Accelerated Primal-Dual Coordinate Descent for Computational Optimal Transport

Wenshuo Guo
UC Berkeley
wguo@eecs.berkeley.edu

Nhat Ho
UC Berkeley
minhnhat@berkeley.edu

Michael I. Jordan
UC Berkeley
jordan@cs.berkeley.edu

Abstract

We propose and analyze a novel accelerated primal-dual coordinate descent framework for computing the optimal transport (OT) distance between two discrete probability distributions. First, we introduce the *accelerated primal-dual randomized coordinate descent* (APDRCD) algorithm for computing OT. Then we provide a complexity upper bound $\widetilde{\mathcal{O}}(\frac{n^{5/2}}{\varepsilon})$ for the APDRCD method for approximating OT distance, where n stands for the number of atoms of these probability measures and $\varepsilon > 0$ is the desired accuracy. This upper bound matches the best known complexities of adaptive primal-dual accelerated gradient descent (APDAGD) and adaptive primal-dual accelerate mirror descent (APDAMD) algorithms while it is better than those of Sinkhorn and Greenkhorn algorithms, which are of the order $\widetilde{\mathcal{O}}(\frac{n^2}{\varepsilon^2})$, in terms of the desired accuracy $\varepsilon > 0$. Furthermore, we propose a greedy version of APDRCD algorithm that we refer to as the *accelerated primal-dual greedy coordinate descent* (APDGCD) algorithm and demonstrate that it has a better practical performance than the APDRCD algorithm. Extensive experimental studies demonstrate the favorable performance of the APDRCD and APDGCD algorithms over state-of-the-art primal-dual algorithms for OT in the literature.

1 Introduction

The computation of optimal transport distance between probability distributions has become a central topic in statistical machine learning, with applications in areas as diverse as Bayesian nonparametrics [17, 18], scalable Bayesian inference [22, 23], topic modeling [15], isotonic regression [20], and deep learning [3, 2, 24]. By viewing the optimal transport distance as a linear programming problem, interior-point methods have been employed as a computational solver, with a best known practical complexity of $\widetilde{\mathcal{O}}(n^3)$ [19]. Recently, [13] proposed to use the Laplace linear system solver to theoretically improve the complexity of interior-point methods to $\widetilde{\mathcal{O}}(n^{5/2})$. It remains a practical challenge, however, to develop efficient interior-point implementations in the high-dimensional problems characteristic of machine learning.

To circumvent the scalability issue of interior-point methods, [4] defined the *entropic (regularized) optimal transportation distance* by regularizing the OT distance by the entropy of the corresponding transportation plan of the probability measures. The most popular algorithm for computing entropic regularized OT distance is the Sinkhorn algorithm [21, 12, 10], which has a complexity upper bound of $\widetilde{\mathcal{O}}(\frac{n^2}{\varepsilon^2})$. where $\varepsilon > 0$ is the desired accuracy [7].

Recently, several algorithms have been proposed to improve the performance of Sinkhorn algorithm. Most notably, [1] introduced the Greenkhorn algorithm, which is a greedy coordinate descent algorithm, for solving the dual form of the entropic regularized OT problem. The Greenkhorn algorithm has a theoretical complexity of $\widetilde{\mathcal{O}}(\frac{n^2}{\varepsilon^2})$ [14], which is comparable to that of the Sinkhorn

Preprint. Under review.

algorithm, while enjoying better practical performance than that of Sinkhorn algorithm on several datasets [1, 14]. However, for large-scale applications of the OT problems, such as computational Wasserstein barycenters [5, 6], particularly in randomized and asynchronous scenarios, existing literature has shown that neither the Sinkhorn nor the Greenkhorn algorithms are sufficiently scalable and flexible.

To improve the flexibility of these algorithms, a recent line of work on accelerated primal-dual algorithms has been proposed for computing OT. This includes the adaptive primal-dual accelerated gradient descent (APDAGD) algorithm [7] and the adaptive primal-dual accelerated mirror descent (APDAMD) algorithms [14], which possess theoretical complexity bounds of $\widetilde{\mathcal{O}}(\frac{n^{2.5}}{\varepsilon})$ and $\widetilde{\mathcal{O}}(\frac{n^2\sqrt{\gamma}}{\varepsilon})$ repectively, where $\gamma \leq \sqrt{n}$ is the inverse of the strong complexity constant of Bregman divergence with respect to the l_∞-norm. These complexity bounds are better than those of Sinkhorn and Greenkhorn algorithms in terms of ε. Nevertheless, when the dimension n is large, the practical performance of these accelerated primal-dual algorithms remains unsatisfying.

Our contributions. The contributions of the paper are three-fold.

1. We introduce a novel *accelerated primal-dual coordinate descent* framework for solving the OT problem. This framework is inspired by the favorable performance of accelerated dual coordinate descent algorithms [16] and recent active research on developing accelerated primal-dual algorithms for the OT problem [7, 14]. Furthermore, this new primal-dual framework possesses the requisite flexibility and scalability compared to the Sinkhorn algorithm, which is crucial for computational OT problems in large-scale application settings [5, 9]. To the best of our knowledge, this is the first accelerated primal-dual coordinate descent framework for OT problems.

2. In addition to the accelerated primal-dual coordinate descent framework, we propose an *accelerated primal-dual randomized coordinate descent* (APDRCD) algorithm. We establish a complexity upper bound of $\widetilde{\mathcal{O}}(\frac{n^{5/2}}{\varepsilon})$ for the APDRCD algorithm, which is comparable to the complexity of state-of-art primal-dual algorithms for OT problems, such as the APDAGD and APDAMD algorithms [7, 14]. Furthermore, that complexity bound is better than the complexities of Sinkhorn and Greenkhorn algorithms in terms of ε.

3. To further improve the practical performance of the APDRCD algorithm, we study a greedy version of that algorithm, which we refer to as the *accelerated primal-dual greedy coordinate descent* (APDGCD) algorithm. Extensive experimental comparisons show that both APDRCD and APDGCD algorithms outperform the APDAGD and APDAMD algorithms on approximating OT problems on both synthetic and real image datasets. As a consequence, APDRCD and APDGCD algorithms achieve the best performance among all the recent accelerated primal-dual algorithms on solving entropic regularized OT problems.

Organization. The remainder of the paper is organized as follows. In Section 2, we provide the formulation of the entropic OT problem as well as its dual form. In Section 3, we introduce an accelerated primal-dual coordinate descent framework for solving the regularized OT problem and provide a complexity upper bound for the APDRCD algorithm. In Section 4, we present comparative experiments between the APDRCD algorithm and the APDAGD and APDAMD algorithms. We conclude the paper with a few future directions in Section 5. Finally, the proofs of all results in the paper are in the Appendix A while the details of the APDGCD algorithm as well as additional experiments are presented in Appendices B and C.

Notation. We denote the probability simplex $\Delta^n := \{u = (u_1, \ldots, u_n) \in \mathbb{R}^n : \sum_{i=1}^n u_i = 1,\ u \geq 0\}$ for $n \geq 2$. Furthermore, $[n]$ stands for the set $\{1, 2, \ldots, n\}$ while $\mathbb{R}^n_+$ stands for the set of all vectors in $\mathbb{R}^n$ with nonnegative components for any $n \geq 1$. For a vector $x \in \mathbb{R}^n$ and $1 \leq p \leq \infty$, we denote $\|x\|_p$ as its ℓ_p-norm and $\mathrm{diag}(x)$ as the diagonal matrix with x on the diagonal. For a matrix $A \in \mathbb{R}^{n\times n}$, the notation $\mathrm{vec}(A)$ stands for the vector in $\mathbb{R}^{n^2}$ obtained from concatenating the rows and columns of A. $\mathbf{1}$ stands for a vector with all of its components equal to 1. $\partial_x f$ refers to a partial gradient of f with respect to x. Lastly, given the dimension n and accuracy ε, the notation $a = \mathcal{O}\left(b(n, \varepsilon)\right)$ stands for the upper bound $a \leq C \cdot b(n, \varepsilon)$ where C is independent of n and ε. Similarly, the notation $a = \widetilde{\mathcal{O}}(b(n, \varepsilon))$ indicates the previous inequality may depend on the logarithmic function of n and ε, and where $C > 0$.

2 Problem Setup

In this section, we provide the necessary background for the entropic regularized optimal transport (OT) problem between two discrete probability measures with at most n components. In particular, the objective function of the entropic regularized OT problem is presented in Section 2.1 while its dual form as well as the key properties of that dual form are given in Section 2.2.

2.1 Entropic regularized OT

As shown in [11], the problem of approximating the OT distance between two discrete probability distributions with at most n components is equivalent to the following linear programming problem

$$\min_{X\in\mathbb{R}^{n\times n}} \langle C, X\rangle \qquad \text{s.t. } X\mathbf{1} = r,\ X^\top\mathbf{1} = l,\ X \geq 0, \tag{1}$$

where X is a *transportation plan*, $C = (C_{ij}) \in \mathbb{R}_+^{n\times n}$ is a cost matrix with non-negative elements, and r and l refer to two known probability distributions in the probability simplex Δ^n. The best known practical complexity bound for (1) is $\widetilde{\mathcal{O}}(n^3)$ [19] while the best theoretical complexity bound is $\widetilde{\mathcal{O}}(n^{2.5})$ [13], achieved via interior-point methods. However, these methods are not efficient with the high dimensional settings of OT problems. This motivates the usage of the entropic regularization for the OT problem (1), which is referred to as the *entropic regularized OT* problem [4]. This problem is given by

$$\min_{X\in\mathbb{R}_+^{n\times n}} \langle C, X\rangle - \eta H(X) \qquad \text{s.t. } X\mathbf{1} = r,\ X^\top\mathbf{1} = l, \tag{2}$$

where $\eta > 0$ is the *regularization parameter* and $H(X)$ is the entropic regularization given by $H(X) := -\sum_{i,j=1}^n X_{ij}\log(X_{ij})$. The main focus of the paper is to determine an *ε-approximate transportation plan* $\hat{X} \in \mathbb{R}_+^{n\times n}$ such that $\hat{X}\mathbf{1} = r$ and $\hat{X}^\top\mathbf{1} = l$ and the following bound holds

$$\langle C, \hat{X}\rangle \ \leq\ \langle C, X^*\rangle + \varepsilon, \tag{3}$$

where X^* is an optimal solution; i.e., an optimal transportation plan for the OT problem (1). To ease the ensuing presentation, we denote $\langle C, \hat{X}\rangle$ an *ε-approximation* for the OT distance. Furthermore, we define matrix A such that $A\text{vec}(X) := \begin{pmatrix} X\mathbf{1} \\ X^\top\mathbf{1}\end{pmatrix}$ for any $X \in \mathbb{R}^{n\times n}$.

2.2 Dual entropic regularized OT

The Lagrangian function for problem (2) is given by

$$\mathcal{L}(X,\alpha,\beta) \quad := \quad \langle\alpha, r\rangle + \langle\beta, l\rangle + \langle C, X\rangle - \eta H(X) - \langle\alpha, X\mathbf{1}\rangle - \langle\beta, X^\top\mathbf{1}\rangle.$$

Given the Lagrangian function, the dual form of the entropic regularized OT problem can be obtained by solving the optimization problem $\min_{X\in\mathbb{R}^{n\times n}}\mathcal{L}(X,\alpha,\beta)$. Since the Lagrangian function $\mathcal{L}(\cdot,\alpha,\beta)$ is strictly convex, that optimizition problem can be solved by setting $\partial_X\mathcal{L}(X,\alpha,\beta) = 0$, which is equivalent to the following equation:

$$C_{ij} + \eta\left(1 + \log(X_{ij})\right) - \alpha_i - \beta_j = 0, \quad \forall i,j \in [n].$$

The above equations lead to the following form of the transportation plan X where $X_{ij} = e^{\frac{-C_{ij}+\alpha_i+\beta_j}{\eta}-1}$ for all $i,j \in [n]$. With this solution, we have $\min_{X\in\mathbb{R}^{n\times n}}\mathcal{L}(X,\alpha,\beta) = -\eta\sum_{i,j=1}^n e^{-\frac{C_{ij}-\alpha_i-\beta_j}{\eta}-1} + \langle\alpha, r\rangle + \langle\beta, l\rangle$. The *dual entropic regularized OT* problem is, therefore, equivalent to the following optimization problem:

$$\min_{\alpha,\beta\in\mathbb{R}^n} \varphi(\alpha,\beta) := \eta\sum_{i,j=1}^n e^{-\frac{C_{ij}-\alpha_i-\beta_j}{\eta}-1} - \langle\alpha, r\rangle - \langle\beta, l\rangle. \tag{4}$$

Building on Lemma 4.1 in [14], the dual objective function $\varphi(\alpha,\beta)$ is smooth with respect to $\|\cdot\|_2$ norm, which is given by the following lemma:

Lemma 2.1. *The dual objective function φ is smooth with respect to $\|.\|_2$ norm:*

$$\varphi(\lambda_1) - \varphi(\lambda_2) - \langle\nabla\varphi(\lambda_2), \lambda_1 - \lambda_2\rangle \leq \frac{2}{\eta}||\lambda_1 - \lambda_2||_2^2.$$

The proof of Lemma 2.1 is provided in Appendix A.1.

3 Accelerated Primal-Dual Coordinate Descent Framework

In this section, we present and analyze an accelerated primal-dual coordinate descent framework to obtain an ε-approximate transportation plan for the OT problem (1). First, in Section 3.1, we introduce the accelerated primal-dual randomized coordinate descent (APDRCD) method for the entropic regularized OT problem and present the detailed pseudo-code in Algorithm 1. Then, following the approximation scheme of [1], we provide the complete pseudo-code to approximate the OT distance based on the APDRCD algorithm in Algorithm 2. Furthermore, we provide theoretical analysis to establish the complexity bound of $\mathcal{O}(\frac{n^{\frac{5}{2}}\sqrt{||C||_\infty \log(n)}}{\varepsilon})$ for the APDRCD algorithm to achieve an ε-approximate transportation plan for the OT problem in Section 3.2. This complexity upper bound of the APDRCD algorithm matches the best known complexity bounds of the APDAGD [7] and APDAMD algorithms [14]. Finally, to further improve the practical performance of APDRCD algorithm, we propose a greedy version of it, which is referred to as the accelerated primal-dual greedy coordinate descent (APDGCD) algorithm. Due to space constraints, the details of the APDGCD algorithm are deferred to Appendix B.

Algorithm 1: APDRCD $(C, \eta, A, b, \varepsilon')$

1 **Input:** $\{\theta_k | \theta_0 = 1, \frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2}, \}, \lambda^0 = \lambda_0 = 0, z^0 = z_0 = 0, k = 0$

2 **while** $\mathbb{E}\big[||Ax^k - b||_2\big] > \varepsilon'$ **do**

3 Set
$$y^k = (1-\theta_k)\lambda^k + \theta_k z^k \tag{5}$$

4 Compute
$$x^k = \frac{1}{C_k}\left(\sum_{j=0}^{k}\frac{x(y^j)}{\theta_j}\right) \text{ where } C_k := \sum_{j=0}^{k}\frac{1}{\theta_j} \tag{6}$$

5 **Randomly sample one coordinate** $i_k \in \{1, 2, ..., 2n\}$:

6 Update
$$\lambda_{i_k}^{k+1} = y_{i_k}^k - \frac{1}{L}\nabla_{i_k}\varphi(y^k) \tag{7}$$

7 Update
$$z_{i_k}^{k+1} = z_{i_k}^k - \frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k) \tag{8}$$

8 Update
$$k = k + 1$$

9 **end**

10 **Output:** X^k where $x^k = vec(X^k)$

3.1 Accelerated primal-dual randomized coordinate descent (APDRCD) algorithm

We denote by L the Lipschitz constant for the dual objective function φ, which means that $L := \frac{4}{\eta}$, and $x(\lambda) := \underset{x\in\mathbb{R}^{n\times n}}{\arg\max}\Big\{ - \langle C, x\rangle - \big\langle A^\top\lambda, x\big\rangle\Big\}$. The APDRCD algorithm is initialized with the auxiliary sequence $\{\theta_k\}$ and two auxiliary dual variable sequences $\{\lambda_i\}$ and $\{\mathbf{z}_i\}$, where the first auxiliary sequence $\{\theta_k\}$ is used for the key averaging step and the two dual variable sequences are used to perform the accelerated randomized coordinate descent on the dual objective function φ as a subroutine. The whole algorithmic framework of APDRCD is composed of two main parts. First, noticing the convexity property of the dual objective function, we perform an randomized accelerated coordinate descent step on the dual objective function as a subroutine in step 7 and 8. In the second part, we take a weighted average over the past iterations to get a good approximate solution for the primal problem from the approximate solutions to the dual problem (4). Notice that the auxiliary sequence $\{\theta_k\}$ is decreasing and the primal solutions corresponding to the more recent dual solutions have more weights in this average.

3.2 Complexity analysis of APDRCD algorithm

Given the updates from APDRCD algorithm in Algorithm 1, we have the following result regarding the difference of the values of φ at λ^{k+1} and y^k:

Lemma 3.1. *Given the updates λ^{k+1} and y^k from the APDRCD algorithm, we have the following inequality*

$$\varphi(\lambda^{k+1}) - \varphi(y^k) \leq -\frac{1}{2L}|\nabla_{i_k}\varphi(y^k)|^2,$$

where i_k is chosen in the APDRCD algorithm.

The proof of Lemma 3.1 is provided in Appendix A.2. The result of Lemma 3.1 is vital to establish an upper bound for $\mathbb{E}_{i_k}\varphi(\lambda^{k+1})$, which is given by the following lemma:

Lemma 3.2. *For each iteration ($k > 0$) of the APDRCD algorithm, we have*

$$\mathbb{E}_{i_k}\left[\varphi(\lambda^{k+1})\right] \leq (1-\theta_k)\varphi(\lambda^k) + \theta_k\left[\varphi(y^k) + (\lambda - y^k)^T\nabla\varphi(y^k)\right] + 2L^2n^2\theta_k^2\left(||\lambda - z^k||^2 - \mathbb{E}_{i_k}\left[||\lambda - z^{k+1}||^2\right]\right),$$

where the outer expectation in the above display is taken with respect to the random coordinate i_k in Algorithm 1.

Algorithm 2: Approximating OT by APDRCD

Input: $\eta = \dfrac{\varepsilon}{4\log(n)}$ and $\varepsilon' = \dfrac{\varepsilon}{8\|C\|_\infty}$.

Step 1: Let $\tilde{r} \in \Delta_n$ and $\tilde{l} \in \Delta_n$ be defined as

$$\left(\tilde{r}, \tilde{l}\right) = \left(1 - \frac{\varepsilon'}{8}\right)(r, l) + \frac{\varepsilon'}{8n}(\mathbf{1}, \mathbf{1}).$$

Step 2: Let $A \in \mathbb{R}^{2n \times n^2}$ and $b \in \mathbb{R}^{2n}$ be defined by

$$Avec(X) = \begin{pmatrix} X\mathbf{1} \\ X^T\mathbf{1} \end{pmatrix} \text{ and } b = \begin{pmatrix} \tilde{r} \\ \tilde{l} \end{pmatrix}$$

Step 3: Compute $\tilde{X} = \text{APDRCD}\,(C, \eta, A, b, \varepsilon'/2)$ with φ defined in (4).

Step 4: Round $\tilde{X}$ to $\hat{X}$ by Algorithm 2 [1] such that $\hat{X}\mathbf{1} = r$, $\hat{X}^\top\mathbf{1} = l$.

Output: $\hat{X}$.

The proof of Lemma 3.2 is in Appendix A.3. Now, equipped with the result of Lemma 3.2, we are ready to provide the convergence guarantee and complexity bound of the APDRCD algorithm for the approximating OT problem. First, we start with the following result regarding an upper bound for the number of iterations k to reach the stopping rule $\mathbb{E}\left[||Avec(X^k) - b||_2\right] \leq \varepsilon'$ for $\varepsilon' = \dfrac{\varepsilon}{8\|C\|_\infty}$. Here, the outer expectation is taken with respect to the random coordinates i_j in Algorithm 1 for $1 \leq j \leq k$.

Theorem 3.3. *The APDRCD algorithm for approximating optimal transport (Algorithm 2) returns an output X^k that satisfies the stopping criterion $\mathbb{E}\left[||Avec(X^k) - b||_2\right] \leq \varepsilon'$ in a number of iterations k bounded as follows:*

$$k \leq 12n^{\frac{3}{2}}\sqrt{\frac{R + 1/2}{\varepsilon}} + 1,$$

where $R := \frac{||C||_\infty}{\eta} + \log(n) - 2\log(\min_{1 \leq i,j \leq n}\{r_i, l_i\})$. Here, ε' and η are chosen in Algorithm 2.

The proof of Theorem 3.3 is provided in Appendix A.4. Given an upper bound for the number of iterations k for the stopping rule $\mathbb{E}\left[||Avec(X^k) - b||_2\right] \leq \varepsilon'$ for $\varepsilon' = \dfrac{\varepsilon}{8\|C\|_\infty}$ in Theorem 3.3. We proceed to present a complexity bound for the APDRCD algorithm.

Theorem 3.4. *The APDRCD algorithm for approximating optimal transport (Algorithm 2) returns* $\hat{X} \in \mathbb{R}^{n \times n}$ *satisfying* $\hat{X}\mathbf{1} = r$, $\hat{X}^T\mathbf{1} = l$ *and (3) in a total of*

$$\mathcal{O}\left(\frac{n^{\frac{5}{2}}\sqrt{||C||_\infty \log(n)}}{\varepsilon}\right)$$

arithmetic operations.

The proof of Theorem 3.4 is provided in Appendix A.5. The result of Theorem 3.4 indicates that the complexity upper bound of APDRCD algorithm matches the best known complexity $\widetilde{\mathcal{O}}(\frac{n^{5/2}}{\varepsilon})$ of the APDAGD [7] and APDAMD [14] algorithms. Furthermore, that complexity of the APDRCD algorithm is better than that of the Sinkhorn and Greenkhorn algorithms, which is $\widetilde{\mathcal{O}}(\frac{n^2}{\varepsilon^2})$, in terms of the desired accuracy $\varepsilon > 0$. Later, in the experiment results (cf. Section 4), we demonstrate that the APDRCD algorithm indeed has better practical performance than APDAGD and APDAMD algorithms on both synthetic and real datasets.

4 Experiments

In this section, we carry out the comparative experiments between the APDRCD, APDGCD algorithms and the existing state-of-art primal-dual algorithms for the OT problem including the APDAGD and APDAMD algorithms, on both synthetic images and real images from the MNIST Digits dataset[1]. Due to space constraints, the comparative experiments between the APDGCD algorithm and APDAGD/APDAMD algorithms are deferred to Appendix C. Finally, in that appendix, we also include the comparisons between the APDRCD and APDGCD algorithms with the Sinkhorn algorithm for completeness. Note that for the above comparisons, we also utilize the default linear programming solver in MATLAB to obtain the optimal value of the original optimal transport problem without entropic regularization.

4.1 APDRCD algorithm with synthetic images

We conduct extensive comparisons of the performance of the APDRCD algorithm with the APDAGD and APDAMD algorithms on synthetic images. The generation of synthetic images follows the procedure of [1, 14]. In particular, the images are of size 20×20 and generated based on randomly placing a foreground square in the otherwise black background. Then, for the intensities of the background pixels and foreground pixels, we choose uniform distributions on [0,1] and [0, 50] respectively. For comprehensive results, we vary the proportion of the size of the foreground square in 0.1, 0.5, 0.9 of the full size of the image and implement all the algorithms on different kinds of synthetic images.

Evaluation metric: Regarding the evaluation metrics, we utilize the popular metrics from [1]. The first metric is the distance between the output of the algorithm and the transportation polytope $d(X) := ||r(X) - r||_1 + ||l(X) - 1||_1$ where $r(X)$ and $l(X)$ are the row and column marginal vectors of the output matrix X while r and l stand for the true row and column marginal vectors. The second metric is the competitive ratio, defined by $log(d(X1)/d(X2))$ where $d(X1)$ and $d(X2)$ refer to the distance between the outputs of two algorithms and the transportation polytope.

Experimental settings and results: We perform two pairwise comparative experiments for the APDRCD algorithm versus the APDAGD and APDAMD algorithms by running these algorithms with ten randomly selected pairs of synthetic images. We also evaluate all the algorithms with varying regularization parameter $\eta \in \{1, 5, 9\}$ and the optimal value of the original optimal transport problem without the entropic regularization, as suggested by [1, 14].

We present the experimental results in Figure 1 and Figure 2. According to these figures, the APDRCD algorithm has better performance than the APDAGD and APDAMD algorithms in terms of the iteration numbers. More specifically, when the number of iteration number is small, the APDRCD algorithm achieves faster and more stable decrements than other two algorithms with regard to both the distance to polytope and the value of OT during the computing process, which is beneficial for easier tuning in practice. These superior behaviors of APDRCD illustrate the improvement achieved

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

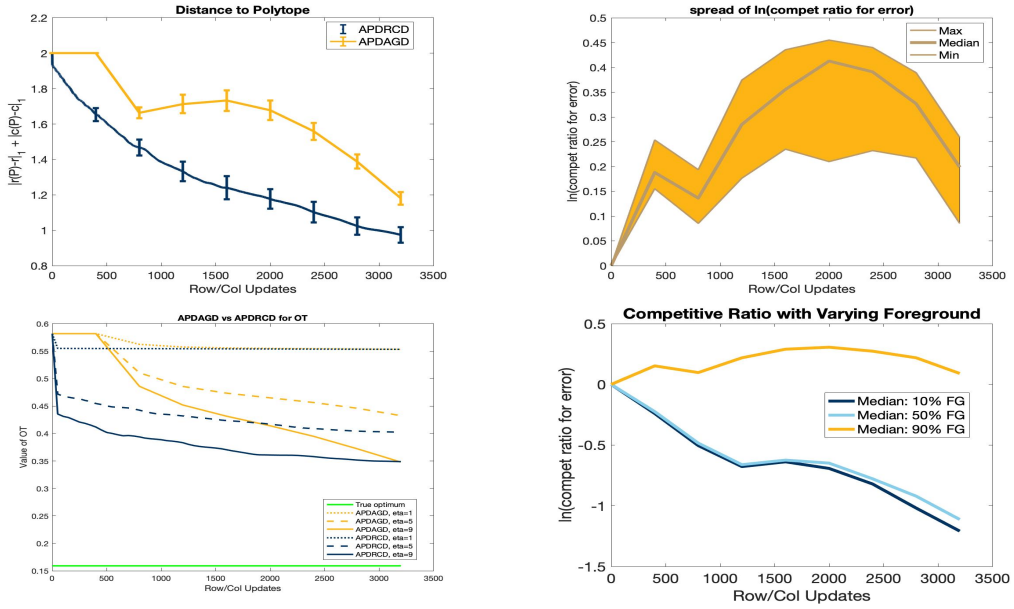

Figure 1: Performance of APDRCD and APDAGD algorithms on the synthetic images. In the top two images, the comparison is based on using the distance $d(P)$ to the transportation polytope, and the maximum, median and minimum of competitive ratios on ten random pairs of images. In the bottom left image, the comparison is based on varying the regularization parameter $\eta \in \{1, 5, 9\}$ and reporting the optimal value of the original optimal transport problem without entropic regularization. Note that the foreground covers 10% of the synthetic images here. In the bottom right image, we compare the algorithms by using the median of competitive ratios with varying coverage ratio of foreground in the range of $\{0.1, 0.5, 0.9\}$.

by using randomized coordinate descent on the dual regularized problem, and support the theoretical assertion that the APDRCD algorithm a complexity bound that matches those of the APDAGD and APDAMD algorithms.

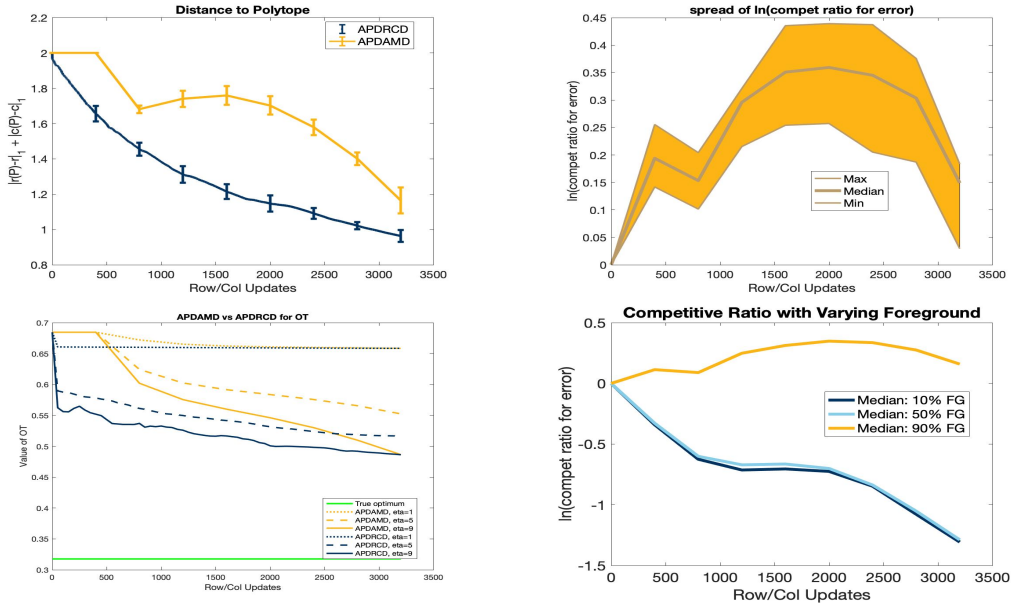

Figure 2: Performance of APDRCD and APDAMD algorithms on the synthetic images. The organization of the images is similar to those in Figure 1.

4.2 APDRCD algorithm with MNIST images

Moving beyond the synthetic images, we present the comparisons between the APDRCD algorithm versus the APDAGD and APDAMD algorithms on MNIST images with the same evaluation metrics as in the synthetic images. The image pre-processing follows the same pre-processing procedure as suggested in [14]; therefore, we will omit the details for the sake of brevity.

We present the experimental results with the MNIST images in Figure 3 with various values for the regularization parameter $\eta \in \{1, 5, 9\}$. We also evaluate all the algorithms with the optimal value of the original optimal transport problem without entropic regularization. As shown in Figure 3, the APDRCD algorithm outperforms both the APDAGD and APDAMD algorithms on the MNIST dataset in terms of the number of iterations. Additionally, the APDRCD algorithm experiences faster and smoother convergence than the other algorithms at small iteration numbers with regard to both the evaluation metrics, which gives it an advantage that it is easier to be tuned in practice. In summary, the consistent superior performance of the APDRCD algorithm over the APDAGD and APDAMD algorithms on both the synthetic and MNIST datasets supports the theoretical assertion on the matched complexity bounds of these three algorithms, and shows the advantage of using randomized coordinate descent for approximating the OT problem.

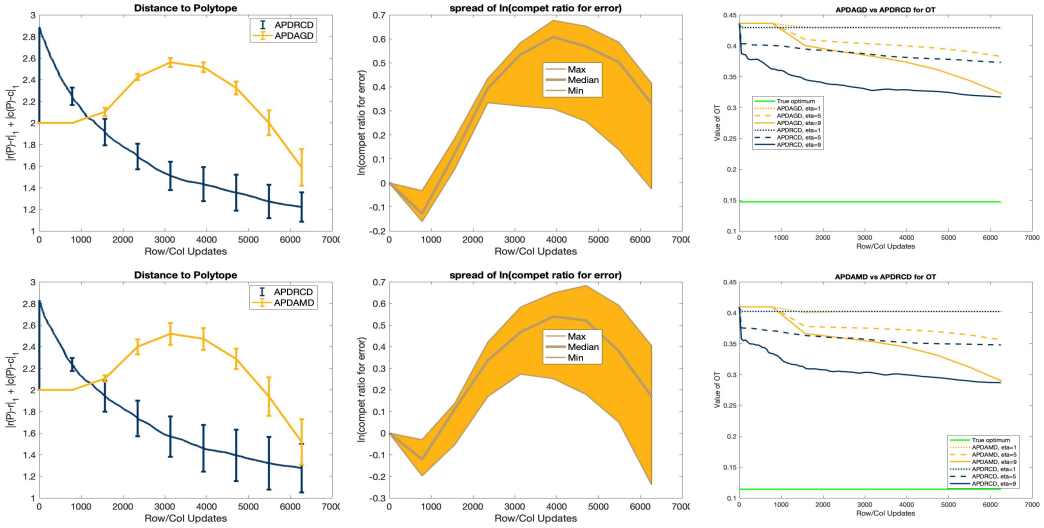

Figure 3: Performance of the APDRCD, APDAGD and APDAMD algorithms on the MNIST images. In the first row of images, we compare the APDRCD and APDAGD algorithms in terms of iteration counts. The leftmost image specifies the distances $d(P)$ to the transportation polytope for two algorithms; the middle image specifies the maximum, median and minimum of competitive ratios on ten random pairs of MNIST images; the rightmost image specifies the values of regularized OT with varying regularization parameter $\eta = \{1, 5, 9\}$. In addition, the second row of images present comparative results for APDRCD versus APDAMD.

5 Discussion

In the paper, we propose and analyze a novel accelerated primal-dual coordinate descent framework for approximating the optimal transport distance between two discrete probability measures. To the best of our knowledge, these algorithms are among the first accelerated primal-dual coordinate descent algorithms proposed for solving the OT problem that share similar complexity upper bounds with existing accelerated primal-dual algorithms while enjoying better experimental performance in practice. There are several future directions arising from the current work. Given the favorable practical performance of the APDRCD and APDGCD algorithms over existing primal-dual algorithms, it is of interest to develop efficient algorithms to search for optimal solutions of computational Wasserstein barycenter problems, which have been used in several applications [5, 23], based on the current primal-dual coordinate descent framework. Furthermore, as large-scale data become prevalent, extending the APDRCD and APDGCD algorithms to asynchronized and distributed computing settings of the OT problems is another interesting and important direction.

Acknowledgements

We would like to thank Tianyi Lin for helpful discussion with the complexity bound of APDRCD algorithm. This work was supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-18-1-2764.

References

[1] J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *NIPS*, pages 1964–1974, 2017. (Cited on pages 1, 2, 4, 5, 6, 15, 16, and 18.)

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017. (Cited on page 1.)

[3] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017. (Cited on page 1.)

[4] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013. (Cited on pages 1 and 3.)

[5] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *ICML*, pages 685–693, 2014. (Cited on pages 2, 8, and 20.)

[6] P. Dvurechenskii, D. Dvinskikh, A. Gasnikov, C. Uribe, and A. Nedich. Decentralize and randomize: Faster algorithm for Wasserstein barycenters. In *NIPS*, pages 10783–10793, 2018. (Cited on pages 2 and 20.)

[7] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorns algorithm. In *ICML*, pages 1367–1376, 2018. (Cited on pages 1, 2, 4, and 6.)

[8] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pages 3440–3448, 2016. (Cited on page 20.)

[9] N. Ho, V. Huynh, D. Phung, and M. I. Jordan. Probabilistic multilevel clustering via composite transportation distance. *AISTATS*, 2019. (Cited on pages 2 and 20.)

[10] B. Kalantari, I. Lari, F. Ricca, and B. Simeone. On the complexity of general matrix scaling and entropy minimization via the RAS algorithm. *Mathematical Programming*, 112(2):371–401, 2008. (Cited on page 1.)

[11] L. V. Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942. (Cited on page 3.)

[12] P. A. Knight. The Sinkhorn–Knopp algorithm: Convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008. (Cited on page 1.)

[13] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\widetilde{O}$(sqrt(rank)) iterations and faster algorithms for maximum flow. In *FOCS*, pages 424–433. IEEE, 2014. (Cited on pages 1 and 3.)

[14] T. Lin, N. Ho, and M. I. Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. *arXiv preprint arXiv:1901.06482*, 2019. (Cited on pages 1, 2, 3, 4, 6, 8, 11, and 14.)

[15] T. Lin, Z. Hu, and X. Guo. Sparsemax and relaxed Wasserstein for topic sparsity. *ArXiv Preprint: 1810.09079*, 2018. (Cited on page 1.)

[16] H. Lu, R. M. Freund, and V. Mirrokni. Accelerating greedy coordinate descent methods. *arXiv preprint arXiv:1806.02476*, 2018. (Cited on page 2.)

[17] X. Nguyen. Convergence of latent mixing measures in finite and infinite mixture models. *Annals of Statistics*, 4(1):370–400, 2013. (Cited on page 1.)

[18] X. Nguyen. Borrowing strength in hierarchical Bayes: posterior concentration of the Dirichlet base measure. *Bernoulli*, 22(3):1535–1571, 2016. (Cited on page 1.)

[19] O. Pele and M. Werman. Fast and robust earth movers distance. In *ICCV*. IEEE, 2009. (Cited on pages 1 and 3.)

[20] P. Rigollet and J. Weed. Uncoupled isotonic regression via minimum Wasserstein deconvolution. *Information and Inference*, To appear. (Cited on page 1.)

[21] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974. (Cited on page 1.)

[22] S. Srivastava, V. Cevher, Q. Dinh, and D. Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *AISTATS*, pages 912–920, 2015. (Cited on page 1.)

[23] S. Srivastava, C. Li, and D. Dunson. Scalable Bayes via barycenter in Wasserstein space. *Journal of Machine Learning Research*, 19(8):1–35, 2018. (Cited on pages 1 and 8.)

[24] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. In *ICLR*, 2018. (Cited on page 1.)

[25] L. Yang, J. Li, D. Sun, and K.-C. Toh. A fast globally linearly convergent algorithm for the computation of wasserstein barycenters. *arXiv preprint arXiv:1809.04249*, 2018. (Cited on page 20.)

Supplement to "Accelerated Primal-Dual Coordinate Descent for Computational Optimal Transport"

In this Supplementary appendix, we first provide detailed proofs of all the key results in Section A. Then, we present the algorithmic framework for accelerated primal-dual greedy coordinate descent (APDGCD) algorithm in Section B. Finally, further comparative experiments between APDGCD algorithm versus APDRCD, APDAGD, APDAMD, and Sinkhorn algorithms are in Section C.

A Proofs for all results

In this appendix, we provide the complete proofs for all results in the main text.

A.1 Proof of Lemma 2.1

The proof is straightforward application of the result from Lemma 4.1 in [14]. Here, we provide the details of this proof for the completeness. Indeed, invoking Lemma 4.1 in [14], we find that

$$\varphi(\lambda_1) - \varphi(\lambda_2) - \langle \nabla\varphi(\lambda_2), \lambda_1 - \lambda_2\rangle \leq \frac{||A||_1^2}{2\eta}||\lambda_1 - \lambda_2||_\infty^2.$$

Since $||A||_1$ equals to the maximum ℓ_1-norm of a column of A and each column of A contains only two nonzero elements which are equal to one, we have $||A||_1 = 2$. Combining with the fact that $||\lambda_1 - \lambda_2||_\infty^2 \leq ||\lambda_1 - \lambda_2||_2^2$, we achieve a conclusion of the lemma.

A.2 Proof of Lemma 3.1

To ease the presentation of proof argument, we denote the vector-valued function $h(i_k) \in \mathbb{R}^{2n}$ such that $\begin{cases} h(i_k)_i = 1 & \text{if } i = i_k \\ h(i_k)_i = 0 & \text{otherwise} \end{cases}$. By the update in Eq. (7) of Algorithm 1, we obtain the following equations

$$\varphi(\lambda^{k+1}) - \varphi(y^k) = \varphi\left(y^k - h(i_k)\frac{1}{L}\nabla_{i_k}\varphi(y^k)\right) - \varphi(y^k). \tag{9}$$

Due to the smoothness of φ with respect to $\|.\|_2$ norm in Lemma 2.1, the following inequalities hold

$$\begin{aligned}
\varphi\left(y^k - h(i_k)\frac{1}{L}\nabla_{i_k}\varphi(y^k)\right) - \varphi(y^k) &\leq \left\langle \nabla\varphi(y^k), -h(i_k)\frac{1}{L}\nabla_{i_k}\varphi(y^k)\right\rangle \\
&\quad + \frac{L}{2}||h(i_k)\frac{1}{L}(\nabla_{i_k}\varphi(y^k))||^2 \\
&= -\frac{1}{L}\left\langle \nabla\varphi(y^k), \nabla_{i_k}\varphi(y^k)h(i_k)\right\rangle + \frac{1}{2}(\nabla_{i_k}\varphi(y^k))^2 \\
&= -\frac{1}{L}(\nabla_{i_k}\varphi(y^k))^2 + \frac{1}{2L}(\nabla_{i_k}\varphi(y^k))^2 \\
&= -\frac{1}{2L}(\nabla_{i_k}\varphi(y^k))^2.
\end{aligned} \tag{10}$$

Combining the results of Eq. (9) and Eq. (10) completes the proof of the lemma.

A.3 Proof of Lemma 3.2

By Eq. (7), we have the following equations

$$\begin{aligned}
|\nabla\varphi_{i_k}(y^k)|^2 &= 2|\nabla\varphi_{i_k}(y^k)|^2 - |\nabla\varphi_{i_k}(y^k)|^2 \\
&= 2L\big(y_{i_k}^k - \lambda_{i_k}^{k+1}\big)\nabla_{i_k}\varphi(y^k) - L^2(\lambda_{i_k}^{k+1} - y_{i_k}^{k+1})^2.
\end{aligned}$$

Combining the above equations with Lemma 3.1, we find that

$$\begin{aligned}
\varphi(\lambda^{k+1}) &\leq \varphi(y^k) - \frac{1}{2L}(\nabla_{i_k}\varphi(y^k))^2 \\
&= \varphi(y^k) + (\lambda_{i_k}^{k+1} - y_{i_k}^k)\nabla_{i_k}\varphi(y^k) + \frac{L}{2}(\lambda_{i_k}^{k+1} - y_{i_k}^{k+1})^2.
\end{aligned} \tag{11}$$

Furthermore, the results from Eq. (7) and Eq. (8) lead to the following equations

$$\lambda_{i_k}^{k+1} - y_{i_k}^k = -\frac{1}{L}\nabla_{i_k}\varphi(y^k),$$
$$z_{i_k}^{k+1} - z_{i_k}^k = -\frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k).$$

Therefore, we have

$$\lambda_{i_k}^{k+1} - y_{i_k}^k = 2n\theta_k(z_{i_k}^{k+1} - z_{i_k}^k).$$

Plugging the above equation into Eq. (11) yields the following inequality

$$\varphi(\lambda^{k+1}) \leq \varphi(y^k) + 2n\theta_k\big(z_{i_k}^{k+1} - z_{i_k}^k\big)\nabla_{i_k}\varphi_{i_k}(y^k) + 2n^2L\theta_k^2\big(z_{i_k}^{k+1} - z_{i_k}^k\big)^2. \tag{12}$$

By the result of Eq. (8), we have

$$(z_{i_k}^{k+1} - z_{i_k}^k) + \frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k) = 0.$$

Therefore, for any $\lambda \in \mathbb{R}^{2n}$, we find that

$$(\lambda_{i_k} - z_{i_k}^{k+1})[(z_{i_k}^{k+1} - z_{i_k}^k) + \frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k)] = 0.$$

The above equation is equivalent to the following equations

$$\begin{aligned}\frac{1}{nL\theta_k}(\lambda_{i_k} - z_{i_k}^{k+1})\nabla_{i_k}\varphi(y^k) &= -2(\lambda_{i_k} - z_{i_k}^{k+1})(z_{i_k}^{k+1} - z_{i_k}^k) \\ &= (\lambda_{i_k} - z_{i_k}^{k+1})^2 - (\lambda_{i_k} - z_{i_k}^k)^2 + (z_{i_k}^{k+1} - z_{i_k}^k)^2\end{aligned}$$

where the second equality in the above display comes from simple algebra. Rewriting the above equality, we have:

$$(z_{i_k}^{k+1} - z_{i_k}^k)^2 = \frac{1}{nL\theta_k}(\lambda_{i_k} - z_{i_k}^{k+1})\nabla_{i_k}\varphi(y^k) - (\lambda_{i_k} - z_{i_k}^{k+1})^2 + (\lambda_{i_k} - z_{i_k}^k)^2$$

Combining the above equation with Eq. (12) yields the following inequality:

$$\varphi(\lambda^{k+1}) \leq \varphi(y^k) + 2n\theta_k(\lambda_{i_k} - z_{i_k}^k)\nabla_{i_k}\varphi(y^k) + 2n^2L\theta_k^2\left[(\lambda_{i_k} - z_{i_k}^k)^2 - (\lambda_{i_k} - z_{i_k}^{k+1})^2\right]. \tag{13}$$

Recall the definition of y^k in Eq. (5) as follows:

$$y^k = (1-\theta_k)\lambda^k + \theta^k z^k,$$

which can be rewritten as:

$$\theta_k(\lambda - z^k) = \theta_k(\lambda - y^k) + (1-\theta_k)(\lambda^k - y^k) \tag{14}$$

for any $\lambda \in \mathbb{R}^{2n}$.

The above equation implies that

$$\begin{aligned}&\varphi(y^k) + \theta_k(\lambda - z^k)^T\nabla\varphi(y^k) \\ &\leq \theta_k[\varphi(y^k) + (\lambda - y^k)^T\nabla\varphi(y^k)] + (1-\theta_k)[\varphi(y^k) + (\lambda^k - y^k)^T\nabla\varphi(y^k)] \\ &\leq \theta_k[\varphi(y^k) + (\lambda - y^k)^T\nabla\varphi(y^k)] + (1-\theta_k)\varphi(\lambda^k)\end{aligned}$$

where the last inequality comes from the convexity of φ. Combining this equation and taking expectation over i_k for the first two terms of Eq. (13), we have:

$$\begin{aligned}\varphi(y^k) + (2n\theta_k)\mathbb{E}_{i_k}[(\lambda_{i_k} - z_{i_k}^k)\nabla_{i_k}\varphi(y^k)] &= \varphi(y^k) + \theta_k(\lambda - z^k)^T\nabla\varphi(y^k) \\ &\leq \theta_k[\varphi(y^k) + (\lambda - y^k)^T\nabla\varphi(y^k)] + (1-\theta_k)\varphi(\lambda^k)\end{aligned} \tag{15}$$

where we use Eq. (14) and the convexity of the dual function in the last step. For the last term in the right hand side of Eq. (13), by taking expectation over i_k, we have:

$$\mathbb{E}_{i_k}\left[2n^2L\theta_k^2\big[(\lambda_{i_k} - z_{i_k}^k)^2 - (\lambda_{i_k} - z_{i_k}^{k+1})^2\big]\right] = 2n^2L\theta_k^2\mathbb{E}_{i_k}\left[||\lambda - z^k||^2 - ||\lambda - z^{k+1}||^2\right] \tag{16}$$

where the last equality comes from the following equations:

$$\begin{aligned}
\mathbb{E}_{i_k}\left[(\lambda_{i_k}-z_{i_k}^k)^2-\left(\lambda_{i_k}-z_{i_k}^{k+1}\right)^2\right] &= \mathbb{E}_{i_k}\left[(\lambda_{i_k}-z_{i_k}^k)^2-\left(\lambda_{i_k}-z_{i_k}^k+\frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k)\right)^2\right] \\
&= \frac{1}{2n}\|\lambda-z^k\|^2-\frac{1}{2n}\sum_{i_k=0}^{2n}\left(\lambda_{i_k}-z_{i_k}^k+\frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k)\right)^2 \\
&= \frac{1}{2n}\|\lambda-z^k\|^2-\frac{1}{2n}\|\lambda-z^k+\frac{1}{2nL\theta_k}\nabla\varphi(y^k)\|^2 \\
&= \frac{1}{2n}\left[-\frac{(\lambda-z^k)}{nL\theta_k}\nabla\varphi(y^k)-\frac{1}{4n^2L^2\theta_k^2}\|\nabla\varphi(y^k)\|^2\right] \\
&= \frac{1}{2n}\left[-4n(\lambda-z^k)\mathbb{E}_{i_k}[z^k-z^{k+1}]-2n\mathbb{E}_{i_k}[||z^k-z^{k+1}||^2]\right]
\end{aligned}$$

where the last inequality is due to the fact that $\nabla\varphi(y^k)=4\mathbb{E}_{i_k}[(z^k-z^{k+1})n^2L\theta_k]$ and Jensen's inequality. Therefore, by simple algebra, we have

$$\begin{aligned}
\mathbb{E}_{i_k}\left[(\lambda_{i_k}-z_{i_k}^k)^2-(\lambda_{i_k}-z_{i_k}^{k+1})^2\right] &= -2(\lambda-z^k)\mathbb{E}_{i_k}[z^k-z^{k+1}]-\mathbb{E}_{i_k}[||z^k-z^{k+1}||^2] \\
&= \mathbb{E}_{i_k}[\|\lambda-z^k\|^2-\|\lambda-z^{k+1}\|^2].
\end{aligned}$$

Notice that equation (13) holds for any value of i_k. Hence, by combining the results from Eq. (15) and Eq. (16) with Eq. (13), at each iteration with a certain value of i_k, we obtain that

$$\begin{aligned}
\mathbb{E}_{i_k}[\varphi(\lambda^{k+1})] \leq (1-\theta_k)\varphi(\lambda^k)+\theta_k[\varphi(y^k)+(\lambda-y^k)^\top\nabla\varphi(y^k)] \\
+2n^2L\theta_k^2\left(||\lambda-z^k||^2-\mathbb{E}_{i_k}[||\lambda-z^{k+1}||^2]\right).
\end{aligned}$$

As a consequence, we achieve the conclusion of the lemma.

A.4 Proof of Theorem 3.3

By the result of Lemma 3.2 and the definition of the sequence $\{\theta_k\}$ in Algorithm 1, we obtain the following bounds:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\theta_k^2}\varphi(\lambda^{k+1})\right] &\leq \mathbb{E}\left[\frac{1-\theta_k}{\theta_k^2}\varphi(\lambda^k)+\frac{1}{\theta_k}[\varphi(y^k)+(\lambda-y^k)^T\nabla\varphi(y^k)]\right. \\
&\qquad \left.+2Ln^2\left(||\lambda-z^k||^2-||\lambda-z^{k+1}||^2\right)\right] \\
&= \mathbb{E}\left[\frac{1}{\theta_{k-1}^2}\varphi(\lambda^k)+\frac{1}{\theta_k}[\varphi(y^k)+(\lambda-y^k)^T\nabla\varphi(y^k)]\right. \\
&\qquad \left.+2Ln^2\left(||\lambda-z^k||^2-||\lambda-z^{k+1}||^2\right)\right]
\end{aligned}$$

where the outer expectations are taken with respect to the random sequence of the coordinate indexes in Algorithm 1. Keep iterating the above bound and using the fact that $\theta_0=1$ and $C_k=1/\theta_k^2$, we

arrive at the following inequalities:

$$
\begin{aligned}
C_k\mathbb{E}\big[\varphi(\lambda^{k+1})\big] &\leq \sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big]+2Ln^2\big(||\lambda-z^0||^2-\mathbb{E}\big[||\lambda-z^{k+1}||^2\big]\big)\\
&\leq \min_{\lambda\in\mathbb{R}^n}\Bigg(\sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big]\\
&\qquad\qquad +2Ln^2\big(||\lambda-z^0||^2-\mathbb{E}\big[||\lambda-z^{k+1}||^2\big]\big)\Bigg)\\
&\leq \min_{\lambda\in B_2(2\hat{R})}\Bigg(\sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big]\\
&\qquad\qquad +2Ln^2\big(||\lambda-z^0||^2-\mathbb{E}\big[||\lambda-z^{k+1}||^2\big]\big)\Bigg)
\end{aligned}
$$

where $\hat{R}:=\eta n(R+\frac{1}{2})$ is the upper bound for l_2-norm of optimal solutions of dual regularized OT problem (4) according to Lemma 3.2 in [14] and $\mathbb{B}_2(r)$ is defined as

$$\mathbb{B}_2(r):=\{\lambda\in\mathbb{R}^{2n}\mid||\lambda||_2\leq r\}.$$

As $\mathbb{E}\big[||\lambda-z^{k+1}||^2\big]\geq 0$, the inequality in the above display can be further rewritten as

$$
\begin{aligned}
C_k\mathbb{E}\big[\varphi(\lambda^{k+1})\big] &\leq \min_{\lambda\in\mathbb{B}_2(2\hat{R})}\Bigg(\sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big]+2Ln^2||\lambda-z^0||^2\Bigg)\\
&\leq \min_{\lambda\in\mathbb{B}_2(2\hat{R})}\Bigg(\sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big]+8Ln^2\hat{R}^2\Bigg) \qquad (17)
\end{aligned}
$$

where the last inequality is due to $z^0=0$. Furthermore, by the definition of the dual entropic regularized OT objective function $\varphi(\lambda)$, we can verify the following equations:

$$
\begin{aligned}
\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle &= \big\langle y^i,b-Ax(y^i)\big\rangle-f(x(y^i))+\big\langle\lambda-y^i,b-Ax(y^i)\big\rangle\\
&= -f(x(y^i))+\big\langle\lambda,b-Ax(y^i)\big\rangle
\end{aligned}
$$

where $f(x):=\langle C,x\rangle$, $x(\lambda):=\underset{x\in\mathbb{R}^{n\times n}}{\arg\max}\Big\{-f(x)-\big\langle A^\top\lambda,x\big\rangle\Big\}$, and $b=\begin{pmatrix}r\\l\end{pmatrix}$. The above equation leads to the following inequality:

$$
\begin{aligned}
\sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[\varphi(y^i)+\big\langle\nabla\varphi(y^i),\lambda-y^i\big\rangle\big] &= \sum_{i=0}^{k}\frac{1}{\theta_i}\mathbb{E}\big[-f(x(y^i))+\big\langle\lambda,b-Ax(y^i)\big\rangle\big]\\
&\leq -C_kf(\mathbb{E}\big[x^k\big])+\sum_{i=0}^{k}\frac{1}{\theta_i}\big\langle\lambda,b-A\mathbb{E}\big[x(y^i)\big]\big\rangle\\
&= C_k\big(-f(\mathbb{E}\big[x^k\big])+\big\langle\lambda,b-A\mathbb{E}\big[x^k\big]\big\rangle\big) \qquad (18)
\end{aligned}
$$

where the second inequality is due to the convexity of f. Combining the results from (17) and (18), we achieve the following bound

$$
\begin{aligned}
C_k\mathbb{E}\big[\varphi(\lambda^{k+1})\big] &\leq -C_kf(\mathbb{E}\big[x^k\big])+\min_{\lambda\in\mathbb{B}_2(2\hat{R})}\big\{C_k\big\langle\lambda,b-A\mathbb{E}\big[x^k\big]\big\rangle\big\}+8Ln^2\hat{R}^2\\
&\leq -C_kf(\mathbb{E}\big[x^k\big])+8Ln^2\hat{R}^2-2C_k\hat{R}\mathbb{E}\big[||Ax^k-b||_2\big].
\end{aligned}
$$

The above inequality is equivalent to

$$f(\mathbb{E}\big[x^k\big])+\mathbb{E}\big[\varphi(\lambda^{k+1})\big]+2\hat{R}\mathbb{E}\big[||Ax^k-b||_2\big]\leq\frac{8Ln^2\hat{R}^2}{C_k}. \qquad (19)$$

Denoting λ^* as the optimal solution for the dual entropic regularized OT problem (4). Then, we can verify the following inequalities

$$\begin{aligned} f(\mathbb{E}[x^k]) + \mathbb{E}[\varphi(\lambda^{k+1})] &\geq f(\mathbb{E}[x^k]) + \varphi(\lambda^*) \\ &= f(\mathbb{E}[x^k]) + \langle \lambda^*, b\rangle + \max_{x\in\mathbb{R}^{n\times n}} \{-f(x) - \langle A^\top \lambda^*, x\rangle\} \\ &\geq f(\mathbb{E}[x^k]) + \langle \lambda^*, b\rangle - f(\mathbb{E}[x^k]) - \langle \lambda^*, A\mathbb{E}[x^k]\rangle \\ &= \langle \lambda^*, b - A\mathbb{E}[x^k]\rangle \\ &\geq -\hat{R}\mathbb{E}[||Ax^k - b||_2] \end{aligned} \tag{20}$$

where the last inequality comes from Hölder inequality and the fact that $||\lambda^*||_2 \leq \hat{R}$. Plugging the inequality in (20) to the inequality in (19) leads to the following bound:

$$\mathbb{E}[||Ax^k - b||_2] \leq \frac{8Ln^2\hat{R}}{C_k} = \frac{8||A||_1^2 n^3(R+1/2)}{C_k} = \frac{32n^3(R+1/2)}{C_k}. \tag{21}$$

It remains to bound C_k. We will use induction to show that $\theta_k \leq \frac{2}{k+2}$ for all $k \geq 0$. The inequality clearly holds for $k = 0$ as $\theta_0 = 1$. Suppose that the hypothesis holds for $k \geq 0$, namely, $\theta_k \leq \frac{2}{k+2}$ for $k \geq 0$. By the definition of θ_{k+1} and simple algebra, we obtain that

$$\theta_{k+1} = \frac{\theta_k^2}{2}\left(\sqrt{1 + \frac{4}{\theta_k^2}} - 1\right) \leq \frac{2}{k+3}$$

where the above inequality is due to $\theta_k \leq \frac{2}{k+2}$. Therefore, we achieve the conclusion of the hypothesis for $k+1$. Now, simple algebra demonstrates that $C_k \geq \frac{1}{4}(k+1)(k+4) \geq \frac{1}{4}(k+1)^2$. Combining this lower bound of C_k and the inequality in (21) leads to the following result:

$$\mathbb{E}[||Ax^k - b||_2] \leq \frac{144n^3(R+1/2)}{(k+1)^2}.$$

As a consequence, we conclude the desired bound on the number of iterations k required to satisfy the bound $\mathbb{E}[||A\text{vec}(X^k) - b||_2] \leq \varepsilon'$.

A.5 Proof of Theorem 3.4

The proof of the theorem follows the same steps as those in the proof of Theorem 1 in [1]. Here, we provide the detailed proof for the completeness. In particular, we denote $\tilde{X}$ the matrix returned by the APDRCD algorithm (Algorithm 1) with $\tilde{r}$, $\tilde{l}$ and $\varepsilon'/2$. Recall that, X^* is a solution to the OT problem. Then, we obtain the following inequalities:

$$\begin{aligned} \left\langle C, \hat{X}\right\rangle - \langle C, X^*\rangle &\leq 2\eta\log(n) + 4\left(\left\|\tilde{X}\mathbf{1} - r\right\|_1 + \left\|\tilde{X}^\top\mathbf{1} - l\right\|_1\right)\|C\|_\infty \\ &\leq \frac{\varepsilon}{2} + 4\left(\left\|\tilde{X}\mathbf{1} - r\right\|_1 + \left\|\tilde{X}^\top\mathbf{1} - l\right\|_1\right)\|C\|_\infty, \end{aligned}$$

where the last inequality in the above display holds since $\eta = \frac{\varepsilon}{4\log(n)}$. Furthermore, we have

$$\begin{aligned} \left\|\tilde{X}\mathbf{1} - r\right\|_1 + \left\|\tilde{X}^\top\mathbf{1} - l\right\|_1 &\leq \left\|\tilde{X}\mathbf{1} - \tilde{r}\right\|_1 + \left\|\tilde{X}^\top\mathbf{1} - \tilde{l}\right\|_1 + \|r - \tilde{r}\|_1 + \left\|l - \tilde{l}\right\|_1 \\ &\leq \frac{\varepsilon'}{2} + \frac{\varepsilon'}{2} = \varepsilon'. \end{aligned}$$

Since $\varepsilon' = \dfrac{\varepsilon}{8\|C\|_\infty}$, the above inequalities demonstrate that $\left\langle C, \hat{X}\right\rangle - \langle C, X^*\rangle \leq \varepsilon$. Hence, we only need to bound the complexity. Following the approximation scheme in Step 1 of Algorithm 2, we achieve the following bound

$$\begin{aligned} R &= \frac{||C||_\infty}{\eta} + \log(n) - 2\log(\min_{1\leq i,j\leq n}\{\widetilde{r}_i, \widetilde{l}_i\}) \\ &\leq \frac{4\|C\|_\infty\log(n)}{\varepsilon} + \log(n) - 2\log\left(\frac{\varepsilon}{64n\|C\|_\infty}\right). \end{aligned}$$

Given the above bound with R, we have the following bound with the iteration count:

$$
\begin{aligned}
k &\leq 1 + 12n^{\frac{3}{2}}\sqrt{\frac{R+1/2}{\varepsilon'}} \\
&\leq 1 + 12n^{\frac{3}{2}}\sqrt{\frac{8||C||_\infty\left(\frac{4||C||_\infty \log(n)}{\varepsilon} + \log(n) - 2\log\left(\frac{\varepsilon}{64n||C||_\infty}\right) + 1/2\right)}{\varepsilon}} \\
&= \mathcal{O}\left(\frac{n^{\frac{3}{2}}||C||_\infty\sqrt{\log(n)}}{\varepsilon}\right).
\end{aligned}
$$

Combining the above result with the fact that each iteration the APDRCD algorithm requires $\mathcal{O}(n)$ arithmetic operations to compute the gradient of one coordinate block, we conclude that the total number of arithmetic operations required for the APDRCD algorithm for approximating optimal transport is $\mathcal{O}\left(\frac{n^{\frac{5}{2}}||C||_\infty\sqrt{\log(n)}}{\varepsilon}\right)$. Furthermore, the column $\tilde{r}$ and row $\tilde{l}$ in Step 2 of Algorithm 2 can be found in $\mathcal{O}(n)$ arithmetic operations while Algorithm 2 in [1] requires $\mathcal{O}(n^2)$ arithmetic operations. As a consequence, we conclude that the total number of arithmetic operations is $\mathcal{O}\left(\frac{n^{\frac{5}{2}}||C||_\infty\sqrt{\log(n)}}{\varepsilon}\right)$.

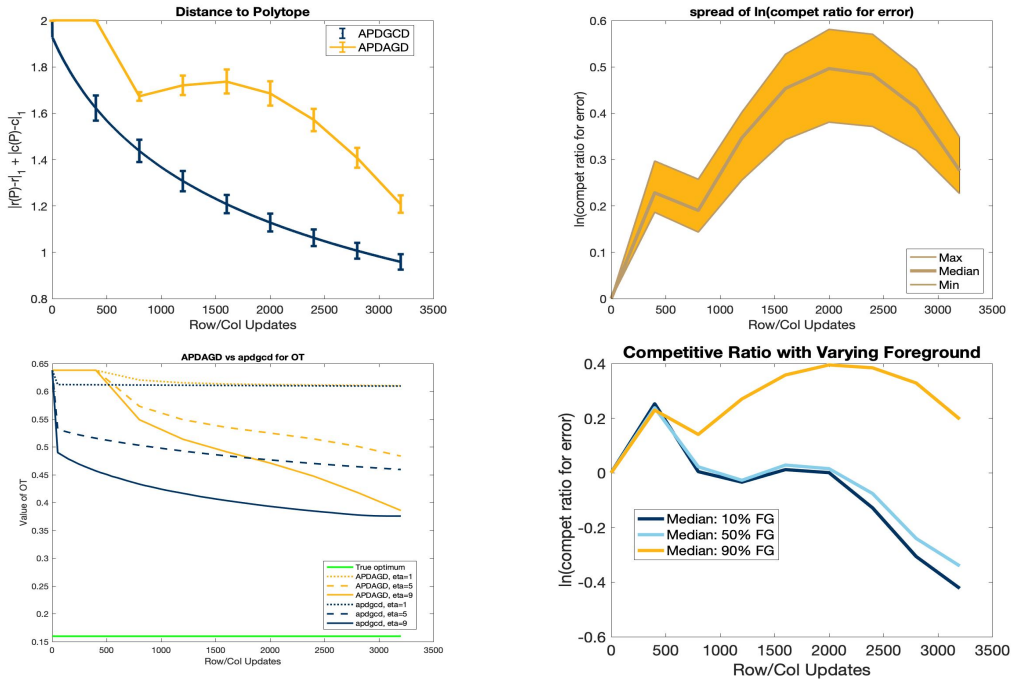

Figure 4: Performance of APDGCD and APDAGD algorithms on the synthetic images. The organization of the images is similar to those in Figure 1.

B Accelerated Primal-Dual Greedy Coordinate Descent (APDGCD) Algorithm

In this appendix, we present a greedy version of APDRCD algorithm, which is termed as the *accelerated primal-dual greedy coordinate descent (APDGCD)* algorithm. The detailed pseudo-code of that algorithm is in Algorithm 3 while an approximating scheme of OT based on the APDGCD algorithm is summarized in Algorithm 4.

Both the APDGCD and APDRCD algorithms follow along the general accelerated primal-dual coordinate descent framework. Similar to the APDRCD algorithm, the algorithmic framework of

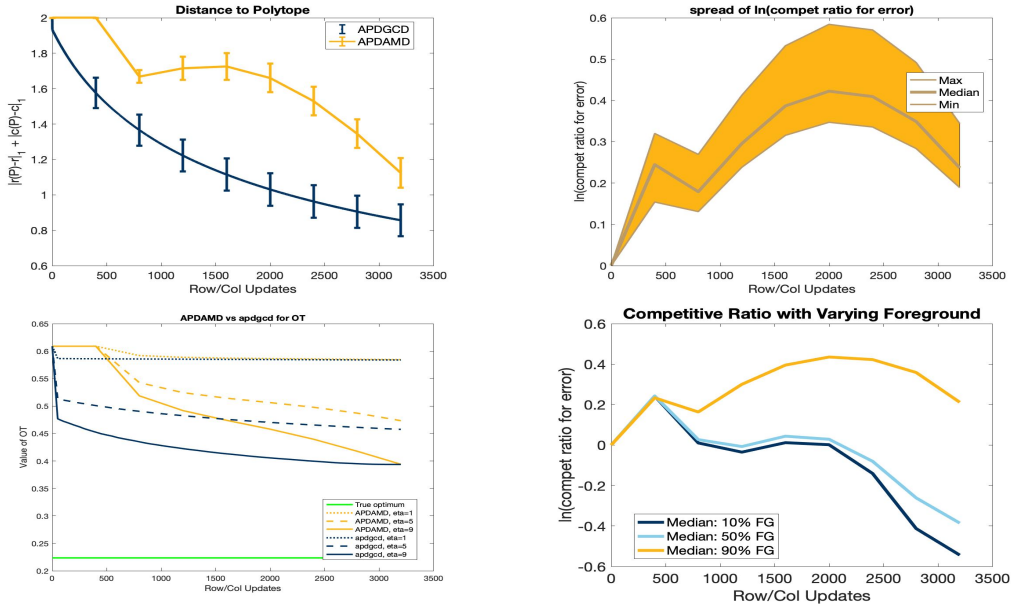

Figure 5: Performance of APDGCD and APDAMD algorithms on the synthetic images. The organization of the images is similar to those in Figure 1.

APDGCD is composed by two main parts: First, instead of performing the randomized accelerated coordinate descent on the dual objective function as a subroutine in step (24) and step (25),the APDGCD algorithm chooses the best coordinate that maximizes the absolute value of the gradient of the dual objective function of regularized OT problem among all the coordinates. In the second part, we follow the key averaging step in the APDRCD algorithm by taking a weighted average over the past iterations to get a good approximated solution for the primal problem from the approximated solutions to the dual problem. Since the auxiliary sequence is decreasing, the primal solutions corresponding to the more recent dual solutions have more weights in this average.

We further demonstrate that the APDGCD algorithm enjoys favorable practical performance than APDRCD algorithm in both synthetic and real datasets (cf. Appendix C).

C Further experiments with the APDRCD and APDGCD algorithms

In this appendix, we provide further comparative experiments between APDGCD algorithm versus APDRCD, APDAGD, APDAMD, and Sinkhorn algorithms. Experimental results (cf. Section 4 and Appendix C) show that APDGCD enjoys favorable practical performance than APDAGD, APDAMD, and APDRCD algorithms on both synthetic and real datasets. This demonstrates the benefit of choosing the best coordinate to descent to optimize the dual objective function of entropic regularized OT problems in the APDGCD algorithm comparing to choosing the random descent coordinate in the APDRCD algorithm.

C.1 APDGCD algorithm with synthetic images

The generation of synthetic images as well as the evaluation metrics are similar to those in Section 4.1. We respectively present in Figure 4, Figure 5 and Figure 6 the comparisons between APDGCD algorithm versus APDAGD, APDAMD and APDRCD algorithms.

According to Figure 4, Figure 5 and Figure 6, the APDGCD algorithm enjoys better performance than the APDAGD, APDAMD and also the APDRCD algorithms in terms of the iteration numbers in terms of both the evaluation metrics. Besides, at the same number of iteration number, the APDGCD algorithm achieves even faster decrements than other three algorithms with regard to both the distance to polytope and the value of OT metrics during the computing process. This is beneficial in practice for easier tuning and smaller error when the update number is limited.

Algorithm 3: APDGCD $(C, \eta, A, b, \varepsilon')$

Input: $\{\theta_k | \theta_0 = 1, \frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2}, \}, \lambda^0 = \lambda_0 = 0, z^0 = z_0 = 0, k = 0$

while $\mathbb{E}\left[||Ax^k - b||_2\right] > \varepsilon'$ **do**

Set

$$y^k = (1 - \theta_k)\lambda^k + \theta_k z^k \tag{22}$$

Compute

$$x^k = \frac{1}{C_k}\left(\sum_{j=0}^{k} \frac{x(y^j)}{\theta_j}\right) \text{ where } C_k := \sum_{j=0}^{k} \frac{1}{\theta_j} \tag{23}$$

Select coordinate $i_k = \underset{i_k \in \{1,2,\ldots,2n\}}{\operatorname{argmax}} |\nabla_{i_k}\varphi(y^k)|$:

Update

$$\lambda_{i_k}^{k+1} = y_{i_k}^k - \frac{1}{L}\nabla_{i_k}\varphi(y^k) \tag{24}$$

Update

$$z_{i_k}^{k+1} = z_{i_k}^k - \frac{1}{2nL\theta_k}\nabla_{i_k}\varphi(y^k) \tag{25}$$

Update

$$k = k + 1$$

end

Output: X^k where $x^k = vec(X^k)$

Algorithm 4: Approximating OT by APDGCD

Input: $\eta = \dfrac{\varepsilon}{4\log(n)}$ and $\varepsilon' = \dfrac{\varepsilon}{8\|C\|_\infty}$.

Step 1: Let $\tilde{r} \in \Delta_n$ and $\tilde{l} \in \Delta_n$ be defined as

$$\left(\tilde{r}, \tilde{l}\right) = \left(1 - \frac{\varepsilon'}{8}\right)(r, l) + \frac{\varepsilon'}{8n}(\mathbf{1}, \mathbf{1}).$$

Step 2: Let $A \in \mathbb{R}^{2n \times n^2}$ and $b \in \mathbb{R}^{2n}$ be defined by

$$Avec(X) = \begin{pmatrix} X\mathbf{1} \\ X^T\mathbf{1} \end{pmatrix} \text{ and } b = \begin{pmatrix} \tilde{r} \\ \tilde{l} \end{pmatrix}$$

Step 3: Compute $\tilde{X} = \text{APDGCD}\,(C, \eta, A, b, \varepsilon'/2)$ with φ defined in 4.

Step 4: Round $\tilde{X}$ to $\hat{X}$ by Algorithm 2 [1] such that $\hat{X}\mathbf{1} = r$, $\hat{X}^\top\mathbf{1} = l$.

Output: $\hat{X}$.

C.2 APDGCD algorithm with MNIST images

Moving beyond the synthetic images, we proceed to present the comparisons between APDGCD algorithm versus APDAGD, APDAMD, and APDRCD algorithms in Figure 7 with MNIST images. In summary, the consistent superior performance of APDGCD over APDAGD, APDAMD and APDRCD on the MNIST dataset shows the advantage of using greedy coordinate descent for optimizing the dual objective function of entropic regularized OT problems.

According to Figure 7, the APDGCD algorithm enjoys better performance than the APDAGD, APDAMD and also the APDRCD algorithms in terms of the iteration numbers in terms of both the evaluation metrics. Furthermore, the convergence of the APDGCD algorithm is faster than other three algorithms with regard to both the distance to polytope and the value of OT metrics during the computing process when the number of iterations are small. This is beneficial in practice for easier tuning and smaller error when the total update number is limited.

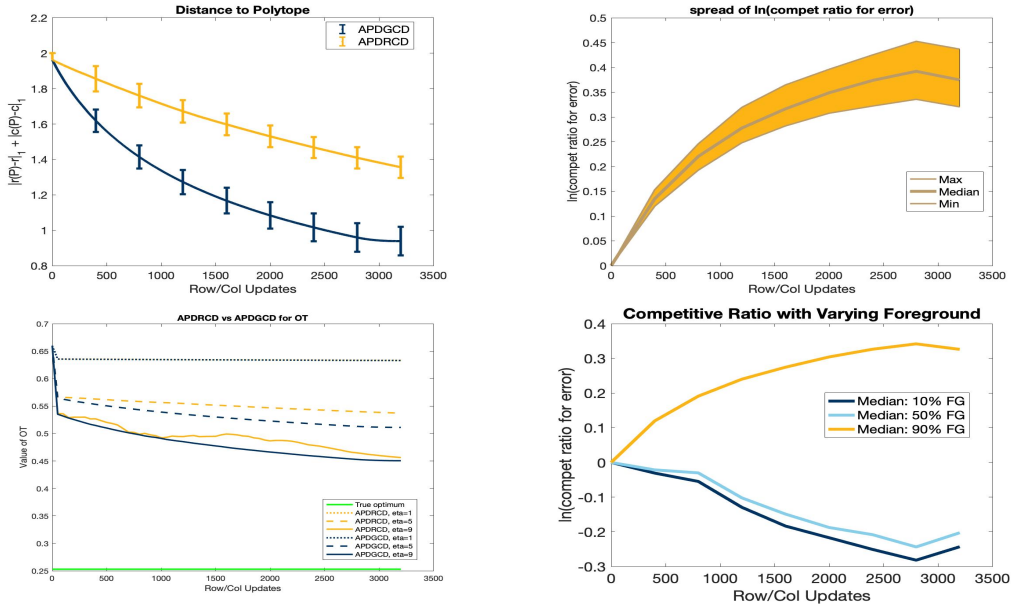

Figure 6: Performance of APDGCD and APDRCD algorithms on the synthetic images. The organization of the images is similar to those in Figure 1.

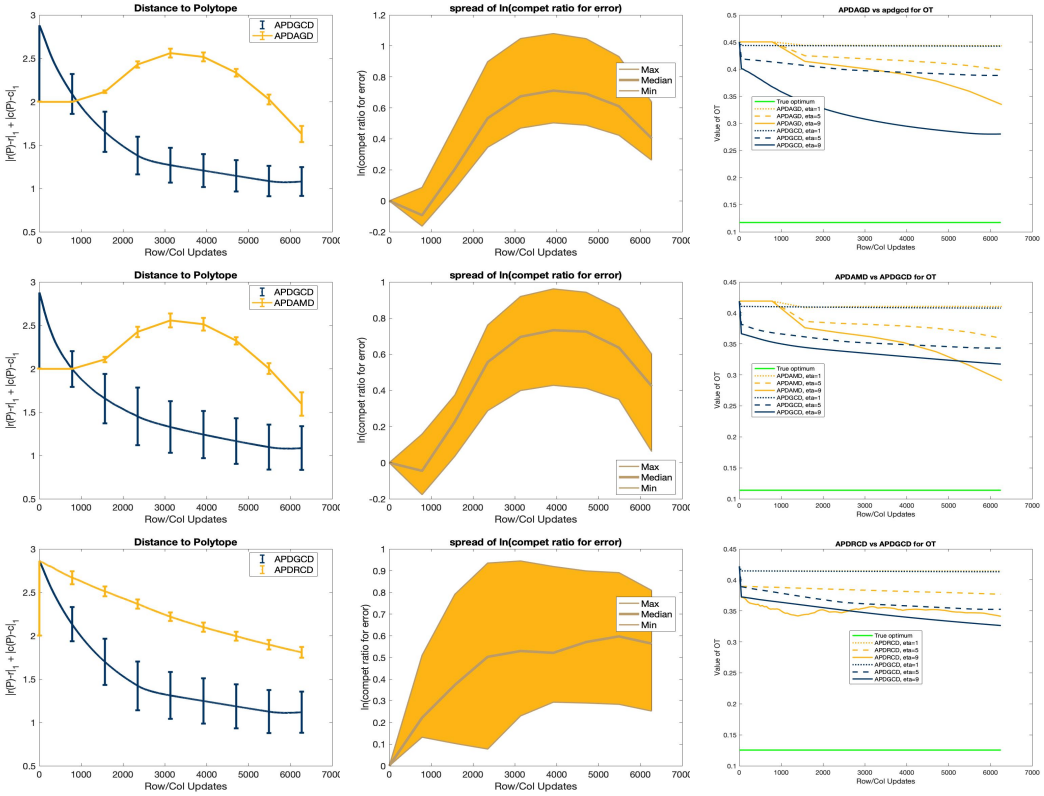

Figure 7: Performance of the APDGCD, APDAGD, APDAMD, and APDRCD algorithms on the MNIST real images. The organization of the images is similar to those in Figure 3.

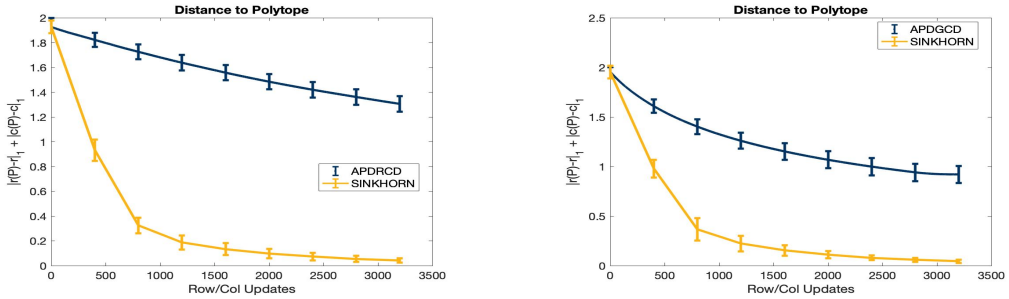

Figure 8: Performance of APDRCD, APDGCD algorithms versus Sinkhorn algorithm on the synthetic images.

C.3 APDRCD and APDGCD algorithms versus Sinkhorn algorithm

For the completeness of the comparative experiments, in this appendix section we present the comparisons of the performance between APDRCD, APDGCD algorithms versus Sinkhorn algorithm with Figure 8.

According to Figure 8, we note in passing that similar to the APDAGD and APDAMD algorithms, the APDRCD and APDGCD methods are not faster in terms of number of iterations than Sinkhorn algorithm in standard settings of OT, while the APDGCD algorithm outperforms the APDRCD algorithm. However, we note that for large-scale applications of the OT problems, such as computational Wasserstein barycenter [5] or multilevel clustering problems [9] where the Sinkhorn algorithm is in general not flexible enough to be adapted to, the accelerated primal-dual algorithms are particularly suitable [6, 8, 25]. Since the APDRCD and APDGCD algorithms obtain the best practical performance among accelerated primal-dual algorithms, we anticipate the superior performance of these proposed algorithms to large-scale applications of OT over state-of-the-art algorithms in the literature.