On Partial Optimal Transport: Revising the Infeasibility of Sinkhorn and Efficient Gradient Methods

Anh Duc Nguyen[1], Tuan Dung Nguyen[2], Quang Minh Nguyen[3], Hoang H. Nguyen[4],
Lam M. Nguyen[5], Kim-Chuan Toh[1, 6]

[1]Department of Mathematics, National University of Singapore, Singapore
[2]Department of Computer and Information Science, University of Pennsylvania, USA
[3]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA
[4]School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
[5]IBM Research, Thomas J. Watson Research Center, USA
[6]Institute of Operations Research and Analytics, National University of Singapore, Singapore
anh_duc@u.nus.edu, joshtn@seas.upenn.edu, nmquang@mit.edu,
hnguyen455@gatech.edu, LamNguyen.MLTD@ibm.com, mattohkc@nus.edu.sg

Abstract

This paper studies the Partial Optimal Transport (POT) problem between two unbalanced measures with at most n supports and its applications in various AI tasks such as color transfer or domain adaptation. There is hence the need for fast approximations of POT with increasingly large problem sizes in arising applications. We first theoretically and experimentally investigate the infeasibility of the state-of-the-art Sinkhorn algorithm for POT due to its incompatible rounding procedure, which consequently degrades its qualitative performance in real world applications like point-cloud registration. To this end, we propose a novel rounding algorithm for POT, and then provide a feasible Sinkhorn procedure with a revised computation complexity of $\widetilde{\mathcal{O}}(n^2/\varepsilon^4)$. Our rounding algorithm also permits the development of two first-order methods to approximate the POT problem. The first algorithm, Adaptive Primal-Dual Accelerated Gradient Descent (APDAGD), finds an ε-approximate solution to the POT problem in $\widetilde{\mathcal{O}}(n^{2.5}/\varepsilon)$, which is better in ε than revised Sinkhorn. The second method, Dual Extrapolation, achieves the computation complexity of $\widetilde{\mathcal{O}}(n^2/\varepsilon)$, thereby being the best in the literature. We further demonstrate the flexibility of POT compared to standard OT as well as the practicality of our algorithms on real applications where two marginal distributions are unbalanced.

Introduction

Optimal Transport (OT) (**??**), which seeks a minimum-cost coupling between two balanced measures, is a well-studied topic in mathematics and operations research. With the introduction of entropic regularization (**?**), the scalability and speed of OT computation have been significantly improved, facilitating its widespread applications in machine learning such as domain adaptation (**?**), and dictionary learning (**?**). However, OT has a stringent requirement that the input measures must have equal total masses (**?**), hindering its practicality in various other machine learning applications, which require an optimal matching between two measures with unbalanced masses, such as averaging of neuroimaging data (**?**) and image classification (**??**).

In response to such limitations, Partial Optimal Transport (POT), which explicitly constrains the mass to be transported between two unbalanced measures, was proposed. It has been studied from the perspective of partial differential equations by theorists (**??**). Practically, the relaxation of the marginal constraints, which are strictly imposed by the standard OT, and the control over the total mass transported grant POT immense flexibility compared to OT (**?**) and more robustness to outliers (**?**). POT has been deployed in various recent AI applications such as color transfer (**?**), graph neural networks (**?**), graph matching (**?**), partial covering (**?**), point set registration (**?**), and robust estimation (**?**).

Despite its potential applicability, POT still suffers from the computation bottleneck, whereby the more intricate structural constraints imposed on admissible couplings have hindered the direct adaptation of any efficient OT solver in the literature. Currently, the literature (**??**) relies on reformulating POT into an extended OT problem under additional assumptions on the input masses, which can then be solved via existing OT methods, and finally retrieves an admissible POT coupling from the solution to the extended OT problem. This approach has two fundamental drawbacks. First, in the reformulated OT problem, the maximum entry of the extended cost matrix is increased (**?**, Proposition 1), which will always worsen the computational complexity since most efficient algorithms for standard OT (**???**) depend on this maximum entry in their complexities. Second, we discover, more details in the Revisiting Sinkhorn section, that although Sinkhorn for POT proposed by (**?**) achieves the best known complexity of $\widetilde{\mathcal{O}}(n^2/\varepsilon^2)$, it in fact always outputs a *strictly infeasible* solution to the POT problem. In brief, by discarding the last row and column of the reformulated OT solution obtained by Sinkhorn, the POT solution, transportation matrix $\mathbf{X}$, will violate the equality constraint $\mathbf{1}^\top \mathbf{X} \mathbf{1} = s$, which controls the total transported mass. Violating this equality constraint can degrade the results of practical applications in robust regimes such as point cloud registration (**?**) and mini-batch OT (**?**) (refer to Remark **??** and our Point Cloud Registration experiment). We theoretically justify the ungroundedness of Sinkhorn for POT in the Revisiting Sinkhorn section and
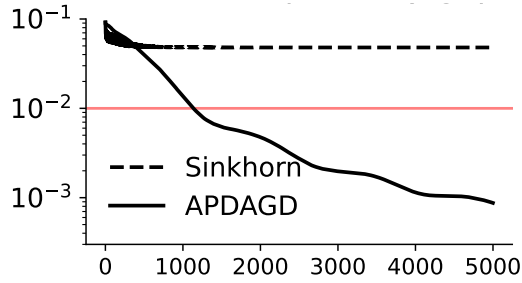
Figure 1: Primal optimality gap against optimization rounds achieved by Sinkhorn (**?**) and APDAGD for POT. The marginal distributions are taken from a color transfer application described in our Numerical Experiments section, and the red horizontal line depicts the pre-defined tolerance (ε) for both algorithms.

empirically verify this claim in several applications in the Numerical Experiment section. Here, in Figure **??**, we specifically investigate Sinkhorn infeasibility in a color transfer example (detailed experimental setup in the later Numerical Experiment section). We show that the optimality gap produced by Sinkhorn is unable to reduce lower than ε, the tolerance of the problem (red line). In other words, Sinkhorn fails to produce an adequate ε-approximate POT solution.

To the best of our knowledge, the invalidity of Sinkhorn means there is currently no efficient method for solving POT in the literature. We attribute this to the fact that while the equivalence between POT and extended OT holds at optimality, all efficient OT solvers instead only output an approximation of the optimum value before projecting it back to the feasible set. However, the well-known rounding algorithm by (**?**), which is specifically designed for OT, does not guarantee to respect the more intricate structural constraints of POT, resulting in the invalidity of Sinkhorn. Motivated by this challenge and the success of optimization literature for OT, we raise the following central question of this paper:

Can we design a rounding algorithm for POT and then utilize it to develop efficient algorithms for POT that even match the best-known complexities of those for OT?

We affirmatively answer this question and formally summarize our contributions as follows.

- We theoretically and experimentally show the infeasibility of the state-of-the-art Sinkhorn algorithm for POT due to its incompatible rounding algorithm. We propose a novel POT rounding procedure ROUND-POT (Rounding Algorithm Section), which projects an approximate solution onto the feasible POT set in $\mathcal{O}(n^2)$ time.
- From our theoretical bounds of the Sinkhorn constraint violations and the newly introduced ROUND-POT, we provide a revised procedure for Sinkhorn which will return a feasible POT solution. We also establish the revised complexity of Sinkhorn for POT (Table **??**).
- Predicated on our novel dual formulation for entropic regularized POT objective, our proposed Adaptive Primal-Dual Accelerated Gradient Descent (APDAGD) algorithm for POT finds an ε-approximate solution in $\widetilde{\mathcal{O}}(n^{2.5}/\varepsilon)$, which is better in ε than the revised Sinkhorn. Various

experiments on synthetic and real datasets and with applications such as point cloud registration, color transfer, and domain adaptation illustrate not only our algorithms' favorable performance against the pre-revised Sinkhorn but also the versatility of POT compared to OT.

- Motivated by our novel rounding algorithm, we further reformulate the POT problem with ℓ_1 penalization as a minimax problem and propose Dual Extrapolation (DE) framework for POT. We prove that DE algorithm can theoretically achieve $\widetilde{\mathcal{O}}(n^2/\varepsilon)$ computational complexity, thereby being the best in the POT literature to the best of our knowledge (Table **??**).

Preliminaries

Notation

The set of non-negative real numbers is $\mathbb{R}_+$. We use bold capital font for matrices (e.g., $\mathbf{A}$) and bold lowercase font for vectors (e.g., $\mathbf{x}$). For an $m \times n$ matrix $\mathbf{X}$, $\mathrm{vec}(\mathbf{X})$ denotes the (mn)-dimensional vector obtained by concatenating the rows of $\mathbf{X}$ and transposing the result. Entrywise multiplication and division for matrices and vectors are respectively denoted by $\odot$ and $\oslash$. For $1 \leq p \leq \infty$, let $\|\cdot\|_p$ be the ℓ_p-norm of matrix or vector. For matrices, $\|\cdot\|_{p \to q}$ is the operator norm: $\|\mathbf{A}\|_{p \to q} = \sup_{\|\mathbf{x}\|_p = 1} \|\mathbf{A}\mathbf{x}\|_q$. Three specific cases are considered in this paper: for $q \in \{1, 2, \infty\}$, $\|\mathbf{A}\|_{1 \to q}$ is the largest ℓ_q norm of any column of $\mathbf{A}$. We use $\|\mathbf{A}\|_{\max}$ and $\|\mathbf{A}\|_{\min}$ to denote the maximum and minimum entries in absolute value of a matrix $\mathbf{A}$, respectively. The n-vectors of zeros and of ones are respectively denoted by $\mathbf{0}_n$ and $\mathbf{1}_n$. The $(n-1)$-dimensional probability simplex is $\Delta_n = \{\mathbf{v} \in \mathbb{R}_+^n : \mathbf{v}^\top \mathbf{1}_n = 1\}$.

Partial Optimal Transport

Consider two discrete distributions $\mathbf{r}, \mathbf{c} \in \mathbb{R}_+^n$ with possibly different masses. POT seeks a transport plan $\mathbf{X} \in \mathbb{R}_+^{n \times n}$ which maps $\mathbf{r}$ to $\mathbf{c}$ at the lowest cost. Since the masses at two marginals may differ, only a total mass s such that $0 \leq s \leq \min\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\}$ is allowed to be transported (**??**). Formally, the POT problem is written as

$$\mathbf{POT}(\mathbf{r}, \mathbf{c}, s) = \min \langle \mathbf{C}, \mathbf{X} \rangle \text{ s.t. } \mathbf{X} \in \mathcal{U}(\mathbf{r}, \mathbf{c}, s), \quad (1)$$

where $\mathcal{U}(\mathbf{r}, \mathbf{c}, s)$ is defined as

$$\left\{ \mathbf{X} \in \mathbb{R}_+^{n \times n} : \mathbf{X}\mathbf{1}_n \leq \mathbf{r}, \mathbf{X}^\top \mathbf{1}_n \leq \mathbf{c}, \mathbf{1}_n^\top \mathbf{X}\mathbf{1}_n = s \right\},$$

i.e. the feasible set for the transport map $\mathbf{X}$ is and $\mathbf{C} \in \mathbb{R}_+^{n \times n}$ is a cost matrix. The goal of this paper is to derive efficient algorithms to find an ε-approximate solution to $\mathbf{POT}(\mathbf{r}, \mathbf{c}, s)$, pursuant to the following definition.

Definition 1 (ε-approximation). For $\varepsilon \geq 0$, the matrix $\mathbf{X} \in \mathbb{R}_+^{n \times n}$ is an ε-approximate solution to $\mathbf{POT}(\mathbf{r}, \mathbf{c}, s)$ if $\mathbf{X} \in \mathcal{U}(\mathbf{r}, \mathbf{c}, s)$ and

$$\langle \mathbf{C}, \mathbf{X} \rangle \leq \min \langle \mathbf{C}, \mathbf{X}' \rangle + \varepsilon \text{ s.t. } \mathbf{X}' \in \mathcal{U}(\mathbf{r}, \mathbf{c}, s).$$

Algorithm	Regularizer	Cost per iteration	Iteration complexity
Iterative Bregman projections (**?**)	Entropic	Unspecified	Unspecified
(Infeasible) Sinkhorn (**?**)	Entropic	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}(1/\varepsilon^2)$
(Feasible) Sinkhorn (**This paper**)	Entropic	$\mathcal{O}\left(n^2\right)$	$\widetilde{\mathcal{O}}\left(1/\varepsilon^4\right)$
APDAGD (**This paper**)	Entropic	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}\left(\sqrt{n}/\varepsilon\right)$
Dual Extrapolation (**This paper**)	Area-convex	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}\left(1/\varepsilon\right)$

Table 1: Type of regularizers and orders of complexity for four algorithms for POT approximation.

To aid the algorithmic design in the following sections, we introduce two new slack variables $\mathbf{p}, \mathbf{q} \in \mathbb{R}_+^n$ and equivalently express problem (**??**) as

$$\min_{\mathbf{X}\geq 0, \mathbf{p}\geq 0, \mathbf{q}\geq 0} \langle \mathbf{C}, \mathbf{X}\rangle \qquad (2)$$

$$\text{s.t. } \mathbf{X}\mathbf{1}_n + \mathbf{p} = \mathbf{r}, \mathbf{X}^\top \mathbf{1}_n + \mathbf{q} = \mathbf{c}, \mathbf{1}_n^\top \mathbf{X}\mathbf{1}_n = s. \qquad (3)$$

We also study a convenient equivalent formulation of this problem:

$$\min_{\mathbf{x}\geq \mathbf{0}} \langle \mathbf{d}, \mathbf{x}\rangle \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \qquad (4)$$

where we perform vectorization with $\mathbf{d}^\top = (\text{vec}(\mathbf{C})^\top, \mathbf{0}_{2n}^\top)$ and $\mathbf{x}^\top = (\text{vec}(\mathbf{X})^\top, \mathbf{p}^\top, \mathbf{q}^\top)$. The constraints in Equation **??** are encoded in $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{(2n+1)\times(n^2+2n)}$ and $\mathbf{b} \in H := \mathbb{R}^{2n+1}$ such that $(\mathbf{A}\mathbf{x})^\top = ((\mathbf{X}\mathbf{1} + \mathbf{p})^\top, (\mathbf{X}^\top\mathbf{1} + \mathbf{q})^\top, \mathbf{1}^\top\mathbf{X}\mathbf{1})$ and $\mathbf{b}^\top = (\mathbf{r}^\top, \mathbf{c}^\top, s)$. In other words, the linear operator $\mathbf{A}$ has the form

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}' & \mathbf{I}_{2n} \\ \mathbf{1}_{n^2}^\top & \mathbf{0}_{2n} \end{pmatrix},$$

where $\mathbf{A}'$ is the edge-incidence matrix of the underlying bipartite graph in OT problems (**??**).

Revisiting Sinkhorn for POT

We can reformulate Problem (**??**) by adding *dummy points* and extending the cost matrix as

$$\widetilde{\mathbf{C}} = \begin{pmatrix} \mathbf{C} & \mathbf{0}_n \\ \mathbf{0}_n^\top & A \end{pmatrix} \in \mathbb{R}_+^{(n+1)\times(n+1)},$$

where $A > \max(C_{i,j})$ (**?**). Then the two marginals are augmented to $(n+1)$-dimensional vectors as $\widetilde{\mathbf{r}}^\top = (\mathbf{r}^\top, \|\mathbf{c}\|_1 - s)$ and $\widetilde{\mathbf{c}}^\top = (\mathbf{c}^\top, \|\mathbf{r}\|_1 - s)$. (**?**, Proposition 1) show that one can obtain the solution POT by solving this extended OT problem with balanced marginals $\widetilde{\mathbf{r}}, \widetilde{\mathbf{c}}$ and cost matrix $\widetilde{\mathbf{C}}$. In particular, if the OT problem admits an optimal solution of the form

$$\widetilde{\mathbf{X}} = \begin{pmatrix} \bar{\mathbf{X}} & \widetilde{\mathbf{p}} \\ \widetilde{\mathbf{q}}^\top & \widetilde{X}_{n+1,n+1} \end{pmatrix} \in \mathbb{R}_+^{(n+1)\times(n+1)},$$

then $\bar{\mathbf{X}} \in \mathbb{R}_+^{n\times n}$ is the solution to the original POT. (**?**) seeks an approximate solution to the extended OT problem using the Sinkhorn algorithm (see Algorithm **??** in the Appendix). Then the rounding procedure by (**?**) is applied to the solution to give a primal feasible matrix. While the two POT inequality constraints are satisfied, we discover in the following Theorem that the equality constraint $\mathbf{1}^\top\bar{\mathbf{X}}\mathbf{1} = s$ is violated. The proof is in Appendix, Revisiting Sinkhorn for POT section.

Theorem 2. *For a POT solution $\bar{\mathbf{X}}$ from (**?**), the constraint violation $V := \mathbf{1}^\top\bar{\mathbf{X}}\mathbf{1} - s$ can be bounded as*

$$\tilde{\mathcal{O}}\left(\frac{\|\mathbf{C}^2\|_{max}}{A}\right) \geq V \geq \exp\left(\frac{-12A\log n}{\varepsilon} - \mathcal{O}(\log n)\right).$$

Feasible Sinkhorn Procedure: With these bounds, we deduce that in order for Sinkhorn to be feasible, one needs to **both** utilize our ROUND-POT and choose a sufficiently large A (Theorem **??**) as opposed to the common practice of picking A a bit larger than 1 (**?**). We derive the revised complexity of Sinkhorn for POT in the following theorem.

Theorem 3. *(Revised Complexity for Feasible Sinkhorn with ROUND-POT) We first derive the sufficient size of A to be $\mathcal{O}\left(\frac{\|\mathbf{C}\|_{max}}{\varepsilon}\right)$. With this large A and ROUND-POT, Sinkhorn for POT has a computational complexity of $\tilde{\mathcal{O}}\left(\frac{n^2\|\mathbf{C}\|_{max}^2}{\varepsilon^4}\right)$ as oppose to $\tilde{\mathcal{O}}\left(\frac{n^2\|\mathbf{C}\|_{max}^2}{\varepsilon^2}\right)$ (**?**).*

The detailed proof for this theorem is included in Appendix, Revisiting Sinkhorn for POT section. We also empirically verify this worsened complexity in section in Feasible Sinkhorn section in Appendix.

Remark 4. Respecting the equality constraint is crucial for various applications that demand strict adherence to feasible solutions such as point cloud registration (**?**) (for avoiding incorrect many-to-many correspondences) and mini-batch OT (**?**) (for minimizing misspecification). Consequently, it is imperative for POT to transport the exact fraction of mass to achieve an optimal mapping, which is vital for the effective performances of ML models.

Algorithm 1: ROUND-POT

Input: $\mathbf{x} = (\text{vec}(\mathbf{X})^\top, \mathbf{p}^\top, \mathbf{q}^\top)^\top$; marginals $\mathbf{r}, \mathbf{c}$; mass s.
1: $\bar{\mathbf{p}} = \text{EP}(\mathbf{r}, s, \mathbf{p})$
2: $\bar{\mathbf{q}} = \text{EP}(\mathbf{c}, s, \mathbf{q})$
3: $\mathbf{g} = \min\{\mathbf{1}, (\mathbf{r} - \bar{\mathbf{p}}) \oslash \mathbf{X}\mathbf{1}\}$
4: $\mathbf{h} = \min\{\mathbf{1}, (\mathbf{c} - \bar{\mathbf{q}}) \oslash \mathbf{X}^\top\mathbf{1}\}$
5: $\mathbf{X}' = \text{diag}(\mathbf{g})\mathbf{X}\text{diag}(\mathbf{h})$
6: $\mathbf{e}_1 = (\mathbf{r} - \bar{\mathbf{p}}) - \mathbf{X}'\mathbf{1}, \mathbf{e}_2 = (\mathbf{c} - \bar{\mathbf{q}}) - \mathbf{X}'^\top\mathbf{1}$
7: $\bar{\mathbf{X}} = \mathbf{X}' + \mathbf{e}_1\mathbf{e}_2^\top / \|\mathbf{e}_1\|_1$
Output: $\bar{\mathbf{x}} = (\bar{\mathbf{X}}, \bar{\mathbf{p}}, \bar{\mathbf{q}})$

Rounding Algorithm

All efficient algorithms for standard OT (**???**) only output an infeasible approximation of the optimum value, and leverage the well-known rounding algorithm (**?**, Algorithm 2)

to project it back to the set of admissible couplings. Nevertheless, its ad-hoc design tailored to the OT's marginal constraints makes generalization to the case of POT with more intricate structural constraints non-trivial. In fact, we attribute the rather limited literature on efficient POT solvers to such lack of a rounding algorithm for POT. Specifically, previous works rely on imposing additional assumptions on the input masses to permit reformulation of POT into standard OT with an additional computational burden (**??**). Deviating from the vast literature, we address this fundamental challenge by proposing a novel rounding procedure for POT, termed ROUND-POT (Algorithm **??**), to efficiently round any approximate solution to a feasible solution of (**??**). Given an approximate solution $\mathbf{x} = (\mathbf{X}, \mathbf{p}, \mathbf{q}) \geq 0$ violating the POT constraints of (**??**) by a predefined error, $\|\mathbf{Ax} - \mathbf{b}\|_1 \leq \delta$ for some δ, ROUND-POT returns $\bar{\mathbf{x}} = (\bar{\mathbf{X}}, \bar{\mathbf{p}}, \bar{\mathbf{q}}) \geq \mathbf{0}$ strictly in the feasible set, i.e., $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$, and close to $\mathbf{x}$ in ℓ_1 distance.

The Enforcing Procedure (EP) (Algorithm **??**) is a novel subroutine to ensure $0 \leq \bar{\mathbf{p}} \leq \mathbf{r}$ and $\|\bar{\mathbf{p}}\|_1 = \|\mathbf{r}\|_1 - s$ (Lemma **??**). Equivalently, a similar procedure is applied to $(\mathbf{c}, s, \mathbf{q})$ in step 2 of Algorithm **??** with similar guarantees for $\bar{\mathbf{q}}$. Step 1 transforms $\mathbf{p}$ (or $\mathbf{q}$) to $\mathbf{p}'$ (or $\mathbf{q}'$) so that $0 \leq \mathbf{p}' \leq \mathbf{r}$ (or $0 \leq \mathbf{q}' \leq \mathbf{c}$). The transformation in steps 2 and 3 ensures that $\|\mathbf{p}''\|_1 \leq \|\mathbf{r}\|_1 - s$ (or $\|\mathbf{q}''\|_1 \leq \|\mathbf{c}\|_1 - s$). The rest of the EP steps ensure the other guarantee $\|\bar{\mathbf{p}}\|_1 = \|\mathbf{r}\|_1 - s$. The proof is included in Rounding Algorithm section in the Appendix.

Lemma 5. *(Guarantees for* EP*) We obtain in $\mathcal{O}(n)$ time $0 \leq \bar{\mathbf{p}} \leq \mathbf{r}$ and $\|\bar{\mathbf{p}}\|_1 = \|\mathbf{r}\|_1 - s$.*

For ROUND-POT, steps 3 through 7 check whether the solutions $\mathbf{X}$ violate each of the two equality constraints $\mathbf{X}\mathbf{1} = \mathbf{r} - \bar{\mathbf{p}}$ and $\mathbf{X}^\top \mathbf{1} = \mathbf{c} - \bar{\mathbf{q}}$; if so, the algorithm projects $\mathbf{X}$ into the feasible set. It is noteworthy that these two constraints directly implies the last needed constraint $\mathbf{1}^\top \mathbf{X} \mathbf{1} = s$. Finally, ROUND-POT returns an output that satisfies the required constraints in Equation (**??**). The following Theorem **??** characterizes the error guarantee of the rounded output $\bar{\mathbf{x}}$. Its detailed proof can be found in Rounding Algorithm section in the Appendix.

Algorithm 2: Enforcing Procedure EP

Input: marginal $\mathbf{r}$; mass s; slack variable $\mathbf{p}$.
  1: $\mathbf{p}' = \min\{\mathbf{p}, \mathbf{r}\}, \alpha = \min\left\{1, (\|\mathbf{r}\|_1 - s)/\|\mathbf{p}'\|_1\right\}$
  2: $\mathbf{p}'' = \alpha \mathbf{p}'$
  3: **if** $1 > (\|\mathbf{r}\|_1 - s)/\|\mathbf{p}'\|_1$ **then**
  4:    $\bar{\mathbf{p}} = \mathbf{p}''$
  5: **else**
  6:    $i = 0$
  7:    **while** $\|\mathbf{p}''\|_1 \leq \|\mathbf{r}\|_1 - s$ **do**
  8:      $i = i + 1$
  9:      $p_i'' = r_i$
 10:    **end while**
 11:    $p_i'' = p_i'' - (\|\mathbf{p}''\|_1 - \|\mathbf{r}\|_1 + s)$
 12:    $\bar{\mathbf{p}} = \mathbf{p}''$
 13: **end if**
Output: $\bar{\mathbf{p}}$.

Theorem 6. *(Guarantees for* ROUND-POT*) Let $\mathbf{A}$, $\mathbf{x}$ (consisting of $\mathbf{X}$, $\mathbf{p}$ and $\mathbf{q}$) and $\mathbf{b}$ be defined as in the preliminaries. If $\mathbf{x}$ satisfies that $\mathbf{x} \geq 0$ and $\|\mathbf{Ax} - \mathbf{b}\|_1 \leq \delta$ for some $\delta \geq 0$, Algorithm **??** outputs $\bar{\mathbf{x}} \geq \mathbf{0}$ (consisting of $\bar{\mathbf{X}}, \bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$) in $\mathcal{O}(n^2)$ time such that $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$ and $\|\mathbf{x} - \bar{\mathbf{x}}\|_1 \leq 23\delta$.*

Adaptive Primal-Dual Accelerated Gradient Descent (APDAGD)

Dual Formulation and Algorithmic Design

Following a similar formulation to (**?**, Section 3.1), we have the following primal problem with entropic regularization

$$\min_{\mathbf{x} \geq \mathbf{0}} \{f(\mathbf{x}) := \langle \mathbf{d}, \mathbf{x} \rangle + \gamma \langle \mathbf{x}, \log \mathbf{x} \rangle\} \ \text{ s.t. } \ \mathbf{Ax} = \mathbf{b}, \quad (5)$$

where $\mathbf{Ax} = \mathbf{b}$ is encoded as explained in Equation (**??**). Since problem (**??**) is a linearly constrained convex optimization problem, strong duality holds.

Lemma 7. *With a dual variable $\boldsymbol{\lambda} \in H^* = \mathbb{R}^{2n+1}$, the dual of (**??**) is given by*

$$\min_{\boldsymbol{\lambda} \in H^*} \left\{ \varphi(\boldsymbol{\lambda}) := \langle \boldsymbol{\lambda}, \mathbf{b} \rangle + \max_{\mathbf{x} \in Q} \left\{ -f(\mathbf{x}) - \langle \mathbf{x}, \mathbf{A}^\top \boldsymbol{\lambda} \rangle \right\} \right\},$$

or equivalently

$$\min_{\mathbf{y}, \mathbf{z}, t} \left\{ -ts - \langle \mathbf{y}, \mathbf{r} \rangle - \langle \mathbf{z}, \mathbf{c} \rangle \right.$$
$$\left. -\gamma \sum_{i,j=1}^{n} e^{-(C_{i,j} + y_i + z_j + t)/\gamma - 1} + e^{-y_i/\gamma - 1} + e^{-z_j/\gamma - 1} \right\},$$
$$(6)$$

*where $\mathbf{y}, \mathbf{z}, t$ are dual variables corresponding the POT constraints in (**??**) as $\boldsymbol{\lambda} = (\mathbf{y}^\top, \mathbf{z}^\top, t)^\top$ (which we simply refer as $(\mathbf{y}, \mathbf{z}, t)$ from now on).*

The detailed dual formulation of Equation (**??**) is found in subsection Dual Formulation for Entropic POT in the Appendix.

More details on the properties (strong convexity, smoothness, etc) of the primal and dual objectives are in Properties of Entropic POT section in Appendix. The APDAGD procedure is described in Algorithm **??** in Appendix. So as to approximate POT, we incorporate our novel rounding algorithm with a similar procedure to (**?**, Algorithm 2), in Algorithm **??**.

Computational Complexity

Now, we provide the computational complexity of APDAGD (Theorem **??**) and its proof sketch. The detailed proof of this result will be presented in Complexity of APDAGD for POT Detailed Proof subsection in Appendix.

Theorem 8. *(Complexity of APDAGD) The APDAGD algorithm returns ε-approximation POT solution $\widehat{\mathbf{X}} \in \mathcal{U}(\mathbf{r}, \mathbf{c}, s)$ in $\widetilde{\mathcal{O}}\left(n^{5/2} \|\mathbf{C}\|_{\max} / \varepsilon\right)$.*

Proof sketch. **Step 1**: we present the reparameterization $\mathbf{u} = -\mathbf{y}/\gamma - \mathbf{1}, \mathbf{v} = -\mathbf{z}/\gamma - \mathbf{1}$ and $w = -t/\gamma + 1$ for the dual

Algorithm 3: Approximating POT by APDAGD

Input: marginals $\mathbf{r}, \mathbf{c}$; cost matrix $\mathbf{C}$.
1: $\gamma = \varepsilon/(4\log(n)), \widetilde{\varepsilon} = \varepsilon/(8\|\mathbf{C}\|_{\max})$
2: **if** $\|\mathbf{r}\|_1 > 1$ **then**
3:     $\widetilde{\varepsilon} = \min\{\widetilde{\varepsilon}, 8(\|\mathbf{r}\|_1 - s)/(\|\mathbf{r}\|_1 - 1)\}$
4: **end if**
5: **if** $\|\mathbf{c}\|_1 > 1$ **then**
6:     $\widetilde{\varepsilon} = \min\{\widetilde{\varepsilon}, 8(\|\mathbf{c}\|_1 - s)/(\|\mathbf{c}\|_1 - 1)\}$
7: **end if**
8: $\widetilde{\mathbf{r}} = (1 - \widetilde{\varepsilon}/8)\mathbf{r} + \widetilde{\varepsilon}\mathbf{1}_n/(8n)$
9: $\widetilde{\mathbf{c}} = (1 - \widetilde{\varepsilon}/8)\mathbf{r} + \widetilde{\varepsilon}\mathbf{1}_n/(8n)$
10: $\widetilde{\mathbf{X}} = \text{APDAGD}(\mathbf{C}, \gamma, \widetilde{\mathbf{r}}, \widetilde{\mathbf{c}}, \widetilde{\varepsilon}/2)$
11: $\bar{\mathbf{X}} = \text{ROUND-POT}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{r}}, \widetilde{\mathbf{c}}, s)$
Output: $\bar{\mathbf{X}}$.

(**??**), leading to the equivalent dual form

$$\min_{\mathbf{u},\mathbf{v},w} \sum_{i,j=1}^{n} \exp\left(-C_{i,j}/\gamma + u_i + v_j + w\right) + \sum_{i=1}^{n} \exp(u_i)$$
$$+ \sum_{j=1}^{n} \exp(v_j) - \langle \mathbf{u}, \mathbf{r} \rangle - \langle \mathbf{v}, \mathbf{c} \rangle - ws.$$

This transformation will facilitate the bounding of the dual variables in later steps.

Step 2: we proceed to bound the ℓ_∞-norm of the transformed optimal dual variables $\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty$. Conventional analyses for OT such as (**?**) are inapplicable to the case of POT due to the addition of the third dual variable w and more intricate dependencies of the dual variables $\mathbf{u}, \mathbf{v}$. To this end, our novel proof technique establishes the tight bound of $\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty = \widetilde{\mathcal{O}}(\|\mathbf{C}\|_{\max})$, which consequently translates to the final bound for original dual variables $\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_2 = \widetilde{\mathcal{O}}(\sqrt{n}\|\mathbf{C}\|_{\max})$ in Lemma **??**. Bounding the ℓ_2-norm (i.e. bounding $\bar{R}$) is crucial because it contributes to the APDAGD guarantees (Theorem **??**) and the final complexity.

Step 3: Combining the $\bar{R}$ bound from **Step 2** in view of (**?**, Proposition 4.10) and the theoretical guarantees of ROUND-POT (Theorem **??**), we conclude the final computational complexity of APDAGD of $\widetilde{\mathcal{O}}(n^{2.5}/\varepsilon)$. $\qquad\square$

Dual Extrapolation (DE)

Our novel POT rounding algorithm permits the development of Dual Extrapolation (DE) for POT. From our analysis, DE is a first-order and parallelizable algorithm that can approximate POT distance up to ε accuracy with $\widetilde{\mathcal{O}}(1/\varepsilon)$ parallel depth and $\widetilde{\mathcal{O}}(n^2/\varepsilon)$ total work.

Setup

For each feasible $\mathbf{x}$, we have $\|\mathbf{x}\|_1 = \|\mathbf{X}\|_1 + \|\mathbf{p}\|_1 + \|\mathbf{q}\|_1 = \|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s$. We can normalize $\mathbf{x} = \mathbf{x}/(\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s)$, $\mathbf{b} = \mathbf{b}/(\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s)$. These imply $\mathbf{x} \in \Delta_{n^2+2n}$. The POT problem formulation (**??**) is now updated as

$$\min_{\mathbf{x} \in \Delta_{n^2+2n}} \langle \mathbf{d}, \mathbf{x} \rangle \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \qquad (7)$$

We then consider the ℓ_1 penalization for the problem (**??**) and show that it has equal optimal value and ε-approximate minimizer to those of the POT formulation (**??**) (more details in ℓ_1 Penalization subsection in Appendix). Through a primal-dual point of view, the ℓ_1 penalized objective (**??**) can be rewritten as

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}) := \mathbf{d}^\top \mathbf{x} + 23\|\mathbf{d}\|_\infty \left(\mathbf{y}^\top \mathbf{A}\mathbf{x} - \mathbf{y}^\top \mathbf{b}\right),$$
$$(8)$$

with $\mathcal{X} = \Delta_{n^2+2n}, \mathcal{Y} = [-1, 1]^{2n+1}$. Note that the term 23 comes from the guarantees of ROUND-POT (Theorem **??**, Lemma **??**). Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ such that $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. For a bilinear objective $F(\mathbf{x}, \mathbf{y})$ that is convex in $\mathbf{x}$ and concave in $\mathbf{y}$, it is natural to define the gradient operator $g(\mathbf{x}, \mathbf{y}) = (\nabla_\mathbf{x} F(\mathbf{x}, \mathbf{y}), -\nabla_\mathbf{y} F(\mathbf{x}, \mathbf{y}))$. Specifically for the objective (**??**), we have $g(\mathbf{x}, \mathbf{y}) = (\mathbf{d} + 23\|\mathbf{d}\|_\infty \mathbf{A}^\top \mathbf{y}, -23\|\mathbf{d}\|_\infty (\mathbf{A}\mathbf{x} - \mathbf{b}))$. This minimax objective can be solved with the dual extrapolation framework (**?**), which requires strongly convex regularizers. This setup can be relaxed with the notion of area-convexity (**?**, Definition 1.2), in the following definition.

Definition 9. (Area Convexity) A regularizer r is κ-area-convex w.r.t an operator g if for any x_1, x_2, x_3 in its domain,

$$\kappa \sum_i^3 r(x_i) - 3\kappa r\left(\frac{\sum_i^3 x_i}{3}\right) \geq \langle g(x_2) - g(x_1), x_2 - x_3 \rangle.$$

The regularizer chosen for this framework is the Sherman regularizer, introduced in (**?**)

$$r(\mathbf{x}, \mathbf{y}) = 2\|\mathbf{d}\|_\infty \left(10\langle \mathbf{x}, \log \mathbf{x} \rangle + \mathbf{x}^\top \mathbf{A}^\top (\mathbf{y}^2)\right), \qquad (9)$$

in which $\mathbf{y}^2$ is entry-wise. While this regularizer has a similar form to that in (**?**), the POT formulation leads to a different structure of $\mathbf{A}$. For instance, $\|\mathbf{A}\|_1 = 3$ instead of 2. This following lemma shows that chosen r is 9-area-convex (its proof is in the Proof of Lemma **??** section in the Appendix).

Lemma 10. *r is 9-area-convex with respect to the gradient operator g, i.e., $\kappa = 9$.*

Algorithmic Development

The main motivation is the DE Algorithm **??** in Appendix, proposed by (**?**). This general DE framework essentially has two proximal steps per iteration, while maintaining a state $\mathbf{s}$ in the dual space. We follow (**?**) and update $\mathbf{s}$ with $1/2\kappa$ rather than $1/\kappa$ (**?**). The proximal steps (steps 2 and 3 in Algorithm **??**) needs to minimize:

$$P(\mathbf{x}, \mathbf{y}) := \langle \mathbf{v}, \mathbf{x} \rangle + \langle \mathbf{u}, \mathbf{y} \rangle + r(\mathbf{x}, \mathbf{y}). \qquad (10)$$

This can be solved efficiently with an Alternating Minimization (AM) approach (**?**). Details for AM are included in the Appendix. Combining both algorithms, we have the DE for POT Algorithm **??**, where each of the two proximal steps is solved by the AM subroutine.

Computational Complexities

Firstly, we bound the regularizer r to satisfy the convergence guarantees (**?**) in Proof of Lemma **??** subsection in Appendix.

Algorithm 4: Dual Extrapolation for POT
___

Input: linearized cost $\mathbf{d}$; linear operator $\mathbf{A}$; constraints $\mathbf{b}$; area-convexity coefficient κ; initial states $\mathbf{s}_\mathbf{x}^0 = \mathbf{0}_{n^2+2n}, \mathbf{s}_\mathbf{y}^0 = \mathbf{0}_{2n+1}$; iterations M (Theorem **??**)

1: $\nabla_\mathbf{x} r(\bar{\mathbf{z}}) = 20 \|\mathbf{d}\|_\infty (1 - \log(n^2 + 2n))\mathbf{1}_{n^2+2n}$
2: $\nabla_\mathbf{y} r(\bar{\mathbf{z}}) = \mathbf{0}_{2n+1}$
3: **for** $t = 0, 1, 2, \ldots, T-1$ **do**
4: $\quad \mathbf{v} = \mathbf{s}_\mathbf{x}^t - \nabla_\mathbf{x} r(\bar{\mathbf{z}})$
5: $\quad \mathbf{u} = \mathbf{s}_\mathbf{y}^t - \nabla_\mathbf{y} r(\bar{\mathbf{z}})$
6: $\quad (\mathbf{z}_\mathbf{x}^t, \mathbf{z}_\mathbf{y}^t) = \texttt{AM}(M, \mathbf{v}, \mathbf{u})$
7: $\quad \mathbf{v} = \mathbf{v} + (\mathbf{d} + 23\|\mathbf{d}\|_\infty \mathbf{A}^\top \mathbf{z}_\mathbf{y}^t)/\kappa$
8: $\quad \mathbf{u} = \mathbf{u} - 23\|\mathbf{d}\|_\infty (\mathbf{A}\mathbf{z}_\mathbf{x}^t - \mathbf{b})/\kappa$
9: $\quad (\mathbf{w}_\mathbf{x}^t, \mathbf{w}_\mathbf{y}^t) = \texttt{AM}(M, \mathbf{v}, \mathbf{u})$
10: $\quad \mathbf{s}_\mathbf{x}^{t+1} = \mathbf{s}_\mathbf{x}^t + (\mathbf{d} + 23\|\mathbf{d}\|_\infty \mathbf{A}^\top \mathbf{w}_\mathbf{y}^t)/(2\kappa)$
11: $\quad \mathbf{s}_\mathbf{y}^{t+1} = \mathbf{s}_\mathbf{y}^t - 23\|\mathbf{d}\|_\infty (\mathbf{A}\mathbf{w}_\mathbf{x}^t - \mathbf{b})/(2\kappa)$
12: **end for**
Output: $\bar{\mathbf{w}}_\mathbf{x} = \sum_{t=0}^{T-1} \mathbf{w}_\mathbf{x}^t/T, \bar{\mathbf{w}}_\mathbf{y} = \sum_{t=0}^{T-1} \mathbf{w}_\mathbf{y}^t/T$.
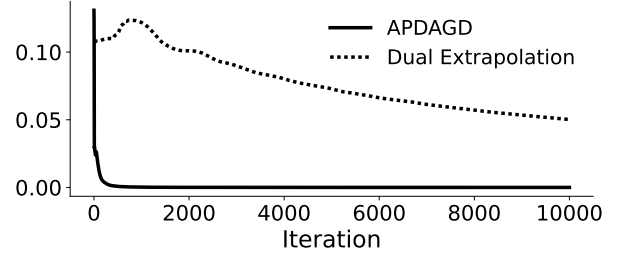___

Figure 2: Primal optimality gap for solutions produced by APDAGD and Dual Extrapolation.
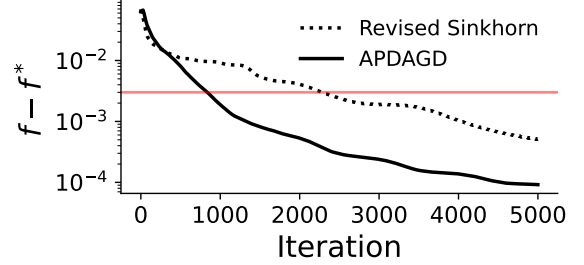

Figure 3: Primal optimality gap against optimization rounds achieved by our **revised** Sinkhorn and APDAGD for POT. The marginal distributions are taken from the later color transfer application in this section, and the red horizontal line depicts the pre-defined tolerance (ε) for both algorithms.

In that same subsection, we also derive the required number of iterations T in DE with respect to Θ, the range of the regularizer. Next, we have this essential lemma that bounds the number of iterations to evaluate a proximal step.

Theorem 11. *(Complexity of* AM*) For* $T = \lceil 36\Theta/\varepsilon \rceil$ *iterations of DE,* AM *Algorithm* **??** *obtains additive error* $\varepsilon/2$ *in*

$$M = 24\log\left(\left(840\|\mathbf{d}\|_\infty/\varepsilon^2 + 6/\varepsilon\right)\Theta + 1336\|\mathbf{d}\|_\infty/9\right)$$

iterations. This is done in wall-clock time $\mathcal{O}(n^2\log\eta)$ *with* $\eta = \log n \|\mathbf{d}\|_\infty/\varepsilon$.

The proof of this theorem can be found in Proof of Theorem **??** subsection in Appendix. The main proof idea is to bound the number of iterations required to solve the proximal steps. This explicit bound for the number of AM iterations is novel as in DE for OT (**?**), the authors runs a while loop and do not analyze its final number of iterations. We can now calculate the computational complexity of the DE algorithm. The proof is in Proof of Theorem **??** subsection in Appendix.

Theorem 12. *(Complexity of DE) In* $\widetilde{\mathcal{O}}(n^2\|\mathbf{C}\|_{max}/\varepsilon)$ *wall-clock time, the DE Algorithm* **??** *returns* $(\bar{\mathbf{w}}_\mathbf{x}, \bar{\mathbf{w}}_\mathbf{y}) \in \mathcal{Z}$ *so that the duality gap (***??***) is less than* ε.

Numerical Experiments

In this section, we provide numerical results on approximating POT and its applications using the algorithms presented above.[1] In all settings, the optimal solution is found by solving the linear program (**??**) using the cvxpy package. Due to space constraint, we also include many extra experiments such as domain adaptation application, large-scale APDAGD, run time for varying ϵ comparison, and revised Sinkhorn performance in Appendix, further solidifying the efficiency and practicality of the proposed algorithms.

___
[1]Implementation and numerical experiments can be found at https://github.com/joshnguyen99/partialot.

Run Time Comparison

APGAGD vs DE: We are able to implement the DE algorithm for POT while the authors of DE for OT faced numerical overflow and had to use mirror prox instead "for numerical stability considerations" (**?**). For the setup of Figure **??**, we use images in the CIFAR-10 dataset (**?**) as the marginals. More details are included in the Further Experiment Setup Details section in Appendix. In Figure **??**, despite having a better theoretical complexity, DE has relatively poor practical performance compared to APDAGD. This is not surprising as previous works on this class of algorithms like (**?**) reach similar conclusions on DE's practical limitations. This can be partly explained by the large constants that are dismissed by the asymptotic computational complexity. Thus, in our applications in the following subsection, we will use APDAGD or revised Sinkhorn.

APGAGD vs Sinkhorn: For the same setting with CIFAR-10 dataset, we report that the ratio between the average per iteration cost of APDAGD and that of Sinkhorn is 0.68. Furthermore, in the Run Time for Varying ε section in Appendix, we reproduce the same result in Figure 6 as **Figure 1 in (Dvurechensky et al., 2018)**, comparing the runtime of APDAGD and Sinkhorn for varying ε.

Revised Sinkhorn

Using the similar setting as the later example of color transfer, we can empirically verify that our revised Sinkhorn procedure can achieve the required tolerance ε of the POT problem in

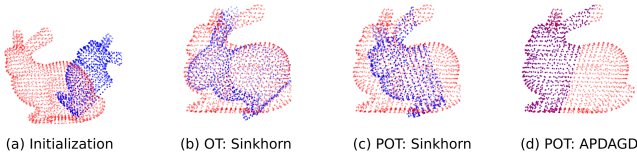(a) Initialization | (b) OT: Sinkhorn | (c) POT: Sinkhorn | (d) POT: APDAGD

Figure 4: Point cloud registration using (partial) optimal transport. (a): Initial point sets with one set (in blue) missing 45% of the points. (b): Registration result obtained after transforming the blue point cloud using the OT plan by Sinkhorn. (c): Registration result using the POT plan by **pre-revised** Sinkhorn (**?**). (d): Registration result using the POT plan by APDAGD.

Figure (as opposed to pre-revised Sinkhorn in Figure **??**).

Point Cloud Registration

We now present an application of POT in point set registration, a common task in shape analysis. Start with two point clouds in three dimensions $R = \{\mathbf{x}_i \in \mathbb{R}^3 \mid i = 1, \ldots, m\}$ and $Q = \{\mathbf{y}_j \in \mathbb{R}^3 \mid j = 1, \ldots, n\}$. The objective is to find a transformation, consisting of a rotation and a translation, that best aligns the two point clouds. When the initial point clouds contain significant noise or missing data, the registration result is often badly aligned. Here, we consider a scenario where one set has missing values. In Figure **??** (a), the blue point cloud is set to contain the front half of the rabbit, retaining about 45% of the original points. A desirable transformation must align the first halves of the two rabbits correctly. Figure **??** compares the point clouds registration result using these methods. If $\mathbf{T}$ is simply the OT matrix, not subject to a total transported mass constraint, the blue cloud is clearly not well-aligned with the red cloud. This is because the points for which the blue cloud is missing (i.e., the right half of the rabbit) are in the red cloud. If the total mass transported is set to $s = \frac{\min\{m,n\}}{\max\{m,n\}}$, then we end up with a POT matrix $\mathbf{T}$ with all entries summing to s. In Figures **??** (c) and (d), the POT solution leads to a much better result than the OT solution: the left halves of the rabbit are closer together. Importantly, the feasible solution obtained by APDAGD leads to an even better alignment, compared to the pre-revised Sinkhorn, due to its infeasibility.

Color Transfer

A frequent application of OT in computer vision is color transfer. POT offers flexibility in transferring colors between two possibly different-sized images, in contrast to OT which requires two color histograms to be normalized (**?**). We follow the setup by (**?**) and the detail of the implementation is described in the Further Experiment Setup Details section in Appendix. Our results are presented in Figure **??**. The third row of Figure **??** displays the resulting image produced by APDAGD with different levels of α (or transported mass s). Increasing α makes the wall color closer to the lighter part of the kiwi in the target image but comes at a cost of saturating the window frames' white. We emphasize that α is a tunable parameter, and the user can pick the most suitable level of transported mass. This can be more flexible than the vanilla OT which stringently requires all marginal masses to be nor-

(a) Source image | (b) Target image | (c) Primal optimality gap | (d) LP ($\alpha = 0.90$)
(e) Sinkhorn ($\alpha = 0.3$) | (f) Sinkhorn ($\alpha = 0.5$) | (g) Sinkhorn ($\alpha = 0.7$) | (h) Sinkhorn ($\alpha = 0.9$)
(i) APDAGD ($\alpha = 0.3$) | (j) APDAGD ($\alpha = 0.5$) | (k) APDAGD ($\alpha = 0.7$) | (l) APDAGD ($\alpha = 0.9$)
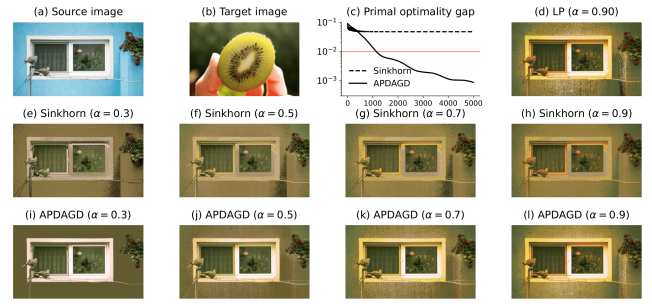
Figure 5: Color transfer using POT. The source and target images are of different sizes, giving rise to unbalanced color histograms. We apply Sinkhorn and APDAGD to solve the entropic regularized POT problem between these histograms. The optimality gap $\langle \mathbf{C}, \mathbf{X} \rangle - f^*$ after each iteration is displayed in (c). The exact solution, retrieved by the Gurobi solver is presented in (d). On the second and third rows, the transformed image is displayed for different levels of α, showing the flexibility of POT as opposed to OT.

malized. We also emphasize that qualitatively, the solution produced by APDAGD closely matches the exact solution in (d), in contrast to **pre-revised** Sinkhorn's (**?**). This highlights the impact of having a feasible and optimal mapping on the quality of the application.

Conclusion

In this paper, we first examine the infeasibility of Sinkhorn for POT. We then propose a a novel rounding algorithm which facilitates our development of feasible Sinkhorn procedure with guarantees, APDAGD and DE. DE achieves the best theoretical complexity while APDAGD and revised Sinkhorn are practically efficient algorithms, as demonstrated in our extensive experiments. We believe the rigor and applicability of our proposed methods will further facilitate the practical adoptions of POT in machine learning applications.

Acknowledgements

Additional Related Literature

Another alternative to OT for unbalanced measures, Unbalanced Optimal Transport (UOT) regularizes the objective by penalizing the marginal constraints via some given divergences (**??**) including Kullback-Leiber (**?**) or squared ℓ_2 norm (**?**). Several works have studied UOT both in terms of computational complexity (**??**) and applications (**???**). In practical applications, the structure of UOT and recent results in the approximation of OT with UOT (**?**) suggest that UOT is best used in the time-varying regimes or strict control of mass is not required. For these reasons, UOT and POT cannot be directly compared to each other, and each serves separate purposes in different ML settings. The relaxation of the marginal constraints—which are strictly imposed by the standard OT and UOT—and the control over the total mass transported have granted POT immense flexibility compared to UOT.

Similar to OT, POT or entropic regularized POT problems belong to the class of minimization problems with linear constraints. Given the structure of linear constraints and the property of strong duality, the prevalent method is to next formulate the Lagrange dual problem. It is then intuitive to solve the dual problem with first-order methods like Quasi-Newton (**?**) or Conjugate Gradient (**?**). Indeed, previous works (**??**) utilized L-BFGS method to find the OT dual Lagrange problem. Based on the Accelerated Gradient Descent method with primal-dual updates, APDAGD (**?**) gives better computational complexity and practicality compared to L-BFGS and other previous first-order methods applied to OT such as Quasi-Newton or Conjugate Gradient. With these APDAGD's theoretical and practical performances, we naturally want to develop a first-order framework based on AGDAGD to solve the POT problem. Nevertheless, there is ample room to further extend this first-order framework. For instance, one may look to develop a stochastic solver to handle stochastic gradient computations (**?**) or extend this method to decentralized settings such as (**?**).

Extra Algorithms

Sinkhorn

Algorithm 5: Sinkhorn for POT

Input: cost matrix $\mathbf{C}$, marginals $\mathbf{r}, \mathbf{c}$, regularization term γ, transported mass s.
1: $\mathbf{C} \leftarrow \widetilde{\mathbf{C}}, \mathbf{r} \leftarrow \widetilde{\mathbf{r}}, \mathbf{q} \leftarrow \widetilde{\mathbf{q}}$ (Updates follow the discussion in Revisiting Sinkhorn section)
2: $\widetilde{\mathbf{X}} \leftarrow \text{SINKHORN}(\widetilde{\mathbf{C}}, \widetilde{\mathbf{r}}, \widetilde{\mathbf{c}}, \gamma)$
3: $\bar{\mathbf{X}} \leftarrow \text{ROUND}(\widetilde{\mathbf{X}}, \mathcal{U}(\widetilde{\mathbf{r}}, \widetilde{\mathbf{c}}))$ (**?**, Algorithm 2)
Output: $\bar{\mathbf{X}}[1:n, 1:n]$

APDAGD

Algorithm 6: Adaptive Primal-Dual Accelerated Gradient Descent (APDAGD) (**?**, Algorithm 3)

Input: accuracy $\varepsilon_f, \varepsilon_{eq} > 0$; initial estimate L_0 such that $0 < L_0 < 2L$; initialize $M_{-1} = L_0, i_0 = k = \alpha_0 = \beta_0 = 0$, $\boldsymbol{\eta}_0 = \boldsymbol{\zeta}_0 = \boldsymbol{\lambda}_0 = \mathbf{0}$.
1: **while** $f(\hat{\mathbf{x}}_{k+1}) + \varphi(\boldsymbol{\eta}_{k+1}) \leq \varepsilon_f, \|\mathbf{A}\hat{\mathbf{x}}_{k+1} - \mathbf{b}\|_2 \leq \varepsilon_{eq}$ **do**
2:    **while** $\varphi(\boldsymbol{\eta}_{k+1}) \leq \varphi(\boldsymbol{\lambda}_{k+1}) + \langle \nabla\varphi(\boldsymbol{\lambda}_{k+1}), \boldsymbol{\eta}_{k+1} - \boldsymbol{\lambda}_{k+1} \rangle + \dfrac{M_k}{2}\|\boldsymbol{\eta}_{k+1} - \boldsymbol{\lambda}_{k+1}\|_2^2$ **do**
3:       $M_k = 2^{i_k - 1} M_k$
         find α_{k+1} such that $\beta_{k+1} := \beta_k + \alpha_{k+1} = M_k \alpha_{k+1}^2$
4:       $\tau_k = \dfrac{\alpha_{k+1}}{\beta_{k+1}}$
5:       $\boldsymbol{\lambda}_{k+1} = \tau_k \boldsymbol{\zeta}_k + (1 - \tau_k)\boldsymbol{\eta}_k$
6:       $\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k - \alpha_{k+1}\nabla\varphi(\boldsymbol{\lambda}_{k+1})$
7:       $\boldsymbol{\eta}_{k+1} = \tau_k \boldsymbol{\zeta}_{k+1} + (1 - \tau_k)\boldsymbol{\eta}_k$
8:    **end while**
9:    $\hat{\mathbf{x}}_{k+1} = \tau_k \mathbf{x}(\boldsymbol{\lambda}_{k+1}) + (1 - \tau_k)\hat{\mathbf{x}}_k$
10:   $i_{k+1} = 0, k = k + 1$
11: **end while**
Output: $\hat{\mathbf{x}}_{k+1}, \boldsymbol{\eta}_{k+1}$.

General Dual Extrapolation

In this context of this work, we can define the Bregman divergence as follows

---

Algorithm 7: General Dual Extrapolation

Input: initial state $\mathbf{s}_0 = \mathbf{0}$; area-convexity coefficient κ; gradient operator g; number of iterations T; regularizer r with minimizer $\bar{\mathbf{z}}$.

  1: **for** $t = 0, 1, 2, \ldots, T - 1$ **do**

  2:    $\mathbf{z}^t = \mathrm{prox}_{\bar{\mathbf{z}}}^r(\mathbf{s}^t)$

  3:    $\mathbf{w}^t = \mathrm{prox}_{\bar{\mathbf{z}}}^r(\mathbf{s}^t + \frac{1}{\kappa}g(\mathbf{z}^t))$

  4:    $\mathbf{s}^{t+1} = \mathbf{s}^t + \frac{1}{2\kappa}g(\mathbf{w}^t)$

  5:    $t = t + 1$

  6: **end for**

Output: $\bar{\mathbf{w}} := \frac{1}{T-1}\sum_{t=0}^{T}\mathbf{w}^t$.

Definition 13 (Bregman divergence). For differentiable convex function $h(\cdot)$ and $\mathbf{x}, \mathbf{y}$ in its domain, the Bregman divergence from $\mathbf{x}$ to $\mathbf{y}$ is $V_{\mathbf{x}}^h(\mathbf{y}) := h(\mathbf{y}) - h(\mathbf{x}) - \langle \nabla h(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle$.

In our context, $h(\cdot)$ is chosen to be $r(\cdot)$, which is area-convex (**?**, Definition 1.2). Similar to (**?**, Definition 2.5), we can define the proximal operator in Dual Extrapolation as follow

Definition 14 (Proximal operator). For differentiable (regularizer) $h(\cdot)$ with $\mathbf{x}, \mathbf{y}$ in its domain, and g in the dual space, the proximal operator is $\mathrm{prox}_{\mathbf{x}}^h(g) := \arg\min_{\mathbf{y}}\{\langle g, \mathbf{y}\rangle + V_{\mathbf{x}}^h(\mathbf{y})\}$.

Alternating Minimization

Algorithm 8: Alternating minimization for (**??**) ($\mathtt{AM}\,(M, \mathbf{v}, \mathbf{u})$)

Input: number of iterations M; vectors $\mathbf{v}, \mathbf{u}$; initialize $\mathbf{x} = \frac{1}{n^2 + 2n}\mathbf{1}_{n^2+2n}$; $\mathbf{y} = \mathbf{0}_{2n+1}$.

  1: **for** $t = 0, 1, 2, \ldots, M - 1$ **do**

  2:    $\boldsymbol{\zeta} = \frac{1}{20\,\|\mathbf{d}\|_{\infty}}\mathbf{v} + \frac{1}{10}\mathbf{A}^{\top}(\mathbf{y}^t)^2$

  3:    $\mathbf{x}^{t+1} = \dfrac{\exp(-\boldsymbol{\zeta})}{\sum_{j=1}^{n^2+2n}[\exp(-\boldsymbol{\zeta})]_j}$

  4:    $\mathbf{y}^{t+1} = \min\left(\mathbf{1}, \max\left(-\mathbf{1}, \dfrac{-\mathbf{u}}{4\,\|\mathbf{d}\|_{\infty}\mathbf{A}\mathbf{x}^{t+1}}\right)\right)$

     (All operations are element-wise)

  5: **end for**

Output: $\mathbf{x}^{t+1}, \mathbf{y}^{t+1}$.

Revisiting Sinkhorn for POT

Proof of Theorem ??

From the augmentation of the marginals (**?**) and the guarantees of the rounding algorithm (**?**, Algorithm 2), we have

$$\widetilde{\mathbf{X}}\mathbf{1} = \|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s, \widetilde{\mathbf{X}}^{\top}\mathbf{1} = \|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s$$
$$\mathbf{1}^{\top}\bar{\mathbf{X}}\mathbf{1} + \|\widetilde{\mathbf{p}}\|_1 = \|\mathbf{r}\|_1, \|\widetilde{\mathbf{q}}\|_1 + \widetilde{X}_{n+1,n+1} = \|\mathbf{c}\|_1 - s$$
$$\mathbf{1}^{\top}\bar{\mathbf{X}}\mathbf{1} + \|\widetilde{\mathbf{q}}\|_1 = \|\mathbf{c}\|_1, \|\widetilde{\mathbf{p}}\|_1 + \widetilde{X}_{n+1,n+1} = \|\mathbf{r}\|_1 - s.$$

Hence we can make a quick substitution

$$\|\widetilde{\mathbf{p}}\|_1 + \widetilde{X}_{n+1,n+1} = \|\mathbf{r}\|_1 - s = \mathbf{1}^{\top}\bar{\mathbf{X}}\mathbf{1} + \|\widetilde{\mathbf{p}}\|_1 - s,$$

which implies the constraint violation V

$$V = \mathbf{1}^{\top}\bar{\mathbf{X}}\mathbf{1} - s = \widetilde{X}_{n+1,n+1}.$$

Hence, we now proceed to bound $\widetilde{X}_{n+1,n+1}$. we thus proceed to lower-bound $\tilde{X}_{n+1,n+1}$. Note that the reformulated OT problem that Sinkhorn solves

$$\min_{\tilde{\mathbf{X}} \geq 0} \langle \tilde{\mathbf{X}}, \tilde{\mathbf{C}} \rangle \quad \text{s.t.} \quad \mathbf{X1} = \tilde{\mathbf{r}}, \ \mathbf{X}^\top \mathbf{1} = \tilde{\mathbf{c}}.$$

With a well known dual formulation for entropic OT (?), we can minimize the Lagrangian w.r.t. $\tilde{\mathbf{X}}$ and obtain

$$\tilde{\mathbf{X}} = \exp\left(-\frac{\tilde{\mathbf{C}}}{\gamma} + \mathbf{u1}^\top + \mathbf{1v}^\top\right),$$

where $\mathbf{u}$, $\mathbf{v}$ are the dual variable associated with each constraints. Now consider only the $\tilde{X}_{n+1,n+1}$ element, we can deduce

$$\tilde{X}_{n+1,n+1} = \exp\left(-\frac{\tilde{C}_{n+1,n+1}}{\gamma} - u_{n+1} - v_{n+1}\right) \geq \exp\left(-\frac{A}{\gamma} - \|\mathbf{u}\|_\infty - \|\mathbf{v}\|_\infty\right).$$

Similar to our Lemma **??** for POT, a similar bound for the infinity norm of OT dual variables (?, Lemma 3.2) shows that

$$\|\mathbf{u}\|_\infty, \|\mathbf{v}\|_\infty \leq R,$$

where

$$R = \frac{A}{\gamma} + \log(n) - 2\log\left(\min_{1 \leq i,j \leq n}\{r_i, c_j\}\right).$$

With $\gamma = \varepsilon/(4\log n)$ set by Sinkhorn, we obtain a lower bound of the constraint violation $\tilde{X}_{n+1,n+1}$

$$\tilde{X}_{n+1,n+1} \geq \exp\left(-\frac{12A\log n}{\varepsilon} - \mathcal{O}(\log n)\right).$$

Next, we establish an upper bound for the violation. We consider a feasible transportation matrix $\hat{\mathbf{X}}$ in the form $\hat{X}_{i,j} = s/n^2$ for $i, j \in [n]$ and $\hat{X}_{i,j} = 0$ otherwise. As $\tilde{\mathbf{X}}$ is a Sinkhorn ε-approximate solution of POT, we have

$$\langle \tilde{\mathbf{C}}, \tilde{\mathbf{X}} \rangle - \gamma H(\tilde{\mathbf{X}}) \leq \langle \tilde{\mathbf{C}}, \hat{\mathbf{X}} \rangle - \gamma H(\hat{\mathbf{X}}) + \varepsilon$$

$$= \frac{s}{n^2} \sum_{i,j}^{n} C_{ij} - \gamma s \log \frac{s}{n^2} + \varepsilon$$

$$\leq s\|\mathbf{C}\|_\infty - \gamma s \log \frac{s}{n^2} + \varepsilon.$$

On the other hand, we also have

$$\langle \tilde{\mathbf{C}}, \tilde{\mathbf{X}} \rangle - \gamma H(\tilde{\mathbf{X}}) \geq A\tilde{X}_{n+1,n+1} + \gamma \sum_{i,j}^{n+1} \tilde{X}_{ij} \log(\tilde{X}_{ij})$$

$$\geq A\tilde{X}_{n+1,n+1} + \gamma \left(\sum_{i,j}^{n+1} \tilde{X}_{ij}\right) \log\left(\frac{1}{n+1} \sum_{i,j}^{n+1} \tilde{X}_{ij}\right)$$

$$= A\tilde{X}_{n+1,n+1} + \gamma \left(\sum_{i,j}^{n+1} \tilde{X}_{ij}\right) \log\left(\sum_{i,j}^{n+1} \tilde{X}_{ij}\right) - \gamma \left(\sum_{i,j}^{n+1} \tilde{X}_{ij}\right) \log(n+1)$$

$$\geq (A + \gamma - \gamma \log(n+1))\tilde{X}_{n+1,n+1} + \gamma \left(\sum_{i,j}^{n} \tilde{X}_{ij} - 1\right)$$

$$- \gamma \left(\sum_{i,j}^{n} \tilde{X}_{ij}\right) \log(n+1),$$

where the second steps follows from Jensen's inequality, and the last step follows from the fact that $y \log y \le y - 1 \; \forall \, y > 0$. Combining the above two inequalities we deduce

$$\tilde{X}_{n+1,n+1} \le \frac{1}{A - \gamma(\log(n+1) - 1)} \left(s\|\mathbf{C}\|_\infty - \gamma s \log \frac{s}{n^2} + \varepsilon \right.$$
$$+ \gamma \left(\sum_{i,j}^n \tilde{X}_{ij} - 1 \right) - \gamma \left(\sum_{i,j}^n \tilde{X}_{ij} \right) \log(n+1) \right)$$
$$= \mathcal{O}\left(\frac{\|\mathbf{C}\|_\infty}{A} \right).$$

$\square$

Proof of Theorem ??

From Theorem **??**, we have the constraint violation V or $\tilde{X}_{n+1,n+1}$ satifying

$$\mathcal{O}\left(\frac{\|\mathbf{C}\|_\infty}{A} \right) \le \tilde{X}_{n+1,n+1}.$$

Hence, in order to achieve the ε tolerance we need

$$\mathcal{O}\left(\frac{\|\mathbf{C}\|_\infty}{A} \right) \le \varepsilon,$$

yielding the sufficient size of A to be

$$A = \mathcal{O}\left(\frac{\|\mathbf{C}\|_\infty}{\varepsilon} \right).$$

The computational complexity of (Infeasible) Sikhorn for POT (**?**) is

$$\tilde{\mathcal{O}}\left(\frac{n^2 \|\tilde{\mathbf{C}}\|_\infty^2}{\varepsilon^2} \right) = \tilde{\mathcal{O}}\left(\frac{n^2 A^2}{\varepsilon^2} \right).$$

Plugging our derived sufficient value of A into the Sinkhorn complexity, we obtain the revised complexity for Feasible Sinkhorn to be

$$\tilde{\mathcal{O}}\left(\frac{n^2 \|\mathbf{C}\|_\infty^2}{\varepsilon^4} \right).$$

$\square$

Rounding Algorithm

Proof of Lemma ?? (Guarantees for the Enforcing Procedure)

In the first case, we have $\alpha = \dfrac{\|\mathbf{r}\|_1 - s}{\|\mathbf{p}'\|_1} (\ge 0)$, implying

$$\|\bar{\mathbf{p}}\|_1 = \|\mathbf{p}''\|_1 = \alpha \|\mathbf{p}'\|_1 = \frac{\|\mathbf{r}\|_1 - s}{\|\mathbf{p}'\|_1} \|\mathbf{p}'\|_1 = \|\mathbf{r}\|_1 - s.$$

Also, consider

$$\bar{\mathbf{p}} = \mathbf{p}'' = \alpha \mathbf{p}' \ge 0$$
$$\bar{\mathbf{p}} = \mathbf{p}'' = \alpha \mathbf{p}' \le \mathbf{p}' \le \mathbf{r}.$$

So we have the guarantees as desired in the first scenario. For the second case, we obtain $\alpha = 1$, implying $\mathbf{p}'' = \mathbf{p}'$. Firstly we want to show the while loop will always stop. This is because the while condition $\|\mathbf{r}\|_1 - s \ge \|\mathbf{p}''\|_1$ will be eventually violated:

$$\sum_{i=1}^n (r_i - p_i'') = \|\mathbf{r}\|_1 - \|\mathbf{p}''\|_1 \ge \|\mathbf{r}\|_1 - s - \|\mathbf{p}''\|_1.$$

During the updates, for some index $1 \le j \le i$, $p_j'' = r_j$, while the rest remain the same. Therefore, we still have $\mathbf{0} \le \mathbf{p}'' \le \mathbf{r}$ after the running the while loop. For the last updated index i, we check that $0 \le p_i'' \le r_i$, and the second inequality is obvious. The first inequality holds because we have $r_i \ge r_i - p_i' \ge \|\mathbf{p}''\|_1 - \|\mathbf{r}\|_1 + s$ as $p_i' = p_i''$ before the update inside the while loop. After the while loop, by updating $p_i'' = p_i'' - (\|\mathbf{p}''\|_1 - \|\mathbf{r}\|_1 + s)$, we now have $\|\mathbf{p}''\|_1 = \|\mathbf{r}\|_1 - s$ as desired. $\square$

Proof of Theorem ?? (Guarantees for the Rounding Algorithm)

First, we show that $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$. Note that $\|\mathbf{e}_1\|_1 = \|\mathbf{e}_2\|_1 = s - \mathbf{1}^\top \mathbf{X}'\mathbf{1}$. From step 10, we have

$$\bar{\mathbf{X}}\mathbf{1} + \bar{\mathbf{p}} = \left(\mathbf{X}' + \frac{1}{\|\mathbf{e}_1\|_1}\mathbf{e}_1\mathbf{e}_2^\top\right)\mathbf{1} + \bar{\mathbf{p}} = \mathbf{X}'\mathbf{1} + \mathbf{e}_1 + \bar{\mathbf{p}} = \mathbf{r},$$

$$\bar{\mathbf{X}}^\top\mathbf{1} + \bar{\mathbf{q}} = \left(\mathbf{X}' + \frac{1}{\|\mathbf{e}_1\|_1}\mathbf{e}_1\mathbf{e}_2^\top\right)^\top \mathbf{1} + \bar{\mathbf{q}} = \mathbf{X}'^\top\mathbf{1} + \mathbf{e}_2 + \bar{\mathbf{q}} = \mathbf{c},$$

this implies

$$\mathbf{1}^\top\bar{\mathbf{X}}\mathbf{1} = \mathbf{1}^\top(\mathbf{r} - \bar{\mathbf{p}}) = \|\mathbf{r}\|_1 - \|\bar{\mathbf{p}}\|_1 = s.$$

Next, we endeavor to bound $\|\mathbf{x} - \bar{\mathbf{x}}\|_1$. Considering from the given information $\delta \geq \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$, we obtain

$$\delta \geq \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$
$$= \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1 + |\mathbf{1}^\top\mathbf{X}\mathbf{1} - s|$$
$$\geq \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1.$$

Since steps 7 through 10 have similar structure to (?, Algorithm 2) with changes in the marginals, by (?, Lemma 7), we have

$$\|\mathrm{vec}(\mathbf{X}) - \mathrm{vec}(\bar{\mathbf{X}})\|_1 \leq 2\left(\|\mathbf{X}\mathbf{1} + (\bar{\mathbf{p}} - \mathbf{r})\|_1 + \|\mathbf{X}^\top\mathbf{1} + (\bar{\mathbf{q}} - \mathbf{c})\|_1\right)$$
$$\leq 2\left(\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\bar{\mathbf{p}} - \mathbf{p}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1 + \|\bar{\mathbf{q}} - \mathbf{q}\|_1\right)$$
$$\leq 2(\delta + \|\bar{\mathbf{p}} - \mathbf{p}\|_1 + \|\bar{\mathbf{q}} - \mathbf{q}\|_1).$$

Hence,

$$\|\mathbf{x} - \bar{\mathbf{x}}\|_1 = \|\mathrm{vec}(\mathbf{X}) - \mathrm{vec}(\bar{\mathbf{X}})\|_1 + \|\bar{\mathbf{p}} - \mathbf{p}\|_1 + \|\bar{\mathbf{q}} - \mathbf{q}\|_1 \tag{11}$$
$$\leq 2\delta + 3(\|\bar{\mathbf{p}} - \mathbf{p}\|_1 + \|\bar{\mathbf{q}} - \mathbf{q}\|_1). \tag{12}$$

Note that

$$\|\bar{\mathbf{p}} - \mathbf{p}\|_1 \leq \|\mathbf{p} - \mathbf{p}'\|_1 + \|\mathbf{p}' - \mathbf{p}''\|_1 + \|\mathbf{p}'' - \bar{\mathbf{p}}\|_1.$$

For bounding $\|\mathbf{p} - \mathbf{p}'\|_1$, consider

$$\|\mathbf{p} - \mathbf{p}'\|_1 = \sum_{i=1}^n (p_i - \min\{p_i, r_i\}) = \sum_{i=1}^n \nu_i,$$

where

$$\nu_i = \begin{cases} 0 & p_i \leq r_i \\ p_i - r_i & p_i > r_i \end{cases}.$$

If $p_i > r_i$, then obviously $(X1)_i + p_i > r_i$ as $(X1)_i > 0$. Hence we obtain

$$\|\mathbf{p} - \mathbf{p}'\|_1 = \sum_{i=1}^n \nu_i = \sum_{i:p_i > r_i}^n \nu_i \leq \sum_{i:p_i > r_i}^n ((X1)_i + p_i - r_i) \leq \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1.$$

Next, consider

$$\|\mathbf{p}' - \mathbf{p}''\|_1 = \sum_{i=1}^n p_i'|1 - \alpha|$$
$$= \|\mathbf{p}'\|_1 - \min\{\|\mathbf{p}'\|_1, \|\mathbf{r}\|_1 - s\}$$
$$= \|\mathbf{p}'\|_1 - \frac{1}{2}(\|\mathbf{p}'\|_1 + \|\mathbf{r}\|_1 - s - |\|\mathbf{p}'\|_1 - \|\mathbf{r}\|_1 + s|),$$

Further simplifying the RHS, we have

$$\|\mathbf{p}' - \mathbf{p}''\|_1 = \frac{1}{2}\left(\|\mathbf{p}'\|_1 - \|\mathbf{r}\|_1 + s + |\|\mathbf{p}'\|_1 - \|\mathbf{r}\|_1 + s|\right)$$
$$\leq |\|\mathbf{p}'\|_1 - \|\mathbf{r}\|_1 + s|$$
$$\leq |\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + \|\mathbf{p}\|_1 - \|\mathbf{p}'\|_1$$
$$= |\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + \|\mathbf{p} - \mathbf{p}'\|_1$$
$$\leq |\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1.$$

Finally, we bound for $\|\mathbf{p}'' - \bar{\mathbf{p}}\|_1$. Note that

$$\|\mathbf{p}'' - \bar{\mathbf{p}}\|_1 = \|\mathbf{r}\|_1 - s - \|\mathbf{p}''\|_1$$
$$= \|\mathbf{r}\|_1 - s - \|\mathbf{p}\|_1 + (\|\mathbf{p}\|_1 - \|\mathbf{p}''\|_1)$$
$$= \|\mathbf{r}\|_1 - s - \|\mathbf{p}\|_1 + \|\mathbf{p} - \mathbf{p}''\|_1,$$

yielding

$$\|\mathbf{p}'' - \bar{\mathbf{p}}\|_1 \leq |\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + \|\mathbf{p} - \mathbf{p}'\|_1 + \|\mathbf{p}' - \mathbf{p}''\|_1$$
$$\leq 2|\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + 2\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1.$$

Combining the bounds for $\|\mathbf{p} - \mathbf{p}'\|_1, \|\mathbf{p}' - \mathbf{p}''\|_1, \|\mathbf{p}'' - \bar{\mathbf{p}}\|_1$, we obtain

$$\|\bar{\mathbf{p}} - \mathbf{p}\|_1 \leq 3|\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + 4\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1.$$

Similarly, we can obtain the bound for $\|\bar{\mathbf{q}} - \mathbf{p}\|_1$

$$\|\bar{\mathbf{q}} - \mathbf{q}\|_1 \leq 3|\|\mathbf{q}\|_1 - \|\mathbf{c}\|_1 + s| + 4\|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1.$$

Next, note that

$$\delta + |\mathbf{1}^\top\mathbf{X}\mathbf{1} - s|$$
$$\geq \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1 + 2|\mathbf{1}^\top\mathbf{X}\mathbf{1} - s|$$
$$= |\mathbf{1}^\top\mathbf{X}\mathbf{1} + \|\mathbf{p}\|_1 - \|\mathbf{r}\|_1| + |\mathbf{1}^\top\mathbf{X}\mathbf{1} - s| + |\mathbf{1}^\top\mathbf{X}\mathbf{1} + \|\mathbf{q}\|_1 - \|\mathbf{c}\|_1| + |\mathbf{1}^\top\mathbf{X}\mathbf{1} - s|$$
$$\geq |\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + |\|\mathbf{q}\|_1 - \|\mathbf{c}\|_1 + s|.$$

Hence we obtain

$$\|\bar{\mathbf{p}} - \mathbf{p}\|_1 + \|\bar{\mathbf{q}} - \mathbf{q}\|_1$$
$$\leq 3(|\|\mathbf{p}\|_1 - \|\mathbf{r}\|_1 + s| + |\|\mathbf{q}\|_1 - \|\mathbf{c}\|_1 + s|) + 4(\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1)$$
$$\leq 3\delta + 3|\mathbf{1}^\top\mathbf{X}\mathbf{1} - s| + 4(\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1)$$
$$= 3\delta + 3\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1 + \|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top\mathbf{1} + \mathbf{q} - \mathbf{c}\|_1$$
$$\leq 7\delta$$

Therefore, from (**??**), we conclude

$$\|\mathbf{x} - \bar{\mathbf{x}}\|_1 \leq 23\delta.$$

$\square$

Properties of Entropic POT

The term $\langle \mathbf{x}, \log \mathbf{x} \rangle$ is the negative entropic regularizer for every $\mathbf{x} \geq 0$, with the convention that $\mathbf{0}\log\mathbf{0} = \mathbf{0}$, and the parameter $\gamma > 0$ controls the strength of regularization. It is well-known that the objective $\varphi(\boldsymbol{\lambda})$ is L-smooth with $L \leq \|\mathbf{A}\|_{1\to 2}^2 / \mu_f$ (**?**), since f is μ_f-strongly convex w.r.t. $\|\cdot\|_1$, where $\mu_f = \gamma/(\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s)$ (Lemma **??**). Instead of this loose estimate, APDAGD uses line search to find and adapt the local Lipschitz value (**?**). Further, the gradient of φ is $\nabla\varphi(\boldsymbol{\lambda}) = \mathbf{b} - \mathbf{A}\mathbf{x}(\boldsymbol{\lambda})$ where $\mathbf{x}(\boldsymbol{\lambda}) := \arg\max_{\mathbf{x}\in Q}\{-f(\mathbf{x}) - \langle \mathbf{x}, \mathbf{A}^\top\boldsymbol{\lambda}\rangle\}$.

Strong Convexity

Lemma 15. *For convenience, denote* $\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s$ *as D. Let* $Q = \{\mathbf{x} \in \mathbb{R}^{n^2+2n}|\mathbf{x} > \mathbf{0}, \|\mathbf{x}\|_1 \leq D\}$. *The entropic regularized objective function* $f(\mathbf{x}) = \sum_{i=1}^{n^2+2n} \gamma x_i log(x_i) + d_i x_i$ *is* $\frac{\gamma}{D}$*-strongly-convex with respect to* ℓ_1 *norm over Q.*

Proof. To prove the Lemma, we first consider the following Observation **??**:

Observation 1. $f(\mathbf{x})$ *is* $\frac{\gamma}{D}$*-strongly-convex with respect to* ℓ_1 *norm over Q if for all* $\mathbf{x}, \mathbf{y} \in Q$,

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \frac{\gamma}{D}\|\mathbf{x} - \mathbf{y}\|_1^2. \tag{13}$$

Proof of Observation **??**. Consider $\mathbf{x} \neq \mathbf{y}$. For $\lambda \in [0,1]$, we define $h_1(\lambda) = f(\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})) - \lambda\langle\nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle$ and $h_2(\lambda) = \dfrac{\lambda^2\gamma}{2D}\|\mathbf{x} - \mathbf{y}\|_1^2$, and further obtain:

$$h_1'(\lambda) = \langle\nabla f(\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})), \mathbf{x} - \mathbf{y}\rangle - \langle\nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle$$
$$= \langle\nabla f(\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle$$
$$h_2'(\lambda) = \frac{\lambda\gamma}{D}\|\mathbf{x} - \mathbf{y}\|_1^2.$$

By given information from (**??**), we have:

$$\lambda h_1'(\lambda) = \langle\nabla f(\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})) - \nabla f(\mathbf{y}), [\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})] - \mathbf{y}\rangle$$
$$\geq \frac{\gamma}{D}\|[\mathbf{y} + \lambda(\mathbf{x} - \mathbf{y})] - \mathbf{y}\|_1^2 = \lambda h_2'(\lambda)$$
$$\therefore h_1'(\lambda) \geq h_2'(\lambda).$$

By Cauchy's mean value theorem, there exists $c \in (0,1)$ such that:

$$h_1(1) - h_1(0) = \frac{h_1'(c)}{h_2'(c)}(h_2(1) - h_2(0)) \geq h_2(1) - h_2(0) \tag{14}$$

$$\therefore f(\mathbf{x}) \geq f(\mathbf{y}) + \langle\nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle + \frac{\gamma}{2D}\|\mathbf{x} - \mathbf{y}\|_1^2. \tag{15}$$

Hence, we have the desired conclusion for any $\mathbf{x} \neq \mathbf{y}$. We also have (**??**) to hold trivially for $\mathbf{x} = \mathbf{y}$. $\quad\square$

Back to our main proof of Lemma **??**, we proceed to prove that for all $\mathbf{x}, \mathbf{y} \in Q$, the inequality (**??**) holds. We consider $\mathbf{x} \neq \mathbf{y}$, while for $\mathbf{x} = \mathbf{y}$ (**??**) trivially holds. Since $\dfrac{\partial f(\mathbf{x})}{\partial x_i} = \gamma log(x_i) + \gamma + d_i$, we rewrite (**??**) as:

$$\sum_{i=1}^{k}\gamma(x_i - y_i)(log(x_i) - log(y_i)) \geq \frac{\gamma}{D}\|\mathbf{x} - \mathbf{y}\|_1^2.$$

For $\lambda \in [0,1]$, we define $s_1(\lambda) = \sum_{i=1}^{k}(x_i - y_i)log(y_i + \lambda(x_i - y_i))$ and $s_2(\lambda) = \lambda\|\mathbf{x} - \mathbf{y}\|_1^2$, and further obtain:

$$s_1'(\lambda) = \sum_{i=1}^{k}\frac{(x_i - y_i)^2}{y_i + \lambda(x_i - y_i)}$$
$$s_2'(\lambda) = \|\mathbf{x} - \mathbf{y}\|_1^2.$$

Using $\sum_{i=1}^{k}[y_i + \lambda(x_i - y_i)] = \sum_{i=1}^{k}[\lambda x_i + (1 - \lambda)y_i] \leq \lambda D + (1 - \lambda)D = D$, we have:

$$D \cdot s_1'(\lambda) \geq \sum_{i=1}^{k}[y_i + \lambda(x_i - y_i)] \cdot \sum_{i=1}^{k}\frac{(x_i - y_i)^2}{y_i + \lambda(x_i - y_i)}$$
$$\geq (\sum_{i=1}^{k}|x_i - y_i|)^2 = \|\mathbf{x} - \mathbf{y}\|_1^2 \text{ (by Cauchy-Schwarz inequality)}$$
$$\geq s_2'(\lambda).$$

By Cauchy's mean value theorem, there exists $c \in (0,1)$ such that:

$$D(s_1(1) - s_1(0)) = \frac{D \cdot s_1'(c)}{s_2'(c)}(s_2(1) - s_2(0)) \geq s_2(1) - s_2(0).$$

Since $s_1(1) - s_1(0) = \sum_{i=1}^{k}(x_i - y_i)(log(x_i) - log(y_i))$ and $s_2(1) - s_2(0) = \|\mathbf{x} - \mathbf{y}\|_1^2$, we get:

$$\sum_{i=1}^{k}(x_i - y_i)(log(x_i) - log(y_i)) \geq \frac{1}{D}\|\mathbf{x} - \mathbf{y}\|_1^2$$

The condition (**??**) thus follows for all $\mathbf{x}, \mathbf{y} \in Q$. By Observation **??**, $f(\mathbf{x})$ is $\dfrac{\gamma}{\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s}$-strongly-convex with respect to ℓ_1 norm over Q. $\quad\square$

Dual Formulation for Entropic POT

First we note the general form of Equation (**??**). Given a simple closed convex set Q in a finite-dimensional real vector space E; H is another finite-dimensional real vector space with an element $\mathbf{b}$; H^* is the dual space of H. $\mathbf{A}$ is a given linear operator from E to H. Consider

$$\min_{\mathbf{x} \in Q \subseteq E} f(\mathbf{x}) \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{16}$$

where $f(\mathbf{x})$ is a μ_f-strongly convex function on Q with respect to some chosen norm $\|\cdot\|_E$ on E. If $\gamma = 0$, we recover Problem (**??**).

Let $\mathbf{y} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n$ and $t \in \mathbb{R}$ be the dual variables corresponding to the three equality constraints. The Lagrangian is

$$\mathcal{L} = \langle \mathbf{C}, \mathbf{X} \rangle + \gamma(\langle \mathbf{X}, \log \mathbf{X} \rangle + \langle \mathbf{p}, \log \mathbf{p} \rangle + \langle \mathbf{q}, \log \mathbf{q} \rangle)+$$
$$\langle \mathbf{y}, \mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r} \rangle + \langle \mathbf{z}, \mathbf{X}^\top \mathbf{1} + \mathbf{q} - \mathbf{c} \rangle + t(\mathbf{1}^\top \mathbf{X}\mathbf{1} - s),$$

Hence, we have the dual problem:

$$\max_{\mathbf{y},\mathbf{z},t} \left\{ -\langle \mathbf{y}, \mathbf{r} \rangle - \langle \mathbf{z}, \mathbf{c} \rangle - ts + \min_{X_{i,j} \geq 0} \sum_{i,j=1}^n X_{i,j}(C_{i,j} + \gamma \log X_{i,j} + y_i + z_j + t) \right.$$
$$\left. + \min_{p_i \geq 0} \sum_{i=1}^n p_i(\gamma \log p_i + y_i) + \min_{q_j \geq 0} \sum_{j=1}^n q_j(\gamma \log q_j + z_j) \right\}. \tag{17}$$

Next, we minimize the Lagrangian w.r.t. $\mathbf{X}$, $\mathbf{p}$ and $\mathbf{q}$ according to the first order condition, which gives

$$X_{i,j} = \exp\left(-\frac{1}{\gamma}(C_{i,j} + y_i + z_j + t) - 1 \right), p_i = \exp\left(-\frac{y_i}{\gamma} - 1 \right), q_j = \exp\left(-\frac{z_j}{\gamma} - 1 \right).$$

Plugging this back to the dual problem (**??**), we have

$$\max_{\mathbf{y},\mathbf{z},t} \left\{ -\langle \mathbf{y}, \mathbf{r} \rangle - \langle \mathbf{z}, \mathbf{c} \rangle - ts - \gamma \sum_{i,j=1}^n \exp\left(-\frac{1}{\gamma}(C_{i,j} + y_i + z_j + t) - 1 \right) \right.$$
$$\left. -\gamma \sum_{i=1}^n \exp\left(-\frac{y_i}{\gamma} - 1 \right) - \gamma \sum_{j=1}^n \exp\left(-\frac{z_j}{\gamma} - 1 \right) \right\},$$

as desired. Plugging this back into the dual problem, suppressing all additive constants and diving the objective by $-\gamma$, we obtain the following new dual problem:

$$\min_{\mathbf{u},\mathbf{v},w} h(\mathbf{u}, \mathbf{v}, w) := \sum_{i,j=1}^n e^w e^{u_i} e^{-C_{i,j}/\gamma} e^{v_j} + \sum_{i=1}^n e^{u_i} + \sum_{j=1}^n e^{v_j} - \langle \mathbf{u}, \mathbf{r} \rangle - \langle \mathbf{v}, \mathbf{c} \rangle - ws. \tag{18}$$

APDAGD

APDAGD Theoretical Guarantees

We have the following guarantees for APDAGD , similar to (**?**, Theorem 3).

Theorem 16. *Let $\mathbf{x}^*$ and f^* be the optimal solution and optimal value of (**??**), respectively. Suppose that $\|\boldsymbol{\lambda}^*\|_2 \leq \bar{R}$, the outputs $\hat{\mathbf{x}}_k$ and $\boldsymbol{\eta}_k$ from Algorithm **??** satisfy*

$$f(\hat{\mathbf{x}}_k) - f^* \leq f(\hat{\mathbf{x}}_k) + \varphi(\boldsymbol{\eta}_k) \leq \frac{16 \|\mathbf{A}\|_{1\to 2}^2 \bar{R}^2}{\mu_f k^2},$$

$$\|\mathbf{A}\hat{\mathbf{x}}_k - \mathbf{b}\|_2 \leq \frac{16\bar{R} \|\mathbf{A}\|_{1\to 2}^2}{\mu_f k^2},$$

$$\|\hat{\mathbf{x}}_k - \mathbf{x}^*\|_1 \leq \frac{8\bar{R} \|\mathbf{A}\|_{1\to 2}}{\mu_f k}.$$

Complexity of APDAGD for POT (Theorem ??) Detailed Proof

Step 1: Change of Dual Variables   For the sake of convenience, we introduce the change of variables $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}$ such that $u_i = -y_i/\gamma - 1$, $v_j = -z_j/\gamma - 1$ and $w = -t/\gamma + 1$. After suppressing all additive constants and diving the objective by $-\gamma$, we obtain the equivalent dual problem:

$$\min_{\mathbf{u},\mathbf{v},w} h(\mathbf{u},\mathbf{v},w) := \sum_{i,j=1}^n \exp\left(-\frac{C_{i,j}}{\gamma} + u_i + v_j + w\right) + \sum_{i=1}^n \exp(u_i)+ \tag{19}$$

$$\sum_{j=1}^n \exp(v_j) - \langle \mathbf{u}, \mathbf{r} \rangle - \langle \mathbf{v}, \mathbf{c} \rangle - ws. \tag{20}$$

Let $\mathbf{u}^*, \mathbf{v}^*, w^*$ be the optimal solutions of the entropic regularized problem (**??**). Denote $u_{max}^* = \max_{1 \le i \le n} u_i^*$, $u_{min}^* = \min_{1 \le i \le n} u_i^*$ and similarly for v_{max}^* and v_{min}^*. Notice that $\forall\, 1 \le i \le n$, $\exp(u_i^*) = p_i \le r_i$. This implies $\forall\, 1 \le i \le n$, $u_i^* \le \log r_i \le \log 1 = 0$. So, $u_{max}^* \le 0$. Similarly, $v_{max}^* \le 0$.

Step 2: Bound for Infinity Norm of Transformed Dual Variables

Lemma 17. *The upper bound for $\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty$ is*

$$\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty \le \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)} - \log\left(\min_{1 \le i,j \le n}(r_i, c_j)\right).$$

Proof. First we lower bound w^*, note that

$$s = \sum_{i,j=1}^n \exp\left(\frac{-C_{i,j}}{\gamma} + u_i^* + v_j^* + w^*\right) \le n^2 \exp\left(u_{max}^* + v_{max}^* + w^*\right).$$

Taking logs on both sides, we have

$$\log s \le 2\log n + u_{max}^* + v_{max}^* + w^*.$$

Since $u_{max}^*, v_{max}^* \le 0$, we obtain the desired lower bound

$$w^* \ge \log s - 2\log n - u_{max}^* - v_{max}^* \ge \log s - 2\log n.$$

For the upper bound, note that if we let h^* be the optimal objective value of $h(\mathbf{u}, \mathbf{v}, w)$ in the transformed dual problem (**??**), then by accounting for the additive constants the optimal value of the original dual problem (**??**) will be

$$-\gamma h^* + \gamma(\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s). \tag{21}$$

Accompanying (**??**) with the fact that the primal objective (**??**) optimal value is equal to the dual objective optimal (**??**) value by strong duality, we have that

$$-\gamma h^* + \gamma(\|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s) \ge 0,$$

which yields

$$h^* \le \|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - s. \tag{22}$$

On the other hand, consider

$$s + \sum_{i=1}^n e^{u_i^*} = \sum_{i,j=1}^n \exp\left(\frac{-C_{i,j}}{\gamma} + u_i^* + v_j^* + w^*\right) + \sum_{i=1}^n e^{u_i^*} = \sum_{i,j=1}^n X_{i,j} + \sum_{i=1}^n p_i = \|\mathbf{r}\|_1,$$

$$s + \sum_{j=1}^n e^{v_j^*} = \sum_{i,j=1}^n \exp\left(\frac{-C_{i,j}}{\gamma} + u_i^* + v_j^* + w^*\right) + \sum_{j=1}^n e^{v_j^*} = \sum_{i,j=1}^n X_{i,j} + \sum_{j=1}^n p_j = \|\mathbf{c}\|_1.$$

This leads to

$$h^* = h(\mathbf{u}^*, \mathbf{v}^*, w^*) = -s + \|\mathbf{r}\|_1 + \|\mathbf{c}\|_1 - \langle \mathbf{u}^*, r \rangle - \langle \mathbf{v}^*, c \rangle - w^* s.$$

Together with the inequality (**??**),

$$-\langle \mathbf{u}^*, r \rangle - \langle \mathbf{v}^*, c \rangle - w^* s \le 0.$$

This implies

$$-w^* s \le \langle \mathbf{u}^*, \mathbf{r} \rangle + \langle \mathbf{v}^*, \mathbf{c} \rangle \le u_{\max}^* \|\mathbf{r}\|_1 + v_{\max}^* \|\mathbf{c}\|_1 \le (u_{\max}^* + v_{\max}^*) \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1),$$

which implies

$$u_{\max}^* + v_{\max}^* \ge \frac{-w^* s}{\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}.$$

We also know that
$$u_i^* + v_j^* + w^* - \frac{C_{ij}}{\gamma} \leq 0 \ \forall i, j.$$
Hence, combining the two results above, we obtain
$$\frac{\|\mathbf{C}\|_{\max}}{\gamma} \geq u_{\max}^* + v_{\max}^* + w^* \geq \frac{-w^* s}{\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)} + w^*,$$
yielding the desire upper bound
$$w^* \leq \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)}.$$
Note that $\forall \ 1 \leq j \leq n$, we have
$$\left(\sum_{j=1}^n \exp\left(v_j^*\right)\right) \exp\left(u_i^* + w^*\right) \geq \sum_{j=1}^n \exp\left(-\frac{C_{i,j}}{\gamma} + u_i^* + v_j^* + w^*\right)$$
$$= \sum_{j=1}^n X_{i,j}$$
$$= r_i$$
$$\geq \min_{1 \leq i,j \leq n}(r_i, c_j).$$
Taking log on both sides, we get
$$\log\left(\sum_{j=1}^n \exp\left(v_j^*\right)\right) + u_i^* + w^* \geq \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right),$$
which implies
$$u_i^* \geq \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) - \log\left(\sum_{j=1}^n \exp\left(v_j^*\right)\right) - w^*$$
$$\geq \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) - \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)}.$$
Similarly, we also obtain the same lower bound for v_j^*, $\forall 1 \leq i \leq n$. And as $\forall 1 \leq i, j \leq n$, $u_i^*, v_j^* \leq 0$, we obtain
$$\|(\mathbf{u}^*, \mathbf{v}^*)\|_\infty \leq \left|\log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) - \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)}\right|$$
$$= -\log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) + \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)}.$$
Combining this with the bounds for w^*, we have
$$\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty \leq \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\gamma(\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s)} - \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right).$$
$\square$

Corollary 18. *The upper bound for* $\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty$ *is*
$$\frac{\|\mathbf{C}\|_{\max} \max\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\}}{\max\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\} - s} - \gamma \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) + \gamma. \tag{23}$$

Proof. We have
$$\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty = \gamma \|(\mathbf{u}^* + \mathbf{1}, \mathbf{v}^* + \mathbf{1}, w^* - 1)\|_\infty$$
$$\leq \gamma \left(\|(\mathbf{u}^*, \mathbf{v}^*, w^*)\|_\infty + 1\right)$$
$$\leq \frac{\|\mathbf{C}\|_{\max} \max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1)}{\max(\|\mathbf{r}\|_1, \|\mathbf{c}\|_1) - s} - \gamma \log\left(\min_{1 \leq i,j \leq n}(r_i, c_j)\right) + \gamma,$$
where the last inequality comes as a result of Step 2. $\square$

Note that $\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_2 \leq \sqrt{n} \|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty$, so we obtain $\bar{R} = \widetilde{\mathcal{O}}\left(\sqrt{n} \|\mathbf{C}\|_{\max}\right)$.

Step 3: Derive the Final APDAGD Computational Complexity (Theorem ??)

With the introduction of $\widetilde{\mathbf{r}}$ and $\widetilde{\mathbf{c}}$ in Algorithm **??**, our POT constraints now becomes $\mathbf{A}\mathbf{x} = \widetilde{\mathbf{b}}$, where $\widetilde{\mathbf{b}} = (\widetilde{\mathbf{r}}^\top, \widetilde{\mathbf{c}}^\top, s)^\top$. We can still ensure that $\|\widetilde{\mathbf{r}}\|_1 \geq s$ as follow

$$
\begin{aligned}
\|\widetilde{\mathbf{r}}\|_1 &= \left(1 - \frac{\widetilde{\varepsilon}}{8}\right)\|\mathbf{r}\|_1 + \frac{\widetilde{\varepsilon}}{8} \\
&= \frac{\widetilde{\varepsilon}}{8}\left(1 - \|\mathbf{r}\|_1\right) + \|\mathbf{r}\|_1.
\end{aligned}
$$

If $\|\mathbf{r}\|_1 \leq 1$, we obtain $\|\widetilde{\mathbf{r}}\|_1 \geq \|\mathbf{r}\|_1 \geq s$. And if $\|\mathbf{r}\|_1 > 1$, note that as $\widetilde{\varepsilon} \leq \dfrac{8(\|\mathbf{r}\|_1 - s)}{\|\mathbf{r}\|_1 - 1}$, we have $\|\widetilde{\mathbf{r}}\|_1 \geq s - \|\mathbf{r}\|_1 + \|\mathbf{r}\|_1 = s$. Similarly, we can ensure $\|\widetilde{\mathbf{c}}\|_1 \geq s$. With these changes to $\widetilde{\mathbf{r}}$ and $\widetilde{\mathbf{c}}$, we obtain from Corollary **??**

$$
\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty \leq \frac{\|\mathbf{C}\|_{\max}\max(\|\widetilde{\mathbf{r}}\|_1, \|\widetilde{\mathbf{c}}\|_1)}{\max(\|\widetilde{\mathbf{r}}\|_1, \|\widetilde{\mathbf{c}}\|_1) - s} - \gamma \log\left(\min_{1 \leq i,j \leq n}(\widetilde{r}_i, \widetilde{c}_j)\right) + \gamma, \tag{24}
$$

implying $\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty = \widetilde{\mathcal{O}}(\|\mathbf{C}\|_{\max})$ since $\log\left(\min_{1 \leq i,j \leq n}(\widetilde{r}_i, \widetilde{c}_j)\right) = \widetilde{O}(\log(n/\varepsilon))$.

From Proposition 4.10 from (**?**), the number of iterations for APDAGD Algorithm required to reach the tolerance ε is

$$
t \leq \max\left\{\mathcal{O}\left(\min\left\{\frac{n^{1/4}\sqrt{\bar{R}\,\|C\|_{max}\log n}}{\varepsilon}, \frac{\bar{R}\,\|C\|_{max}\log n}{\varepsilon^2}\right\}\right), \mathcal{O}\left(\frac{\bar{R}\log n}{\varepsilon}\right)\right\} \tag{25}
$$

From the bound (**??**) and the fact that $\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_2 \leq \sqrt{n}\|(\mathbf{y}^*, \mathbf{z}^*, t^*)\|_\infty$, we obtain $\bar{R} = \widetilde{\mathcal{O}}(\sqrt{n}\|\mathbf{C}\|_{\max})$. Therefore, from (**??**), the total iteration complexity is $\mathcal{O}(n^{1/2}\|\mathbf{C}\|_{\max}\sqrt{\log n}/\varepsilon)$. Since each iteration of APDAGD algorithm requires $\mathcal{O}(n^2)$ arithmetic operations, and the rounding algorithm also requires $\mathcal{O}(n^2)$ arithmetic operations, we have the total number of arithmetic operations is the desired $\mathcal{O}(n^{5/2}\|\mathbf{C}\|_{\max}\sqrt{\log n}/\varepsilon)$. $\qquad\square$

Dual Extrapolation

Second Order Characterization of Area Convexity

In our context, the second order characterization of area-convexity, which was proven in (**?**, Theorem 1.6) and restated in (**?**, Theorem 3.2), is quite useful

Theorem 19. *For bilinear minimax objective and twice-differentiable regularizer r, if for all $\mathbf{z}$ in the domain*

$$
\begin{pmatrix} \kappa\nabla^2 r(\mathbf{z}) & -J \\ J & \kappa\nabla^2 r(\mathbf{z}) \end{pmatrix} \succeq 0,
$$

where J is the Jacobian of the gradient operator g, then r is 3κ-area-convex with respect to g.

$\ell 1$ Penalization

Lemma 20. *The ℓ_1 penalized POT objective,*

$$
\min_{\mathbf{x} \in \Delta_{n^2+2n}} \mathbf{d}^\top\mathbf{x} + 23\|\mathbf{d}\|_\infty\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1, \tag{26}
$$

*has an equal optimal value to that of the POT formulation (**??**). Moreover, any ε-approximate minimizer $\widetilde{\mathbf{x}}$ of the ℓ_1 penalized objective (**??**) yields in $\mathcal{O}(n^2)$ time an ε-approximate transport plan $\bar{\mathbf{x}}$ for the POT formulation (**??**) (after applied the Rounding Algorithm **??**).*

Proof. Note that, intuitively, multiplicative constant 23 comes from the guarantees of ROUND-POT (Theorem **??**).

Let $\tilde{\mathbf{x}} \in \arg\min_{\mathbf{x} \in \Delta_{n^2+2n}} \mathbf{d}^\top\mathbf{x} + 23\|\mathbf{d}\|_\infty\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$. Following the approach from (**?**, Lemma 1), we claim there is some $\tilde{\mathbf{x}}$ satisfying $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$. Suppose otherwise, let $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_1 = \delta > 0$. Then, let $\bar{\mathbf{x}}$ be the output after rounding $\tilde{\mathbf{x}}$ with Algorithm **??**, so that $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$, $\|\tilde{\mathbf{x}} - \bar{\mathbf{x}}\|_1 \leq 23\delta$ (Theorem **??**). Consider

$$
\mathbf{d}^\top\bar{\mathbf{x}} + 23\|\mathbf{d}\|_\infty\|\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}\|_1 = \mathbf{d}^\top(\bar{\mathbf{x}} - \tilde{\mathbf{x}}) + \mathbf{d}^\top\tilde{\mathbf{x}} \leq \|\mathbf{d}\|_\infty\|\bar{\mathbf{x}} - \tilde{\mathbf{x}}\|_1 + \mathbf{d}^\top\tilde{\mathbf{x}} \leq 23\|\mathbf{d}\|_\infty\delta + \mathbf{d}^\top\tilde{\mathbf{x}}.
$$

The ℓ_1 penalized objective value of $\bar{\mathbf{x}}$ is smaller than that of $\tilde{\mathbf{x}}$, hence this implies contradiction. From this claim, it is evident that

$$
\min_{\mathbf{x} \in \Delta_{n^2+2n}} \mathbf{d}^\top\mathbf{x} + 23\|\mathbf{d}\|_\infty\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1 = \mathbf{d}^\top\tilde{\mathbf{x}} + 23\|\mathbf{d}\|_\infty\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_1 = \mathbf{d}^\top\tilde{\mathbf{x}}.
$$

Let $\hat{\mathbf{x}} \in \arg\min_{\mathbf{Ax}=\mathbf{b}} \mathbf{d}^\top \mathbf{x}$. If we consider the objective (**??**) for $\hat{\mathbf{x}}$, we have

$$\min_{\mathbf{Ax}=\mathbf{b}} \mathbf{d}^\top \mathbf{x} = \mathbf{d}^\top \hat{\mathbf{x}}$$
$$= \mathbf{d}^\top \hat{\mathbf{x}} + 23 \|\mathbf{d}\|_\infty \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_1$$
$$\geq \min_{\mathbf{x} \in \Delta_{n^2+2n}} \mathbf{d}^\top \mathbf{x} + 23 \|\mathbf{d}\|_\infty \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$
$$= \mathbf{d}^\top \tilde{\mathbf{x}}.$$

Enforcing equality sign, we obtain the first claim that problems (**??**) and (**??**) have the same optimal value. Therefore, we can take any approximate minimizer to (**??**) and round it to a transport plan $\bar{\mathbf{x}}$ without increasing the POT objective. $\square$

Proof of Lemma ??

With the second order condition for area convexity (Theorem **??**) it suffices to show that for any z in the domain

$$\xi = \begin{pmatrix} 3\nabla^2 r(\mathbf{z}) & -J \\ J & 3\nabla^2 r(\mathbf{z}) \end{pmatrix} \succeq 0,$$

where J is the Jacobian of the operator g.

First, note that we can scale down both J and $\nabla^2 r(\mathbf{z})$ with a common factor of $2\|\mathbf{d}\|_\infty$ and still maintain the positive-semidefiniteness of the matrix ξ. By straightforward differentiation, we obtain $J = \mathbf{A}^\top$ and

$$\nabla^2 r(z) = \begin{pmatrix} 10\,\text{diag}(\frac{1}{x_j}) & 2\,\mathbf{A}^\top \text{diag}(y_i) \\ 2\,\text{diag}(y_i)\mathbf{A} & 2\,\text{diag}(\mathbf{A}_i^\top \mathbf{x}) \end{pmatrix}.$$

Hence we get

$$\xi = \begin{pmatrix} 30\,\text{diag}(\frac{1}{x_j}) & 6\,\mathbf{A}^\top \text{diag}(y_i) & \mathbf{0} & -\mathbf{A}^\top \\ 6\,\text{diag}(y_i)\mathbf{A} & 6\,\text{diag}(\mathbf{A}_i^\top \mathbf{x}) & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & 30\,\text{diag}(\frac{1}{x_j}) & 6\,\mathbf{A}^\top \text{diag}(y_i) \\ -\mathbf{A} & \mathbf{0} & 6\,\text{diag}(y_i)\mathbf{A} & 6\,\text{diag}(\mathbf{A}_i^\top \mathbf{x}) \end{pmatrix}.$$

As the linear operator $\mathbf{A}$ has the structure (**??**), we obtain $\|\mathbf{A}_{:j}\|_1 \leq 3$. Given any arbitrary vector $(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})^\top$, since $x_j \geq 0 \,\forall\, j$, we note that

$$\mathbf{a}\xi\mathbf{a}^\top = \sum_j \frac{30a_j^2}{x_j} \geq \sum_j \|\mathbf{A}_{:j}\|_1 \frac{10a_j^2}{x_j} = \sum_{i,j} A_{i,j} \frac{10a_j^2}{x_j},$$
$$\mathbf{c}\xi\mathbf{c}^\top = \sum_j \frac{30c_j^2}{x_j} \geq \sum_j \|\mathbf{A}_{:j}\|_1 \frac{10c_j^2}{x_j} = \sum_{i,j} A_{i,j} \frac{10c_j^2}{x_j}.$$

Therefore expand the terms and simplify $(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})\,\xi\,(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})^\top$ as follows

$$(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})\,\xi\,(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})^\top$$
$$\geq \sum_{i,j} A_{ij} \left(\frac{10a_j^2}{x_j} + 6b_i^2 x_j + \frac{10c_j^2}{x_j} + 6d_i^2 x_j + 12a_j b_i y_i + 12c_j d_i y_i - 2a_j d_i + 2c_j b_i \right).$$

Note that as $y_i \in [-1, 1]$, we have

$$\frac{9a_j^2}{x_j} + 12a_j b_i y_i + 4b_i^2 x_j \geq 0, \quad \frac{9c_j^2}{x_j} + 12c_j d_i y_i + 4d_i^2 x_j \geq 0.$$

So we obtain

$$(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})\,\xi\,(\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,\mathbf{d})^\top$$
$$\geq \sum_{i,j} A_{ij} \left(\frac{a_j^2}{x_j} + 2b_i^2 x_j + \frac{c_j^2}{x_j} + 2d_i^2 x_j - 2a_j d_i + 2c_j b_i \right)$$
$$= \sum_{i,j} A_{ij} \left(\left(\frac{a_j^2}{x_j} + d_i^2 x_j - 2a_j d_i \right) + \left(b_i^2 x_j + \frac{c_j^2}{x_j} + 2c_j b_i \right) + b_i^2 x_j + d_i^2 x_j \right)$$
$$\geq 0,$$

since $x_j \geq 0$. Hence $\xi \succeq 0$ as desired. $\square$

Proof of Lemma ??

By (**?**, Lemma 2), we can evaluate the proposed algorithm with output $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \mathcal{X} \times \mathcal{Y}$ by the duality gap

$$\max_{\mathbf{y} \in \mathcal{Y}} F(\bar{\mathbf{x}}, \mathbf{y}) - \min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}, \bar{\mathbf{y}}) \leq \varepsilon. \tag{27}$$

As we want to obtain the convergence guarantees in terms of duality gap (**??**), we consider the following lemma which has been stated and proven in (**?**, Corollary 1):

Lemma 21. *Suppose that the proximal steps in General Dual Extrapolation Algorithm* **??** *have an additive error ε' during implementation and the regularizer r is κ-area-convex with respect to the gradient operator g. Suppose for some $\mathbf{u} \in \mathcal{Z}, \Theta \geq r(\mathbf{u}) - r(\bar{\mathbf{z}})$. Then the output $\bar{\mathbf{w}}$ of Algorithm* **??** *satisfies*

$$\langle g(\bar{\mathbf{w}}), \bar{\mathbf{w}} - \mathbf{u} \rangle \leq \frac{2\kappa\Theta}{T} + \varepsilon'.$$

From this lemma, with a chosen additive error $\varepsilon' = \dfrac{\varepsilon}{2}$, if the aim is $\langle g(\bar{\mathbf{w}}), \bar{\mathbf{w}} - \mathbf{u} \rangle \leq \varepsilon$, then we want to have $\dfrac{2\kappa\Theta}{T} \leq \dfrac{\varepsilon}{2}$. So T is required to be $\left\lceil \dfrac{4\kappa\Theta}{\varepsilon} \right\rceil$. Next, we look for a suitable Θ.

Lemma 22. *Lemma* **??** *will apply to $\forall\, \mathbf{u} \in \mathcal{Z}$ if $\Theta = 60 \|\mathbf{d}\|_\infty \log(n) + 6 \|\mathbf{d}\|_\infty$.*

Proof. Note that it is obvious that we want to choose

$$\Theta \geq \sup_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}) - \inf_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}).$$

Denote the negative entropy as $h(\mathbf{x})$, it is well-known that

$$\max_{\mathbf{x} \in \Delta_d} h(\mathbf{x}) - \min_{\mathbf{x} \in \Delta_d} h(\mathbf{x}) = \log d.$$

Next, we consider

$$
\begin{aligned}
\mathbf{x}^\top \mathbf{A}^\top (\mathbf{y}^2) &= \langle \mathbf{x}, \mathbf{A}^\top (\mathbf{y}^2) \rangle \\
&\leq \|\mathbf{x}\|_1 \left\| \mathbf{A}^\top (\mathbf{y}^2) \right\|_\infty \\
&= \left\| \mathbf{A}^\top (\mathbf{y}^2) \right\|_\infty \\
&\leq \left\| \mathbf{A}^\top \mathbf{1}_{2n+1} \right\|_\infty \\
&= \left\| \mathbf{A}^\top \right\|_\infty \\
&= 3.
\end{aligned}
$$

On the other hand, it is obvious that $\mathbf{x}^\top \mathbf{A}^\top (\mathbf{y}^2) \geq 0$. Combining the previous few inequalities, we have the following bound for the range of regularizer $r(\mathbf{x}, \mathbf{y})$

$$\sup_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}) - \inf_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}) \leq 2 \|\mathbf{d}\|_\infty \left(10 \log(n^2 + 2n) + 3 \right) = 20 \|\mathbf{d}\|_\infty \log(n^2 + 2n) + 6 \|\mathbf{d}\|_\infty.$$

It is noteworthy that with $n \geq 2, n^3 = n \cdot n^2 \geq n \cdot (n + 2) = n^2 + 2n$. Hence, we choose

$$\Theta = 60 \|\mathbf{d}\|_\infty \log(n) + 6 \|\mathbf{d}\|_\infty \geq 20 \|\mathbf{d}\|_\infty \log(n^2 + 2n) + 6 \|\mathbf{d}\|_\infty \geq \sup_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}) - \inf_{\mathbf{z} \in \mathcal{Z}} r(\mathbf{z}).$$

$\square$

Proof of Theorem ??

By (**?**, Lemma 7, 8), the suboptimality gap can be reduced by a factor of $1/24$. There are two proximal step, and the second one deals with $\mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t)$ which is bigger than $\mathbf{s}^t$ in the first step. Hence, we endeavor to bound the number of iterations for the second bigger proximal step of $\mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t)$. We have the proximal objective

$$
\begin{aligned}
prox_{\bar{\mathbf{z}}}^r\left(\mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t)\right) &= \arg\min_{\mathbf{z} \in \mathcal{Z}} \langle \mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t), \mathbf{z} \rangle + V_{\bar{\mathbf{z}}}^r(\mathbf{z}) \\
&= \arg\min_{\mathbf{z} \in \mathcal{Z}} \langle \mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t), \mathbf{z} \rangle + r(\mathbf{z}) - r(\bar{\mathbf{z}}) - \langle \nabla r(\bar{\mathbf{z}}), \mathbf{z} - \bar{\mathbf{z}} \rangle \\
&= \arg\min_{\mathbf{z} \in \mathcal{Z}} \langle \mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t) - \nabla r(\bar{\mathbf{z}}), \mathbf{z} \rangle + r(\mathbf{z}).
\end{aligned}
$$

Let $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$, $g(\mathbf{z}) = (g_{\mathbf{x}}(\mathbf{z}), g_{\mathbf{y}}(\mathbf{z}))$, and $\mathbf{s}^t = (\mathbf{s}_{\mathbf{x}}^t, \mathbf{s}_{\mathbf{y}}^t)$, we obtain

$$prox_{\bar{\mathbf{z}}}^r \left(\mathbf{s}^t + \frac{1}{\kappa} g(\mathbf{z}^t) \right) = \arg\min_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} \langle \mathbf{s}_{\mathbf{x}}^t - \nabla_{\mathbf{x}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) + \frac{1}{\kappa} g_{\mathbf{x}}(\mathbf{z}^t), \mathbf{x} \rangle +$$

$$\langle \mathbf{s}_{\mathbf{y}}^t - \nabla_{\mathbf{y}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) + \frac{1}{\kappa} g_{\mathbf{y}}(\mathbf{z}^t), \mathbf{y} \rangle + r(\mathbf{x}, \mathbf{y}).$$

Note that $\nabla_{\mathbf{y}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \mathbf{0}$. Let $\mathbf{x}^*$ and $\mathbf{y}^*$ be the minimizer of the proximal objective, while $\mathbf{x}_0$ and $\mathbf{y}_0$ are the initializations, the suboptimality gap is

$$\delta = \langle \mathbf{s}_{\mathbf{x}}^t - \nabla_{\mathbf{x}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) + \frac{1}{\kappa} g_{\mathbf{x}}(\mathbf{z}^t), \mathbf{x}_0 - \mathbf{x}^* \rangle + \langle \mathbf{s}_{\mathbf{y}}^t + \frac{1}{\kappa} g_{\mathbf{y}}(\mathbf{z}^t), \mathbf{y}_0 - \mathbf{y}^* \rangle + r(\mathbf{x}_0, \mathbf{y}_0) - r(\mathbf{x}^*, \mathbf{y}^*).$$

Now the goal is to bound this suboptimality gap. By Lemma **??**, as proximal steps are implemented with $\varepsilon/2$ accuracy, we get that $T = \lceil 36\Theta/\varepsilon \rceil$ iterations would suffice. We can bound $\mathbf{s}_{\mathbf{x}}^t$ and $\mathbf{s}_{\mathbf{y}}^t$ as follows

$$\left\| \mathbf{s}_{\mathbf{x}}^t \right\|_\infty \leq T \frac{1}{2\kappa} \left\| g_{\mathbf{x}}(\mathbf{z}) \right\|_\infty \leq \frac{2\Theta}{\varepsilon} \left\| \mathbf{d} + 23 \left\| \mathbf{d} \right\|_\infty \mathbf{A}^\top \mathbf{y} \right\|_\infty \leq \frac{2\Theta}{\varepsilon} \left(\left\| \mathbf{d} \right\|_\infty + 23 \left\| \mathbf{d} \right\|_\infty \left\| \mathbf{A}^\top \mathbf{y} \right\|_\infty \right)$$

$$\leq \frac{140\Theta \left\| \mathbf{d} \right\|_\infty}{\varepsilon},$$

$$\left\| \mathbf{s}_{\mathbf{y}}^t \right\|_1 \leq T \frac{1}{2\kappa} \left\| g_{\mathbf{y}}(\mathbf{z}) \right\|_1 \leq \frac{2\Theta}{\varepsilon} \left\| -23 \left\| \mathbf{d} \right\|_\infty (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_1 \leq \frac{46\Theta \left\| \mathbf{d} \right\|_\infty}{\varepsilon} \left(\left\| \mathbf{A}\mathbf{x} \right\|_1 + \left\| \mathbf{b} \right\|_1 \right)$$

$$\leq \frac{276\Theta \left\| \mathbf{d} \right\|_\infty}{\varepsilon}.$$

Similarly, we have

$$\frac{1}{\kappa} \left\| g_{\mathbf{x}}(\mathbf{z}) \right\|_\infty \leq \frac{1}{9} \left\| \mathbf{d} + 23 \left\| \mathbf{d} \right\|_\infty \mathbf{A}^\top \mathbf{y} \right\|_\infty \leq \frac{1}{9} \left(\left\| \mathbf{d} \right\|_\infty + 23 \left\| \mathbf{d} \right\|_\infty \left\| \mathbf{A}^\top \mathbf{y} \right\|_\infty \right) \leq \frac{70 \left\| \mathbf{d} \right\|_\infty}{9},$$

$$\frac{1}{\kappa} \left\| g_{\mathbf{y}}(\mathbf{z}) \right\|_1 \leq \frac{1}{9} \left\| -23 \left\| \mathbf{d} \right\|_\infty (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_1 \leq \frac{23 \left\| \mathbf{d} \right\|_\infty}{9} \left(\left\| \mathbf{A}\mathbf{x} \right\|_1 + \left\| \mathbf{b} \right\|_1 \right) \leq \frac{46 \left\| \mathbf{d} \right\|_\infty}{3}.$$

Also note that

$$\left\| \nabla_{\mathbf{x}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \right\|_\infty = \left\| 20 \left\| \mathbf{d} \right\|_\infty (1 - \log(n^2 + 2n)) \mathbf{1}_{n^2+2n} \right\|_\infty \leq 20 \left\| \mathbf{d} \right\|_\infty (1 + 3\log(n))$$

Hence, we can now proceed to bound the suboptimality gap

$$\delta \leq \left\| \mathbf{s}_{\mathbf{x}}^t - \nabla_{\mathbf{x}} r(\bar{\mathbf{x}}, \bar{\mathbf{y}}) + \frac{1}{\kappa} g_{\mathbf{x}}(\mathbf{z}^t) \right\|_\infty \left\| \mathbf{x}_0 - \mathbf{x}^* \right\|_1 + \left\| \mathbf{s}_{\mathbf{y}}^t + \frac{1}{\kappa} g_{\mathbf{y}}(\mathbf{z}^t) \right\|_1 \left\| \mathbf{y}_0 - \mathbf{y}^* \right\|_\infty + \Theta$$

$$\leq \left(\frac{140\Theta \left\| \mathbf{d} \right\|_\infty}{\varepsilon} + 20 \left\| \mathbf{d} \right\|_\infty (1 + 3\log(n)) + \frac{70 \left\| \mathbf{d} \right\|_\infty}{9} + \frac{70\Theta \left\| \mathbf{d} \right\|_\infty}{\varepsilon} + \frac{46 \left\| \mathbf{d} \right\|_\infty}{3} \right) \times 2 + \Theta$$

$$= \left(\frac{420 \left\| \mathbf{d} \right\|_\infty}{\varepsilon} + 1 \right) \Theta + 40 \left\| \mathbf{d} \right\|_\infty + 120 \left\| \mathbf{d} \right\|_\infty \log(n) + \frac{416 \left\| \mathbf{d} \right\|_\infty}{9}$$

$$= \left(\frac{420 \left\| \mathbf{d} \right\|_\infty}{\varepsilon} + 3 \right) \Theta + \frac{668 \left\| \mathbf{d} \right\|_\infty}{9}.$$

Therefore, the number of iterations to obtain the desired output with $\varepsilon/2$-accuracy is

$$M = 24 \log_{24/23} \left(\frac{2\delta}{\varepsilon} \right) \leq 24 \log \left(\left(\frac{840 \left\| \mathbf{d} \right\|_\infty}{\varepsilon^2} + \frac{6}{\varepsilon} \right) \Theta + \frac{1336 \left\| \mathbf{d} \right\|_\infty}{9} \right) = \mathcal{O}(\log \eta),$$

where $\eta = \log n \times \left\| \mathbf{d} \right\|_\infty \times \varepsilon^{-1}$. Each iteration can be done in $\mathcal{O}(n^2)$ time, so we obtain the desired complexity. □

Proof of Theorem ??

Proof. From Theorem **??**, each proximal step in Algorithm **??** requires $\mathcal{O}(n^2 \log \eta)$ time where $\eta = \log n \left\| \mathbf{d} \right\|_\infty / \varepsilon$. Hence, with 2 proximal steps, Algorithm **??** can be done in the total of $\mathcal{O}(n^2 \log n \left\| \mathbf{d} \right\|_\infty \log \eta / \varepsilon) = \widetilde{\mathcal{O}}(n^2 \left\| \mathbf{d} \right\|_\infty / \varepsilon)$ time. This is $\widetilde{\mathcal{O}}(n^2 \left\| \mathbf{C} \right\|_{max} / \varepsilon)$ in term of the original cost matrix $\mathbf{C}$. □

Further Experiment Setup Details

In this section, we will provide the more detailed experimental setup for the several applications in the main text.

Computational Resources

For all numerical experiments in this paper, we use Python 3.9 with all algorithms implemented using `numpy` version 1.23.2. We run all experiments on a Linux machine with an Intel Xeon Silver 4114 CPU with 40 threads and 128GB of RAM. No GPU is used.

APDAGD vs DE

Each RGB image of size $32 \times 32 \times 3$ is transformed into grayscale, downscaled to 10×10, and flattened into a histogram of 100 bins, all using the `scikit-image` library. We add 10^{-6} to every bin to avoid zeros. For every chosen pair of marginals, we divide each histogram by the maximum mass of the two, giving one marginal with a total mass of 1 and other possibly less than 1. For the cost matrix $\mathbf{C}$, we use the squared Euclidean distance between pixel locations and normalize so $\|\mathbf{C}\|_{\max} = 1$. The total transported mass is 0.8 times the minimum total mass between the marginals.

Color Transfer

Each RGB image can be considered a point cloud of pixels, and applying the color palette of one image into another is equivalent to applying the OT transport map to convert one color distribution to another. We choose the source image (Figure **??**a, https://flic.kr/p/Lm6gFA) and the target image (Figure **??**b, https://flic.kr/p/c89LBU) images, taken from Flickr. These images are chosen because they have quite distinct color palettes and different sizes. We follow the setup by (**?**) and implement the experiments as follows. Each image of size $h \times w \times 3$ is considered a collection of hw pixels in 3 dimensions. Each image is quantized using k-means with $n = 100$ centroids. We apply k-means clustering to find n centroids and assign to each pixel the centroid it is closest to. The image now becomes a color histogram with n bins representing the centroids. For a pair of source and target images, similar to the previous subsection, we divide each histogram by the maximum total mass, yielding $\max\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\} = 1$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_n$ and $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be the collections of centroids for the source and target images. The cost matrix is defined as $C_{i,j} = \|\mathbf{a}_i - \mathbf{b}_j\|_2^2$. With a partial transport map $\mathbf{X} \in \mathbb{R}_+^{n \times n}$, every centroid $\mathbf{a}_i$ in the source image is transformed to $\hat{\mathbf{a}}_i = (\sum_{j=1}^n X_{i,j}\mathbf{b}_j)/(\sum_{j=1}^n X_{i,j})$. All pixels in the source image previously assigned to $\mathbf{a}_i$ are now assigned to $\hat{\mathbf{a}}_i$. The total transported mass is set to $s = \alpha \min\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\}$, where $\alpha \in [0, 1]$. To approximate the POT solution, we set $\varepsilon = 10^{-2}$ and run both Sinkhorn (**?**) and APDAGD for 1,000 iterations. We set $\alpha = 0.1$, corresponding to transporting exactly 10% of the allowed mass. We plot the optimality gap $\langle \mathbf{C}, \mathbf{X} \rangle - f^*$ for each transport map after every iteration in Figure **??** (c). As explained earlier, Sinkhorn with the accompanying rounding algorithm by (**?**) does not produce a primal feasible solution, and its primal gap does not satisfy an error of $\varepsilon = 10^{-2}$ (indicated by the red line).

Point Cloud Registration

Previously, (**?**) proposed a procedure to find such a transformation using partial optimal transport. Let the two marginal distributions be $\mathbf{r} = \frac{1}{m}\mathbf{1}_m$ and $\mathbf{c} = \frac{1}{n}\mathbf{1}_n$ and the cost matrix be $C_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|_2^2$. Given an optimal transport matrix $\mathbf{T} \in \mathbb{R}^{m \times n}$ between these marginals, the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ are obtained by minimizing the energy:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^m \sum_{j=1}^n T_{i,j} \|\mathbf{x}_i - (\mathbf{R}\mathbf{y}_j + \mathbf{t})\|_2^2.$$

This problem admits the closed-form solution

$$\mathbf{R} = \mathbf{V}\mathbf{S}\mathbf{U}^\top, \quad \mathbf{t} = \mathbf{u}_x - \mathbf{R}\mathbf{u}_y,$$

where $\mathbf{u}_x = \frac{1}{m}\sum_{i=1}^m \mathbf{x}_i$, $\mathbf{u}_y = \frac{1}{n}\sum_{j=1}^n \mathbf{y}_j$. The matrices $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$ are obtained as follows. Let $\hat{\mathbf{X}} \in \mathbb{R}^{m \times 3}$ whose ith row is $(\mathbf{x}_i - \mathbf{u}_x)^\top$. Similar for $\hat{\mathbf{Y}} \in \mathbb{R}^{n \times 3}$. Then, obtain the singular value decomposition of the matrix $\hat{\mathbf{X}}\mathbf{T}^\top\hat{\mathbf{X}}^\top$ as $\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$. Finally, $\mathbf{S} = \text{diag}(1, 1, \det(\mathbf{V}\mathbf{U}^\top))$.

To find the the matrix $\mathbf{T}$, we follow the iterative procedure in (**?**, Algorithm 1) in which the point cloud Q is updated gradually until convergence. In our implementation, we set the initial value for γ (strength of entropic regularization) to 0.004 and the annealing rate of γ to $\lambda = 0.99$. With each γ, we find the OT matrix using Sinkhorn, and the POT matrix using two methods: Sinkhorn and APDAGD. The iterative process ends when the change of $\mathbf{R}$ in Frobenius norm falls below 10^{-6}.

Further Numerical Experiments

Run Time for Varying ε

With the similar setting to Figure **??** of using images in the CIFAR-10 dataset, we provide additional experiments showcasing that APDAGD is better than Sinkhorn in wall-clock time in terms of both aggregate run time cost in Figure **??**. Moreover, for aggregate wall-clock time, we note that previous works showed that gradient methods and especially APDAGD are practically faster than Sinkhorn (**?**, Figure 1). This claim is further supported by Figure **??** since it reproduce the results from (**?**, Figure 1) which compare the runtime of Sinkhorn and APDAGD in the context of OT.
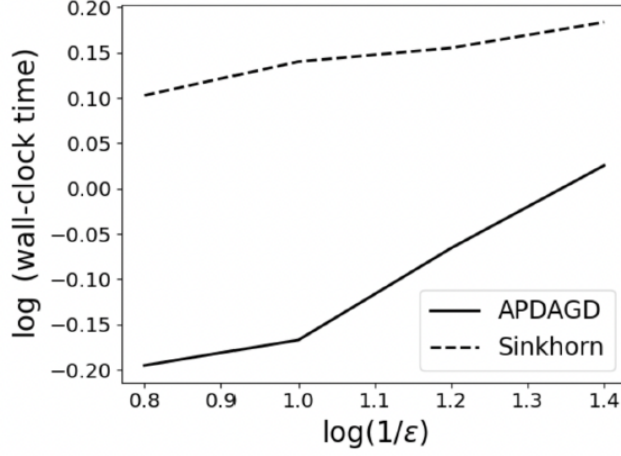
Figure 6: Comparison of wall-clock running time between APDAGD and Sinkhorn for varying ε
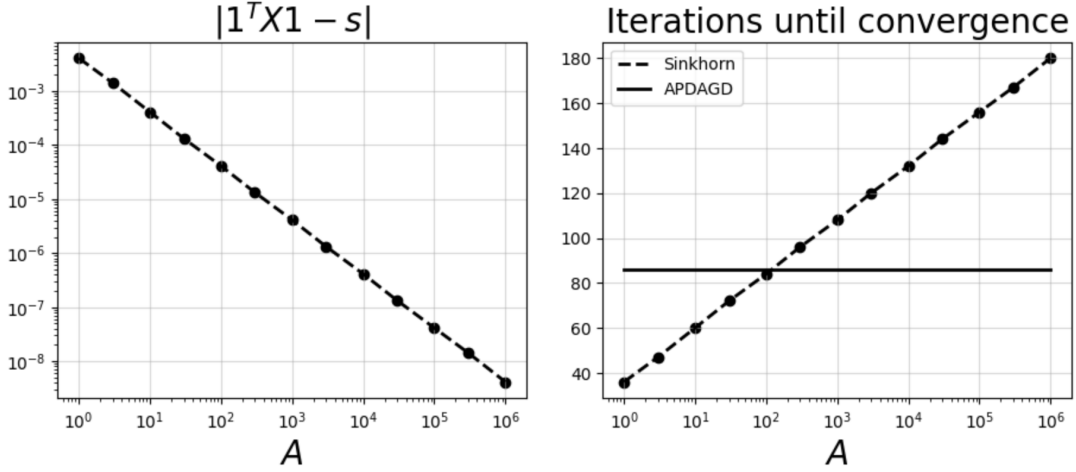

Figure 7: Left: Sinkhorn constraint violation with respect to A. Right: Number of iterations for both APDAGD and Sinkhorn until a pre-defined suboptimality is achieved.

Revised Sinkhorn

We again utilize the same setting Figure **??** and Run Time for Varying ε with CIFAR-10 dataset. On the left, we empirically verify our theoretical bounds in Theorem **??**, by showing that increasing A potentially lead to a better feasibility of Sinkhorn. For the figure on the right, we show that the increase of A to a sufficient size can lead to more number of iterations to convergence of Sinkhorn. This is also suggested by the worsened theoretical complexity of revised Sinkhorn (Theorem **??**).

Domain Adaptation

In our setting, the source and target datasets contain different numbers of data points ($N_s = 300$ and $N_t = 400$, respectively). For OT, we use Sinkhorn to approximate the OT matrix of size 300×400, then transform the source examples according to Courty et al. (2017). For POT, we use K-means clustering to transform the target domains to a histogram of 300 bins. The two unbalanced marginals are then normalized so that $\|r\|_1 = 0.75$, $\|c\|_1 = 1.0$, and we set $s = 0.999 \times \min\{\|r\|_1, \|c\|_1\}$ then use APDAGD to find the POT matrix between these two unbalanced marginals. The source examples are transformed similarly. Finally, for both methods (OT and POT), we train a support vector machine with the radial basis kernel (parameter $\sigma^2 = 1$) using the transformed source domain. In the Figure **??** we report the accuracy on the target domain using 20,000 test examples. All hyperparameters are set to be equal.

Here we implement and compare Domain Adaptation with OT to Domain Adaptation with POT, which is calculated with two different algorithms Sinkhorn (infeasible rounding) and APDAGD (with our novel ROUND-POT). In (**?**), the authors presented an OT-based method to transform a source domain to a reference target domain such that the their densities are approximately equal. We present a binary classification setting involving the "moons" dataset. The source dataset is displayed as colored scatter
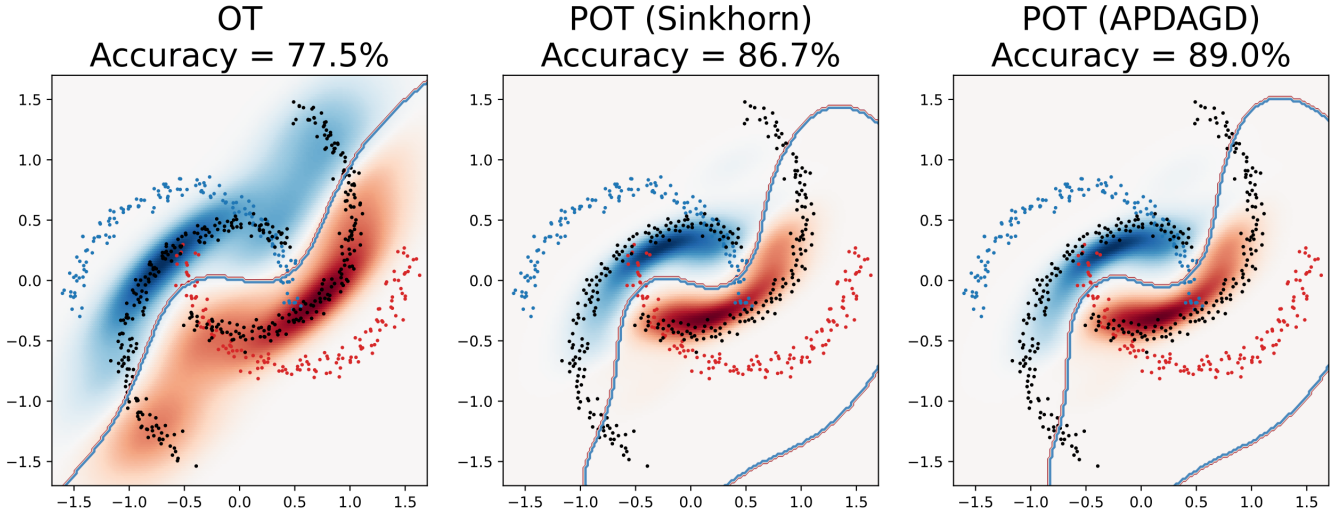
Figure 8: Domain adaptation with OT and with POT. POT offers a flexibility in choosing the mass transported and helps avoid the need to normalize both marginals to have a sum of 1, resulting in better complexity than OT. APDAGD demonstrates better accuracy in the novel domain than Sinkhorn, which suffers from infeasibility.

points. The target dataset comes from the same distribution, but the points go to a rotation of 60 degrees (black scatter points), representing a domain covariate shift. Additional experimental details are further described in Further Experiments Setup Details section in Appendix. To summarize the results, POT offers much more practicality as it does not require normalizing both marginals to have a sum of 1 in contrast to OT. More importantly, POT avoids bad matchings to outliers due to the flexibility in choosing the amount of mass transported, resulting in a higher accuracy than OT. Furthermore, linking back to Remark **??**, the infeasibility Sinkhorn degrades the practical performance of POT for Domain Adaption and consequently leads to a worse accuracy compared APDAGD. This is because the optimal mapping between the source and target domains requires the exact fraction of mass to be transported which APDAGD fulfills practically in contrast to Sinkhorn.
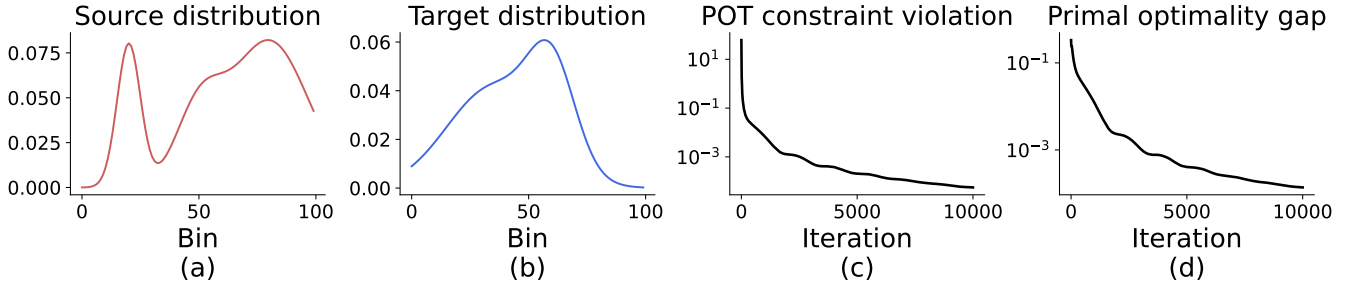
Synthetic Data

Figure 9: Convergence of APDAGD on a synthetic setting.

In this section we design a POT setting and evaluate the convergence of APDAGD. The two marginals are unnormalized Gaussian mixtures. We then discretize them into histograms of $n = 100$ bins each and perform normalization so that $\|\mathbf{r}\|_1 = 5$, $\|\mathbf{c}\|_1 = 3$. The total transported mass is set to $s = 0.9 \min\{\|\mathbf{r}\|_1, \|\mathbf{c}\|_1\} = 2.7$, and the cost matrix represents squared Euclidean distances between every pair of bins in the source and target histograms. In other words, $C_{i,j} = (i - j)^2$ for all $i, j = 1, \ldots, n$. Finally, we divide $\mathbf{C}$ by its maximum entry to have $\|\mathbf{C}\|_{\max} = 1$.

We use `cvxpy` with the GUROBI backend to solve the linear problem (**??**), and denote by f^* the optimal objective value. This will be used to calculate the primal optimality gap.

The approximate POT solution is solved using APDAGD, where we set the additive suboptimality error ε to 10^{-3}. We run APDAGD for 10,000 iterations and keep track of two errors. The first error is constraint violation, equal to $\|\mathbf{X}\mathbf{1} + \mathbf{p} - \mathbf{r}\|_1 + \|\mathbf{X}^\top \mathbf{1} + \mathbf{q} - \mathbf{c}\|_1 + |\mathbf{1}^\top \mathbf{X}\mathbf{1} - s|$ where $\mathbf{X}$, $\mathbf{p}$ and $\mathbf{q}$ are part of the solution $\mathbf{x}_k$ found after each iteration. The second error is the optimality gap for the primal objective. To calculate this, we need to round the solution so that it becomes primal feasible.
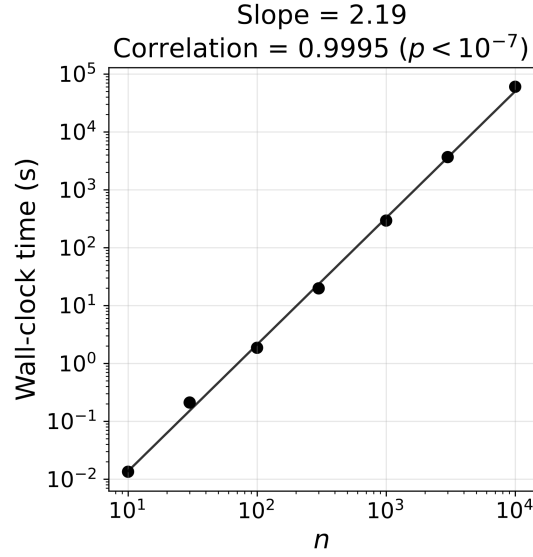
Figure 10: Wall-clock time of solving POT with APDAGD against n

We use our rounding algorithm described in Rounding Algorithm Section to round $\mathbf{X}$, $\mathbf{p}$, and $\mathbf{q}$ after each iteration, giving us $\bar{\mathbf{X}}$, $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$. The primal gap then is $\langle \mathbf{C}, \bar{\mathbf{X}} \rangle - f^*$.

Scalability of APDAGD

Given the complexity claims in the paper, it is important to show how running time scales with n, the number of supports in each marginal. In the below Figure **??**, we show the wall-clock time of running APDAGD to solve the POT problem between two Gaussian mixtures (shown in the end of the appendix). We fix the error tolerance to $\varepsilon = 10^{-1}$ and vary n in the set $\{10, 30, 100, 300, 1000, 3000, 10000\}$. The figure plots wall-clock time in seconds against n. The expected slope of the best-fit line between $\log(\text{time})$ and $\log(n)$ should be at most 2.5, as the complexity is $\widetilde{O}(n^{2.5}/\varepsilon)$. We find that the slope is 2.19, which is consistent with the upper bound. Furthermore, the correlation is statistically significant.