{\epsilon}(n+l)\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/l}\right)\right) = \widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right).$$

Case 2: $\|\cdot\|_H = \|\cdot\|_\infty$. Then $\|\cdot\|_{H,*} = \|\cdot\|_1$, and Theorem 1 implies that

$$\|\mathbb{E}[b - A\widetilde{x}]\|_1 \leq \frac{2\left[l\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{49}$$

$$f(\mathbb{E}(\widetilde{x})) - f(x^*) \leq \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right], \tag{50}$$

where $\bar{L} = \frac{5}{\eta}$. Now R is an upper bound for $\|\lambda^*\|_\infty$, and again by Lemma 3.2 in Lin et al. (2019), $R = \eta(R' + .5)$ for $R' = \mathcal{O}(\|C\|_\infty \log(n)/\epsilon)$.

Thus the stopping criteria in step 2 of Algorithm 3 is satisfied for

$$\begin{aligned}
S =& \max\left\{\mathcal{O}\left(\sqrt{\frac{\bar{L}R}{\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R\gamma}{l\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2}{\epsilon}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2\gamma}{l\epsilon}}\right)\right\} \\
=& \max\left\{\mathcal{O}\left(\sqrt{\frac{\log(n)\|C\|_\infty^2}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\gamma\|C\|_\infty^2}{l\epsilon^2}}\right),\right. \\
&\left.\mathcal{O}\left(\sqrt{\frac{\|C\|_\infty^2\log n}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\|C\|_\infty^2\gamma}{l\epsilon^2}}\right)\right\} \\
=& \mathcal{O}\left(\frac{\|C\|_\infty}{\epsilon}\max\left(\sqrt{\log n}, \sqrt{\log(n)\gamma/n}\right)\right) \\
=& \mathcal{O}\left(\frac{\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/n}\right)\right).
\end{aligned}$$

Now each outer loop of PDASMD has $\mathcal{O}(n^2 + nl) = \mathcal{O}(n^2)$ operations, thus the total number of arithmetic operations of Algorithm 3 is

$$\mathcal{O}\left(\frac{n^2\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/n}\right)\right) = \widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right).$$

□

E Stochastic Sinkhorn Algorithm and Proof of Computational Complexity

We first describe the Stochastic Sinkhorn algorithm. In the Stochastic Sinkhorn algorithm, the following definitions are used:

Definition 4 (Increasing probability function)**.** *An increasing probability function $\Psi : \mathbb{R}_+^p \to \Delta_p$ is such that*

$$\Psi(\boldsymbol{h}) = \left(\frac{g(h_i)}{\sum_i g(h_i)} \right)_i ,$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is an increasing positive function.

Definition 5 (KL violation)**.** *For a matrix $M \in \mathbb{R}_+^{p\times p}$ and two vectors $\boldsymbol{p}, \boldsymbol{q} \in \Delta_p$, define the KL violation*

$$\rho(M; \boldsymbol{p}, \boldsymbol{q}) = \left[\begin{array}{l} (\mathcal{KL}(p_i \| (M\mathbf{1})_i))_{i=1,\dots,p} \\ (\mathcal{KL}(q_j \| (M^T\mathbf{1})_j))_{j=1,\dots,p} \end{array} \right]. \tag{51}$$

The Stochastic Sinkhorn algorithm for solving problem (2) is as following:

Algorithm 4: Stochastic Sinkhorn

```
Input: $C, \boldsymbol{p}', \boldsymbol{q}', \Psi, \eta$
Calculate $A = \exp(-C/\eta)$ where all the operations are element-wise;
Initialize: $\boldsymbol{u}^0 = \boldsymbol{v}^0 = \mathbf{1}$;
for k=0,...,K-1 do
    Calculate $\boldsymbol{h} = \Psi(\rho(X(\boldsymbol{u}^k, \boldsymbol{v}^k); \boldsymbol{p}', \boldsymbol{q}'))$, where $X(\boldsymbol{u}^k, \boldsymbol{v}^k) = diag(\boldsymbol{u}^k) A diag(\boldsymbol{v}^k)$;
    Sample index $I$ with
        $P(I = i) = h_i, \forall i \in \{1, 2, ..., 2n\}.$
    if $I \leq n$ then
        $\boldsymbol{u}^{k+1} = (u_1^k, \dots, u_{I-1}^k, p_I'/(A\boldsymbol{v}^k)_I, u_{I+1}^k, \dots, u_n^k)^T$, $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k$;
    else
        $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k$, $\boldsymbol{v}^{k+1} = (v_1^k, \dots, v_{I-n-1}^k, q_{I-n}'/(A^T\boldsymbol{u}^k)_{I-n}, v_{I-n+1}^k, \dots, v_n^k)^T$.
    end if
end for
Output: $\widetilde{X} = diag(\boldsymbol{u}^K) A diag(\boldsymbol{v}^K)$.
```

To find a ϵ-solution to OT, an extra rounding step is required. The full procedure is given in Algorithm 5.

Algorithm 5: Approximating OT by Stochastic Sinkhorn

```
Input: Accuracy $\epsilon > 0$, $\eta = \frac{\epsilon}{4\log(n)}$ and $\epsilon' = \frac{\epsilon}{8\|C\|_\infty}$.
Step 1: Let $\boldsymbol{p}' \in \Delta_n$ and $\boldsymbol{q}' \in \Delta_n$ be
    $\begin{pmatrix} \boldsymbol{p}' \\ \boldsymbol{q}' \end{pmatrix} = \left(1 - \frac{\epsilon'}{8}\right) \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{pmatrix} + \frac{\epsilon'}{8n} \begin{pmatrix} \mathbf{1}_n \\ \mathbf{1}_n \end{pmatrix}.$
Step 2: Compute $\widetilde{X}$ by Stochastic Sinkhorn until $\|\widetilde{X}\mathbf{1} - \boldsymbol{p}'\|_1 + \|\widetilde{X}^T\mathbf{1} - \boldsymbol{q}'\|_1 \leq \frac{\epsilon'}{2}$.
Step 3: Round $\widetilde{X}$ to $\widehat{X}$ by Algorithm 2 in Altschuler et al. (2017) such that $\widehat{X}\mathbf{1}_n = \boldsymbol{p}$, $\widehat{X}^T\mathbf{1}_n = \boldsymbol{q}$.
Output: $\widehat{X}$.
```

We now prove the computational complexity of Stochastic Sinkhorn in Theorem 3. To prove it, we first need the convergence of Algorithm 4, which we show in following Lemma.

Lemma 6. *For a given $\epsilon > 0$, we have that Algorithm 4 returns a matrix $\widetilde{X}$ such that*

$$\mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] \leq \epsilon$$

in the number of iterations

$$k \leq 2 + 112nR/\epsilon.$$

Proof. Denote $(x^k, y^k) := (\log u^k, \log v^k)$ and the dual function $f(x,y) = \sum_{i,j} A_{i,j}\exp(x_i + y_j) - \langle p', x\rangle - \langle q', y\rangle$. Denote $E_k = \mathbb{E}[\|X(u^k,v^k)\mathbf{1} - p'\|_1 + \|X(u^k,v^k)^T\mathbf{1} - q'\|_1]$. By (21) in Abid and Gower (2018), Algorithm 4 has

$$E[f(x^k,y^k) - f(x^{k+1},y^{k+1})] > \frac{E_k^2}{28n}. \tag{52}$$

Since Algorithm 4 only updates one element in u or v, and the updating rule for that element is the same as Greenkhorn, we have that Corollary 3.3 in Lin et al. (2019) holds. Adding expectations to both sides, we get:

$$E[f(x^k,y^k) - f(x^*,y^*)] \leq 4RE_k. \tag{53}$$

Let $\delta_k = E[f(x^k,y^k) - f(x^*,y^*)]$, then by inequalities (52) and (53) we have

$$\delta_k - \delta_{k+1} \overset{(52)}{\geq} \frac{E_k^2}{28n} \overset{(53)}{\geq} \frac{\delta_k^2}{448nR^2}. \tag{54}$$

That is,

$$\delta_k - \delta_{k+1} \geq \max\left\{\frac{\epsilon^2}{28n}, \frac{\delta_k^2}{448nR^2}\right\}. \tag{55}$$

We adopt the strategy in Dvurechensky et al. (2018) to split the process of $\{\delta_k\}$ into two halves:

First, consider the process from δ_1 to δ_t:

$$\frac{\delta_t}{448nR^2} \leq \frac{\delta_{t-1}}{448nR^2} - \left(\frac{\delta_{t-1}}{448nR^2}\right)^2 \leq \frac{1}{t-1+448nR^2/\delta_1} \Rightarrow t \leq 1 + \frac{448nR^2}{\delta_t} - \frac{448nR^2}{\delta_1}.$$

Second, consider the process from δ_t to δ_{t+m}:

$$\delta_{t+m} \leq \delta_t - \frac{\epsilon^2 m}{28n} \Rightarrow m \leq \frac{28n(\delta_t - \delta_{t+m})}{\epsilon^2}.$$

So the total number of iterations $k = t+m$ can be optimized over δ_t, i.e.

$$k \leq \min_{\delta_t \in (0,\delta_1]} \left(2 + \frac{448nR^2}{\delta_t} - \frac{448nR^2}{\delta_1} + \frac{28n\delta_t}{\epsilon^2}\right) \leq 2 + \frac{112nR}{\epsilon}.$$

□

Then we can prove Theorem 3 as follows:

Proof. By Lemma 6, we have $\mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] \leq \epsilon'/2$ for the number of iterations $k = 2 + 224nR/\epsilon'$. Thus for this k, we also have

$$\mathbb{E}[\|\widetilde{X}\mathbf{1} - p\|_1 + \|\widetilde{X}^T\mathbf{1} - q\|_1] \leq \mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] + \|p - p'\|_1 + \|q - q'\|_2 \leq \epsilon'.$$

By Theorem 1 in Altschuler et al. (2017),

$$\mathbb{E}\langle C, \widehat{X}\rangle - \langle C, X^*\rangle \leq \epsilon/2 + 4(\mathbb{E}[\|\widetilde{X}\mathbf{1} - p\|_1 + \|\widetilde{X}^T\mathbf{1} - q\|_1])\|C\|_\infty \leq \epsilon,$$

which is an $\epsilon-$solution.

Now calculate the number of arithmetic operations, step 1 requires $\mathcal{O}(n)$; step 2 requires k iterations of Algorithm 4, each iteration requires $\mathcal{O}(n)$ operation (Abid and Gower, 2018); step 3 requires $\mathcal{O}(n^2)$ operations (Altschuler et al., 2017). Thus the total number of operations is

$$\mathcal{O}(n^2R/\epsilon') = \mathcal{O}(n^2\|C\|_\infty^2 \log n/\epsilon^2).$$

□

F PDASMD-B Algorithm and the Convergence Rate

In this Section, we prove the convergence of PDASMD-B. The convergence rate of PDASMD-B is in the following theorem:

Theorem 4 (Convergence of PDASMD-B)**.** *In Algorithm 2, assume that the dual optimal solution has $\|\lambda^*\|_H \leq R$. Then we have the convergence of Algorithm 2 as:*

$$\|\mathbb{E}[\boldsymbol{b} - A\boldsymbol{x}^{S-1}]\|_{H,*} \leq \frac{2\left[(1+(l-1)/B)\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{56}$$

$$f(\mathbb{E}(\boldsymbol{x}^{S-1})) - f(\boldsymbol{x}^*) \leq \frac{2}{S^2 l}\left[(1+(l-1)/B)\bar{L}R^2 + 18\bar{L}R^2\gamma\right]. \tag{57}$$

The key to prove Theorem 4 is to find an analogue to Lemma 2, which we do in following steps.

Lemma 7 (Variance upper bound)**.**

$$\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2] \leq \frac{8\bar{L}}{B}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle). \tag{58}$$

Proof. Each ϕ_i is convex and L_i-smooth, then by Theorem 2.1.5. in Nesterov (2003) we have

$$\|\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v}^s)\|_{H,*}^2 \leq 2L_i(\phi_i(\widetilde{v}^s) - \phi_i(v_{k+1}) - \langle\nabla\phi_i(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle). \tag{59}$$

Take expectation with respect to the randomness of index set I, note that all indexes in I are independently selected, we have

$$\begin{aligned}
&\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2] \\
=&\frac{1}{B}\mathbb{E}\left[\left\|\nabla\phi(\widetilde{v}^s) + \frac{1}{mp_i}(\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v}^s)) - \nabla\phi(v_{k+1})\right\|_{H,*}^2\right] \\
\leq&\frac{1}{B}\mathbb{E}\left[2\left\|\frac{1}{mp_i}(\nabla\phi_i(\widetilde{v}^s) - \nabla\phi_i(v_{k+1}))\right\|_{H,*}^2 + 2\|\nabla\phi(\widetilde{v}^s) - \nabla\phi(v_{k+1})\|_{H,*}^2\right] \\
\overset{(59)}{\leq}&\frac{1}{B}\mathbb{E}\left[4\frac{L_i}{m^2p_i^2}(\phi_i(\widetilde{v}^s) - \phi_i(v_{k+1}) - \langle\nabla\phi_i(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle) + 2\|\nabla\phi(\widetilde{v}^s) - \nabla\phi(v_{k+1})\|_{H,*}^2\right] \\
=&\frac{1}{B}[4\bar{L}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle) + 2\|\nabla\phi(\widetilde{v}^s) - \nabla\phi(v_{k+1})\|_{H,*}^2] \\
\overset{(59)}{\leq}&\frac{8\bar{L}}{B}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle)
\end{aligned}$$

□

Lemma 8 (Coupling step 1, batch version)**.** *Consider one inner loop of Algorithm 2, where the randomness only comes from the choice of I. It satisfies that for $\forall u$:*

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle \\
\leq&\frac{\alpha_s}{\tau_{1,s}}\left\{\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k-1}\rangle\right\} \\
&+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

Proof. One can easily check that the Lemma E.1. and Lemma E.3. in Allen-Zhu (2017) holds for the batch version of PDASMD, where $\psi(\cdot) = 0$ in these two lemmas for our case. Then we have

$$\begin{aligned}
\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] \geq&\mathbb{E}\left[-\min_y\left\{\frac{9\bar{L}}{2}\|y - v_{k+1}\|_H^2 + \langle\widetilde{\nabla}_{k+1}, y - v_{k+1}\rangle\right\}\right] \\
&- \frac{1}{16\bar{L}}\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2].
\end{aligned} \tag{60}$$

$$\alpha_s\langle\widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \leq -\frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u). \tag{61}$$

Then

$$\begin{aligned}\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - u\rangle &= \alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle + \alpha_s\langle\widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \\ &\overset{(61)}{\leq} \alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u).\end{aligned} \tag{62}$$

To bound $\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2$, consider the variable $v := \tau_{1,s}z_{k+1} + \tau_2\widetilde{v}^s + (1 - \tau_{1,s} - \tau_2)y_k$, then $v_{k+1} - v = \tau_{1,s}(z_k - z_{k+1})$. We have that

$$\begin{aligned}&\mathbb{E}\left[\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2\right] \\ =&\mathbb{E}\left[\frac{\alpha_s}{\tau_{1,s}}\langle\widetilde{\nabla}_{k+1}, v_{k+1} - v\rangle - \frac{1}{2\tau_{1,s}^2}\|v_{k+1} - v\|_H^2\right] \\ =&\mathbb{E}\left[\frac{\alpha_s}{\tau_{1,s}}\left(\langle\widetilde{\nabla}_{k+1}, v_{k+1} - v\rangle - \frac{9\bar{L}}{2}\|v_{k+1} - v\|_H^2\right)\right] \\ \overset{(60)}{\leq}& \frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \frac{1}{16\bar{L}}\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2]\right) \\ \overset{(58)}{\leq}& \frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \frac{1}{2B}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle)\right).\end{aligned} \tag{63}$$

Take expectation on both sides of inequality (62), plug in inequality (63) and notice that $\mathbb{E}[\langle\widetilde{\nabla}_{k+1}, z_k - u\rangle] = \langle\nabla\phi(v_{k+1}), z_k - u\rangle$ and $\tau_2 = \frac{1}{2B}$, we get the desired bound. □

The rest of the proof for Theorem 4 is simply repeating the steps in Appendix B, except that we replace Lemma 2 with Lemma 8. So we omit the details of the proof.

G Details of Numerical Study

Data description. We use both synthetic and real grey-scale images as the marginal distribution. For the simulated data, we follow the data generation mechanism in Altschuler et al. (2017); Xie et al. (2022). The images are generated by randomly positioning a square foreground on a background, with the foreground occupying about 20% of the space. The foreground has each pixel value randomly drawn from uniform $[0, 3]$, and the background has each pixel value randomly drawn from uniform $[0, 1]$. Figure 2 shows some examples of the generated images.

For the real data, we randomly sample from the hand-written MNIST data set. Then we downscale the images to adjust the size of the marginal distribution. We also add a background with relatively small intensity to the down-scaled images to avoid numerical issue. With the marginal distribution determined, the cost matrix has each element calculated as the l_1 distance between the pixel locations on the image.

Algorithm implementation. We compare the computational efficiency of the algorithms by measuring the number of arithmetic operations they use for finding an ϵ-solution of the OT between two marginal distributions for a fixed ϵ. To achieve this, all the algorithms are run with a rounding step. Thus for PDASMD, we run Algorithm 3. In particular, for step 2 of Algorithm 3, the PDASMD algorithm is run with the number of inner loops set to the problem size, $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$. We run the PDASGD algorithm by changing $\|\cdot\|_H$ to $\|\cdot\|_2$ compared to the PDASMD. Note that the PDASGD algorithm is essentially equivalent to that of Xie et al. (2022). We also run APDAGD (Dvurechensky et al., 2018), AAM (Guminov et al., 2021), Sinkhorn (Dvurechensky et al., 2018), APDRCD (Guo et al., 2020) and Stochastic Sinkhorn (Algorithm 5) for comparison. The implementation of all the algorithms above follows their standard definitions; there is no hyper-parameter to tune.

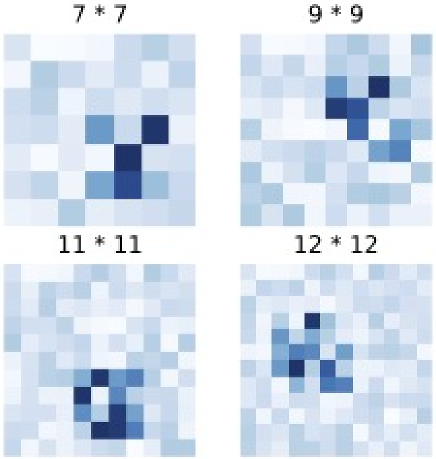


Figure 2: Synthetic image example.

We also implement experiment for PDASMD-B on both synthetic and real
data. For a fixed pair of marginals, PDASMD-B is implemented for a sequence of batch sizes. The number of inner loops is set to be the problem size divided by the batch size, which matches the setting in Corollary 1, and we take $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$, which are the same as the experiment of PDASMD. For each batch size, the total number of computations and the running time are recorded.

All our experiments are run on Google Colab using **NO** GPU or TPU accelerator. We attach the full code and data to reproduce our results in the Supplemental Material.

H Application of the PDASMD Algorithm to Machine Learning Tasks

The Optimal Transport can be applied to some modern machine learning tasks such as domain adaptation and color transfer. In this section, we illustrate that our PDASMD algorithm, when applied to OT, can solve those problems.

Domain Adaptation. This experiment aims to show that our PDASMD algorithm, when applied to OT, can successfully perform domain adaptation. In short, domain adaptation means transferring knowledge from a source domain to a target domain for which data have different probability density functions. For more details on the domain adaptation problem description and its OT formulation, see Courty et al. (2015).

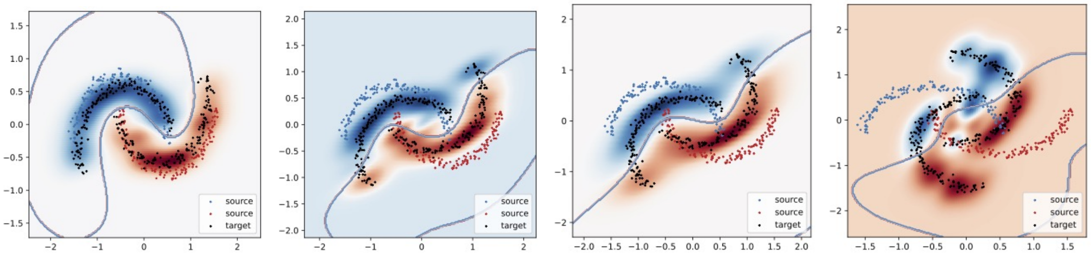
We use the two-moons example to illustrate the application of our PDASMD algorithm on domain adaptation. The two moons example uses simulated data. The source domain consists of two entangled moons, where each moon represents one class. The target domain is built by applying a rotation to the two moons. We sample 150 labeled data points from each moon as our source domain. The target domain consists of the same number of samples, where the samples are independent of the source domain and are unlabeled. We use the labeled source domain data, transfer them to the target domain by OT using our PDASMD algorithm, and learn an SVM classifier with the Gaussian kernel using the transferred source data on the target domain. We test the generalization performance on 2,000 samples that follow the same distribution as the target domain.

Figure 3 shows the domain adaptation result. In Figure 3, we plot the source domain, target domain (for different rotation angles), the transformed density, and decision boundaries. From the plots, we see that the transformed density reasonably fits the major parts of the target domain when the rotation angle is not too large ($\leq 50°$). This shows that the PDASMD algorithm successfully performs the domain adaptation.

We report the generalization performance of the domain adaptation in Table 2. We have three columns: the rotation degree of the target domain in the two moons example, the mean classification error when the domain adaptation is performed using our PDASMD algorithm, and the mean classification error of OT-IT in Courty et al. (2015) (where they solve entropic OT by the Sinkhorn algorithm). From Table 2, we see that our PDASMD algorithm performs better than the Sinkhorn when the rotation degree is large ($> 50°$).

Table 2: Mean Classification Error over 10 Repetitions of the Two Moons Example.

ROTATION DEGREE	PDASMD - CLASSIFICATION ERROR	OT-IT (COURTY ET AL., 2015) CLASSIFICATION ERROR
$10°$	0.022	**0**
$20°$	0.054	**0.007**
$30°$	**0.043**	0.054
$40°$	0.169	**0.102**
$50°$	**0.221**	**0.221**
$70°$	**0.317**	0.398
$90°$	**0.488**	0.508



(a) rotation = 20° (b) rotation = 40° (c) rotation = 50° (d) rotation = 90°

Figure 3: Two Moons Example

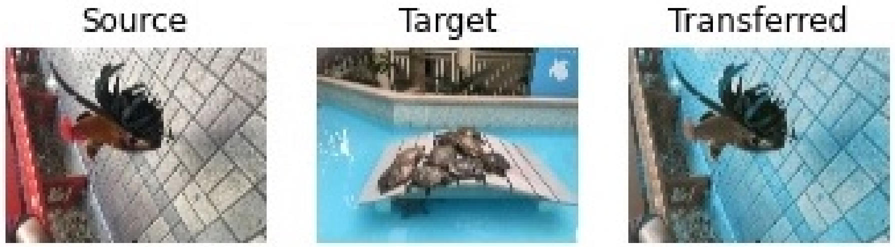


Figure 4: Color Transfer Example

Color Transfer. This experiment shows that our PDASMD algorithm successfully performs the color transfer task. The color transfer takes two input images and imposes the color palette of the first image onto a second one. Color transfer can be formulated as an OT problem. For more details see references Ferradans et al. (2014); Rabin et al. (2014).

For an example of the color transfer problem, we apply our PDASMD algorithm to solve the corresponding OT problem. We show the color transfer result in Figure 4. Though we cannot evaluate the color transfer result quantitatively, one can tell from Figure 4 that the color of the target image has been successfully transferred to the source image. This shows that our PDASMD algorithm successfully performs the color transfer task.