
A Fast Proximal Point Method for Computing Exact Wasserstein Distance

Yujia Xie

Georgia Tech
Atlanta, GA 30332

Xie.Yujia000@gmail.com

Xiangfeng Wang

East China Normal University
Shanghai, China 200026

xfwang@sei.ecnu.edu.cn

Ruijia Wang

Georgia Tech
Atlanta, GA 30332

rwang@gatech.edu

Hongyuan Zha

Georgia Tech
Atlanta, GA 30332

zha@cc.gatech.edu

Abstract

Wasserstein distance plays increasingly important roles in machine learning, stochastic programming and image processing. Major efforts have been under way to address its high computational complexity, some leading to approximate or regularized variations such as Sinkhorn distance. However, as we will demonstrate, regularized variations with large regularization parameter will degrade the performance in several important machine learning applications, and small regularization parameter will fail due to numerical stability issues with existing algorithms. We address this challenge by developing an Inexact Proximal point method for exact Optimal Transport problem (IPOT) with the proximal operator approximately evaluated at each iteration using projections to the probability simplex. The algorithm (a) converges to exact Wasserstein distance with theoretical guarantee and robust regularization parameter selection, (b) alleviates numerical stability issue, (c) has similar computational complexity to Sinkhorn, and (d) avoids the shrinking problem when apply to generative models. Furthermore, a new algorithm is proposed based on IPOT to obtain sharper Wasserstein barycenter.

1 INTRODUCTION

Many practical tasks in machine learning rely on computing a Wasserstein distance between probability measures

Corresponds to: Y. Xie, X. Wang, H. Zha.

Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence, Tel Aviv, Isreal, 2019. Copyright 2019 by the author(s).

or between their sample points [3, 41, 43, 37]. However, the high computational cost of Wasserstein distance has been a thorny issue and has limited its application to challenging machine learning problems.

In this paper we focus on Wasserstein distance for discrete distributions the computation of which amounts to solving the following discrete *optimal transport* (OT) problem,

$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\Gamma \in \Sigma(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{C}, \Gamma \rangle. \quad (1)$$

Here $\boldsymbol{\mu}, \boldsymbol{\nu}$ are two probability vectors, $W(\boldsymbol{\mu}, \boldsymbol{\nu})$ is the Wasserstein distance between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Matrix $\mathbf{C} = [c_{ij}] \in \mathbb{R}_+^{m \times n}$ is the *cost matrix*, whose element c_{ij} represents the distance between the i -th support point of $\boldsymbol{\mu}$ and the j -th one of $\boldsymbol{\nu}$. The optimal solution Γ^* is referred as *optimal transport plan*. Notation $\langle \cdot, \cdot \rangle$ represents the Frobenius dot-product and $\Sigma(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\Gamma \in \mathbb{R}_+^{m \times n} : \Gamma \mathbf{1}_m = \boldsymbol{\mu}, \Gamma^\top \mathbf{1}_n = \boldsymbol{\nu}\}$, where $\mathbf{1}_n$ represents n -dimensional vector of ones. This is a linear programming problem with typical super $O(n^3)$ complexity¹.

An effort by Cuturi to reduce the complexity leads to a regularized variation of (1) giving rise the so-called Sinkhorn distance [8]. It aims to solve an entropy regularized optimal transport problem

$$W_\epsilon(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\Gamma \in \Sigma(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{C}, \Gamma \rangle + \epsilon h(\Gamma). \quad (2)$$

The entropic regularizer $h(\Gamma) = \sum_{i,j} \Gamma_{ij} \ln \Gamma_{ij}$ results in an optimization problem (2) that can be solved efficiently by iterative Bregman projections [5],

$$\mathbf{a}^{(l+1)} = \frac{\boldsymbol{\mu}}{\mathbf{G}\mathbf{b}^{(l)}}, \quad \mathbf{b}^{(l+1)} = \frac{\boldsymbol{\nu}}{\mathbf{G}^\top \mathbf{a}^{(l+1)}}$$

starting from $\mathbf{b}^{(0)} = \frac{1}{n} \mathbf{1}_n$, where $\mathbf{G} = [G_{ij}]$ and $G_{ij} = e^{-C_{ij}/\epsilon}$. The optimal solution Γ^* then takes the form $\Gamma_{ij}^* = a_i G_{ij} b_j$. The iteration is also referred as *Sinkhorn iteration*, and the method is referred as *Sinkhorn algorithm* which, recently, is proven to achieve a near- $O(n^2)$ complexity [2].

¹Assume $O(n) = O(m)$.

The choice of ϵ cannot be arbitrarily small. Firstly, $G_{ij} = e^{-C_{ij}/\epsilon}$ tends to underflow if ϵ is very small. The methods in [5, 6, 19] try to address this numerical instability by performing the computation in log-space, but they require a significant amount of extra exponential and logarithmic operations, and thus, compromise the advantage of efficiency. More significantly, even with the benefits of log-space computation, the linear convergence rate of the Sinkhorn algorithm is determined by the *contraction ratio* $\kappa(\mathbf{G})$, which approaches 1 as $\epsilon \rightarrow 0$ [14]. Consequently, we observe drastically increased number of iterations for Sinkhorn method when using small ϵ .

Can we just employ Sinkhorn distance with a moderately sized ϵ for machine learning problems so that we can get the benefits of the reduced complexity? Some applications show Sinkhorn distance can generate good results with a moderately sized ϵ [15, 20]. However, we show that in several important problems such as generative model learning and Wasserstein barycenter computation, a moderately sized ϵ will significantly degrade the performance while the Sinkhorn algorithm with a very small ϵ becomes prohibitively expensive (also shown in [36]).

In this paper, we propose a new framework, Inexact Proximal point method for Optimal Transport (IPOT) to compute the Wasserstein distance using generalized proximal point iterations based on Bregman divergence. To enhance efficiency, the proximal operator is inexactly evaluated at each iteration using projections to the probability simplex, leading to an **inexact** update at each iteration yet converging to the **exact** optimal transport solution.

Regarding the theoretical analysis of IPOT, we provide conditions on the number of inner iterations that will guarantee the linear convergence of IPOT. In fact, empirically, IPOT behaves better than the analysis: the algorithm seems to be linearly convergent with just one inner iteration, demonstrating its efficiency. We also perform several other tests to show the excellent performance of IPOT. As we will discuss in Section 6.2, the computation complexity is almost indistinguishable comparing to the Sinkhorn method. Yet again, IPOT avoids the lengthy and experience-based tuning of the ϵ and can converge to the true optimal transport solution robustly with respect to its own parameters. This is unquestionably important in applications where the exact sparse transport plan is preferred. In applications where only Wasserstein distance is needed, the bias caused by regularization might also be problematic. As an example, when applying Sinkhorn to generative model learning, it causes the shrinkage of the learned distribution towards the mean, and therefore cannot cover the whole support of the target distribution adequately.

Furthermore, we develop another new algorithm based

on the proposed IPOT to compute Wasserstein barycenter (see Section 4). Better performance is obtained with much sharper images. It turns out that the inexact evaluation of the proximal operator blends well with Sinkhorn-like barycenter iteration.

2 PRELIMINARIES

We then provide some background on optimal transport and proximal point method.

2.1 Wasserstein Distance and Optimal Transport

Wasserstein distance is a metric for two probability distributions. Given two distributions μ and ν , the p -Wasserstein distance between them is defined as

$$W_p(\mu, \nu) := \left\{ \inf_{\gamma \in \Sigma(\mu, \nu)} \int_{\mathcal{M} \times \mathcal{M}} d^p(x, y) d\gamma(x, y) \right\}^{\frac{1}{p}}, \quad (3)$$

where $\Sigma(\mu, \nu)$ is the set of joint distributions whose marginals are μ and ν , respectively. The above optimization problem is also called the Monge-Kantorovich problem or *optimal transport* problem [17]. In the following, we focus on the 2-Wasserstein distance, and for convenience we write $W(\cdot, \cdot) = W_2^2(\cdot, \cdot)$.

When μ and ν both have finite supports, we can represent the distributions as vectors $\boldsymbol{\mu} \in \mathbb{R}_+^m, \boldsymbol{\nu} \in \mathbb{R}_+^n$, where $\|\boldsymbol{\mu}\|_1 = \|\boldsymbol{\nu}\|_1 = 1$. Then the Wasserstein distance between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is computed by (1). In other cases, given realizations $\{x_i\}_{i=1}^m$ and $\{y_j\}_{j=1}^n$ of μ and ν , respectively, we can approximate them by empirical distributions $\hat{\mu} = \frac{1}{m} \sum_{x_i} \delta_{x_i}$ and $\hat{\nu} = \frac{1}{n} \sum_{y_j} \delta_{y_j}$. The supports of $\hat{\mu}$ and $\hat{\nu}$ are finite, so similarly we have $\boldsymbol{\mu} = \frac{1}{m} \mathbf{1}_{\{x_i\}}, \boldsymbol{\nu} = \frac{1}{n} \mathbf{1}_{\{y_j\}}$, and $\mathbf{C} = [c(x_i, y_j)] \in \mathbb{R}_+^{m \times n}$.

The optimization problem (1) can be solved by linear programming (LP) methods. LP tends to provide a sparse solution, which is preferable in applications like histogram calibration or color transferring [24]. However, the cost of LP scales at least $O(n^3 \log n)$ for general metric, where n is the number of data points [22]. As aforementioned, an alternative optimization method is the Sinkhorn algorithm in [8]. Following the same strategy, many variants of the Sinkhorn algorithm have been proposed [2, 10, 40]. Unfortunately, all these methods only approximate original optimal transport by its regularized version and their performance both in terms of numerical stability and computational complexity is sensitive to the choice of ϵ .

2.2 Generalized Proximal Point Method

Proximal point methods are widely used in optimization [1, 21, 28, 29]. Here, we introduce its generalized form. Given a convex objective function f defined on \mathcal{X} with optimal solution set $\mathcal{X}^* \subset \mathcal{X}$, generalized proximal

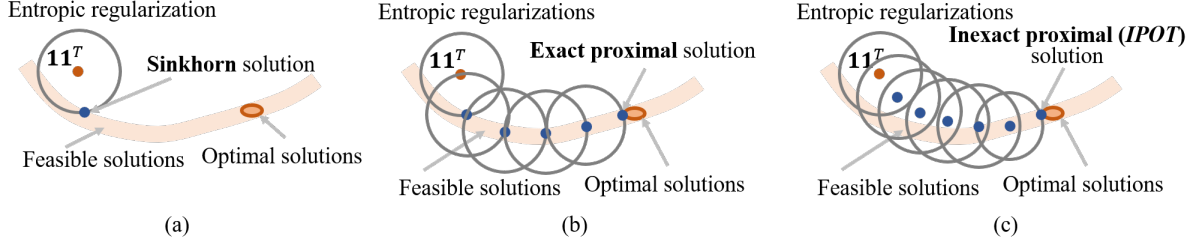


Figure 1: Schematic of the convergence path of (a) Sinkhorn algorithm, (b) exact proximal point algorithm and (c) inexact proximal point algorithm (IPOT). The distance shown is in Bregman sense. Sinkhorn solution is feasible and the closest to optimal solution set within the D_h constraints, but is not in the optimal solution set. However, proximal point algorithm, no matter exact or inexact, solves optimization with D_h constraints iteratively, until an optimal solution is reached.

point algorithm aims to solve

$$\arg \min_{x \in \mathcal{X}} f(x). \quad (4)$$

In order to solve Problem (4), the algorithm generates a sequence $\{x^{(t)}\}_{t=1,2,\dots}$ by the following generalized proximal point iterations:

$$x^{(t+1)} = \arg \min_{x \in \mathcal{X}} f(x) + \beta^{(t)} d(x, x^{(t)}), \quad (5)$$

where d is a regularization term used to define the proximal operator, usually defined to be a closed proper convex function. For classical proximal point method, d adopts the square of Euclidean distance, i.e., $d(x, y) = \|x - y\|_2^2$, in which case the sequence $\{x^{(t)}\}$ converges to an element in \mathcal{X}^* almost surely.

The generalized proximal point method has many advantages, e.g. it has a robust convergence behavior – a fairly mild condition on $\beta^{(t)}$ guarantee its convergence for some given d , and the specific choice of $\beta^{(t)}$ generally just affects its convergence rate. Moreover, even if the proximal operator defined in (5) is not exactly evaluated in each iteration, giving rise to inexact proximal point methods, the global convergence of which with local linear rate is still guaranteed under certain conditions [34, 31].

3 BREGMAN DIVERGENCE BASED PROXIMAL POINT METHOD

In this section we will develop the main algorithm IPOT. Specifically, we will use generalized proximal point method to solve the optimal transport problem (1). Recall the proximal point iteration (5), we take $f(\Gamma) = \langle C, \Gamma \rangle$, $\mathcal{X} = \Sigma(\mu, \nu)$, and $d(\cdot, \cdot)$ to be Bregman divergence D_h based on entropy function $h(x) = \sum_i x_i \ln x_i$, i.e.,

$$D_h(x, y) = \sum_{i=1}^n x_i \log \frac{x_i}{y_i} - \sum_{i=1}^n x_i + \sum_{i=1}^n y_i. \quad (6)$$

As a result, the proximal point iteration for problem (1) can be written as

$$\Gamma^{(t+1)} = \arg \min_{\Gamma \in \Sigma(\mu, \nu)} \langle C, \Gamma \rangle + \beta^{(t)} D_h(\Gamma, \Gamma^{(t)}). \quad (7)$$

However, it is still not trivial to solve the above optimization problem in each iteration, since optimization problems with such complicated constraints generally does not have a closed-form solution. Fortunately, with some reorganization, we can solve it with Sinkhorn algorithm.

Substituting Bregman divergence (6) into proximal point iteration (7), with simplex constraints, we obtain

$$\Gamma^{(t+1)} = \arg \min_{\Gamma \in \Sigma(\mu, \nu)} \langle C - \beta^{(t)} \log \Gamma^{(t)}, \Gamma \rangle + \beta^{(t)} h(\Gamma). \quad (8)$$

Denote $C' = C - \beta^{(t)} \ln \Gamma^{(t)}$. Note that for optimization problem (8), $\Gamma^{(t)}$ is a fixed value that is not relevant to optimization variable Γ . Therefore, C' can be viewed as a new cost matrix that is known, and problem (8) is an entropy regularized optimal transport problem. Comparing to (2), problem (8) can be solved by Sinkhorn iteration by replacing G_{ij} by $G'_{ij} = e^{-C'_{ij}/\beta^{(t)}} = \Gamma_{ij}^{(t)} e^{-C_{ij}/\beta^{(t)}}$. As we will later shown in Section 5, as $t \rightarrow \infty$, $\Gamma^{(t)}$ will converge to an optimal transport plan.

Figure 1 illustrates how Sinkhorn and IPOT solutions approach optimal solution with respect to number of iterations in sense of Bregman divergence. First, let's consider Sinkhorn algorithm. The objective function of Sinkhorn (2) has regularization term $\epsilon h(\Gamma)$, which can be equivalently rewritten as constraint $D_h(\Gamma, \mathbf{11}^T) \leq \eta$ for some $\eta > 0$. Therefore, Sinkhorn solution is feasible within the D_h constraints and the closest to optimal solution set, as shown in Figure 1 (a).

Proximal point algorithms, on the other hand, solves optimization with D_h constraints iteratively as shown in Figure 1 (b)(c). Different from Sinkhorn algorithm, proximal point algorithms converge to the optimal solution

Algorithm 1 IPOT($\boldsymbol{\mu}, \boldsymbol{\nu}, \mathbf{C}$)

Input: Probabilities $\{\boldsymbol{\mu}, \boldsymbol{\nu}\}$ on support points $\{x_i\}_{i=1}^m$, $\{y_j\}_{j=1}^n$, cost matrix $\mathbf{C} = [\|x_i - y_j\|]$
 $\mathbf{b} \leftarrow \frac{1}{m} \mathbf{1}_m$
 $G_{ij} \leftarrow e^{-\frac{C_{ij}}{\beta}}$
 $\boldsymbol{\Gamma}^{(1)} \leftarrow \mathbf{1} \mathbf{1}^T$
for $t = 1, 2, 3, \dots$ **do**
 $\mathbf{Q} \leftarrow \mathbf{G} \odot \boldsymbol{\Gamma}^{(t)}$
 for $l = 1, 2, 3, \dots, L$ **do** // Usually set $L = 1$
 $\mathbf{a} \leftarrow \frac{\boldsymbol{\mu}}{\mathbf{Q}\mathbf{b}}, \mathbf{b} \leftarrow \frac{\boldsymbol{\nu}}{\mathbf{Q}^T \mathbf{a}}$
 end for
 $\boldsymbol{\Gamma}^{(t+1)} \leftarrow \text{diag}(\mathbf{a})\mathbf{Q}\text{diag}(\mathbf{b})$
end for

with nested iterative loops. Exact proximal point method, i.e., solving (8) exactly as shown in Figure 1 (b), provides a feasible solution that is closest to the optimal solution set in each proximal iteration until the optimal solution reached. However, the disadvantage for exact proximal point method is that it's not efficient.

The proposed inexact proximal point method (IPOT) does not solve (8) exactly. Instead, a very small amount of Sinkhorn iteration, e.g., only one iteration, is suggested. The reason for this is three-fold. First, the convergence of Sinkhorn algorithm in each proximal iteration is not required, since it is just intermediate step. Second, usually in numerical optimization, the first a few iterations achieve the most decreasing in the objective function. Performing only the first a few iterations has high cost performance. Last and perhaps the most important, it is observed that IPOT can still converge to an exact solution with small amount of inner iterations².

The algorithm is shown in Algorithm 1. For simplicity we use $\beta = \beta^{(t)}$. Denote $\text{diag}(\mathbf{a})$ the diagonal matrix with a_i as its i -th diagonal elements. Denote \odot as element-wise matrix multiplication and $\frac{(\cdot)}{(\cdot)}$ as element-wise division. We use warm start to improve the efficiency, i.e. in each proximal point iteration, we use the final value of \mathbf{a} and \mathbf{b} from last proximal point iteration as initialization instead of $\mathbf{b}^{(0)} = \mathbf{1}_m$. Later we will show empirically IPOT will converge under a large range of β with $L = 1$, a single inner iteration will suffice.

4 WASSERSTEIN BARYCENTER BY IPOT

We now extend IPOT method to a related problem – computing the Wasserstein barycenter. Wasserstein barycenter

²Similar ideas are also used in accelerating the expectation-maximization algorithm with only one iteration used in the maximization step [18].

is widely used in machine learning and computer vision [5, 25]. Given a set of distributions $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$, their Wasserstein barycenter is defined as

$$\mathbf{q}^*(\mathcal{P}, \boldsymbol{\lambda}) = \arg \min_{\mathbf{q} \in \mathcal{Q}} \sum_{k=1}^K \lambda_k W(\mathbf{q}, \mathbf{p}_k) \quad (9)$$

where \mathcal{Q} is in the space of probability distributions, $\sum_{k=1}^K \lambda_k = 1$, and $W(\mathbf{q}, \mathbf{p}_k)$ is the Wasserstein distance between the barycenter \mathbf{q} and distribution \mathbf{p}_k , which takes the form

$$W(\mathbf{q}, \mathbf{p}_k) = \min_{\boldsymbol{\Gamma}} \langle \mathbf{C}, \boldsymbol{\Gamma} \rangle, \quad \text{s.t.} \quad \boldsymbol{\Gamma} \mathbf{1} = \mathbf{p}_k, \boldsymbol{\Gamma}^T \mathbf{1} = \mathbf{q}. \quad (10)$$

The idea of IPOT method can also be used to compute Wasserstein barycenter. Substitute (10) into (9) and reorganize, we have

$$\begin{aligned} \mathbf{q}^*(\mathcal{P}, \boldsymbol{\lambda}) &= \arg \min_{\mathbf{q} \in \mathcal{Q}} \sum_{k=1}^K \lambda_k \langle \mathbf{C}, \boldsymbol{\Gamma}_k \rangle, \\ \text{s.t.} \quad &\boldsymbol{\Gamma}_k \mathbf{1} = \mathbf{p}_k, \text{ and } \exists \mathbf{q}, \boldsymbol{\Gamma}_k^T \mathbf{1} = \mathbf{q}. \end{aligned}$$

Analogous to IPOT, we take

$$f(\{\boldsymbol{\Gamma}_k\}) = \sum_{k=1}^K \lambda_k \langle \mathbf{C}, \boldsymbol{\Gamma}_k \rangle,$$

take \mathcal{X} to be the corresponding constraints, and take $d(\{\boldsymbol{\Gamma}_k\}, \{\boldsymbol{\Gamma}_k^{(t)}\})$ to be $\sum_{k=1}^K \lambda_k D_h(\boldsymbol{\Gamma}_k, \boldsymbol{\Gamma}_k^{(t)})$. The proximal point iteration for barycenter is

$$\begin{aligned} \boldsymbol{\Gamma}_k^{(t+1)} &= \arg \min_{\boldsymbol{\Gamma}_k} \sum_{k=1}^K \lambda_k \{ \langle \mathbf{C}, \boldsymbol{\Gamma}_k \rangle + \beta^{(t)} D_h(\boldsymbol{\Gamma}_k, \boldsymbol{\Gamma}_k^{(t)}) \} \\ \text{s.t.} \quad &\boldsymbol{\Gamma}_k \mathbf{1} = \mathbf{p}_k, \text{ and } \exists \mathbf{q}, \boldsymbol{\Gamma}_k^T \mathbf{1} = \mathbf{q}. \end{aligned} \quad (11)$$

With further organization, we have

$$\begin{aligned} \boldsymbol{\Gamma}_k^{(t+1)} &= \arg \min_{\boldsymbol{\Gamma}_k} \sum_{k=1}^K \lambda_k \{ \langle \mathbf{C} - \beta^{(t)} \log \boldsymbol{\Gamma}_k^{(t)}, \boldsymbol{\Gamma}_k \rangle \\ &\quad + \beta^{(t)} h(\boldsymbol{\Gamma}_k) \} \\ \text{s.t.} \quad &\boldsymbol{\Gamma}_k \mathbf{1} = \mathbf{p}_k, \text{ and } \exists \mathbf{q}, \boldsymbol{\Gamma}_k^T \mathbf{1} = \mathbf{q}. \end{aligned} \quad (12)$$

On the other hand, analogous to Sinkhorn algorithm, [5] propose *Bregman iterative projection* that seeks to solve an entropy regularized barycenter,

$$\mathbf{q}_\epsilon^*(\mathcal{P}, \boldsymbol{\lambda}) = \arg \min_{\mathbf{q} \in \mathcal{Q}} \sum_{k=1}^K \lambda_k W_\epsilon(\mathbf{q}, \mathbf{p}_k). \quad (13)$$

Comparing (12) and (13), the minimization in each proximal point iteration in (12) can be solved by Bregman iterative projection [5] using the same change-of-variable technique in Section 3.

Algorithm 2 IPOT-WB($\{p_k\}$)

1: **Input:** The probability vector set $\{p_k\}$ on grid $\{y_i\}_{i=1}^n$
2: $\mathbf{b}_k \leftarrow \frac{1}{n} \mathbf{1}_n, \forall k = 1, 2, \dots, K$
3: $C_{ij} \leftarrow c(y_i, y_j) := \|y_i - y_j\|_2^2$
4: $G_{ij} \leftarrow e^{-\frac{C_{ij}}{\beta}}$
5: $\mathbf{\Gamma}_k \leftarrow \mathbf{1}\mathbf{1}^T$
6: **for** $t = 1, 2, 3, \dots$ **do**
7: $\mathbf{H}_k \leftarrow \mathbf{G} \odot \mathbf{\Gamma}_k, \forall k = 1, 2, \dots, K$
8: **for** $l = 1, 2, 3, \dots, L$ **do**
9: $\mathbf{a}_k \leftarrow \frac{\mathbf{q}}{\mathbf{H}_k \mathbf{b}_k}, \forall k = 1, 2, \dots, K,$
10: $\mathbf{b}_k \leftarrow \frac{\mathbf{p}_k}{\mathbf{H}_k^T \mathbf{a}_k}, \forall k = 1, 2, \dots, K$
11: $\mathbf{q} \leftarrow \prod_{k=1}^K (\mathbf{a}_k \odot (\mathbf{H}_k \mathbf{b}_k))^{\lambda_k}$
12: **end for**
13: $\mathbf{\Gamma}_k \leftarrow \text{diag}(\mathbf{a}_k) \mathbf{H}_k \text{diag}(\mathbf{b}_k), \forall k = 1, 2, \dots, K$
14: **end for**
15: **Return** \mathbf{q}

We provide the detailed algorithm in Algorithm 2, and name this algorithm *IPOT-WB*. Same as Algorithms 1, IPOT-WB algorithm can converge with $L = 1$ and a large range of β .

Since the sketch in Figure 1 does not have restrictions on f and \mathcal{X} , the sketch and the corresponding analysis for IPOT also applies to IPOT-WB, except the distance is in sense of convex combination of Bregman divergences instead of a single Bregman divergence.

5 THEORETICAL ANALYSIS

Classical proximal point algorithm has sublinear convergence rate. However, after we replace the square of Euclidean distance in classical proximal point algorithm by Bregman distance, we can prove stronger convergence rate - a linear rate for both IPOT and IPOT-WB. First, we consider when the optimization problem (8) is solved exactly, we have a linear convergence rate guaranteed by the following theorem.

Theorem 5.1. *Let $\{x^{(t)}\}$ be a sequence generated by the proximal point algorithm*

$$x^{(t+1)} = \arg \min_{x \in \mathcal{X}} f(x) + \beta^{(t)} D_h(x, x^{(t)}),$$

where f is continuous and convex. Assume $f^* = \min f(x) > -\infty$. Then, with $\sum_{t=0}^{\infty} \beta^{(t)} = \infty$, we have

$$f(x^{(t)}) \downarrow f^*.$$

If we further assume f is linear and \mathcal{X} is bounded, the algorithm has linear convergence rate.

More importantly, the following theorem gives us a guarantee of convergence when (8) is solved inexactly.

Theorem 5.2. *Let $\{x^{(t)}\}$ be the sequence generated by the Bregman distance based proximal point algorithm with inexact scheme (i.e., finite number of inner iterations are employed). Define an error sequence $\{e^{(t)}\}$ where*

$$e^{(t+1)} \in \beta^{(t)} \left[\nabla f(x^{(t+1)}) + \partial \iota_{\mathcal{X}}(x^{(t+1)}) \right] + \left[\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right],$$

where $\iota_{\mathcal{X}}$ is the indicator function of set \mathcal{X} , and $\partial \iota_{\mathcal{X}}(\cdot)$ is the subdifferential of the indicator function $\iota_{\mathcal{X}}$. If the sequence $\{e^k\}$ satisfies $\sum_{k=1}^{\infty} \|e^k\| < \infty$ and $\sum_{k=1}^{\infty} \langle e^k, x^{(t)} \rangle$ exists and is finite, then $\{x^{(t)}\}$ converges to x^{∞} with $f(x^{\infty}) = f^*$. If the sequence $\{e^{(t)}\}$ satisfies that exist $\rho \in (0, 1)$ such that $\|e^{(t)}\| \leq \rho^t$, $\langle e^{(t)}, x^{(t)} \rangle \leq \rho^t$ and with assumptions that f is linear and \mathcal{X} is bounded, then $\{x^{(t)}\}$ converges linearly.

The proofs of both theorems are given in the supplementary material. Theorem 5.2 guarantees the convergence of inexact proximal point method — as long as the inner iteration number L satisfies the given conditions, IPOT and IPOT-WB algorithm would converge linearly. Note that although Theorem 5.2 manages to prove the linear convergence in inexact case, in practice the conditions is not trivial to verify. In practice we usually just adopt $L = 1$.

Now we know IPOT and IPOT-WB can converge to the exact Wasserstein distance and Wasserstein barycenter. What if an entropic regularization is wanted? Please refer to the supplementary material for how IPOT can achieve regularizations with early stopping.

6 EMPIRICAL ANALYSIS

In this section we will illustrate the convergence behavior with respect to inner iteration number L and parameter β , the scalability of IPOT, and the issue with entropy regularization. We leverage the implementation of Sinkhorn iteration and LP solver based on Python package POT [13], and use Pytorch to parallel some of the implementation.

6.1 Convergence Rate

A simple illustration task of calculating the Wasserstein distance of two 1D distribution is conducted as numerical validation of the convergence theorems proved in Section 5. The two input margins are mixtures of Gaussian distributions shown in the figure in the right lower of Figure 2 (a): the red one is $0.4\phi(\cdot|60, 8) + 0.6\phi(\cdot|40, 6)$, and the blue one is $0.5\phi(\cdot|35, 9) + 0.5\phi(\cdot|70, 9)$, where $\phi(\cdot|\mu, \sigma^2)$ is the probability density function of 1 dimensional Gaussian distribution with mean μ and variance σ^2 . Input vectors μ and ν is the two function values on the

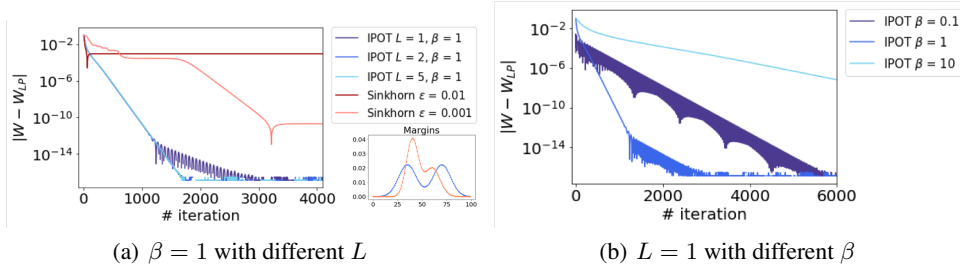


Figure 2: The plots of differences in computed Wasserstein distances w.r.t. number of iterations. Here, W are the Wasserstein distance computed at current iteration. W_{LP} is computed by simplex method, and is used as ground truth. The test adopts $c(x, y) = \|x - y\|_2$. (a) The plot of the convergence trajectories of IPOT with different L . The right lower figure is the two input margins for the test. We also plot the ones for Sinkhorn method in comparison. (b) The plot of differences in computed Wasserstein distances with different β .

uniform discretization of interval $[1, 100]$ with grid size 1. To be clear, the use of two 1D distribution is only for visualization purpose. We also did tests on empirical distribution of 64D Gaussian distributed data, and the result shows the same trend. We include more discussion in the supplementary material.

Figure 2 shows the convergence of IPOT under different L and β . We also include the result of Sinkhorn method for comparison. IPOT algorithm has empirically linear convergence rate even under very small L .

The convergence rate increases w.r.t. β when β is small, and decreases when β is large. This is because the choice of β is a trade-off between inner and outer convergence rates. On the one hand, a smaller β usually lead to quicker convergence of proximal point iterations. On the other hand, the convergence of inner Sinkhorn iteration, is quicker when β is large.

Furthermore, the choice of L also appears to be a trade-off. While a larger L takes more resources in each step, it also achieves a better accuracy, so less proximal point iterations are needed to converge. So the choice of best L is relevant to the choice of β . For large β , the inner Sinkhorn iteration can converge faster, so smaller L should be used. For small β , larger L should be used, which is not efficient, and also improve the risk of underflow for the inner Sinkhorn algorithm. So unless there are specific need for accuracy, we do not recommend using very small β and large L .

For simplicity, we use $L = 1$ for later tests.

6.2 Scalability

We conduct the following scalability test to show the computation time of the proposed IPOT comparing to the state-of-art benchmarks. The optimal transport problem is conducted between the two empirical distributions of 16D

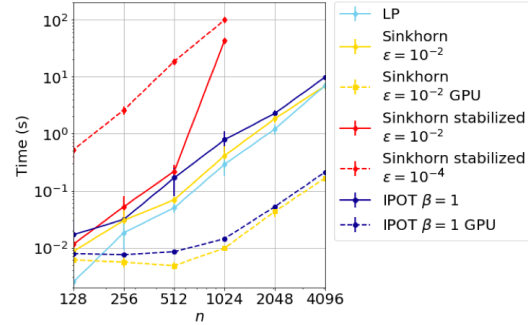


Figure 3: Log-log plot of average time used to achieve $1e-4$ relative precision with error bar. Each point is obtained by the average of 6 tests on different datasets.

uniformly distributed data (See Section 2.1 for formulation). Besides proposed IPOT algorithm (see Algorithm 1 with $L = 1$), the Sinkhorn algorithm follows [8] and the stabilized Sinkhorn algorithm follows [6]. The result of the scalability test is shown in Figure 3. The LP solver has a good performance under the current experiment settings. But LP solver is not guaranteed to have good scalability as shown here. Moreover, LP method is difficult to parallel. Readers who are interested please refer to experiments in [8].

Sinkhorn and IPOT can be paralleled conveniently, so we provide both CPU and GPU tests here. Under this setting, IPOT takes approximately the same resources as Sinkhorn at $\epsilon = 0.01$. For smaller ϵ , original Sinkhorn will underflow, and we need to use stabilized Sinkhorn. Stabilized Sinkhorn is much more expensive than IPOT, especially for large datasets and small ϵ , as demonstrated by the experiment result of stabilized Sinkhorn at $\epsilon = 10^{-2}$ and 10^{-4} .

Note that we also try to use the method proposed in [32]

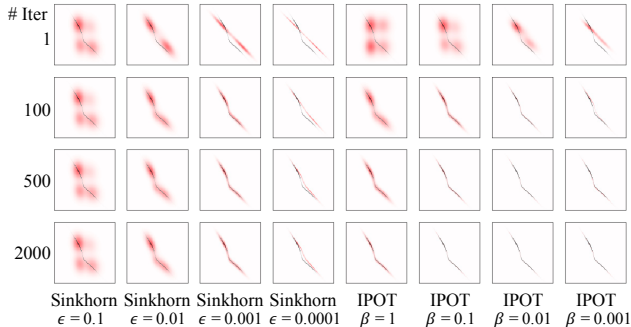


Figure 4: The transport plan generated by Sinkhorn and IPOT methods at different iteration number. The red colormap is the result from Sinkhorn or IPOT method, while the black wire is the result of simplex method for comparison. In the right lower plans, the red and the black is almost identical.

for ϵ scaling, to help the convergence when $\epsilon \rightarrow 0$. However, although it is faster than Sinkhorn method when data size is smaller than 1024, the time used at 1024 is already around 2×10^3 s. Therefore we didn't include this method in the figure.

6.3 Effect of Entropy Regularization

We have shown that IPOT can converge to exact Wasserstein distance with complexity comparable to Sinkhorn (see Figure 1 and 3) and as we claimed in Section 1 this is important in some of the learning problems.

But in what cases is the exact Wasserstein distance truly needed? How will the entropy regularization term affect the result in different applications? In this section, we will discuss the exact transport plan with sparsity preference and the advantage of exact Wasserstein distance in learning the generative models.

6.3.1 Sparsity of the Transport Plan

In applications such as histogram calibration and color transferring, an exact and sparse transport plan is wanted. In this section we conduct tests on the sparsity of the transport plan using the two distributions shown in Figure 2 for both IPOT and Sinkhorn methods with different regularization coefficients. Figure 4 visualize the different transport plans. The red colormap is the result from Sinkhorn or IPOT method, where the black wire beneath is the result by simplex method as ground truth. To be clear, the different number of interaction of IPOT means the number of the outer iteration with still $L = 1$ inner iteration.

The proposed IPOT method can always converge to the sparse ground truth with enough iteration and it is very robust with respect to the parameter β , i.e., there is little

visual difference with β changing from 0.1 to 0.001. Furthermore, even with large $\beta = 1$, the optimal plan is still sparse and acceptable. In addition, if some smoothness is wanted, IPOT method would also be able to work with early stopping. The degree of smoothness can be easily adjusted by adjusting the number of iterations if needed.

On the other hand, the optimal plans obtained by Sinkhorn has two issues. If the ϵ is chosen to be large (i.e., $\epsilon = 0.1$ or 0.01), the optimal plan are blur i.e., neither exact nor sparse. In downstream applications, the non-sparse structure of transport plan make it difficult to extract the transport map from source distribution to target distribution. However if the ϵ is chosen to be small (i.e., $\epsilon = 0.0001$), it needs more iterations to converge. For example, the Sinkhorn $\epsilon = 0.0001$ case still cannot converge after 2000 iterations. So in Sinkhorn applications, ϵ needs to be selected carefully. This fine tuning issue can be avoid by the proposed IPOT method, since IPOT is robust to the parameter β .

6.3.2 Shrinkage Problem in Generative Models

As shown in Equation (2), Sinkhorn method use entropy to penalize the optimization target and has biased evaluation of Wasserstein distance. The inaccuracy will affect the performance of the learning problem where Wasserstein metric is served as loss function.

In order to better illustrate the affect of the inaccurate Wasserstein distance, we consider the task of learning generative models, specifically, Wasserstein GAN [3]. Similar to other GAN, WGAN seeks to learn a generated distribution to approximate a target distribution, except using Wasserstein distance as the loss that measures the distance between the generated distribution and target distribution. It uses the Kantorovitch dual formulation to compute Wasserstein distance.

In this section, we train a Wasserstein GAN with the dual formulation substituted by Sinkhorn and IPOT methods. Detailed derivation can be found in supplementary material. Meanwhile, the standard approach of using dual form proposed in [3] is also compared. Note that the purpose of this section is not to propose a new GAN but visualize how proposed IPOT can avoid the possible negative influence introduced by the inaccuracy of the entropy regularization in the Sinkhorn method.

We claim that result of Sinkhorn method with moderate size ϵ tends to shrink towards the mean, so the learned distribution cannot cover all the support of target distribution. To demonstrate the reason of this trend, consider the extreme condition when $\epsilon \rightarrow \infty$, the loss function becomes

$$\Gamma^* = \arg \min_{\Gamma} h(\Gamma) = \arg \min_{\Gamma} D_h(\Gamma, \mathbf{1}\mathbf{1}^T/mn).$$

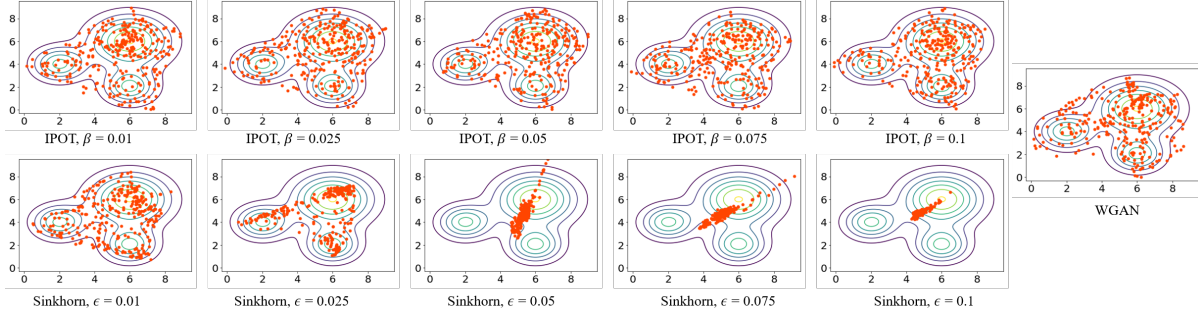


Figure 5: The sequences of learning results using IPOT, Sinkhorn, and original WGAN. In each figure, the orange dots are samples of generated data, while the contour represents the ground truth distribution.

So $\Gamma^* = \mathbf{1}\mathbf{1}^T/mn$. If we view $\{x_i\}$ and $\{y_j : y_j = g_\theta(z_j)\}$ as the realizations of random variables X and Y , the optimal Sinkhorn distance W_ϵ is expected to be

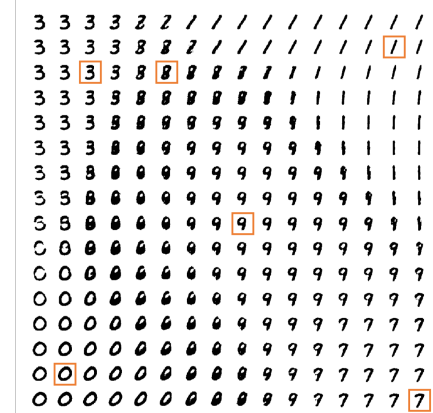
$$\begin{aligned} \mathbb{E}_{X,Y}[W_\epsilon] &= \mathbb{E}_{X,Y}[\langle \Gamma^*, \mathbf{C} \rangle] \\ &= \mathbb{E}_{X,Y}[\sum_{i,j} \|x_i - y_j\|_2^2] \\ &= n^2(\text{Var}(X) + (\bar{X} - \bar{Y})^2 + \text{Var}(Y)), \end{aligned}$$

where n is the data size, $\bar{(\cdot)}$ is the mean of random variable, and $\text{Var}(\cdot)$ is the variance. At the minimum of the distance, the mean of generated data $\{y_j\}$ is the same as $\{x_i\}$, but the variance is zero. Therefore, the learned distribution would shrink asymptotically toward the data mean due to smoothing the effect of regularization.

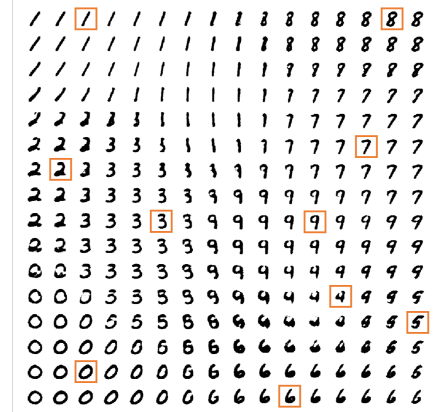
However, the proposed IPOT method is free from the above shrinking issue since the exact Wasserstein distance can be found with the approximately the same cost (see Section 6.1 and Section 6.2). Now we illustrate the shrinkage problem by the following experiments.

Experiments on 2D Synthetic Data First, we conduct a 2D toy example to demonstrate the affect of regularization. We use a 2D-2D NN as generator to learn a mapping from uniformly distributed noise to mixture of Gaussian distributed real data. Since as shown in Figure 2, Sinkhorn may need more iteration to converge, in this experiment, IPOT uses 200 iterations and Sinkhorn uses 500 iterations.

Figure 5 shows the results. As ϵ varies from 0.01 to 0.1, the learned distribution of Sinkhorn gradually shrinks to the mean of target distribution, again this is because the inaccuracy in calculating the Wasserstein distance. On the contrary, since IPOT can converge to the exact Wasserstein distance regardless of different β , the result robustly cover the whole support of target distribution. Furthermore, comparing to the dual form method used in the WGAN, the proposed IPOT method is better in small scale cases and can achieve similar performance in large scale cases [15]. This is mainly because the discriminator neural network used in WGAN is susceptible to overfit-



(a) Sinkhorn $\epsilon = 1$: digits 0,1,3,7,8,9 are covered



(b) IPOT $\beta = 1$: all digits are covered

Figure 6: Plots of MNIST learning result under comparable resources. They both use batch size=200, number of hidden layer=1, number of nodes of hidden layer=500, number of iteration=200, learning rate = 1e-4.

ting in low dimensional cases, and it exceeds the objective of this paper.

Experiments on Higher Dimensional Data For higher dimensional data, we cannot visualize the final generated









































Dataset										
State-of-art										
										
Proposed										

Figure 7: The result of barycenter. For each digit, we randomly choose 8 of 50 scaled and shifted images to demonstrate the input data. From the top to the bottom, we show (top row) the demo of input data; (second row) the results based on [9]; (third row) the result based on [35]; (fourth row) the result based on [5]; (bottom row) the results based on IPOT-WB.

distribution as done in the 2D test. So in order to demonstrate IPOT has little shrinkage issue, we set the latent space to be 2D, and visualize it by plotting the images generated at dense grid points on the latent space. Due to the low dimensional latent space, We perform the experiment using MNIST dataset. Note that this is mainly for the convenience in visualization, the whole shrinkage-free property of IPOT method is also extendable to more complex learning problems. Associated with the MNIST dataset, we use a generator $g_\theta : \mathbb{R}^2 \mapsto \mathbb{R}^{784}$, noise data $\{z_j\} \sim \text{Unif}([0, 1]^2)$ as input, and one fully connected hidden layer with 500 nodes.

Figure 6 shows an example of generated results. The Sinkhorn results look authentic, but we can only find some of the digits in it. This is exactly the consequence of shrinkage due to the inaccurate calculation of Wasserstein distance - in the domain where the density of learned distribution is nonzero, the density of target distribution is usually nonzero; but in some part of the domain where the density of target distribution is nonzero, the learned distribution is zero. In the example of Figure 6 (a), the learned distribution cannot cover the support of digits 2,4,5,6 while when using IPOT to calculate the Wasserstein loss, all ten digits are can be recovered in 6 (b), which shows the coverage of the whole domain of the target distribution. In supplementary material we provide more examples, e.g., if a larger ϵ is used, Sinkhorn generator would shrink to one point, and hence cannot learn anything, while the IPOT method is robust to its parameter β and covers more digits.

6.4 Computing Barycenter

We test our proximal point barycenter algorithm on MNIST dataset, borrowing the idea from [9]. Here, the images in MNIST dataset is randomly uniformly reshape to half to double of its original size, and the reshaped

images have random bias towards corner. After that, the images are mapped into 50×50 grid. For each digit we use 50 of the reshaped images with the same weights as the dataset to compute the barycenter. All results are computed using 50 iterations and under $\epsilon, \beta = 0.001$. So for proximal point method, the regularization is approximately the same as $\epsilon = 2 \times 10^{-5}$, which is pretty small. We compare our method with state-of-art Sinkhorn based methods [9], [35] and [5]. Among the four methods, the convolutional method [35] is different in terms of that it only handles structural input tested here and does not require $O(n^2)$ storage, unlike other three general purpose methods.

We are also aware of that there are other literatures for Wasserstein barycenter, such as [38] and [7], but they are targeting a more complicated setting, and has a different convergence rate (i.e. sublinear rate) than the methods we provide here. The results (Figure 7) from proximal point algorithm are clear, while the results of Sinkhorn based algorithms suffer blurry effect due to entropic regularization. While the time complexity of our method is in the same order of magnitude with Sinkhorn algorithm [5], the space complexity is K times of it, because K different transport maps need to be stored. This might cause pressure to memory for large K . Therefore, a sequential method is needed. We left this to future work.

7 CONCLUSION

We proposed a proximal point method - IPOT - based on Bregman distance to solve optimal transport problem. Different from the Sinkhorn method, IPOT algorithm can converge to ground truth even if the inner optimization iteration only performs once. This nice property results in similar convergence and computation time comparing to Sinkhorn method. However, IPOT provides a robust and accurate computation of Wasserstein distance and associated transport plan, which leads to a better performance in image transformation and avoids the shrinkage in generative models. We also apply the IPOT idea to calculate the Wasserstein barycenter. The proposed method can generate much sharper results than state-of-art due to the exact computation of the Wasserstein distance.

ACKNOWLEDGEMENT

This work is partially supported by the grant NSF IIS 1717916, NSF CMMI 1745382, NSFC 61672231, NSFC U1609220 and 19ZR1414200.

References

- [1] S. Afriat. Theory of maxima and the method of Lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.
- [2] J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. pages 1964–1974, 2017.
- [3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.
- [4] Federico Bassetti, Antonella Bodini, and Eugenio Regazzini. On minimum kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302, 2006.
- [5] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.
- [6] L. Chizat, G. Peyré, B. Schmitzer, and F. Vialard. Scaling algorithms for unbalanced transport problems. *Mathematics of Computation*, 2018.
- [7] Sebastian Clatici, Edward Chien, and Justin Solomon. Stochastic wasserstein barycenters. *arXiv preprint arXiv:1802.05757*, 2018.
- [8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.
- [9] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.
- [10] P. Dvurechensky, S. Omelchenko, and A. Tiurin. Adaptive similar triangles method: a stable alternative to Sinkhorn’s algorithm for regularized optimal transport. *arXiv preprint arXiv:1706.07622*, 2017.
- [11] J. Eckstein. Nonlinear proximal point algorithms using bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.
- [12] J. Eckstein. Approximate iterations in bregman-function-based proximal algorithms. *Mathematical programming*, 83(1-3):113–123, 1998.
- [13] R. Flamary and N. Courty. POT Python Optimal Transport Library. 2017.
- [14] J. Franklin and J. Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114:717–735, 1989.
- [15] A. Genevay, G. Peyré, and M. Cuturi. Sinkhorn-AutoDiff: Tractable Wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*, 2017.
- [16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [17] L. Kantorovich. On mass transfer problem. *Doklady Akademii Nauk SSSR*37, pages 199–201, 1942.
- [18] Kenneth Lange. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):425–437, 1995.
- [19] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (TOG)*, 36(4):39, 2017.
- [20] M. Martinez, M. Haurilet, Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen. Relaxed earth mover’s distances for chain-and tree-connected spaces and their use as a loss function in deep learning. *arXiv preprint arXiv:1611.07573*, 2016.
- [21] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.
- [22] O. Pele and M. Werman. Fast and robust earth mover’s distances. In *IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, 2009.
- [23] B. Polyak. *Introduction to optimization*. Optimization Software Inc., New York, 1987.
- [24] J. Rabin, S. Ferradans, and N. Papadakis. Adaptive color transfer with relaxed optimal transport. In *IEEE International Conference on Image Processing*, pages 4852–4856. IEEE, 2014.
- [25] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011.

- [26] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.
- [27] Stephen M Robinson. Some continuity properties of polyhedral multifunctions. In *Mathematical Programming at Oberwolfach*, pages 206–214. Springer, 1981.
- [28] R. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.
- [29] R. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.
- [30] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *The Sixth International Conference on Computer Vision*, pages 59–66. IEEE, 1998.
- [31] M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems*, pages 1458–1466, 2011.
- [32] B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *arXiv preprint arXiv:1610.06519*, 2016.
- [33] A. M.-C. So and Z. Zhou. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *Optimization Methods and Software*, 32(4):963–992, 2017.
- [34] M. Solodov and B. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22(7-8):1013–1035, 2001.
- [35] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and Leonidas L. Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):66, 2015.
- [36] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, pages 306–314, 2014.
- [37] S. Srivastava, V. Cevher, Q. Dinh, and D. Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pages 912–920, 2015.
- [38] M. Staib, S. Claiici, J. Solomon, and S. Jegelka. Parallel streaming Wasserstein barycenters. *arXiv preprint arXiv:1705.07443*, 2017.
- [39] Marc Teboulle. Entropic proximal mappings with applications to nonlinear programming. *Mathematics of Operations Research*, 17(3):670–690, 1992.
- [40] A. Thibault, L. Chizat, C. Dossal, and N. Papadakis. Overrelaxed Sinkhorn-Knopp algorithm for regularized optimal transport. *arXiv preprint arXiv:1711.01851*, 2017.
- [41] M. Thorpe, S. Park, S. Kolouri, G. Rohde, and D. Slepčev. A transportation L^p distance for signal analysis. *Journal of Mathematical Imaging and Vision*, 59(2):187–210, 2017.
- [42] X. Xiao and L. Ma. Color transfer in correlated color space. In *ACM international Conference on Virtual Reality Continuum and its Applications*, pages 305–309. ACM, 2006.
- [43] J. Ye, P. Wu, J. Wang, and J. Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

A More Analysis on IPOT

A.1 Convergence w.r.t. L

As mentioned in Section 6.1, we provide the test result of 64D Gaussian distributed data here. We choose the computed Wasserstein distance $\langle \Gamma, \mathbf{C} \rangle$ as the indicator of convergence, because while the optimal transport plan might not be unique, the computed Wasserstein distance at convergence must be unique and minimized to ground truth. We use the empirical distribution as input distributions, i.e.,

$$\begin{aligned} W(\{x_i\}, \{g_\theta(z_j)\}) &= \min_{\Gamma} \langle \mathbf{C}(\theta), \Gamma \rangle \\ \text{s.t. } \Gamma \mathbf{1}_n &= \frac{1}{n} \mathbf{1}_n, \Gamma^T \mathbf{1}_n = \frac{1}{n} \mathbf{1}_n. \end{aligned} \quad (14)$$

As shown in Figure 8, the convergence rate is also linear. For comparison, we also provide the convergence path of Sinkhorn iteration. The result cannot converge to ground truth because the method is essentially regularized.

Remark. When we are talking about amount of regularization, usually we are referring to the magnitude of ϵ for Sinkhorn, or the equivalent magnitude of ϵ computed from remark in Section 3 for IPOT method. However, the amount of regularization in a loss function should be quantified by $\epsilon/\|\mathbf{C}\|$, instead of ϵ alone. That is why in this paper, different magnitude of ϵ is used for different application.

A.2 How IPOT Avoids Instability

Heuristically, if Sinkhorn does not underflow, with enough iteration, the result of IPOT is approximately the same as Sinkhorn with $\epsilon^{(t)} = \beta/t$. The difference lies in IPOT is a principled way to avoid underflow and can converge to arbitrarily small regularization, while Sinkhorn always causes numerical difficulty when $\epsilon \rightarrow 0$, even with scheduled decreasing ϵ like [6]. More specifically, in IPOT, we can factor $\Gamma = \text{diag}(\mathbf{u}_1) \mathbf{G}^t \text{diag}(\mathbf{u}_2)$, where $(\cdot)^t$ is element-wise exponent operation, and \mathbf{u}_1 and \mathbf{u}_2 are two scaling vectors. So we have $\epsilon^{(t)} = \beta/t$. As t goes infinity, all entries of \mathbf{G}^t would underflow if we use Sinkhorn with $\epsilon^{(t)} = \beta/t$. But we know Γ^* is neither all zeros nor contains infinity. So instead of computing \mathbf{G}^t , \mathbf{u}_1 and \mathbf{u}_2 directly, we use Γ^t to record the multiplication of \mathbf{G}^t with part of \mathbf{u}_1 and \mathbf{u}_2 in each step, so the entries of Γ^t will not over/underflow. The explicit computation of \mathbf{G}^t is not needed.

Therefore, by tuning β and iteration number, we can achieve the result of arbitrary amount of regularization with IPOT.

B Learning Generative Models

In this section, we show the derivation for the learning algorithm, and more tests result.

For simplicity, we assume $|\{x_i\}| = |\{z_j\}| = n$. Given a dataset $\{x_i\}$ and some noise $\{z_j\}$ [4, 16], our goal is to find a

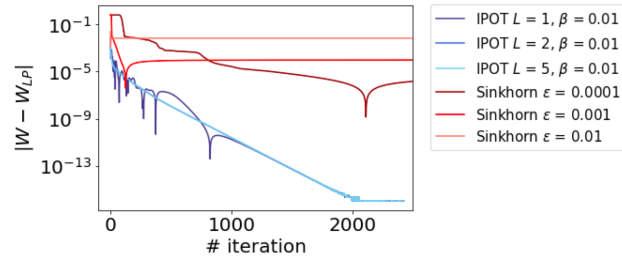


Figure 8: The plot of differences in computed Wasserstein distances w.r.t. number of iterations for 64D Gaussian distributed data. Here, W are the Wasserstein distance computed at current iteration. W_{LP} is computed by simplex method, and is used as ground truth. The test adopts $c(x, y) = \|x - y\|_2$. Due to random data is used, the number of iteration that the algorithm reaches 10^{-17} varies from 1000 to around 5000 according to our tests.

parameterized function $g_\theta(\cdot)$ that minimize $W(\{x_i\}, \{g_\theta(z_j)\})$,

$$\begin{aligned} W(\{x_i\}, \{g_\theta(z_j)\}) &= \min_{\Gamma} \langle \mathbf{C}(\theta), \Gamma \rangle \\ \text{s.t. } \Gamma \mathbf{1}_n &= \frac{1}{n} \mathbf{1}_n, \Gamma^T \mathbf{1}_n = \frac{1}{n} \mathbf{1}_n, \end{aligned} \quad (15)$$

where $\mathbf{C}(\theta) = [c(x_i, g_\theta(z_j))]$. Usually, g_θ is parameterized by a neural network with parameter θ , and the minimization over θ is done by stochastic gradient descent.

In particular, given current estimation θ , we can obtain optimum Γ^* by IPOT, and compute the Wasserstein distance by $\langle \mathbf{C}(\theta), \Gamma^* \rangle$ accordingly. Then, we can further update θ by the gradient of current Wasserstein distance. There are two ways to solve the gradient: One is auto-diff based method such as [15], the other is based on the envelope theorem [1]. Different from the auto-diff based methods, the back-propagation based on envelope theorem does not go into proximal point iterations because the derivative over Γ^* is not needed, which accelerates the learning process greatly. This also has significant implications numerically because the derivative of a computed quantity tends to amplify the error. Therefore, we adopt envelope based method.

Theorem B.1. Envelope theorem. *Let $f(x, \theta)$ and $l(x)$ be real-valued continuously differentiable functions, where $x \in \mathbb{R}^n$ are choice variables and $\theta \in \mathbb{R}^m$ are parameters. Denote x^* to be the optimal solution of f with constraint $l = 0$ and fixed θ , i.e.*

$$x^* = \arg \min_x f(x, \theta) \quad \text{s.t.} \quad l(x) = 0.$$

Then, assume that V is continuously differentiable function defined as $V(\theta) \equiv f(x^*(\theta), \theta)$, the derivative of V over parameters is

$$\frac{\partial V(\theta)}{\partial \theta} = \frac{\partial f}{\partial \theta}.$$

In our case, because Γ^* is the minimization of $\langle \Gamma, \mathbf{C}(\theta) \rangle$ with constraints, we have

$$\begin{aligned} \frac{\partial W(\{x_i\}, \{g_\theta(z_j)\})}{\partial \theta} &= \frac{\partial \langle \Gamma^*, \mathbf{C}(\theta) \rangle}{\partial \theta} \\ &= \langle \Gamma^*, \frac{\partial \mathbf{C}(\theta)}{\partial \theta} \rangle = \langle \Gamma^*, 2(g_\theta(z_j) - x_i) \frac{\partial g_\theta(z_j)}{\partial \theta} \rangle, \end{aligned}$$

where we assume $C_{ij}(\theta) = \|x_i - g_\theta(z_j)\|_2^2$, but the algorithm can also adopt other metrics. The derivation is in supplementary materials. The flowchart is shown in Figure 9, and the algorithm is shown in Algorithm 3.

Note Sinkhorn distance is defined as $S(\{x_i\}, \{g_\theta(z_j)\}) = \langle \mathbf{C}(\theta), \Gamma^* \rangle$, where $\Gamma^* = \arg \min_{\Gamma \in \Sigma(\frac{1}{n}, \frac{1}{n})} \langle \mathbf{C}(\theta), \Gamma \rangle + \epsilon h(\Gamma)$. If Sinkhorn distance is used in learning generative models, envelope theorem cannot be used because the loss function for optimizing θ and Γ is not the same.

In the tests, we observe the method in [15] suffers from shrinkage problem, i.e. the generated distribution tends to shrink towards the target mean. The recovery of target distribution is sensitive to the weight of regularization term ϵ . Only relatively small ϵ can lead to a reasonable generated distribution.

Algorithm 3 Learning generative networks

Input: real data $\{x_i\}$, initialized generator g_θ
while not converged **do**
 Sample a batch of real data $\{x_i\}_{i=1}^n$
 Sample a batch of noise data $\{z_j\}_{i=1}^n \sim q$
 $C_{ij} := c(x_i, g_\theta(z_j)) := \|x_i - g_\theta(z_j)\|_2^2$
 $\Gamma = \text{IPOT}(\frac{1}{n} \mathbf{1}_n, \frac{1}{n} \mathbf{1}_n, \mathbf{C})$
 Update θ with $\langle \Gamma, [2(x_i - g_\theta(z_j)) \frac{\partial g_\theta(z_j)}{\partial \theta}] \rangle$
end while

B.1 Synthetic Test

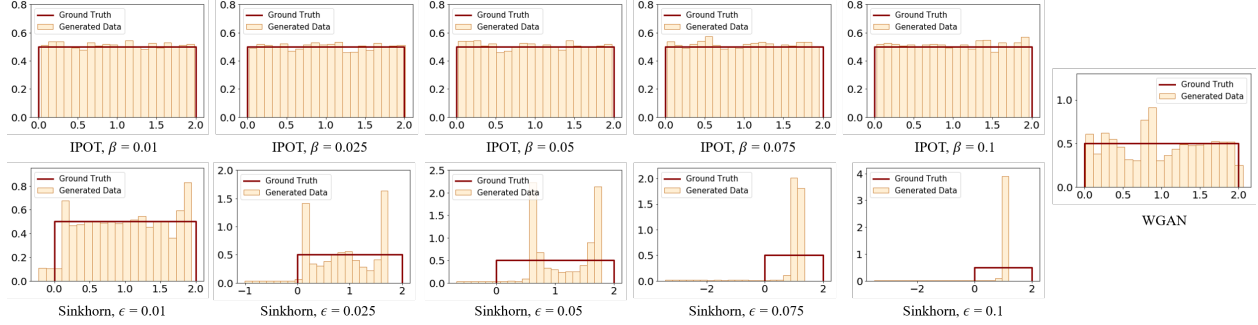


Figure 10: The sequences of learning result of IPOT, Sinkhorn. In each figure, the orange histogram is the histogram of generated data, while the red line represents the PDF of the ground truth of target distribution.

In section 5.1, we show the learning result of Sinkhorn and IPOT in 2D case. In Figure 10 we show sequences of results for a 1D-1D generator, respectively. The upper sequence is IPOT with $\beta = 0.01, 0.025, 0.05, 0.075, 0.1$. The results barely change w.r.t. β . The lower sequence is the corresponding Sinkhorn results. The results shrink to the mean of target data, as expected. Also, we observe the learned distribution tends to have a tail that is not in the range of target data (also in 2D result, we do not include that part for a better view). It might be because the range of support that has a small probability has very small gradient when updated. Once the distribution is initialized to have a tail with small probability, it can hardly be updated. But this theory cannot explain why larger ϵ corresponds to longer tails. The tails can be on the left or right. We pick the ones on the left for easier comparison.

B.2 MNIST Test

The same shrinkage can be observed in MNIST data as well. See figure 11. While $\epsilon = 0.1$ covers most shapes of the numbers, $\epsilon = 1$ only covers a fraction, and $\epsilon = 10$ seems to cover only the mean of images.

C Color Transferring

Optimal transport is directly applicable to many applications, such as color transferring and histogram calibration. We will show the result of color transferring and why accurate transport plan is superior to entropically regularized ones.

The goal of color transferring is to transfer the tonality of a target image into a source image. This is usually done by imposing the histogram of the color palette of one image to another image. Since Reinhard et al. [26], many methods [24, 42] are developed to do so by learning the transformation between the two histograms. Experiments in [30] have shown that transformation based on optimal transport map outperforms state-of-the-art techniques for challenging images.

Same as other prime-form Wasserstein distance solvers [22, 8], the proximal point method provide a transport plan. By definition, the plan is a transport from the source distribution to a target one with minimum cost. Therefore it provides a way to transform a histogram to another.

One example is shown in figure 12. We use three different maps to transform the RGB channels, respectively. For each channel, there are at most 256 bins. Therefore, using three channels separately is more efficient than treating the colors

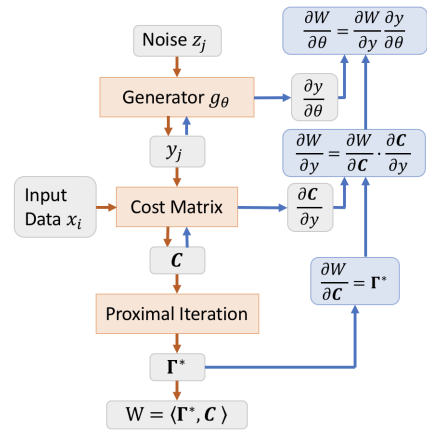


Figure 9: The architecture of the learning model using Envelope theorem in detail. According to Envelope theorem, we do not need to compute $\frac{\partial W}{\partial \Gamma^*}$, so we do not need to back-propagate into the iteration.

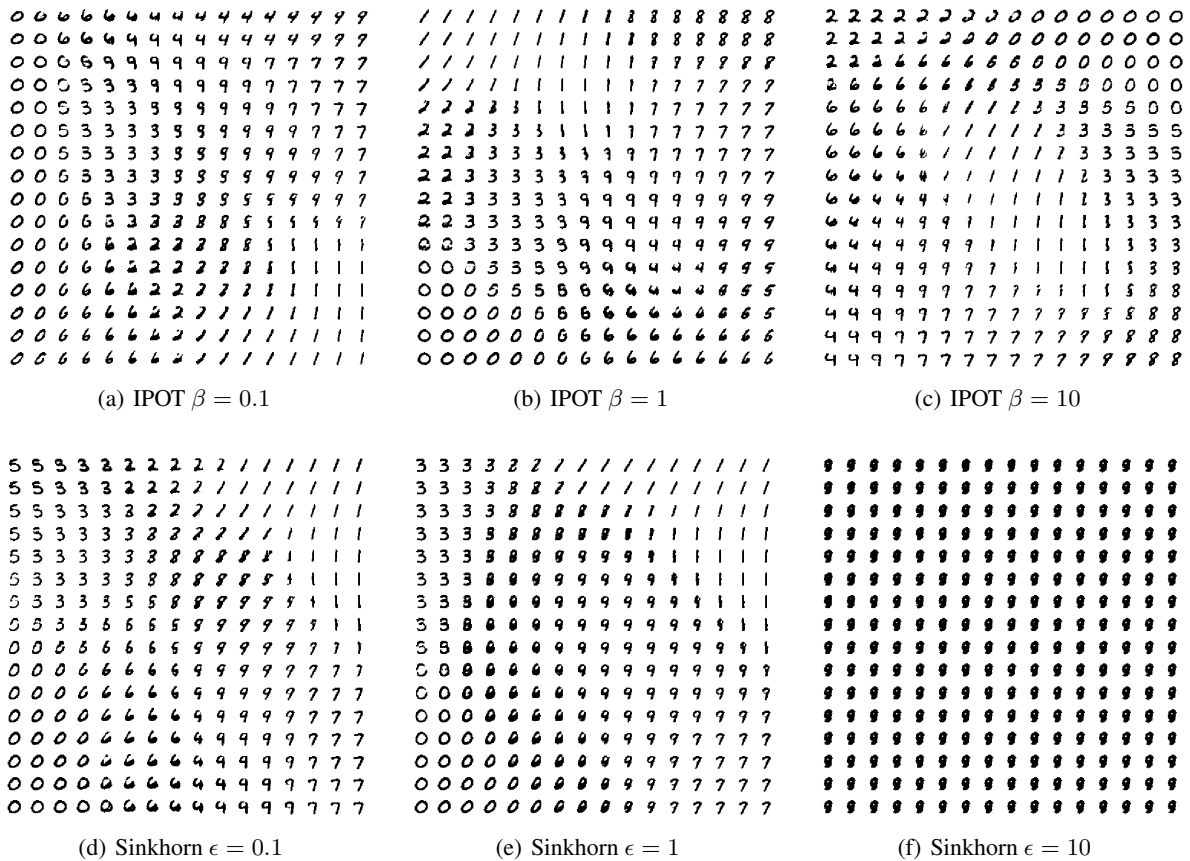


Figure 11: Plots of MNIST learning result under comparable resources with different ϵ . They both use batch size=200, number of hidden layer=1, number of nodes of hidden layer=500, number of iteration=500, learning rate = $1e-4$. Note that despite we show result of $\epsilon = 0.1$ here, the algorithm does not run stably. It would sometimes fail due to numerical issue.

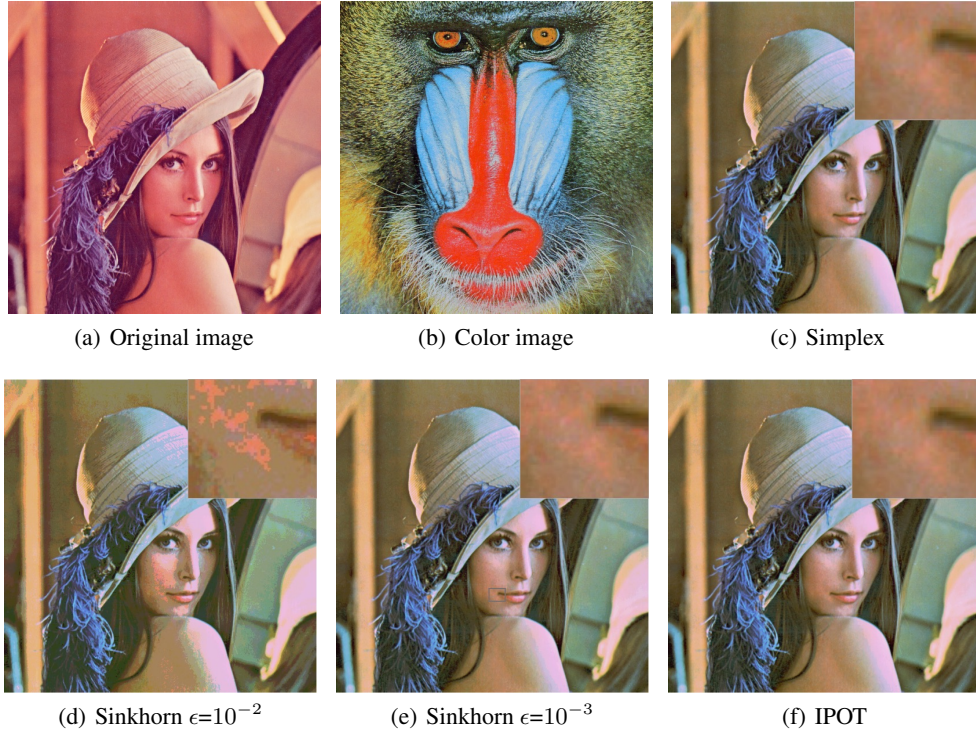


Figure 12: An example of color transferring. The right upper corner of each generated image shows the zoom-in of the color detail of the mouth corner.

as 3D data. Figure 12 shows proximal point method can produce identical result as linear programming at convergence, while the results produced by Sinkhorn method differ w.r.t. ϵ .

D General Bregman Proximal Point Algorithm

In the main body of the paper, we discussed the proximal point algorithm with specific Bregman distance, which is generated through the traditional entropy function. In this section, we generalize our results by proving the effectiveness of proximal point algorithm with general Bregman distance. Bregman distance is applied to measure the discrepancy between different matrices which turns out to be one of the key ideas in regularized optimal transport problems. Its special structure also give rise to proximal-type algorithms and projectors in solving optimization problems.

D.1 Basic Algorithm Framework and Preliminaries

The fundamental iterative scheme of general Bregman proximal point algorithm can be denoted as

$$x^{(t+1)} = \arg \min_{x \in X} \left\{ f(x) + \beta^{(t)} D_h(x, x^{(t)}) \right\}, \quad (16)$$

where $t \in \mathbb{N}$ is the index of iteration, and $D_h(x, x^{(t)})$ denotes a general Bregman distance between x and $x^{(t)}$ based on a Legendre function h (The definition is presented in the following). In the main body of the paper, h is specialized as the classical entropy function and as follows the related Bregman distance reduces to the generalized KL divergence. Furthermore, the Sinkhorn-Knopp projection can be introduced to compute each iterative subproblem. In the following, we present some fundamental definitions and lemmas.

Definition D.1. *Legendre function:* Let $h : X \rightarrow (-\infty, \infty]$ be a lsc proper convex function. It is called

1. Essentially smooth: if h is differentiable on $\text{int dom } h$, with moreover $\|\nabla h(x^{(t)})\| \rightarrow \infty$ for every sequence $\{x^{(t)}\} \subset \text{int dom } h$ converging to a boundary point of $\text{dom } h$ as $t \rightarrow +\infty$;

2. Legendre type: if h is essentially smooth and strictly convex on $\text{int dom } h$.

Definition D.2. Bregman distance: any given Legendre function h ,

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle, \quad \forall x \in \text{dom } h, \forall y \in \text{int dom } h, \quad (17)$$

where D_h is strictly convex with respect to its first argument. Moreover, $D_h(x, y) \geq 0$ for all $(x, y) \in \text{dom } h \times \text{int dom } h$, and it is equal to zero if and only if $x = y$. However, D_h is in general asymmetric, i.e., $D_h(x, y) \neq D_h(y, x)$.

Definition D.3. Symmetry Coefficient: Given a Legendre function $h : X \rightarrow (-\infty, \infty]$, its symmetry coefficient is defined by

$$\alpha(h) = \inf \left\{ \frac{D_h(x, y)}{D_h(y, x)} \mid (x, y) \in \text{int dom } h \times \text{int dom } h, x \neq y \right\} \in [0, 1]. \quad (18)$$

Lemma D.4. Given $h : X \rightarrow (-\infty, +\infty]$, D_h is general Bregman distance, and $x, y, z \in X$ such that $h(x), h(y), h(z)$ are finite and h is differentiable at y and z ,

$$D_h(x, z) - D_h(x, y) - D_h(y, z) = \langle \nabla h(y) - \nabla h(z), x - y \rangle \quad (19)$$

Proof. The proof is straightforward as one can easily verify it by simply subtracting $D_h(y, z)$ and $D_h(x, y)$ from $D_h(x, z)$. \square

D.2 Theorem 5.1 and Theorem 5.2

In this section, we first establish the convergence of Bregman proximal point algorithm, i.e., **Theorem 5.1**, while our analysis is based on ([11, 12, 39]). Further, we establish the convergence of inexact version Bregman proximal point algorithm, i.e., **Theorem 5.2**, in which the subproblem in each iteration is computed inexactly within finite number of sub-iterations.

Note that here for simplicity we provide proof of $d(\Gamma, \Gamma^{(t)}) = D_h(\Gamma, \Gamma^{(t)})$, i.e., the IPOT case. We can analogously prove it for $d(\{\Gamma_k\}, \{\Gamma_k^{(t)}\}) = \sum_{k=1}^K \lambda_k D_h(\Gamma_k, \Gamma_k^{(t)})$, i.e., the IPOT-WB case, with very similar proof. This is because the latter is essentially just the weighted version of the former.

Before proving both theorems, we propose several fundamental lemmas. The first Lemma is the fundamental descent lemma, which is popularly used to analysis the convergence result of first-order methods.

Lemma D.5. (Descent Lemma) Consider a closed proper convex function $f : X \rightarrow (-\infty, \infty]$ and for any $x \in X$ and $\beta^{(t)} > 0$, we have:

$$f(x^{(t+1)}) \leq f(x) + \beta^{(t)} \left[D_h(x, x^{(t)}) - D_h(x, x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) \right], \quad \forall x \in X. \quad (20)$$

Proof. The optimality condition of (16) can be written as

$$\left(x - x^{(t+1)} \right)^T \left[\nabla f(x^{(t+1)}) + \beta^{(t)} \left(\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right) \right] \geq 0, \quad \forall x \in X.$$

Then with the convexity of f , we obtain

$$f(x) - f(x^{(t+1)}) + \beta^{(t)} \left(x - x^{(t+1)} \right)^T \left(\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right) \geq 0. \quad (21)$$

With (19) it follows that

$$\left(x - x^{(t+1)} \right)^T \left(\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right) = D_h(x, x^{(t)}) - D_h(x, x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}).$$

Substitute the above equation into (21), we have

$$f(x^{(t+1)}) \leq f(x) + \beta^{(t)} \left[D_h(x, x^{(t)}) - D_h(x, x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) \right], \quad \forall x \in X.$$

\square

Next, we prove the convergence result in **Theorem 5.1**.

Theorem 5.1 *Let $\{x^{(t)}\}$ be the sequence generated by the general Bregman proximal point algorithm with iteration (16) where f is assumed to be continuous and convex. Further assume that $f^* = \min f(x) > -\infty$. Then we have that $\{f(x^{(t)})\}$ is non-increasing, and $f(x^{(t)}) \rightarrow f^*$. Further assume there exists η , s.t.*

$$f^* + \eta d(x) \leq f(x), \quad \forall x \in X, \quad (22)$$

The algorithm has linear convergence.

Proof. 1. First, we prove the sufficient decrease property:

$$f(x^{(t+1)}) \leq f(x^{(t)}) - \beta^{(t)}(1 + \alpha(h))D_h(x^{(t+1)}, x^{(t)}). \quad (23)$$

Let $x = x^{(t)}$ in (20), we obtain

$$\begin{aligned} f(x^{(t+1)}) &\leq f(x^{(t)}) - \beta^{(t)} \left[D_h(x^{(t)}, x^{(t+1)}) + D_h(x^{(t+1)}, x^{(t)}) \right] \\ &\leq f(x^{(t)}) - \beta^{(t)}(1 + \alpha(h))D_h(x^{(t+1)}, x^{(t)}). \end{aligned}$$

With the sufficient decrease property, it is obvious that $\{f(x^{(t)})\}$ is non-decreasing.

2. Summing (23) from $i = 0$ to $i = t - 1$ and for simplicity assuming $\beta^{(t)} = \beta$, we have

$$\begin{aligned} \sum_{i=0}^{t-1} \left[\frac{1}{\beta^{(i)}} \left(f(x^{(i+1)}) - f(x^{(i)}) \right) \right] &\leq - [1 + \alpha(h)] \sum_{i=0}^{t-1} D_h(x^{(i+1)}, x^{(i)}) \\ \Rightarrow \sum_{i=0}^{\infty} D_h(x^{(i+1)}, x^{(i)}) &< \frac{1}{\beta(1 + \alpha(h))} f(x^{(0)}) < \infty, \end{aligned}$$

which indicates that $D_h(x^{(i+1)}, x^{(i)}) \rightarrow 0$. Then summing (20) from $i = 0$ to $i = t - 1$, we have

$$k \left(f(x^{(t)}) - f(x) \right) \leq \sum_{i=0}^{t-1} \left(f(x^{(i+1)}) - f(x) \right) \leq \beta D_h(x, x^{(0)}) < \infty, \quad \forall x \in X.$$

Let $t \rightarrow \infty$, we have $\lim_{t \rightarrow \infty} f(x^{(t)}) \leq f(x)$ for every x , as a result we have $\lim_{k \rightarrow \infty} f(x^{(t)}) = f^*$.

3. Finally, we prove the convergence rate is linear. Assume $x^* = \arg \min_x f(x)$ is the unique optimal solution. Denote $d(x) = D_h(x^*, x)$. Let also $\beta^{(t)} = \beta$, we will prove

$$\frac{d(x^{(t+1)})}{d(x^{(t)})} \leq \frac{1}{1 + \frac{\eta}{\beta}} \quad (24)$$

Replace x with x^* in inequality (20), we have

$$f(x^{(t+1)}) \leq f^* + \beta \left[d(x^{(t)}) - d(x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) \right]. \quad (25)$$

Using assumption 22, we have

$$f^* + \eta d(x^{(t+1)}) \leq f(x^{(t+1)}) \quad (26)$$

Sum 25 and 26 up, we have

$$\begin{aligned} \frac{\eta}{\beta} d(x^{(t+1)}) &\leq d(x^{(t)}) - d(x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) \\ &\leq d(x^{(t)}) - d(x^{(t+1)}) \end{aligned}$$

Therefore,

$$\frac{d(x^{(t+1)})}{d(x^{(t)})} \leq \frac{1}{1 + \frac{\eta}{\beta}}$$

Therefore, we have a linear convergence in Bregman distance sense.

□

Assumption (22) does not always hold when f is linear. In our specific case, x is bounded in $[0, 1]^{m \times n}$. More rigorously, we can prove the following lemma.

Lemma D.6. *Assume \mathcal{X} is a bounded polyhedron, x^* is unique, $d(x)$ is an arbitrary nonnegative convex function with $d(x^*) = 0$. If f is linear, then there exist η , s.t. $f^* + \eta d(x) \leq f(x)$.*

Proof. Since \mathcal{X} is a bounded polyhedron, any $x \in \mathcal{X}$ can be expressed as $x = \sum_{i=0}^n \lambda_i e_i$, where e_i is the vertices of \mathcal{X} , n is finite, and $\sum \lambda_i = 1$. Also f is linear, so $f(x) = \sum_{i=0}^n \lambda_i f(e_i)$

Since f is linear, \mathcal{X} is polyhedral and x^* is unique, x^* is a vertex of X . Denote $e_0 = x^*$.

Denote $\delta = \min_{i>0} f(e_i) - f^*$, then $\delta > 0$, or else x^* is not unique. Denote $d_{max} = \max_{i>0} d(e_i)$. Take

$$\eta = \delta/d_{max},$$

we have

$$\begin{aligned} f^* + \eta d(x) &= f^* + \eta d\left(\sum_{i=0}^n \lambda_i e_i\right) \\ \text{(Jensen's Inequality)} &\leq f^* + \eta \sum_{i=0}^n \lambda_i d(e_i) \\ (d(e_0) = 0) &\leq f^* + (1 - \lambda_0) \eta d_{max} \\ &= f^* + (1 - \lambda_0) \delta \\ &= \sum_{i=1}^n \lambda_i (f^* + \delta) + \lambda_0 f^* \\ &\leq \sum_{i=0}^n \lambda_i f(e_i) \\ &= f(x). \end{aligned}$$

□

For more general cases, if x^* is not unique, we can divide the vertices as optimal vertices and the rest vertices, instead of e_0 and the rest as above, the conclusion can be proved analogously. Furthermore, if \mathcal{X} is not a polyhedron, as long as \mathcal{X} is bounded, we can always prove the conclusion in a polyhedron \mathcal{A} s.t. $\mathcal{X} \in \mathcal{A}$ and x^* is also the optimal solution of $\min_{x \in \mathcal{A}} f(x)$. Proof of more general cases can be found in [27] (This paper points out some fairly strong continuity properties that polyhedral multifunctions satisfy).

Inequality (24) shows how the convergence rate is linked to β . This is the reason we claim in Section 4.1 that a smaller β would lead to quicker convergence in exact case.

From above, we showed that the general Bregman proximal point algorithm with constant step size can guarantee convergence to the optimal solution f^* , and has linear convergence rate with some assumptions. Further, we prove the convergence result for the general Bregman proximal point algorithm with inexact scheme in **Theorem 5.2**.

Theorem 5.2 *Let $\{x^{(t)}\}$ be the sequence generated by the general Bregman proximal point algorithm with inexact scheme (i.e., finite number of inner iterations are employed). Define an error sequence $\{e^{(t)}\}$ where*

$$e^{(t+1)} \in \beta^{(t)} \left[\nabla f(x^{(t+1)}) + \partial \iota_X(x^{(t+1)}) \right] + \left[\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right], \quad (27)$$

where ι_X is the indicator function of set X . If the sequence $\{e^{(t)}\}$ satisfies $\sum_{k=1}^{\infty} \|e^{(k)}\| < \infty$ and $\sum_{k=1}^{\infty} \langle e^{(k)}, x^{(k)} \rangle$ exists and is finite, then $\{x^{(t)}\}$ converges to x^∞ with $f(x^\infty) = f^*$. If the sequence $\{e^{(t)}\}$ satisfies that exist $\rho \in (0, 1)$ such that $\|e^{(t)}\| \leq \rho^t$, $\langle e^{(t)}, x^{(t)} \rangle \leq \rho^t$ and with assumption (22), then $\{x^{(t)}\}$ converges linearly.

Remark: If exact minimization is guaranteed in each iteration, the sequence $\{x^{(t)}\}$ will satisfy that

$$0 \in \beta^{(t)} \left[\nabla f(x^{(t+1)}) + \partial \iota_X(x^{(t+1)}) \right] + \frac{1}{\beta^{(t)}} \left[\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right].$$

As a result, with enough inner iteration, the guaranteed $e^{(t)}$ will goes to zero.

Proof. This theorem is extended from [12, Theorem 1], and we propose a brief proof here. The proof contains the following four steps:

1. We have for all $k \geq 0$, through the three point lemma

$$D_h(x, x^{(t+1)}) = D_h(x, x^{(t)}) - D_h(x^{(t+1)}, x^{(t)}) - \langle \nabla h(x^{(t)}) - \nabla h(x^{(t+1)}), x^{(t+1)} - x \rangle,$$

which indicates

$$D_h(x, x^{(t+1)}) = D_h(x, x^{(t)}) - D_h(x^{(t+1)}, x^{(t)}) - \langle \nabla h(x^{(t)}) - \nabla h(x^{(t+1)}) + e^{(t+1)}, x^{(t+1)} - x \rangle + \langle e^{(t+1)}, x^{(t+1)} - x \rangle.$$

Since $\frac{1}{\beta^{(t)}} \left[e^{(t+1)} + \nabla h(x^{(t)}) - \nabla h(x^{(t+1)}) \right] \in \nabla f(x^{(t+1)}) + \partial \iota_X(x^{(t+1)})$ and $0 \in \nabla f(x^*) + \partial \iota_X(x^*)$ if x^* be the optimal solution, we have

$$\begin{aligned} & \langle \nabla h(x^{(t)}) - \nabla h(x^{(t+1)}) + e^{(t+1)}, x^{(t+1)} - x^* \rangle \\ &= \beta^{(t)} \left\langle \left[\frac{1}{\beta^{(t)}} \left(\nabla h(x^{(t)}) - \nabla h(x^{(t+1)}) + e^{(t+1)} \right) \right] - 0, x^{(t+1)} - x^* \right\rangle \geq 0, \end{aligned}$$

because $\nabla f + \partial \iota_X$ is monotone ($f + \iota_X$ is convex). Further we have

$$D_h(x^*, x^{(t+1)}) \leq D_h(x^*, x^{(t)}) - D_h(x^{(t+1)}, x^{(t)}) + \langle e^{(t+1)}, x^{(t+1)} - x^* \rangle.$$

2. Summing the above inequality from $i = 0$ to $i = t - 1$, we have

$$D_h(x^*, x^{(t)}) \leq D_h(x^*, x^{(0)}) - \sum_{i=0}^{t-1} D_h(x^{(i+1)}, x^{(i)}) + \sum_{i=0}^{t-1} \langle e^{(i+1)}, x^{(i+1)} - x^* \rangle.$$

Since $\sum_{t=1}^{\infty} \|e^{(t)}\| < \infty$ and $\sum_{t=1}^{\infty} \langle e^{(t)}, x^{(t)} \rangle$ exists and is finite, we guarantee that

$$\bar{E}(x^*) = \sup_{t \geq 0} \left\{ \sum_{i=0}^{t-1} \langle e^{(i+1)}, x^{(i+1)} - x^* \rangle \right\} < \infty.$$

Together with $D_h(x^{(i+1)}, x^{(i)}) > 0$, we have

$$D_h(x^*, x^{(t)}) \leq D_h(x^*, x^{(0)}) + \bar{E}(x^*) < \infty,$$

which indicates

$$0 \leq \sum_{i=0}^{\infty} D_h(x^{(i+1)}, x^{(i)}) < D_h(x^*, x^{(0)}) + \bar{E}(x^*) < \infty,$$

and hence $D_h(x^{(i+1)}, x^{(i)}) \rightarrow 0$.

3. Based on the above two items, we know that the sequence $\{x^{(t)}\}$ must be bounded and has at least one limit point x^∞ . The most delicate part of the proof is to establish that $0 \in \nabla f(x^\infty) + \partial \iota_X(x^\infty)$. Let $T = \nabla f + \partial \iota_X$, then T denotes the subdifferential mapping of a closed proper convex function $f + \iota_X$ (f is a linear function and X is a closed convex set). Let $\{t_j\}$ be the sub-sequence such that $x^{t_j} \rightarrow x^\infty$. Because $x^{t_j} \in X$ and X is a closed convex set, we know $x^\infty \in X$. We know that $D_h(x^*, x^{(t+1)}) \leq D_h(x^*, x^{(t)}) + \langle e^{(t+1)}, x^{(t+1)} - x^* \rangle$

and $\sum_{k=0}^{\infty} \langle e^{(t+1)}, x^{(t+1)} - x^* \rangle$ exists and is finite. From [23, Section 2.2], we guarantee that $\{D_h(x^*, x^{(t)})\}$ converges to $0 \leq d(x^*) < \infty$. Define $y^{(t+1)} := \lambda_k \left(\nabla h(x^{(t)}) - \nabla h(x^{(t+1)}) + e^{(t+1)} \right)$, we have

$$\lambda_k \langle y^{(t+1)}, x^{(t+1)} - x^* \rangle = D_h(x^*, x^{(t)}) - D_h(x^*, x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) + \langle e^{(t+1)}, x^{(t+1)} - x^* \rangle.$$

By taking the limit of both sides and $\lambda_k = \lambda > 0$, we obtain that

$$\langle y^{(t+1)}, x^{(t+1)} - x^* \rangle \rightarrow 0.$$

For the reason that y^{k_j+1} is a subgradient of $f + \iota_X$ at x^{k_j+1} , we have

$$f(x^*) \geq f(x^{k_j+1}) + \langle y^{k_j+1}, x^* - x^{k_j+1} \rangle, \quad x^* \in X, x^{k_j+1} \in X.$$

Further let $j \rightarrow \infty$ and using f is lower semicontinuous, $\langle y^{(t+1)}, x^{(t+1)} - x^* \rangle \rightarrow 0$, we obtain

$$f(x^*) \geq f(x^\infty), \quad x^\infty \in X$$

which implies that $0 \in \nabla f(x^\infty) + \iota_X(x^\infty)$.

4. Recall the inexact scheme (27), we can equivalently guarantee that

$$(x - x^{(t+1)})^T \left\{ \beta^{(t)} \nabla f(x^{(t+1)}) + \left[\nabla h(x^{(t+1)}) - \nabla h(x^{(t)}) \right] - e^{(t+1)} \right\} \geq 0, \quad \forall x \in X.$$

Together the convexity of f and the three point lemma, we obtain

$$f(x^{(t+1)}) \leq f(x) + \frac{1}{\beta^{(t)}} \left[D_h(x, x^{(t)}) - D_h(x, x^{(t+1)}) - D_h(x^{(t+1)}, x^{(t)}) - (x - x^{(t+1)})^T e^{(t+1)} \right].$$

Let $x = x^*$ in the above inequality and recall the assumption (22), i.e.,

$$f(x) - f(x^*) \geq \eta d(x),$$

we have with $\beta^{(t)} = \beta$

$$\begin{aligned} \eta d(x^{(t+1)}) &\leq \frac{1}{\beta} \left[d(x^{(t)}) - d(x^{(t+1)}) \right] + \frac{1}{\beta} \left((x^{(t+1)} - x^*)^T e^{(t+1)} \right) \\ &\leq \frac{1}{\beta} \left[d(x^{(t)}) - d(x^{(t+1)}) \right] + \frac{1}{\beta} \left(C \|e^{(t+1)}\| + \langle x^{(t+1)}, e^{(t+1)} \rangle \right), \end{aligned}$$

where $C := \sup_{x \in X^*} \{\|x\|\}$. The second inequality is obtained through triangle inequality. Then

$$d^{(t+1)} \leq \mu d^{(t)} + \mu \left(C \|e^{(t+1)}\| + \langle x^{(t+1)}, e^{(t+1)} \rangle \right),$$

where $\mu = \frac{1}{1+\beta\eta} < 1$. With our assumptions and according to Theorem 2 and Corollary 2 in [33], we guarantee the generated sequence converges linearly in the order of $\mathcal{O}(c^t)$, where $c = \sqrt{\frac{1+\max\{\mu, \rho\}}{2}} \in (0, 1)$.

Based on the above four items, we guarantee the convergence results in this theorem. □